The QPACE Supercomputer

Applications of Random Matrix Theory in Two-Colour Quantum Chromodynamics

Dissertation

zur Erlangung des Doktorgrades
der Naturwissenschaften (Dr. rer. nat.)
der Fakultät für Physik
der Universität Regensburg

vorgelegt von

Nils Meyer

aus Kempten im Allgäu

Mai 2013

Die Arbeit wurde angeleitet von: Prof. Dr. T. Wettig
Das Promotionsgesuch wurde eingereicht am: 07.05.2013
Datum des Promotionskolloquiums: 14.06.2016

Prüfungsausschuss:
Vorsitzender: Prof. Dr. C. Strunk
Erstgutachter: Prof. Dr. T. Wettig
Zweitgutachter: Prof. Dr. G. Bali
weiterer Prüfer: Prof. Dr. F. Evers

Für Denise.

Contents

List of Figures

List of Tables

List of Acronyms

API	Application Programming Interface
ASIC	Application-Specific Integrated Circuit
BRAM	Block Random Access Memory
BAUD	Symbols per second
DDR	Double Data Rate
DMA	Direct Memory Access
DRAM	Dynamic Random Access Memory
CB	Control Box
CDT-DCR	Clock Domain Transition Device Control Register (bus)
Cell BE	Cell Broadband Engine
CPLD	Complex Programmable Logic Device
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CTS	Clear To Send
DCR	Device Control Register (bus)
DP	Double precision
ECC	Error-Correcting Code
EIB	Element Interconnect Bus
EXC	Exception
FES	Front-End System
FIFO	First In First Out
FIR	Fault Isolation Register
FLOP	Floating-Point Operation
FLOPS	Floating-Point Operations per Second
FP	Floating-Point
FPGA	Field Programmable Gate Array
FSM	Finite State Machine
FUQD	Front-end Utilities for QPACE Daemon
GBIF	General Buffer Interface
GPU	Graphics Processing Unit
GS	Global Signals

HDL	Hardware Description Language
I/O	Input/Output
IWC	Inbound Write Controller
IWCEM	IWC Extension Module
IP	Internet Protocol
IR	Interrupt
IRQ	Interrupt Request
MAC	Media Access Controller
MDIO	Management Data Input/Output
MIC	Memory Interface Controller
MM	Main Memory
MSB	Most Significant Bit
MUX	Multiplexer
NC	Node-card
NUMA	Non-Uniform Memory Access
NWP	Network Processor
LS	Local Store
LSB	Least Significant Bit
LUT	Look-up Table
OWC	Outbound Write Controller
OWCEM	OWC Extension Module
OS	Operating System
OSI	Open Systems Interconnection
PHY	Physical Layer (in OSI model), Physical Transceiver
PLB	Processor Local Bus
PPE	Power Processing Element
PSU	Power Supply Unit
QFC	QPACE Front-end Client
QPACE	QCD Parallel on the Cell Broadband Engine
RAM	Random Access Memory
RC	Root-card
RCC	Root-card Controller
RF	Register File
RTL	Register Transfer Level
RTS	Request To Send
RX	Receive
RXD	Received Data
SDK	Software Development Kit
SIMD	Single-Instruction Multiple-Data
SLOF	Slimline Open Firmware

SP	Service Processor, Single Precision
SPE	Synergistic Processing Element
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
SDRAM	Synchronous Dynamic Random Access Memory
SRC	Superroot-card
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
TNW	Torus Network
TX	Transmit
TXD	Transmitted Data
UART	Universal Asynchronous Receiver Transmitter
UCF	User Constraint File
VPD	Vital Product Data
XAUI	10 (Gigabit) Attachment Unit Interface

Outline

QPACE is a massively parallel and scalable supercomputer designed to meet the requirements of applications in Lattice Quantum Chromodynamics. The project was carried out by several academic institutions in collaboration with IBM Germany and other industrial partners. The project officially started in 2008, and the final design was completed in 2010. In November 2009 and June 2010 QPACE was the leading architecture on the Green 500 list of the most energy-efficient supercomputers in the world.

The first part of my thesis is dedicated to QPACE. This part is structured as follows:

- In the first chapter I give a brief overview on the diversity of supercomputer architectures recently sighted on the market. I also give an overview on my contributions to the QPACE supercomputer project.
- In the second chapter I give a general overview on the QPACE architecture. I introduce the node-card design, discuss the system networks, and explain the cooling concept. The networks setup is discussed and auxiliary system components are introduced.
- The third chapter is dedicated to the IBM PowerXCell 8i. I give an overview on the most important architectural highlights of this microprocessor and point out its impact on supercomputing. I introduce an abstraction model for the hardware architecture that allows to estimate the sustained performance of the Wilson-Dirac Lattice QCD application kernel on the PowerXCell 8i. I also introduce non-linear models for data exchange amongst the processor's functional units which adequately predict the execution times of elementary DMA transactions.
- In the fourth chapter I briefly comment on FPGA technology and give an overview on the recent implementation of the QPACE network processor on the Xilinx Virtex-5 FPGA. After a short introduction of the relevant entities I discuss in detail the DCR device tree, the high-speed Inbound Write Controller logic, and the low-speed serial communication UART device logic.
- In the fifth chapter I comment on the verification of the QPACE design and introduce some of the software tools designed for administration. I give a brief overview on the diversity of system tests applied and, as an illustration for the design of the test software, I discuss the implementation of a simple test case for the node-cards. A software tool designed for booting of the node-cards is introduced afterward. Finally the QFC software tool is discussed in some detail. The QFC was designed for elementary support of maintenance operations and administration of QPACE.
- The project is briefly summarized in chapter nine.

Additional reference to the QPACE hardware and software is provided in Appendix A. In Appendix A.1 references to the most important source code files for discovery tools, test cases, and

VHDL entities are provided. In Appendix A.2 the public DCR memory map is summarized. The API of the *witchlib* software library, which provides of a set of high-level functions for root-card access from the front-end, is provided in Appendix A.3. Full reference to the QFC functionality is given in Appendix A.4.

In the second part of this thesis applications of random matrix theory in two-colour Quantum Chromodynamics and fermions in the fundamental representation are studied. Hermitian and non-Hermitian formulations of chiral random matrix theory of the orthogonal ensemble provide predictions for the microscopic limit of the spectral density of the Dirac operator. The latter formulation is associated with QCD at non-zero baryon chemical potential. Lattice simulations were carried out in the quenched approximation at zero and non-zero baryon chemical potential and the spectral properties of the overlap operator were compared to the results of chiral random matrix theory.

The second part is structured as follows:

- In chapter six I introduce the formalities of non-Abelian gauge theories and give a formal overview on chiral symmetry breaking in the continuum. As a motivation for Hermitian and non-Hermitian chiral random matrix theory I introduce pion effective theory associated with QCD with gauge group SU(2) and fermions in the fundamental representation. I also sketch the formulation of the effective theory including a baryon chemical potential. Next I present the essential ideas behind QCD on the lattice in the quenched approximation. I introduce the Wilson-Dirac and the massless overlap operator as an implementation of fermions on the lattice. I close the chapter with the implementation of the baryon chemical potential in the overlap operator.

- Chapter seven is about chiral random matrix theory. First I introduce the essentials of Hermitian chiral random matrix theory and discuss the random matrix model for the Dirac operator. Then I introduce the non-Hermitian random matrix model which includes a symmetry-breaking parameter associated with the baryon chemical potential in QCD. The breaking of Hermiticity renders the spectrum of the Dirac operator complex-valued. For both formulations of chiral random matrix theory I present the microscopic spectral density. The distribution of the lowest-lying eigenvalue at zero chemical potential is also introduced.

- In chapter eight lattice simulations of two-colour QCD in the quenched approximation are evaluated. The spectrum of the overlap operator obtained from simulations of 8^4, 10^4, and 12^4 lattices is compared to the microscopic spectral density derived from Hermitian chiral random matrix theory. The distribution of the lowest-lying eigenvalue is also evaluated. The study is carried out for several choices of the Wilson mass parameter. Next I evaluate the symmetries of the spectrum of the overlap operator and its flow at non-zero baryon chemical potential. Afterward I compare the spectrum of the operator operator at non-zero baryon chemical potential to the microscopic spectral density derived from non-Hermitian chiral random matrix theory. Simulations at non-zero chemical potential were carried out on 4^4 and 8^4 lattices.

- The results are summarized in chapter nine.

Appendix B provides details of the evaluation of the spectrum of the overlap operator. In Appendix B.1 the Krylov-Ritz method for evaluation of the matrix sign function is described. The statistical bootstrap method was applied to estimate statistical errors on eigenvalue distributions. The bootstrap method is summarized in Appendix B.2.

Part I

The QPACE Supercomputer

Chapter 1

Introduction

1.1 Supercomputers

Today's large-scale supercomputers are giant computer installations providing the processing power of hundreds of thousands of compute cores tightly interconnected within a single network. According to the 40th release of the Top 500 Supercomputer Sites [1] the largest installation on earth in the year 2012 is the Cray XK7 Titan at the Oak Ridge National Laboratory (ORNL) [2], United States, delivering impressive 18 PetaFlops – 18 quadrillion floating-point calculations per second – on the Linpack benchmark. The system consists of more than 18000 compute nodes equipped with AMD 16-core Opteron CPUs and nVidia Tesla K20 GPUs. The largest European supercomputer installations in the year 2012 are hosted at computing sites in Germany. One is the IBM BlueGene/Q Juqueen at the Forschungszentrum Jülich (FZJ) [3], and the other one is the SuperMUC cluster, another machine developed by IBM, installed at the Leibniz Rechenzentrum (LRZ) in Garching [4]. The architectures achieve up to 4 PetaFlops in the same benchmark and rank positions 5 and 6, respectively, in the Top 500 list.

Supercomputers are used in various areas of applications. Among the performance-hungry applications of industrial and financial interest are, e.g., energy and oil exploration, digital content creation, computer aided design, financial analysis and trading. Numerical simulations have also become the third pillar in sciences besides theoretical and experimental research. Research applications such as brain simulations, climate research, material sciences, and quantum field theory require huge amounts of processing power. In a coarse-grained classification two types of supercomputers can be identified that satisfy this demand for compute performance. Capacity machines, such as Japan's flagship K computer [5] and Germany's powerful SuperMUC, typically provide a huge amount of processing power, with the nodes interconnected by custom solutions or industry standards such as InfiniBand and Gigabit Ethernet. Capacity machines are designed to provide the compute power for a large number of applications. In contrast, capability machines are application-optimized supercomputers satisfying the demands of only a small set of applications. Examples for such machines are the Cray XMT [6], a shared memory massive multi-threading architecture optimized for data analysis and data mining, and QPACE, a massively parallel architecture with custom interconnect especially designed to meet the requirements of applications in lattice quantum chromodynamics.

A variety of compute hardware can be found in the supercomputer landscape. The design of most supercomputers operated nowadays is homogeneous, i.e., a single processor architecture provides all the processing power. Off-the-shelf commodity superscalar microprocessors from leading vendors such as Intel and AMD are common choices. However, in the recent years an increasing wealth of heterogeneous architectures entered the landscape backing up server processors with graphics processors, predominantly from the nVidia company. Today graphics cards provide an enormous amount of compute performance, outranging server processor architectures by an order of magnitude.

For example, in the year 2012 nVidia introduced the Kepler GK110 architecture which supports for an aggregate compute power of several TeraFlops in single precision delivered by thousands of cores embedded onto a single chip [7]. The Tesla G20 GPU, one of the implementations of the GK110 architecture, found its way into the Cray Titan supercomputer. The manifold of upcoming heterogeneous supercomputer architectures is expected to increase even more with Intel's powerful first commercial Many Integrated Core (MIC) architecture, the so-called Xeon Phi coprocessor [8].

The performance of the supercomputer installations is steadily increasing. If one extrapolates the evolution of technology then the compute performance of large-scale installations reaches the Exascale, i.e., peak performance on the order of ExaFlops, around the year 2020. However, the design of such an Exascale architecture is an open question unlikely to be resolved by tomorrow. The financial support for further developments in supercomputing is tremendous. The US government has granted financial support for research on supercomputing for the year 2012 on the order of 100 million Dollar [9]. The European Union has doubled its annual investment into supercomputing to 1.2 billion Euro [10] and pursues research on new hardware architectures, e.g., within the DEEP project [11] which is financially supported by eight million Euro.

The challenges coming with the steady increase of compute performance are manifold. On the application software level even homogeneous supercomputer architectures are challenging to program for high performance, e.g., due to the necessity of on-chip and off-chip parallelization on multi-core processor architectures. Heterogeneous supercomputers introduce even more complexity because of non-trivial memory hierarchies, asymmetric node interconnect, and also the generic problem of code portability. Although compiler technology has been steadily improved within the last decades, there exists no convincing compiler-driven parallelization of arbitrary application for neither homogeneous nor heterogeneous architectures, and it is highly doubtful whether this huge step towards automated optimization for any kind of underlying hardware will be performed within the next few years. On the hardware level one of the most concerning issues is the extreme amount of power consumed even by today's installations. Data center operators have to struggle with huge expenses on power and cooling. Today the world wide costs for power and cooling of IT equipment exceeds 25 billion Dollar per year and are comparable to the costs for new hardware [12]. For comparison, in the early 90's the infrastructure and energy cost for standard server boards accounted only for about 20% of the total cost of ownership [13]. Supercomputers, however, are on the edge of technology and some of them do not only exhibit high compute performance but are also very energy-efficient. If one extrapolates nowadays bleeding-edge technology to larger scales then one can expect a potential Exascale platform to consume hundreds of MegaWatt. Therefore, energy efficiency and energy-aware system operation are considered not only main goals, but also main limitations for future supercomputer architectures. The common consensus is to consider a power consumption around 20 MegaWatt for operation of an Exascale system to be realistic. Thus the energy efficiency of such a platform should be somewhat around 100 GigaFlops per Watt. However, such performance is out of scope using the technology available today. In 2012 the most energy-efficient supercomputers listed in the Green 500 deliver an energy efficiency of about 2 GigaFlops per Watt in the Linpack benchmark [14].

In the years 2009 and 2010 the world-leading architecture with respect to energy efficiency was QPACE, an innovative supercomputer designed in joint effort of academia and industry. Its performance relies on IBM's powerful PowerXCell 8i microprocessor tightly coupled to a custom-designed network coprocessor implemented on FPGA technology. The aggregate compute performance provided by this supercomputer is about 200 TeraFlops in double precision. The total power consumption is only about 200 kiloWatt, achieved by low-power chip technology, voltage reduction, highly efficient power supplies, and a water-cooling solution. QPACE achieved a remarkable energy efficiency of 773 MegaFlops per Watt in the Linpack benchmark [14]. Today QPACE is used for research in particle physics. The first part of this thesis provides insight into some details of this supercomputer.

1.2 Contributions to QPACE

My personal contributions to the QPACE design and development cover a variety of fields. These include the hardware and software development, system bring-up, as well as extensive system testing and verification of the system design. The following items provide a brief overview on my contributions to the supercomputer project. Some are discussed in more detail within this thesis:

- Performance model for the Wilson-Dirac operator on the Cell BE.
- Non-linear models for execution time functions of DMA transactions for data transfer between the Local Stores, and also between Local Store and main memory.
- VHDL design of the Inbound Write Controller (IWC) logic and initial design of an extension module, continued by T. Maurer.
- VHDL design of the Universal Asynchronous Receiver Transmitter (UART) logic, which supports for communication via RS-232.
- Initial design of the DCR slave interface logic in VHDL, both for synchronous and asynchronous components, continued by T. Maurer.
- VHDL design of the SPI backdoor logic, which essentially consists of a multi-master arbiter for the DCR device tree, both in synchronous and asynchronous versions.
- Initial version of the MDIO clause 45 logic in VHDL used to control XAUI PHYs in a dual-FPGA setup designed for tests of the torus network logic.
- VDHL design verification using Mentor Graphics ModelSim, and debugging using Xilinx ChipScope logic analyzer.
- High-level *witchlib* library for access to the root-card, based on the low-level *feenlib* library written by S. Solbrig.
- Finalization of system software libraries for access to the superroot-card and power supplies, initially designed by D. Hierl and M. Drochner.
- Development of hardware discovery tools for node-cards, root-cards, and superroot-cards.
- Development of a variety of software for node-cards, root-cards, and superroot-cards that allows to test the design.
- Extensive tests of the system hardware and software in all phases of the design.
- Development of a software tool for booting the node-cards.
- Design of the QPACE Front-end Client, a software tool that provides unified access to the system components.

Chapter 2

Design Overview

2.1 Architecture

QPACE is a massively parallel and scalable supercomputer designed for applications in lattice quantum chromodynamics. The building block of the QPACE supercomputer is the node-card. Each node-card hosts one IBM PowerXCell 8i processor, one Xilinx Virtex-5 FPGA, and six PMC Sierra 10 Gigabit Ethernet transceivers. Thirty-two node-cards and two root-cards are connected to a backplane. Each root-card manages and controls 16 node-cards. The QPACE rack houses eight backplanes. The number of compute nodes per rack is 256. The compute nodes are cooled by water cooling. The water-cooling concept allows for high packaging density by populating both the front- and the backside of a rack with 128 node-cards each. Power is distributed along the backplanes to node- and root-cards. Three power supply units (PSU) are attached to each backplane, one of them being redundant. The PSUs are managed and controlled within the rack by one superroot-card. Node-card, root-card, superroot-card, and backplane are shown in Fig. 2.1.

Each node-card is attached to three communication networks. One high-speed network, the torus network, connects nearest-neighbour nodes in a 3-dimensional toroidal mesh. The Ethernet network connects the node-cards, root-cards, and superroot-cards to the front-end system. The global signals network is a simple two-wire tree network that is used for fast evaluation of global conditions and distribution of a compute-critical exception signal.

QPACE was entirely built with commodity hardware. A custom I/O fabric that is directly connected to the PowerXCell 8i processor was implemented on a Xilinx Virtex-5 FPGA. Data is communicated between nearest-neighbour nodes along six links, each with a peak throughput of 1 GByte/s bi-directional. The nodes are operated by a standard Linux distribution with a few architecture-specific drivers. The Linux kernel runs on the PPE and supports for the widely used SPE runtime management library *libspe2* [15].

In November 2009 and June 2010 QPACE was the leading architecture on the Green 500 list of the most energy-efficient supercomputers in the world [14]. The architecture achieved up to 773 MFlops per Watt in the High Performance Linpack benchmark. In the summer of 2009 QPACE has been deployed at the Jülich Supercomputing Centre and at the University of Wuppertal [16, 17]. The installations consist of four racks each. The aggregate peak performance of QPACE is 200 TFlops in double precision and 400 TFlops in single precision. The average power consumption per rack is on the order of 29 kW. The QPACE installation at the Jülich Supercomputing Centre is shown in Fig. 2.2.

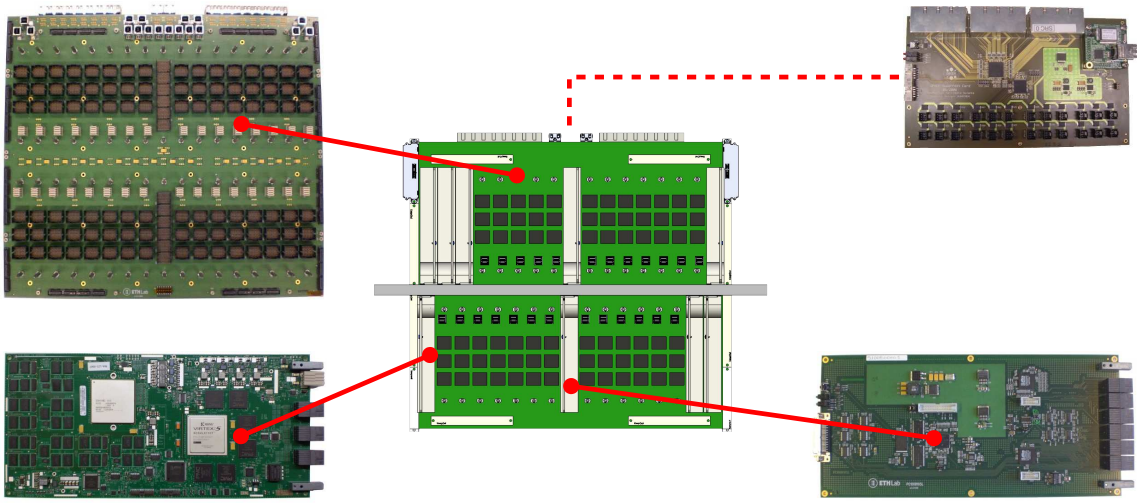

Figure 2.1: Main functional units of QPACE. Thirty-two node-cards (bottom left) and two root-cards (bottom right) are connected to the backplane (top left). Each root-card manages and controls 16 node-cards. One superroot-card (top right) per rack manages and controls the PSUs.

Figure 2.2: The QPACE cluster at the Jülich Supercomputing Centre. In total the 4 racks comprise 1024 IBM PowerXCell 8i processors and deliver an aggregate peak performance of about 100 TFlops for calculations in double precision.

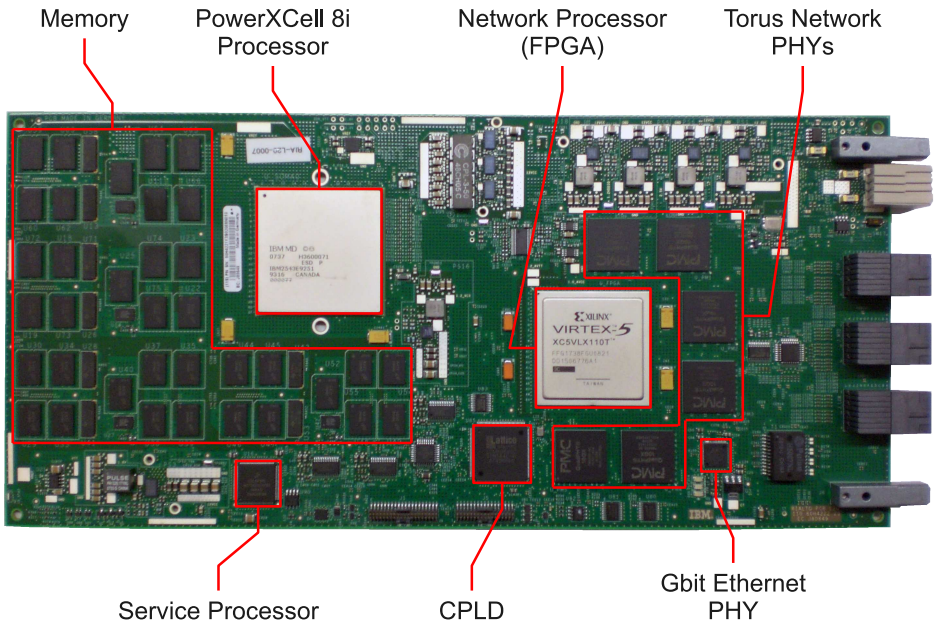

Figure 2.3: Node-card top view with important functional devices highlighted. Each node-card provides one PowerXCell 8i processor and 4 GByte DDR2 memory onboard. The processor is directly connected to the Xilinx Virtex-5 FPGA which acts as an I/O fabric. Six PMC Sierra XAUI PHYs connect to nearest-neighbour nodes. Support for Gigabit Ethernet is provided by the Gigabit Ethernet PHY. The node-card is controlled by the Service Processor and CPLD.

2.2 Node-card

The node-card is the building block of QPACE. It hosts the components necessary for computation of parallel applications. A photograph of the node-card with important functional devices highlighted is shown in Fig. 2.3. Among them are

- 1 IBM PowerXCell 8i
- 4 GByte DDR2-800-SDRAM onboard
- 1 Xilinx Virtex-5 FPGA (network processor)
- 6 PMC Sierra XAUI physical transceivers (PHY) for nearest-neighbour interconnect
- 1 Gigabit Ethernet PHY
- 1 Service Processor and 1 CPLD for management and control
- Ethernet magnetics, flash memory, voltage connector and regulators, board connectors etc.

Each node-card is managed and controlled by an onboard microcontroller, the so-called Service Processor (SP). Node-card-specific information – such as serial number, voltage settings, boot options, and critical error information – are stored on the Vital Product Data memory (VPD) which is accessible by the SP. The images for the Cell BE Slimline Open Firmware (SLOF) [18] and the FPGA bitstream are stored on the onboard flash memory. The firmware boots the node-card via multicast TFTP netboot into a standard Fedora Linux distribution that was extended to support for the QPACE proprietary devices.

The energy efficiency of the node-card was increased by individual voltage tuning. The core voltage of each individual Cell BE was tuned for operation at low voltage setting (which includes a safety margin). Additional reduction of the energy consumption was achieved by optimization of the main memory voltage margins. Voltage tuning improves the power consumption of QPACE by about 10% and therefore increases the performance per Watt ratio. Details about the voltage tuning are provided in Ref. [19].

2.3 Cooling

Roughly 115 Watts have to be dissipated from each node-card. The node-cards are cooled by water, whereas all other system components rely on conventional passive and active cooling by air. Although liquid cooling is not a revolutionary concept in the history of supercomputers, its realization in QPACE addresses the typical disadvantages. Liquid cooling solutions tend to be expensive, maintenance of critical components may become difficult, and electronic devices are seriously damaged when exposed to water. Innovative strategies were introduced to tackle these drawbacks.

The main producers of heat, the node-cards, are packed in housings made of aluminium. The heat is conducted from the components – in particular the Cell BE, FPGA, PHYs, memory chips, and voltage converters – to the surface of the housings. Up to 32 housings are mounted on a flat, water-cooled coldplate made of aluminium. Water-conducting channels inside the coldplate transfer the heat from the node-cards to the liquid circuit. Only a small interfacing area of about 40 cm^2 is necessary for heat transfer from each housing to the coldplate. Thermal contact between the housing and the coldplate is improved by a thin film of synthetical oil. The benefits of the two-component cooling design are summarized in the following:

- The temperature difference between the water inlet and the processor cores is lower than 40°C.
- Water inlet temperature on the order of 40°C is possible (without violation of the chips' temperature specifications).
- The cooling design allows for high packaging density.
- The housings and electronic components are never exposed to water.
- The water circuit stays closed on maintenance operations.
- The node-cards are easily maintained. Hot-plugging is supported.
- No expensive mechanical parts are required.

The concept of the cooling solution and a photograph of a fully populated coldplate are shown in Fig. 2.4. Further details on the QPACE cooling mechanism are provided in Refs. [19, 20], and general aspects of liquid cooling in high-performance computing are discussed in Refs. [21, 22].

2.4 Communication networks

2.4.1 Torus network

2.4.1.1 Characteristics

The torus network was designed for high-speed inter-node communication with low latency. For each node the torus network provides the connections to six nearest-neighbour nodes in a 3-dimensional mesh. In contrast to high-speed networks of other supercomputers based on the Cell BE, the QPACE architecture is optimized for direct communication between the Local Stores (LS) of adjacent processors. Characteristics of the torus network are:

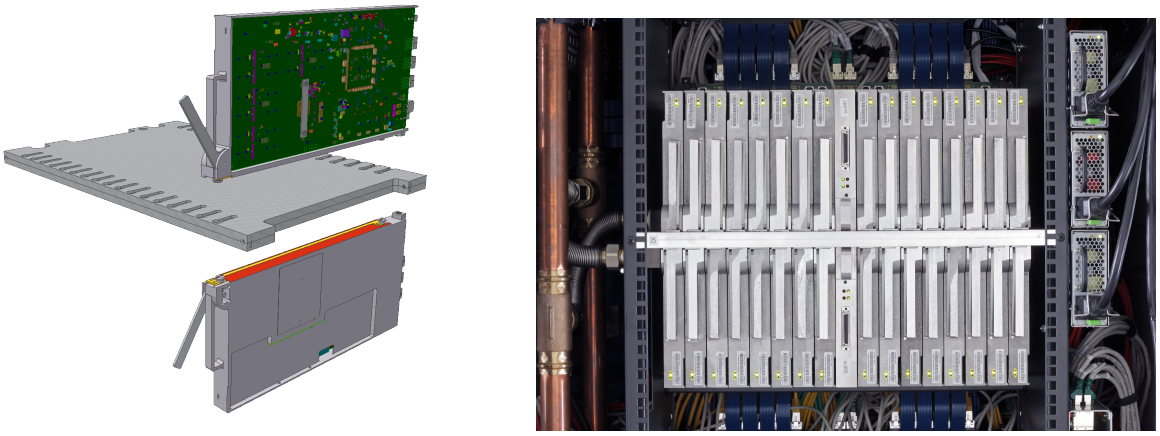

Figure 2.4: The figure on the left shows the QPACE cooling concept. Node-cards mounted in aluminium housings are attached to an aluminium coldplate. Heat is conducted from the housings to the coldplate at the interfacing area (indicated by the red stripe). The photograph on the right shows the backplane fully populated with 32 node-cards and two root-cards. Manifolds are mounted on the left side of the rack and connect the coldplate to the cooling circuit. Three power supply units per backplane are attached to the right side of the rack.

- High-speed connection based on 10 Gigabit Ethernet physical layer with a peak throughput of 1 GByte/s per link bi-directional.
- Application-optimized link layer with support for slimline datagrams of fixed size of $128 + 4 + 4$ bytes (payload + header + CRC).
- Optimization for direct LS-to-LS and LS-to-main memory (MM) communication: Direct memory access (DMA) from the local LS to the remote LS or MM without additional buffering. LS-to-LS latency of $\mathcal{O}(3)$ µs.
- Runtime support for partitioning.

2.4.1.2 Communication concept

In the following a simplified overview on the communication concept for the torus network is provided. Detailed reviews of the underlying hardware concepts, their implementation, and the performance of the network are provided in Refs. [20, 23, 24, 25].

The QPACE torus network is optimized for direct communication between two SPEs on adjacent Cell BEs without the need for additional buffering. The data transfer is driven directly by the SPE's DMA engine. No copy operations from/to main memory are required. This approach grants the application access to the full memory bandwidth, because network I/O operations are removed from this performance-critical data path.

To allow for low-latency communication between adjacent Cell BEs any communication overhead was removed from the torus data path. The sending device pushes data onto its network processor, which autonomously passes the message along the 10 Gigabit Ethernet link to one of its six neighbours. Data is finally pushed by the network processor logic onto the receiving device on the Cell BE. There is no negotiation between the sender and receiver. Minimal support for flow control is granted by a credit mechanism: the network processor streams data onto the receiving device only if a credit is provided to the network processor. After the copy operation a notification is sent to the receiver.

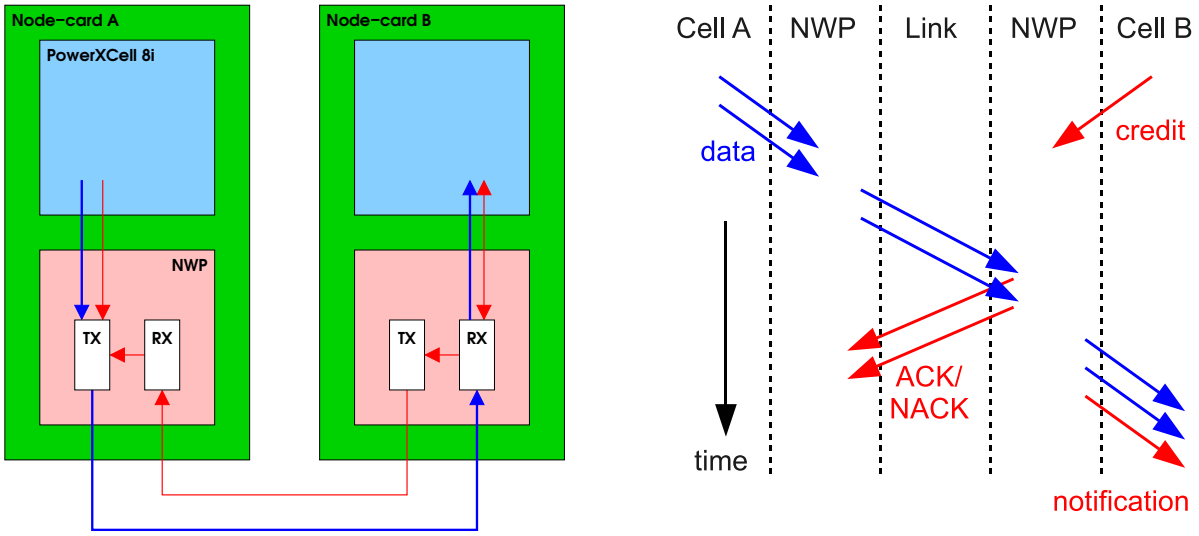

Figure 2.5: The left panel shows the data (blue) and control (red) paths for the torus network. Each datagram transmitted between adjacent network processors (NWP) is authenticated by a (not) acknowledge packet ACK (NACK). In case of not-acknowledge (NACK) the packet is re-transmitted. The right panel shows the time flow of the torus communication process.

Any SPE can be source- and endpoint for communication. This feature requires multiple pairs of SPEs on neighbouring nodes to share the same physical link for communication along the 3-dimensional torus. The QPACE torus network supports for eight *virtual channels* for each link and direction, which effectively allows for logical separation of up to eight sender and receiver pairs per link and per direction.

The torus network software stack comprises a set of low-level communication primitives designed for the SPE. Details about DMA operations and the hardware structure of the network processor are hidden from the user. The simple but effective torus communication pattern is based on matched send and receive commands, see Ref. [26] for full reference of the torus API. The low-level interface requires the following steps to be executed:

1. Initialization of the receive operation by providing for a given link and virtual channel a credit to the network processor.

2. Data is copied to the network processor by the sender. The largest packet size supported is 2048 bytes.

3. Polling of the notification address, typically in the LS, while waiting for the data to arrive.

After completion of the first two steps the network processor handles the communication autonomously. The time between sending the data and receiving the data can be used by the communicating devices for other operations, e.g., for computation or other data load and store operations, including further network transfers. Data, control, and time flow of the torus communication are shown in Fig. 2.5.

2.4.1.3 Partitioning

The torus network supports for a variety of partitions within the 3-dimensional mesh. On a single backplane the logical volume of nodes can be as large $(x, y, z) = (1, 4, 8)$. In y-direction the number of nodes is increased by vertical connections of adjacent backplanes using cables (intra-rack). In a

similar fashion the x-direction is extended by horizontal connections of adjacent backplanes using cables (inter-rack). Different partitions are created by switching between two serial interfaces of the PMC Sierra PHY. Some of the serial interfaces are connected to additional traces on the backplane that allow for various connections between the nodes. Switching between the primary and redundant interface of the PHYs is performed by software during runtime. The following partitions are supported by the QPACE architecture:

- 1, 2, 4, 8 nodes in the z-direction.
- 1, 2, 4, 8, or 16 nodes in the y-direction.
- 1, 2, or $2N$ nodes in the x-direction (here N is the number of interconnected racks).

2.4.2 Ethernet network

The switched Ethernet network connects the node-cards, root-cards, and superroot-cards to the front-end system. The front-end comprises one master server used for machine control and management, one login server which serves as central user access point, and several I/O servers dedicated to the Lustre parallel file system. The Ethernet network is divided into I/O, machine management, and control sub-networks. It also provides connection to the outside world. A closer look on the arrangement of these sub-networks will be taken in Sect. 2.5.2.

Eack QPACE rack provides 24 1-Gigabit Ethernet uplinks for connection of the node-cards to the front-end system. The external bandwidth is on the order of 2 GByte/s per rack, which is also supported by the local disk storage systems. The throughput is sufficient to sustain the compute performance for non-I/O intense applications such as Lattice QCD.

2.4.3 Global signals network

The global signals (GS) network is a simple but effective two-wire tree network with support for partitioning. Information exchange is based on the state of the global signals network. There is no (native) support for packet-based communication. The capability of the GS network is limited to three active states and one idle state. The active states are used to evaluate global conditions and to distribute an exception signal in the case of a compute-critical error. Global conditions are used, e.g., to synchronize the nodes by creation of a barrier. Parallel jobs can be terminated by the global kill signal. Communication along the GS network proceeds in the following steps:

1. Node-cards assigned to the partition propagate the signals `NOP`, `TRUE`, `FALSE`, or `KILL` to the so-called root logic.
2. The root logic is coded into programmable devices hosted by the root-cards and superroot-cards. These cards serve as end-points for local (and global) `OR` and `AND` operations within the GS network hierarchy. The reduced information is propagated up the hierarchy until the end-point is reached.
3. The result of the global reduction operation is propagated down along the hierarchy to the node-cards.

The global signals network was designed to support for all partitioning options granted by the torus network. See Ref. [27] for more information on the global signals tree network.

2.5 System setup

2.5.1 Front-end system

The front-end system (FES) is the interface between the user and the back-end, i.e., the QPACE racks. The FES consists of dedicated servers and interconnects. The FES is integrated into a single rack and serves the following purposes:

- Provide the physical connections between the QPACE racks, front-end servers, and the outside world by the Gigabit Ethernet switch layer.
- Grant access to the Lustre parallel file-system.
- Support for job scheduling by the batch queueing system.
- Support for maintenance operations and administration.
- Control and monitoring of the system by automated software services.

The master server acts as the portal to the system. The master server runs the batch queueing system, the front-end service daemon, and system monitoring services. The login server allows the users to login to the system from the outside world, to schedule and maintain jobs, and access data stored on the Lustre parallel file-system. Support for the Lustre file-system is provided by dedicated metadata and object storage servers. More information about the FES aiming at both administrators and users is provided at the QPACE homepage at DESY, Ref. [28].

2.5.2 Ethernet networks

The Ethernet network is organized as a layered switched network system. It is divided into multiple virtual sub-networks. Multiple switches per layer are stacked, i.e., several physical switches are combined and act as one logical switch with the port capacity of the sum of the single switches. The advantages of the stacked setup are the simplified administration by reduced IP numbers, reduced cabling, and the high inter-switch throughput provided by the dedicated stacking bus.

The layered Ethernet network and the virtual sub-networks for the QPACE installation at the University of Wuppertal are shown in Fig. 2.6. The three layers consist of:

1st layer Each QPACE rack consists of stacked switches that connect to the node-cards. Per rack another dedicated switch connects to the root-cards and the superroot-card.

2nd layer One stacked switch connects the front-end servers, I/O servers, and the racks.

3rd layer Additional switches connect the FES to the outside world and to the administrative networks.

Four Virtual Local Area Networks (VLAN) are configured on top of the switched network. VLANs allow for multiple private networks sharing the same hardware resources. The Ethernet network is subdivided into

- I/O network: Connection between front-end servers and node-cards.
- Control network: Connection between master server, root-cards, and superroot-cards.
- Management network: Connection between the management Ethernet devices in the front-end.
- External network: Connection to the outside world.

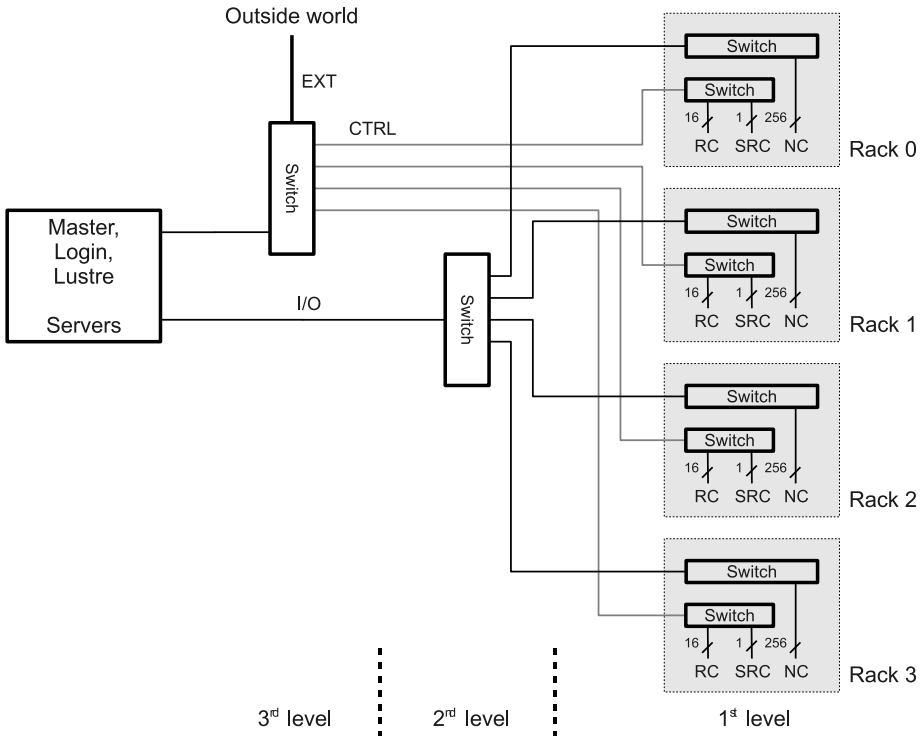

Figure 2.6: Simplified overview of the Ethernet switched network. The external network (EXT) connects the front-end system to the outside world. The I/O network connects the front-end servers and node-cards (NC). The control network (CTRL) grants the master server access to the root-cards (RC) and superroot-cards (SRC).

2.6 Other system components

2.6.1 Root-card

The root-card serves multiple purposes. One of them is the support for administration of the node-cards. Up to 16 node-cards are controlled by the root-card. Two root-cards handle the thirty-two node-cards attached to the backplane. In total 16 root-cards control a complete QPACE rack comprising 256 node-cards. The root-cards are attached to the Ethernet control network described in Sect. 2.5.2 and are controlled by the master server.

The root-card also provides the logic for handling of the global signals network and distributes a global clock signal. This clock signal, which is generated by an oscillator on the master root-card and gets distributed amongst all other root-cards within a tree network, serves as the reference clock for the torus network logic on the network processor on the node-card. The purpose of the global clock is to maximize the clock alignment at the torus transmit and receive logic. The torus reference clock can be switched between the global clock signal and a local clock signal that is generated individually on each node-card.

The root-card provides connections to the node-cards, the Ethernet control network, and the global signals network. Relevant data paths are shown in Fig. 2.7. Each root-card hosts one micro-controller (RCC) and two complex programmable logic devices (CPLD). CPLD 0 acts as switch for reset lines, UART, and SPI. CPLD 1 hosts the logic that handles the global signals (root logic). In the following each connection shown in Fig. 2.7 is briefly described:

A 100 Megabit Ethernet connection to the Ethernet control network.

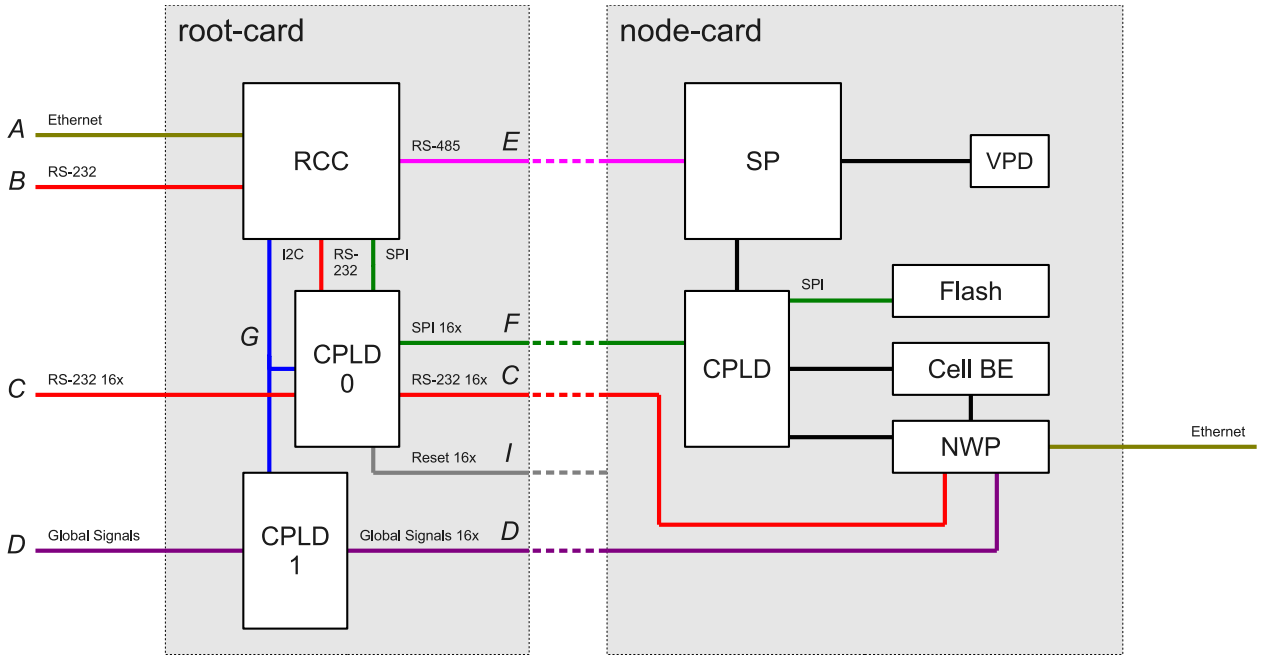

Figure 2.7: Root-card data paths and connections to the node-cards. Only a single node-card is shown.

B One connection to the RCC via RS-232 (mode 8N1), serving as failover data path. The UART of the RCC is accessible by the debug connector.

C Sixteen connections RS-232 (mode 8N1) from CPLD 0 to the node-card network processors (NWP). This data path allows for monitoring and control of the Cell BE firmware (SLOF) during the boot process and grants access to the Linux console. Each NWP is accessible by either the RCC or an external connection by the debug connector. Only one NWP can be accessed by the RCC at a time.

D Connection to the global signals network. CPLD 1 handles the uplinks and downlinks within the tree network. The global signals logic performs the reduction and distribution of the signals along the partition.

E The RS-485 multi-drop bus provides connections between the RCC and 16 Service Processors (SP). The RCC acts as the master, while the SPs are slaves on the bus. Communication is limited to one RCC-SP connection at a time. The communication protocol is proprietary.

F The SPI interface of the RCC is multiplexed by CPLD 0 and allows for access to the flash memory of the node-cards. The flash memory stores the SLOF image and the FPGA bitstream. Only one flash memory device is accessible at a time.

G Control of CPLD 0 and 1 by the RCC via I^2C.

I Sixteen reset lines to the node-cards. Each node-card can be hard-reset individually.

The RCC is operated by an embedded Linux operating system with support for the common Ethernet software stack [29]. The Linux image is loaded on power-on of the RCC via TFTP netboot. Custom-designed software libraries provide remote access to the root-card from the master server. The

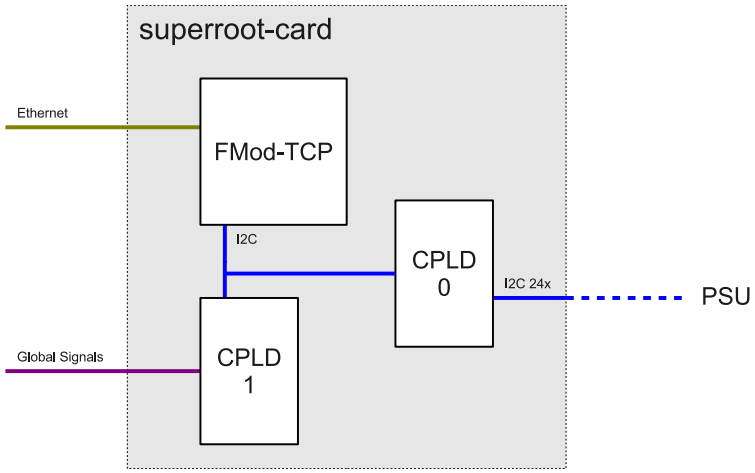

Figure 2.8: Superroot-card data paths. Only a single PSU is shown.

feenlib software library[1] provides low-level functions for control of the RCC and CPLDs. The *witchlib* software library was built on top of the *feenlib* library. It provides a series of high-level functions that, e.g., allow for control of the node-cards. The libraries are accessed by several test and administration tools, see Sect. 5.3 for an example. The API of the *witchlib* library is provided in Appendix A.3.

2.6.2 Superroot-card

Each QPACE rack hosts one superroot-card. The superroot-card monitors and controls the power supply units (PSU). It also handles the global signals at rack top level. The superroot-card connects to the Ethernet control network by the FMod-TCP device [30]. The FMod-TCP essentially consists of a microcontroller unit that allows for remote control of its interfaces via the embedded TCP/IP stack. The controller comes with a variety of standard interfaces, e.g., Ethernet and I^2C. The I^2C bus establishes a connection between the FMod-TCP and two CPLDs mounted on the superroot-card. CPLD 0 provides another 24 I^2C connections accessible by the FMod-TCP. Each of them is dedicated to one of 24 PSUs (with corresponding I^2C expansion cards) attached to the rack, effectively providing support for remote power monitoring and PSU management. CPLD 1 hosts the global signals root logic and connects to the global signals network. Multiple superroot-cards are interconnected such that the global signals tree spans all racks. See Ref. [27] for more information on the global signals tree network. The software libraries for the superroot-card were custom-designed and allow for control of the superroot-card from the master server.

[1] *feenlib* was written by S. Solbrig.

Chapter 3

The IBM Cell Broadband Engine

3.1 The Cell Broadband Engine and the supercomputer league

The Cell Broadband Engine (Cell BE) is a powerful microprocessor developed by Sony, Toshiba, and IBM – the STI alliance – in the years 2001 to 2005. The initial target platform of the Cell BE was the consumer electronics market, especially the PlayStation 3 gaming platform. Whilst the initial version of the processor had limited floating-point performance for calculations in double precision, the revised variant called PowerXCell 8i removed this barrier in 2008. The high floating-point performance of the PowerXCell 8i rendered this processor an interesting option for scientific applications [31]. Besides the outstanding performance in number crunching its superior energy efficiency, typically measured in Flops per Watt, kicked supercomputers based on the PowerXCell 8i on top of the Green 500 list from June 2008 to June 2010 [14].

The most prominent supercomputer that comprised Cell BE technology was the RoadRunner cluster at the Los Alamos National Laboratory [32]. The hybrid architecture hosted more than 6000 AMD Opteron and 12000 IBM PowerXCell 8i processors connected by Infiniband technology. With more than 100000 processor cores distributed amongst 296 racks the cluster hungered for more than 2 MW of power. In the year 2008 RoadRunner was not only the first supercomputer to break the PFlops barrier in sustained performance, it was also amongst the top ten supercomputers in terms of energy efficiency achieving 458 MFlops per Watt [1, 14]. Roadrunner was decommisioned in early 2013.

In the years 2009 and 2010 the success of the Cell BE continued with a new world record. The QPACE supercomputer at the Jülich Supercomputer Centre and the University of Wuppertal achieved 773 MFlops per Watt in the Linpack benchmark. The QPACE architecture was almost 60% more energy efficient than the Chinese Nebulae hybrid CPU/GPU supercomputer at the National Supercomputing Centre in Shenzhen (NSCS). Nebulae was the second-best system on the Green 500 list in June 2010 and achieved 492 MFlops per Watt [14].

3.2 PowerXCell 8i overview

Due to the complexity of the Cell BE architecture this section focuses only on its most striking characteristics. Further details on introductory level are presented, e.g., in Ref. [33]. For full architecture and programming details see Refs. [34, 35].

A schematic diagram for the processor is shown in Fig. 3.1. The PowerXCell 8i is a heterogeneous microprocessor that contains nine physical processing units, namely one 64-bit PowerPC Processor Element (PPE) and eight Synergistic Processor Elements (SPE). Each SPE has 256 kByte of local memory (the Local Store, LS) available, accessible by a dedicated direct memory access (DMA) engine. The register file comprises 128 general-purpose registers, each of them 128 bits wide. The

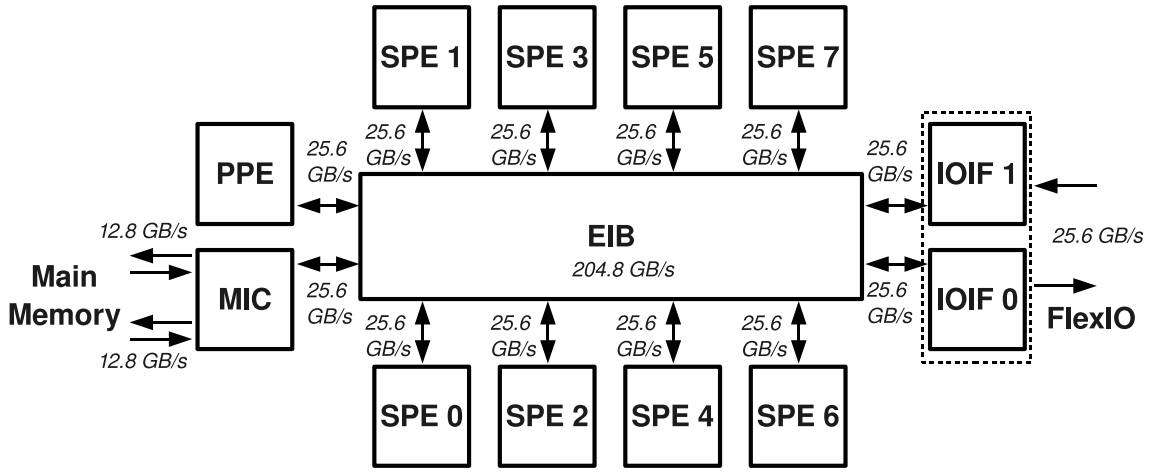

Figure 3.1: Schematic diagram for the IBM PowerXCell 8i. The processor consists of one Power Processor Element (PPE) based on a PowerPC core, eight Synergistic Processor Elements (SPEs), coherent and non-coherent I/O interface (IOIF), and a memory controller (MIC). All units are connected by the element interconnect bus (EIB). The bandwidth values are shown for the 3.2 GHz system clock.

SPE is a single-threaded, dual-issue in-order processor with support for single-instruction multiple-data (SIMD) operations. SIMD instructions perform up to four single precision (SP) or two double precision (DP) floating-point operations simultaneously. The Cell BE's floating-point arithmetic is compliant to IEEE standard 754.[1] At a system clock frequency of 3.2 GHz the outstanding aggregate SP (DP) peak performance of all eight SPEs is 204.8 (102.4) GFlops.[2]

The PowerXCell 8i's on-chip dual-channel DDR2-SDRAM memory interface controller (MIC) delivers a peak memory bandwidth of 25.6 GByte/s. The Rambus configurable I/O interface (IOIF) supports a coherent and a non-coherent protocol with a peak bi-directional bandwidth of 25.6 GByte/s. Internally, all twelve units are connected to the coherent element interconnect bus (EIB) that handles multiple DMA requests simultaneously.

It is worth to mention that the PowerXCell 8i supports for error-correcting code (ECC) of the Local Stores (LS) and also main memory (MM). Data integrity seriously suffers from spontaneous bit flips caused by electromagnetic interference and background radiation. ECC-capable memory allows to mitigate this problem (at the cost of extra storage) and typically supports for either detection and correction of a single bit error, or detection of a double bit error, per 64-bit word.

3.3 Lattice QCD on the Cell BE

The performance of any application relies on the efficient implementation of relevant compute kernels on the underlying hardware structure. Performance models serve as guides to the best implementation of the algorithms and give hints on the optimal data layout and code-optimization strategies. A hardware abstraction model along the lines of Ref. [40] was developed in the initial phase of QPACE

[1] A biased comment by Intel with emphasis on the importance of floating-point arithmetics standardization is provided in Ref. [36]. It is illuminating to continue reading with Ref. [37].

[2] For comparison, the Intel Xeon E7-8870 Westmere-EX server CPU introduced in the year 2011 achieves an aggregate peak performance of 96 (48) GFlops in SP (DP) executing up to 4 (2) Flop per core per cycle clocking all 10 cores at nominal frequency of 2.4 GHz [38]. The nVidia Tesla M2090 GPU, also introduced in the year 2011, achieves 1331 (665) GFlops peak in SP (DP) running 512 compute cores in parallel at a clock frequency of 1.3 GHz [39].

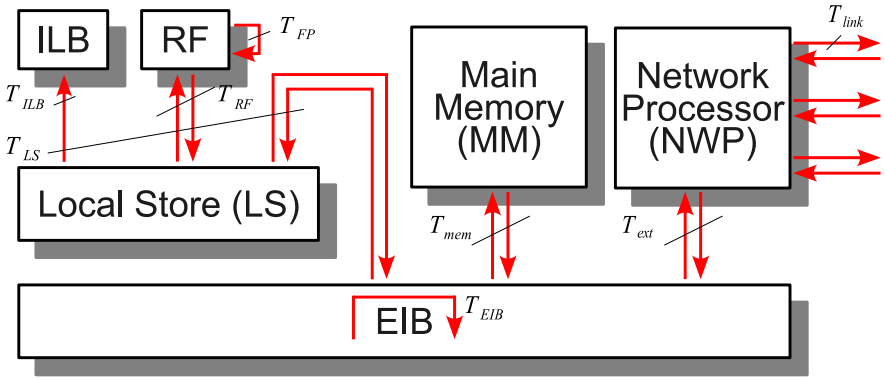

Figure 3.2: Data-flow paths and execution times T_i of micro-tasks considered in the performance model. For simplicity only a single SPE is shown.

that allowed to estimate the performance of the most relevant Lattice QCD kernel for Wilson-type fermions. The model was published in Refs. [41, 42] and is discussed in the following.

3.3.1 Performance model

The performance model is based on the abstraction of the Cell BE hardware architecture. The model relies on the abstraction of the hardware into two classes of devices:

(*i*) *Storage devices*, storing data and/or instructions (e.g., registers, LS, and MM).

(*ii*) *Processing devices*, acting on data or transferring data/instructions from one device to another (e.g., FP units and data buses).

The operations performed by the Lattice QCD kernel can be decomposed into "micro-tasks" which are mapped to the hardware devices. It is assumed that all micro-tasks i performed by the processing devices are running concurrently at maximal throughput β_i, and all latencies λ_i are hidden by suitable scheduling. Any data dependencies are neglected. The execution time T_i of each task i is estimated by the linear ansatz

$$T_i \simeq I_i/\beta_i + \mathcal{O}(\lambda_i) \,, \tag{3.1}$$

where I_i is the amount of data to be processed. The hardware devices, data-flow paths, and associated micro-task execution times T_i considered in the performance model for the Lattice QCD kernel are shown in Fig. 3.2. Among them are

- Execution of floating-point operations (T_{FP}).
- Load of instructions into the instruction look-aside buffer (T_{ILB}).
- Load/store operations between the register file and LS (T_{RF}).
- Off-chip memory access (T_{mem}).
- Internal communications between SPEs (T_{int}).
- External communications between adjacent Cell BEs (T_{ext}).
- Transfers via the EIB (T_{EIB}).

All devices are running in parallel in this model (at maximal throughput) and the kernel is mapped to eight SPEs.[3] The estimate for the total execution time T_{exe} of the kernel is

$$T_{\text{exe}} \simeq \max_i T_i \ , \tag{3.2}$$

thus the execution time is dominated by the most time-consuming micro-task. The identification of this task allows to determine the floating-point efficiency ε_{FP}. Let T_{peak} be the minimal compute time of the kernel, achieved by an ideal implementation that saturates the floating-point performance of the Cell BE, then the floating-point efficiency is defined as $\varepsilon_{\text{FP}} = T_{\text{peak}}/T_{\text{exe}}$.

3.3.2 Lattice QCD kernel

For applications in Lattice QCD with Wilson-type fermions the 4-dimensional hopping term D_h of the Wilson-Dirac operator is the most performance-relevant kernel. The Wilson-Dirac operator is discussed in more detail in Sect. 6.6.1. All compute-intensive tasks, e.g., solving a system of linear equations, involve its repeated application to a quark field ψ,

$$\psi'_x = D_h \psi_x = \sum_{\mu=1}^{4} \left\{ U_{x,\mu} \left(1+\gamma_\mu\right) \psi_{x+\hat{\mu}} + U^\dagger_{x-\hat{\mu},\mu} \left(1-\gamma_\mu\right) \psi_{x-\hat{\mu}} \right\} \ . \tag{3.3}$$

Here $x = (x_1, x_2, x_3, x_4)$ is a 4-tuple of coordinates labelling the lattice sites in a discretized 4-dimensional Euclidian space-time volume. The quark fields ψ'_x and ψ_x are spin-colour vectors assigned to each lattice site x. The vectors consist of twelve complex numbers each. A colour matrix $U_{x,\mu} \in \mathrm{SU}(3)$, consisting of nine complex numbers, is assigned to each link from site x in one out of four space-time directions $\hat{\mu}$. The γ_μ are 4×4 Dirac matrices.

The most efficient way to floating-point performance is to exploit the SIMD instruction set of the SPE. The dual-issue SPE supports for concurrent execution of up to two fused multiply-add operations that compute $a \times b + c$ with real operands.[4] Thus up to four floating-point operations in double precision can be executed per clock cycle. On a single lattice site the computation of Eq. (3.3) amounts to 1320 floating-point operations (neglecting sign flips and complex conjugation). Let the computation perfectly match the instruction set architecture (ISA), i.e., the peak floating-point throughput is saturated, then the minimal compute time is $T_{\text{peak}} = 330$ cycles per site. However, the multiply and add arithmetics arising in the hopping term are not adequately balanced, and the implementation of the operator requires at least 840 fused multiply-add operations. The execution time is $T_{\text{FP}} \geq 420$ cycles per site due to the imbalance. Therefore any implementation of Eq. (3.3) cannot exceed 78% of the peak performance of the Cell BE.

3.3.3 Data layout analysis

State-of-the-art simulations of Lattice QCD carry out calculations on global volumes that consist of $64^3 \times 64$ or even more lattice sites. Thus massive amounts of data need to be distributed on a large number of nodes to be efficiently dealt with. The choice for the data layout on the nodes is crucial for the performance of the Lattice QCD kernel, and the optimal choice for the layout can be found with the performance model introduced above.

Let the lattice sites be partitioned regularly among the nodes, and the local lattice volume assigned to each Cell BE be denoted by V_{Cell}. According to the definition of the hopping term Eq. (3.3), data has to be communicated along four dimensions. However, only the information associated with the surface of the local lattice volume has to be communicated between neighbouring

[3] The PPE is rather weak in floating-point performance and is not taken into account.

[4] There is no native support for complex numbers on the Cell BE.

nodes. The time spent on possible remote communications and on load/store operations of the 9×12 (for the ψ-fields) and 8×9 (for the U-fields) operands of the hopping term strongly depends on the data layout. For the following discussion of the data layout it is assumed that all floating-point numbers are represented in double precision (8 bytes).

The design of the QPACE torus network allows each Cell BE to exchange data with six nearest-neighbours in the mesh. With the constraint of a 3-dimensional physical network that connects neighbouring nodes, one space-time direction must be distributed locally within each Cell BE. For convenience the 4-direction is chosen here. Each Cell BE is assigned the fraction

$$V_{\text{Cell}} = L_1 \times L_2 \times L_3 \times L_4 \tag{3.4}$$

of the global lattice volume. The number of sites locally distributed amongst the SPEs is

$$V_{\text{SPE}} = (L_1/s_1) \times (L_2/s_2) \times (L_3/s_3) \times (L_4/s_4) = V_{\text{Cell}}/8 \,, \tag{3.5}$$

with the SPEs logically arranged as $s_1 \times s_2 \times s_3 \times s_4 = 8$. There are strong limitations to the amount of application data stored on the SPEs. Besides the application data, the LS must also hold the program code, intermediate data, and the run-time environment. For the implementation of a solver one needs access to 8 Dirac spinors and 3×4 colour matrices per lattice site.

Data stored in on-chip memory

One possibility for the data layout is to keep all data in the LS only. Due to the strong constraints on storage the local lattice volume is then restricted to $V_{\text{SPE}} = \mathcal{O}(70)$ lattice sites assigned to each SPE. Since all data associated with the 4-dimension must be kept within the Cell BE, a reasonable choice is $L_4 = 64$ and a logical arrangement of the SPEs by $1^3 \times 8$. This data layout yields an asymmetric local lattice with $V_{\text{Cell}} = 2^3 \times 64$ and $V_{\text{SPE}} = 2^3 \times 8$.

Data stored in off-chip memory

Another possibility is to store the data in main memory. The implementation of a multiple buffering scheme allows for concurrent computation and data load/store from/to memory main and SPEs. The hopping term Eq. (3.3) is computed on a 3-dimensional slice of the local lattice that moves along the 4-direction. Each SPE stores all sites along the 4-direction, and the SPEs are logically arranged as a $2^3 \times 1$ grid. If the U- and ψ-fields associated with all sites of three such slices are kept in the LS at the same time, then all relevant operands are available in the LS. This optimization requirement constrains the local lattice to $V_{\text{Cell}}/L_4 \approx 800$ sites.

Given the ansatz for the data layout, the execution time of the micro-tasks arising in the computation of the hopping term Eq. (3.3) were estimated by Eq. (3.1). The throughput for external communication was assumed to sustain[5] $\beta_{\text{ext}} = 6$ GByte/s with a link bandwidth of $\beta_{\text{link}} = 1$ GByte/s per link and direction. Other relevant information about the Cell BE was taken from public sources. The estimated execution times are shown in Table 3.1 for reasonable sizes of the local lattice. If all data is stored on-chip the sustained performance is limited by remote communications, and the floating-point efficiency is about 27%. In contrast, if all data is stored off-chip the memory wall limits the maximum performance and one can expect to achieve up to 34% efficiency for the implementation of the kernel. Although the performance estimate for both data layouts does not differ significantly, data storage in main memory is the favoured choice. The constraints on remote communications are relaxed, but code optimizations for efficient main memory access become crucial.

[5] The design goal for the external throughput was 6 GByte/s. However, in the recent implementation of the QPACE network processor the external throughput is about 3 GByte/s.

data in on-chip LS		data in off-chip MM			
V_{Cell}	$2 \times 2 \times 2 \times 64$	$L_1 \times L_2 \times L_3$	$8 \times 8 \times 8$	$4 \times 4 \times 4$	$2 \times 2 \times 2$
A_{int}	16	A_{int}/L_4	48	12	3
A_{ext}	192	A_{ext}/L_4	48	12	3
T_{peak}	21	T_{peak}/L_4	21	2.6	0.33
T_{FP}	27	T_{FP}/L_4	27	3.4	0.42
T_{RF}	12	T_{RF}/L_4	12	1.5	0.19
T_{mem}	—	T_{mem}/L_4	**61**	**7.7**	0.96
T_{int}	2	T_{int}/L_4	5	1.2	0.29
T_{ext}	**79**	T_{ext}/L_4	20	4.9	**1.23**
T_{EIB}	20	T_{int}/L_4	40	6.1	1.06
ε_{FP}	**27%**	ε_{FP}	**34%**	**34%**	**27%**

Table 3.1: Comparison of the execution time estimates T_i in 1000 SPE cycles for the micro-tasks arising in the computation of the hopping term Eq. (3.3). The left part corresponds to the on-chip data layout, the right part corresponds to the off-chip data layout. The number of neighbouring sites within and outside each Cell BE is indicated by A_{int} and A_{ext}, respectively. Estimated floating-point efficiencies, $\varepsilon_{\text{FP}} = T_{\text{peak}}/\max_i T_i$, are shown in the last row. Performance bottlenecks are indicated in boldface.

As initial steps for code optimizations simple linear algebra and realistic memory access patterns (neglecting any computation) had been implemented on the Cell BE, see also Ref. [41]. It was found that the implementation of pure linear algebra is uncritical for the performance, leaving enough options for optimizations on other compute-relevant tasks. However, main memory access introduced execution times up to 20% higher than the model predictions for T_{mem}. Recent implementations of a scalable solver for a system of linear equations of type $M\phi = b$ and $M^\dagger M\phi = b$, with M being a huge but sparse complex matrix, achieve a sustained performance of about 20% of peak in single precision, see Refs. [43, 44] for details.

3.4 DMA transaction models

Efficient data transfer on the Cell BE can only be achieved by direct memory access (DMA). The responsibility for data load/store is forwarded to the programmer by full control over DMA engines on the SPEs. One is forced to optimize DMA read and write operations by hand. As discussed in Sect. 3.3.3, the optimization of data exchange, and thus DMA transactions, is crucial for the performance of applications in Lattice QCD.

A suitable method to guide optimizations of the application code is to model the DMA data transfer. However, at the time of investigation the documentation on the DMA mechanisms on the Cell BE was fairly limited. The only fruitful option was to perform benchmarks on Cell BE-based systems, i.e., to measure the execution times of DMA transactions in well-defined environments. The data then served as a template for accurate models.[6]

On the basis of the performance model Eq.(3.1), the simplest ansatz for an execution time function modelling a DMA transfer of size s – from the start of the DMA issue to its completion as defined by the ISA – is

$$T_{\text{DMA}}(s) = \lambda_0 + s/\beta \; . \tag{3.6}$$

Here β corresponds to the limiting bandwidth along the data path. The latency λ_0 is associated

[6] The Cell BE SDK [45] did not accurately model the execution time of data transfers.

with a transfer of zero-size and effectively absorbs the hardware and software management overhead into a single parameter. However, by comparison of measurements on real hardware with the model function Eq. (3.6), it turned out that the linear ansatz is too simplistic to accurately describe, and finally predict, the behaviour of the non-trivial DMA mechanism of the Cell BE. For this reason more accurate models were developed for both transactions between the Local Stores (LS-to-LS) and the Local Stores and main memory (LS-to-MM). The LS-to-LS model was published in Refs. [41, 42].

3.4.1 Local Store to Local Store

The refined model for LS-to-LS data transfer does not only take into account size, latency, and bandwidth, but also the buffers' source and destination addresses A_s and A_d, respectively. Furthermore the native size of one LS line of 128 bytes, and also the fragmentation of the data on the Cell BE in chunks of 128 bytes are essential ingredients. The refined model is based on the following assumptions:

1. Each DMA transfer of size $s > 128$ bytes is fragmented into 128-byte blocks.

2. For a DMA transfer, or residue, of size $s < 128$ bytes that starts at arbitrary position within a single LS line of the source LS, the complete LS line of 128 bytes is copied from the source LS to the destination LS by the EIB.

3. Each data block (≤ 128 bytes) copied to the destination LS that exceeds the 128-byte boundary of a single LS line introduces additional latency λ_a.

According to the assumptions the refined execution time is

$$T_{\mathrm{DMA}}(A_s, A_d, s) = \lambda_0 + \lambda_a N_a(A_s, A_d, s) + N_f(A_s, s)\frac{128 \text{ bytes}}{\beta} \ . \tag{3.7}$$

The model parameters λ_0, λ_a, and β were determined by fits to measurements performed on IBM QS20 blade servers. Each LS-to-LS DMA transfer has a latency of $\lambda_0 \approx 200$ cycles and data is copied with approximately the peak bandwidth, $\beta \approx \beta_{\mathrm{peak}} = 8$ bytes/cycle. Data is fragmented into

$$N_f(A, s) = \mathrm{ceil}\,\frac{(A \bmod 128) \text{ bytes} + s}{128 \text{ bytes}} \tag{3.8}$$

128-byte blocks aligned at LS lines.[7] An additional latency $\lambda_a \approx 16$ cycles is introduced for each copied block that exceeds the 128-byte boundary of a LS line if the source and destination addresses are misaligned, i.e., $\delta A \equiv (A_s - A_d) \neq 0 \pmod{128}$. The model function for N_a, which counts the number of 128-byte blocks that exceed the LS line boundary, is given by

$$N_a(A_s, A_d, s) = [N_f(A_d, s) - 1]\,\vartheta_0(\delta A) \ , \tag{3.9}$$

where ϑ_0 is the Heaviside function and $\vartheta_0(0) = 0$. The contributions to the execution time are visualized in Fig. 3.3 on a transfer of 256 bytes from the source LS to the destination LS for aligned addresses $A_s = A_d = 0 \pmod{128}$, unaligned addresses $A_s = A_d \neq 0 \pmod{128}$, and misaligned addresses $A_s \neq A_d \neq 0 \pmod{128}$.

The impact of the buffer addresses on the execution time can be understood in terms of the effective bandwidth

$$\beta_{\mathrm{eff}} = \frac{s}{T_{\mathrm{DMA}} - \lambda_0} \ . \tag{3.10}$$

[7] The ceiling function is defined as $\mathrm{ceil}(x) \equiv \lceil x \rceil = \min\{n \in \mathbb{Z} \mid n \geq x\}$, giving the smallest following integer of x.

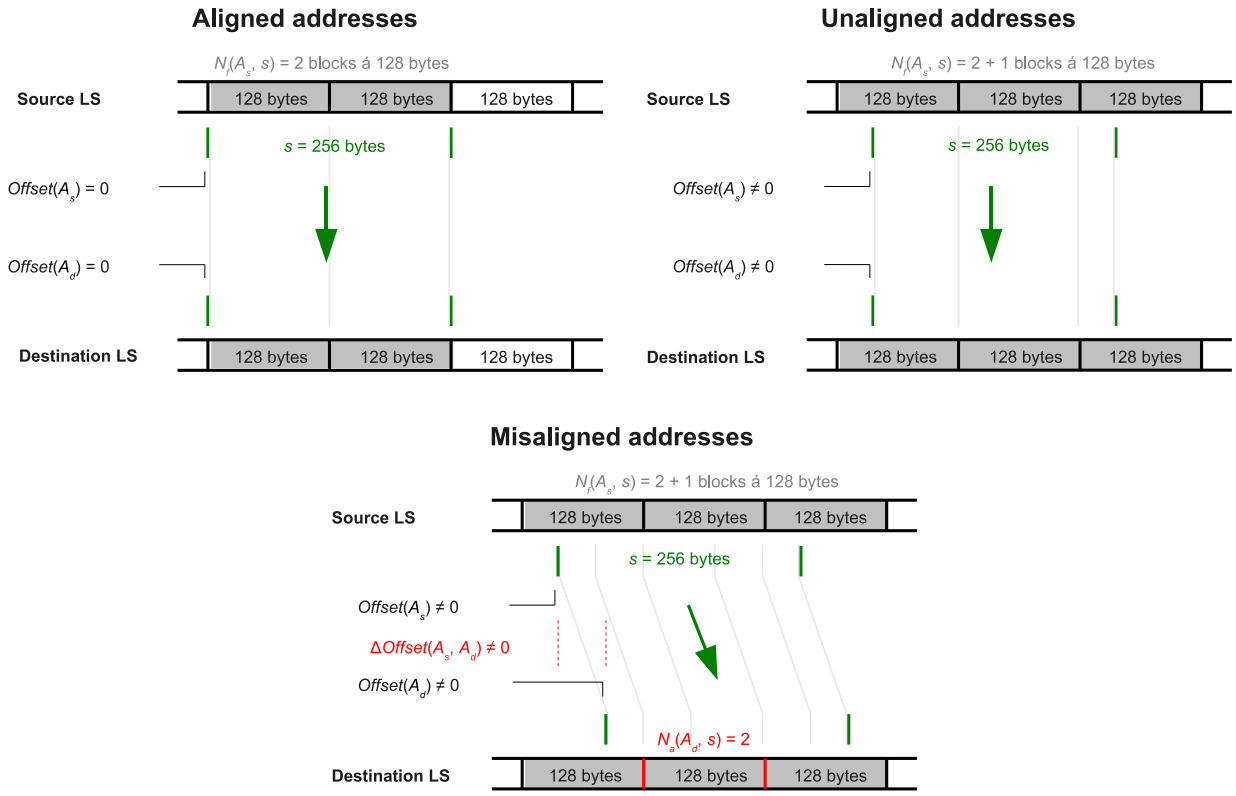

Figure 3.3: Visualization of the LS-to-LS DMA model for a data transfer of 256 bytes. Each LS line consists out of 128 bytes. Contributions to the execution time are indicated by grey lines and show the data fragmentation and the alignment to the LS line. Top left: buffer addresses are aligned at LS lines ($\delta A = 0$). Two 128-byte blocks are copied from the source LS to the destination LS. Top right: buffer addresses are not aligned at LS lines ($\delta A = 0$). Three 128-byte blocks are copied from the source LS to the destination LS. Bottom: buffer addresses are not aligned at LS lines and the address offsets differ ($\delta A \neq 0$). Three 128-byte blocks are copied from the source LS to the destination LS. Due to the misalignment additional latency is introduced.

If the source and destination addresses are identically aligned (with respect to the LS line boundary), i.e. $\delta A = 0$, then data is copied with approximately the peak bandwidth, and therefore $\beta_{\text{eff}} \approx \beta_{\text{peak}} = 8$ bytes/cycle. If the addressed are misaligned ($\delta A \neq 0$) then the additional latency λ_a reduces the effective bandwidth to approximately half of the peak value, $\beta_{\text{eff}} \approx \beta_{\text{peak}}/2 = 4$ bytes/cycle. This effect is clearly seen in Fig. 3.4. The figure shows the execution times as function of the transfer size for both aligned and misaligned buffer addresses. Since the performance of the processor-internal communication strongly depends on the choice for the buffer addresses, misaligned buffers should be avoided by any implementation of (parallel) algorithms on the Cell BE.

3.4.2 Local Store to main memory

Access to main memory from the SPEs is the limiting factor for the performance of the Lattice QCD kernel on the Cell BE. However, modelling of main memory access turned out to be complicated. The main memory subsystem of the Cell BE provides many degrees of freedom to be taken into account by the model, e.g., the logical partitioning of the memory into memory banks, the setup of the dual-channel memory interface controller, and also the Cell BE's resource allocation policy. The

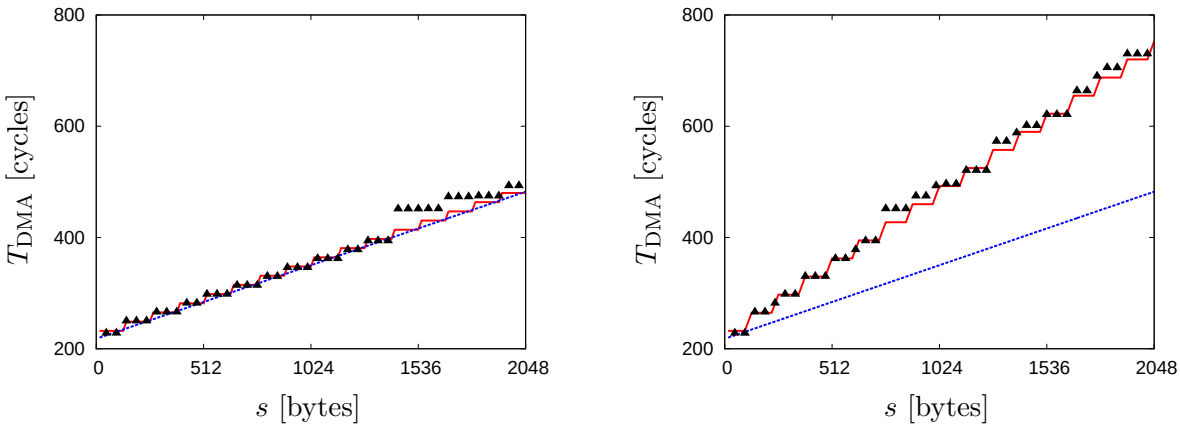

Figure 3.4: Execution time of LS-to-LS DMA transfers as function of the transfer size. The dashed blue line corresponds to the linear model Eq. (3.6). The solid red line corresponds to the refined model Eq. (3.7). Black triangles show measurements on the IBM QS20 blade server. Left panel: source and destination addresses are aligned ($\delta A = 0$). Right panel: source and destination addresses are misaligned ($A_s = 32$, $A_d = 16$, $\delta A \neq 0$). Buffer addresses are relative to the 128-byte boundary of the LS lines.

setup is not necessarily identical on commercially available Cell BE-based platforms.[8]

The results presented here are restricted to IBM QS20 blade servers, which are dual-processor servers with distributed shared memory architecture. As typical for systems that grant for non-uniform memory access (NUMA), access to the local off-chip memory is faster than access to the non-local memory. Spoiling of the measurements due to non-local access to off-chip memory was eliminated by memory allocation and thread binding using a proper NUMA policy supported by `numactl`.

A non-trivial model that describes the execution time of a data transfer of size s between the LS and main memory is given by

$$T_{\mathrm{DMA}}(A, s) = \lambda_0 + \lambda_p N_p(A, s) + N_f(A, s)\frac{128 \text{ bytes}}{\beta} \ . \tag{3.11}$$

This model does not take into account arbitrary buffer addresses, and therefore is limited in the description of main memory access. The execution time function is only valid for buffers equally aligned relative to 128-byte boundaries, $A_s = A_d \equiv A \pmod{128}$. The zero-size latency λ_0 collects the hardware and software overhead and was measured to be on the order of 600 cycles. The bandwidth reaches approximately the peak value, $\beta \approx \beta_{\mathrm{peak}} = 8$ bytes/cycle. Again, data is fragmented into

$$N_f(A, s) = \mathrm{ceil}\,\frac{(A \bmod 128) \text{ bytes} + s}{128 \text{ bytes}} \tag{3.12}$$

chunks of 128 bytes size. A feature of main memory access are plateaus with a fixed size of 2048 bytes found in the measurements. Each plateau contributes an additional latency of $\lambda_p \approx 300$ cycles

[8] Benchmarks for Lattice QCD applications were carried out on the IBM QS20 blade server, Mercury CAB and PlayStation 3. Depending on the system the results for main memory access differ. See Ref. [46] for details.

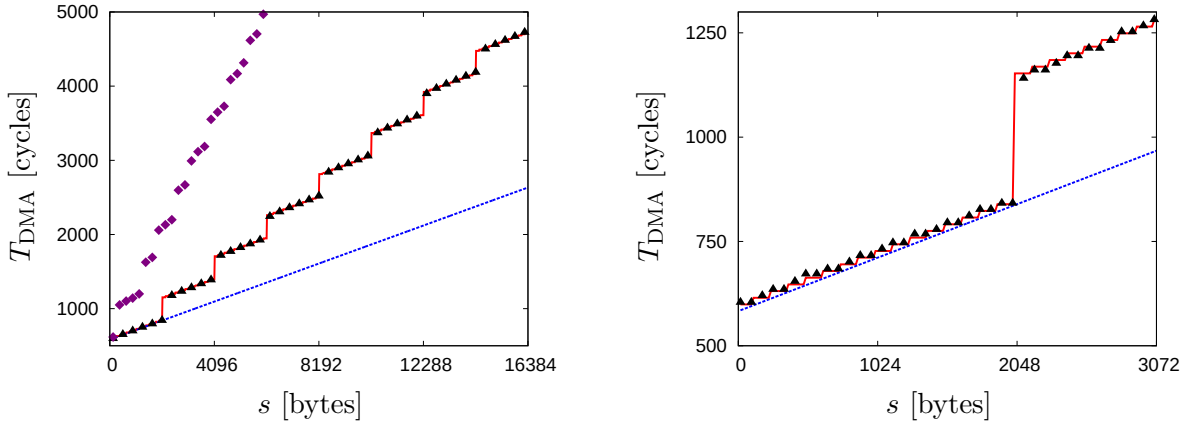

Figure 3.5: Execution time of LS-to-MM DMA transfers as function of the transfer size. The dashed blue line corresponds to the linear model Eq. (3.6). The solid red line corresponds to the refined model Eq. (3.11). Black triangles and purple diamonds show measurements on the IBM QS20 system. Buffer addresses are relative to 128-byte boundaries. Left panel: source and destination addresses are aligned ($\delta A = 0$). Purple diamonds show the effect of misaligned addresses ($A_s = 0$, $A_d = 32$, $\delta A \neq 0$). Right panel: source and destination addresses are unaligned ($A_s = A_d = 32$, $\delta A = 0$). The staircase structure within each plateau is adequately described by the refined model.

to the execution time. The number of plateaus is

$$N_p(A, s) = \mathrm{ceil}\,\frac{(A \bmod 128)\ \text{bytes} + s}{2048\ \text{bytes}} - 1\ , \tag{3.13}$$

counting from zero for convenience. The refined model Eq. (3.11) is shown in Fig. 3.5. DMA transactions with both aligned addresses ($A = 0$) and unaligned addresses ($A \neq 0$) are accurately described. Within each plateau the peak bandwidth is almost saturated, but the contributions of the additional latency λ_p to the execution time limit the effective bandwidth Eq. (3.10) to approximately half of peak for the largest supported DMA transfer size on the Cell BE (16 kByte). The effective bandwidth reaches the peak value only within the zeroth plateau and otherwise decreases as a function of the transfer size s, $\beta_{\mathrm{peak}} \gtrsim \beta_{\mathrm{eff}}(s) \gtrsim \beta_{\mathrm{peak}}/2$. Fig. 3.5 also shows a measurement with misaligned buffer addresses ($\delta A \neq 0$). In this case data transfer is seriously slowed down. The usage of buffers with misaligned addresses introduces prohibitively expensive execution times for main memory access.

It is likely that the behaviour of main memory access from a single SPE is related to the resource allocation and coherency policy of the IBM QS20 dual-processor system. Comparable limitations to the effective bandwidth were found for multiple DMA requests scheduled concurrently on a single SPE. However, the peak bandwidth was measured on all test systems if more than one SPE accesses the MM. A thorough study of main memory-specific parameters and multiple SPEs accessing the memory is provided in Ref. [46].

Chapter 4

The QPACE Network Processor

4.1 FPGA technology

A necessity for performance in parallel applications is the balancing of computation and communication. The PowerXCell 8i supports for outstanding floating-point performance. Therefore a high-speed network with sufficiently low latency and high throughput is required that allows the parallel algorithms to sustain the floating-point pipelines of the processor. Unfortunately for the Cell BE there is no suitable southbridge available that enables for scalable parallel architecture. For this reason an appropriate I/O fabric was custom-designed for QPACE.

In general hardware designs realized with Application-Specific Integrated Circuit (ASIC) technology are expensive in terms of both finances and development. This approach has been pursued in other Lattice QCD machines, e.g., QCDOC [47] and apeNEXT [48]. The requirements for the QPACE network processor are much lower in terms of complexity and resource usage than these system-on-chip designs. Field-Programmable Gate Arrays (FPGA), pioneered by the Xilinx company since the mid-eighties, have become powerful competitors to ASICs on the silicon market. The high logic count and processing capacity render these chips very attractive solutions to a large variety of fields including communications, medical, chip design prototyping, aerospace, military, high performance computing, and many more.

At the highest level FPGAs are reprogrammable silicon chips. Unlike integrated circuits, FPGAs do not have any predefined functionality. Instead, these chips allow to implement virtually any kind of (synchronous) logic circuitry. The logic design is described by a hardware description language (HDL). Amongst the most prominent languages is VHDL,[1] an IEEE standard. VHDL allows to create text-based logic design suitable for synthesis as well as simulation programs for the logic itself. For QPACE almost all application logic is written in VHDL in register-transfer level (RTL) representation.[2] RTL is a high-level representation of the circuitry defined in terms of logical operations on signals and their flow between hardware registers.

Typically, complex designs are broken down into smaller blocks of logic called “entities”. Each entity describes the behaviour of the logic as an interplay between combinational and sequential logic. HDL code is not comparable to program code developed for microprocessors: code written for standard microprocessors is generically executed in serial unless parallelized by the hardware, compiler or programmer. In contrast, HDL design is generically running in parallel unless explicitely serialized. This inherent parallelism enforces extensive testing cycles for non-trivial implementations such as the QPACE network processor. One major advantage of FPGA over ASIC technology is the possibility to verify the design in the field due to reprogrammability. In principle any design can be tested, debugged, and enhanced in a real-world environment with reasonable effort and especially

[1] Very high speed integrated circuits HDL

[2] Few parts of the IBM logic are designed in Verilog, another HDL standardized by IEEE.

no financial risk – a serious advantage over ASIC design. As a penalty the final FPGA design is typically slower than a comparable ASIC design, resources are far more limited, and the price per device is typically higher.

The FPGA that was chosen for QPACE is the Virtex-5 LX110T from Xilinx [49, 50]. This FPGA is medium-sized with an amount of logic blocks sufficient for the design of the network processor. Each logic block consists of two "slices" which provide the necessary basis for combinational and sequential logic: a look-up table, storage elements, and a multiplexer. The logic blocks are interconnected by a switch matrix, see Ref. [50] for the details on the internal structure of this particular type of FPGA. The Virtex-5 also provides "hard cores" embedded into the silicon that come with predefined functionality, e.g., block RAM, Ethernet media access controller (MAC), and RocketIO transceivers. Within some limitations each hard core can be configured according to the needs of the application. Several of these hard cores are used by the implementation of the QPACE network processor.

The resource usage of the QPACE network processor is analyzed in Ref. [20]. About 90% of the available slices are used by the recent logic design. The mapping of the circuitry to the actual hardware, i.e., the generation of a bitstream image which appropriately configures the FPGA and satisfies the target constraints, becomes more and more difficult and even impossible as the resource usage increases. Most of the FPGA's resources are used by the logic interfacing the network processor and Cell BE. The high register count and the large number of clocks consumed by this IBM design, originally not targeting FPGA devices, constrains the resources available for other functional units. Although the effort on optimizations on the performance-critical parts of the logic circuitry was high, the target clock frequencies could not be reached for all parts in the recent design.

4.2 Network processor overview

An overview on the internal structure of the QPACE network processor design is given in Fig. 4.1. Entities depicted in the upper part contain the IBM logic which connects to the Cell BE's non-coherent I/O interface. The lower part shows the application logic designed by academic partners and the connections to the outside world. The following items give a brief overview of the most important entities and their functionality:

- The PowerXCell 8i and the network processor are connected by the Rambus FlexIO high-speed interface. Two FlexIO links with eight data lines per direction each are used for communication between the processors. The Xilinx RocketIO high-speed transceivers are used to interface the FlexIO. This setup is non-trivial and is described in detail in Ref. [51]. The goal for QPACE was to operate the FlexIO links at a frequency of 2.5 GHz. However, the frequency recently used is 2.0 GHz. The resulting gross bandwidth between the FPGA and Cell BE is 32 Gbit/s per direction. Complex logic is required for handling of the communication between the Cell BE and the network processor. The logic design was provided by IBM. All communication details are hidden from the application logic behind the proprietary General Buffer Interface (GBIF). In the recent implementation of the network processor the GBIF runs at a frequency of 200 MHz and supports for data rates up to $\mathcal{O}(3)$ GByte/s bi-directional.

- The IBM logic provides two Device Register Control bus (DCR) masters. One master is accessible by the Service Processor, the other one is accessible by the Cell BE. A DCR arbiter provides access to the DCR device tree from both data paths. The majority of the entities in the network processor are controlled by DCR. The implementation of the DCR bus is discussed in detail in Sect. 4.3. The public DCR memory map is provided in Appendix A.2.

- The Inbound Write Controller (IWC) handles inbound write requests, i.e., data transfers from the Cell BE to the network processor. It acts as a slave to the GBIF. Data written from the

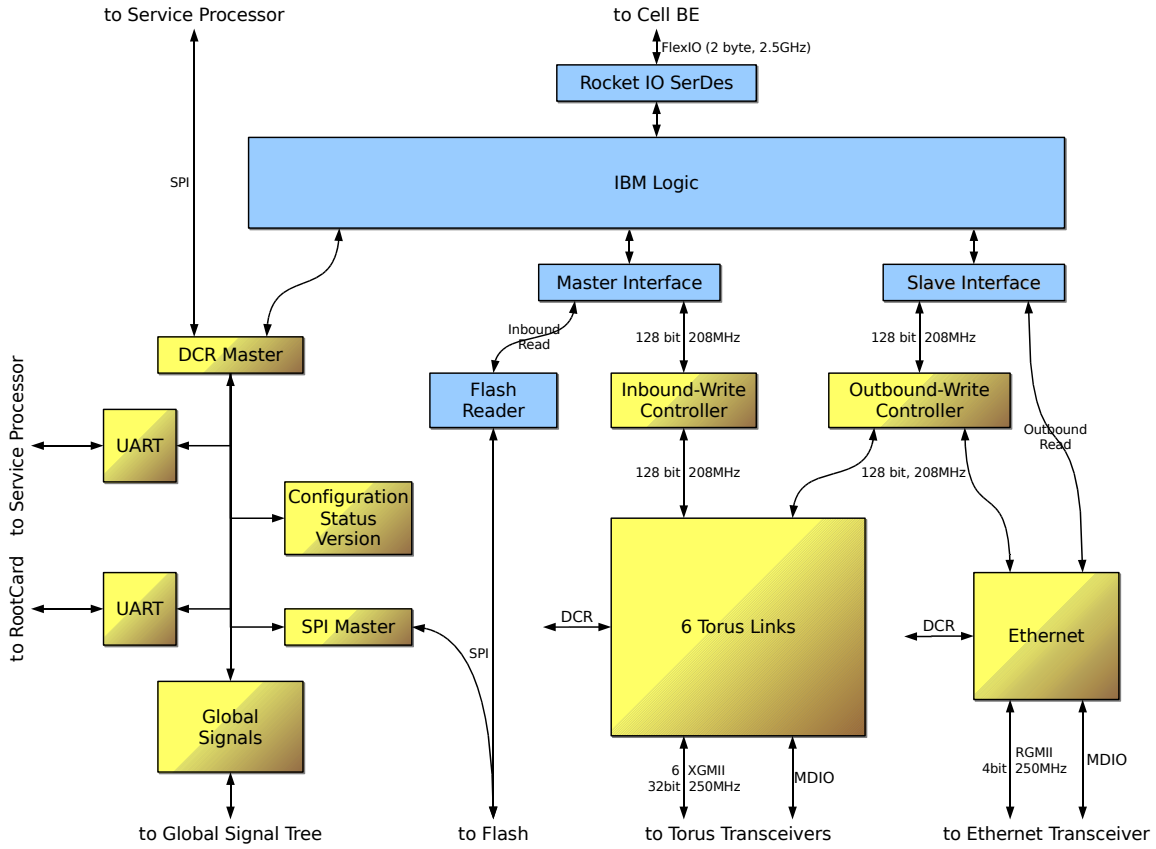

Figure 4.1: Simplified overview of the architecture of the QPACE network processor as published in Ref. [20]. Blue boxes refer to entities provided by IBM, yellow boxes refer to entities designed by academic partners. The recent operating frequency of the IBM high-speed interfaces is slightly lower than indicated.

SPEs to the network processor are transferred from the GBIF to the IWC to one of six torus links. As a design convention the size of data packets written by the SPEs is restricted to multiples of 128 bytes. The IWC logic is discussed in detail in Sect. 4.4.

- The Outbound Write Controller (OWC) is a master on the GBIF. It handles the requests from the Ethernet and the torus network logic using a round-robin arbitration scheme. Similar to inbound data transfer via the IWC, the packet size for outbound data transfer is restricted. The OWC handles data packets in chunks of 128 bytes. Additionally notification packets of 16 bytes size are supported. These packets are dedicated to the communication process along the torus network: after the transfer of block data of predefined size, i.e., n chunks of 128 bytes, a notification is sent to the Cell BE. The polling for notification is performed by the target unit, e.g., the SPE. The OWC design is described in detail in Ref. [52].

- The torus network logic handles the data transfer within the 3-dimensional torus high-speed network. Data provided by the IWC is stored intermediately in one of six transmit FIFOs. The data is passed autonomously from the FIFOs to one of six PMC Sierra 10 Gigabit Ethernet transceiver PHYs along the 32-bit wide 10 Gigabit Media Independent Interface (XGMII). The PHYs are controlled by the torus logic using the Management Data I/O (MDIO) interface. Each PHY drives four serial lines that support for a raw throughput of 2.5 Gbit/s each, transmitting 8b/10b encoded data to the PMC PHY on the neighbouring node in the toroidal

mesh. Thus the data rate per link and direction is 1 GByte/s. The receiver logic stores incoming frames in a dedicated receive buffer. Received data is moved to the destination SPE (or main memory) by the OWC via the GBIF using the credit-based flow control mechanism described in Sect. 2.4.1.

- The Ethernet core supports for Gigabit Ethernet. The GBIF outbound read data path is used to fetch frames to be transmitted directly from main memory. Frames received are written to main memory via the outbound write data path managed by the OWC. The Ethernet logic uses the embedded Tri-mode Gigabit EMAC hard core to manage the data transfer. The Ethernet core interfaces the onboard Gigabit Ethernet PHY by the Reduced Gigabit Media Independent Interface (RGMII). The PHY is controlled by another MDIO interface. The Ethernet logic is described in detail in Ref. [53].

- The global signals (GS) core connects to the GS tree network. Per direction two differential lines are used to propagate the information. The core logic decodes and encodes the four signal states `NOP`, `TRUE`, `FALSE`, and `KILL`.

- Two Universal Asynchronous Receiver Transmitters (UART) are used for low-speed serial communication. One UART is dedicated to communication between the Service Processor and the IBM logic. Communication is necessary during the FlexIO training sequence. The other UART is dedicated to communication with the root-card. The latter allows for monitoring and control of the Cell BE during the boot process. It also grants access to the Linux shell during runtime. The UART logic is discussed in Sect. 4.5. The connections between the node-card and root-card are shown in Sect. 2.6.1.

- Other entities in the network processor comprise the logic necessary for reset handling, interrupt handling, SPI flash memory access, and also addressing of various registers dedicated to configuration and status control.

4.3 Device Control Register bus

The Device Control Register bus (DCR) is a standard system bus especially designed for data exchange between a DCR master controller unit and other logic devices in the system. The DCR bus specification is provided in Ref. [54]. The original purpose is to move configuration registers from the IBM PowerPC Processor Local Bus (PLB) to the DCR bus, thereby improving the PLB bandwidth and reducing the latency introduced by access to configuration registers. Although the hardware setup is quite different in QPACE, the implementation of the DCR bus pursues the same objective: decoupling of configuration registers and low-speed I/O devices from the high-speed torus data path. The DCR bus is also used for passing credits for torus communication from the Cell BE to the torus logic. The most relevant features of the DCR bus are:

- Separate control bus, address bus, inbound and outbound data bus. In the design of the QPACE network processor the width of the address bus is 16 bit and the width of the data buses is 32 bit.

- Master/slave architecture with support for multi-master arbitration.

- Simple but flexible bus interface: four-phase handshake supports for simple interfacing of both synchronous and asynchronous devices.

- Serial, parallel, or mixed bus topology.

Two DCR master controllers are provided by the IBM logic. Both grant access to the DCR device tree. One master (M0) is accessed by the Cell BE. This master is controlled by the Cell BE firmware during the boot process. After booting into Linux the data path is used by the device drivers for runtime management, e.g., access to configuration registers and handling of interrupt requests. Applications use this data path to pass credits to the torus network logic.

Another DCR master is accessible by the Service Processor. This data path is used to monitor and control the DCR device tree from an external source and therefore serves as a "backdoor entry" into the FPGA logic. For technical reasons the latter master is directly attached to another master (M1). Both DCR masters M0 and M1 are attached to a multi-master arbiter that provides unified access to the DCR device tree.

The DCR bus structure and the public DCR device tree are shown Fig. 4.2. In total nine devices are directly attached to the public DCR device tree. The devices are listed in Table 4.1. Detailed description for each device is given in the corresponding reference. The public DCR memory map is provided in Appendix A.2. Two UART devices support for low-speed I/O and three devices are dedicated to Ethernet support. Interrupts and resets are handled by the Control Box. Debug information is accessible by the Extension Modules of the Inbound and Outbound Write Controllers. The torus network configuration is accessible by the torus DCR sub-device tree. Credits for high-speed data transfer are passed into the torus logic via this data path.

Device	Description	Reference
UART 0	UART to Cell BE	Sect. 4.5.2
UART 1	UART to Service Processor	Sect. 4.5.2
Ethernet EMAC	Ethernet EMAC	Ref. [53]
Ethernet GBIF	Ethernet GBIF	Ref. [53]
Ethernet GTX	Ethernet GTX	Ref. [53]
CB	Control Box	Ref. [53]
IWCEM	Inbound Write Controller Extension Module	Ref. [53]
OWCEM	Outbound Write Controller Extension Module	Ref. [52]
Torus	Torus DCR sub-device tree	Ref. [55]

Table 4.1: Devices embedded into the public DCR device tree.

4.3.1 Device tree topology

The DCR bus supports for several structural topologies. The bus specification Ref. [54] suggests two common configurations to meet constraints on the chip physical design and timing closure. One suggestion is the arrangement of the DCR devices in a daisy-chain topology. The data flows through a chain network consisting of DCR slave devices. If the slaves are instantiated physically around the chip, such an approach to the bus topology allows for easier wiring within the network processor's complex logic network. The drawback of the daisy-chain topology are the cumulative data path delays introduced by the chain network. These delays effectively limit the DCR bus clock frequency. It was not clear whether this approach to the device tree structure offers the best choice of bus layout for the network processor. Instead, a different topology was chosen from the beginning of the design cycle. In the "distributed OR" topology each DCR slave device receives all outbound master signals directly from the DCR master controller. The inbound DCR master signals are reduced by a logical OR multiplexor in one or more stages based on the geography and wiring constraints of the physical floorplan before connecting to the DCR master. Fig. 4.2 shows the DCR device tree with distributed OR topology connected to the multi-master arbiter. Only a single OR reduction stage is required to meet the timing constraints. The desired multiplexing functionality of the OR reduction only works if the inbound master signals are deasserted by the DCR slave devices (except for the slave addressed).

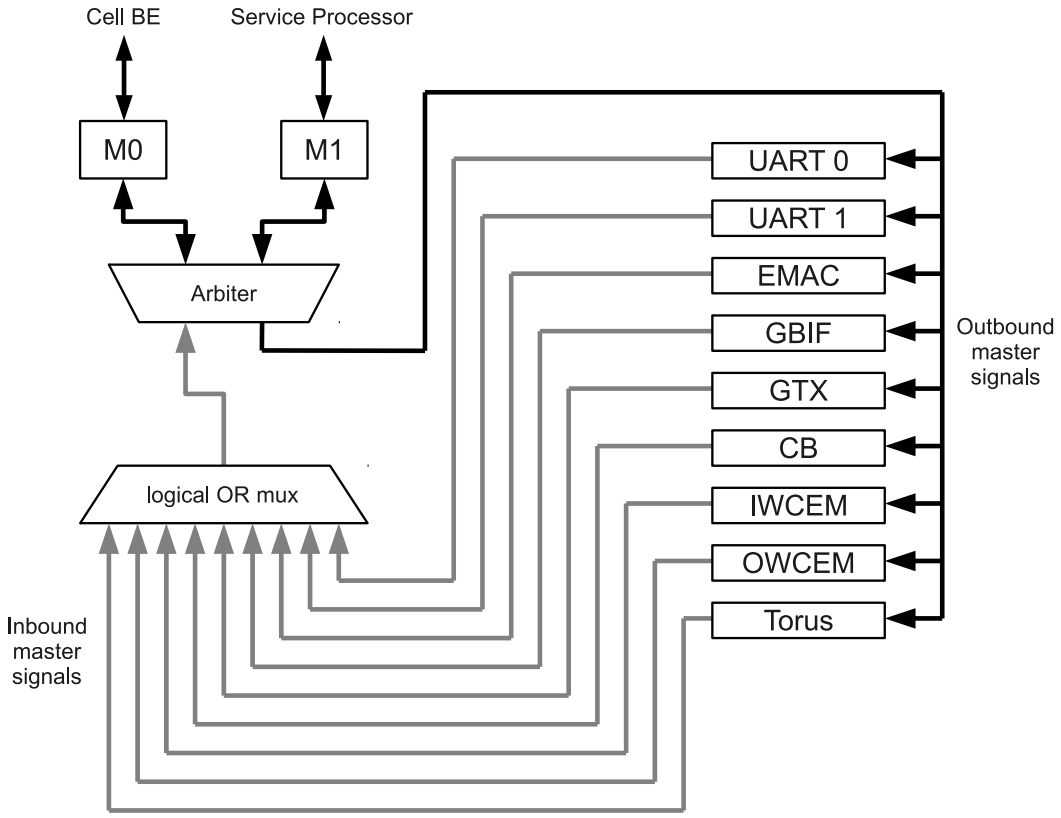

Figure 4.2: Public DCR device tree with distributed OR topology and multi-master arbiter.

4.3.2 Bus interface and protocol

The DCR bus protocol uses a simple but flexible four-phase handshake sequence to regulate the data transfer between the master and the slaves. Data is transferred along separate data buses for read and write transactions. Each transfer is initiated by the DCR master controller and must be acknowledged by the addressed slave device.

The DCR bus signals used in the QPACE network processor implementation are defined in Table 4.2.[3] Other signals defined by the DCR specification Ref. [54] are not required. The data bus directions are defined entity-centric by convention, i.e., *outbound* signals from the DCR master perspective are *inbound* from the DCR slave perspective and vice versa. On read transaction data is transferred from the DCR slave to the DCR master, on write transaction data is transferred from the DCR master to the DCR slave.

Signal	Description
`Read`	Read enable
`Write`	Write enable
`ABus`	16-bit address bus
`DBusOut`	32-bit outbound data bus
`DBusIn`	32-bit inbound data bus
`Ack`	Acknowledge

Table 4.2: DCR bus signals.

[3] Naming conventions in the recent VHDL implementation slightly differ. The differences are due to changes in the design along the development phase.

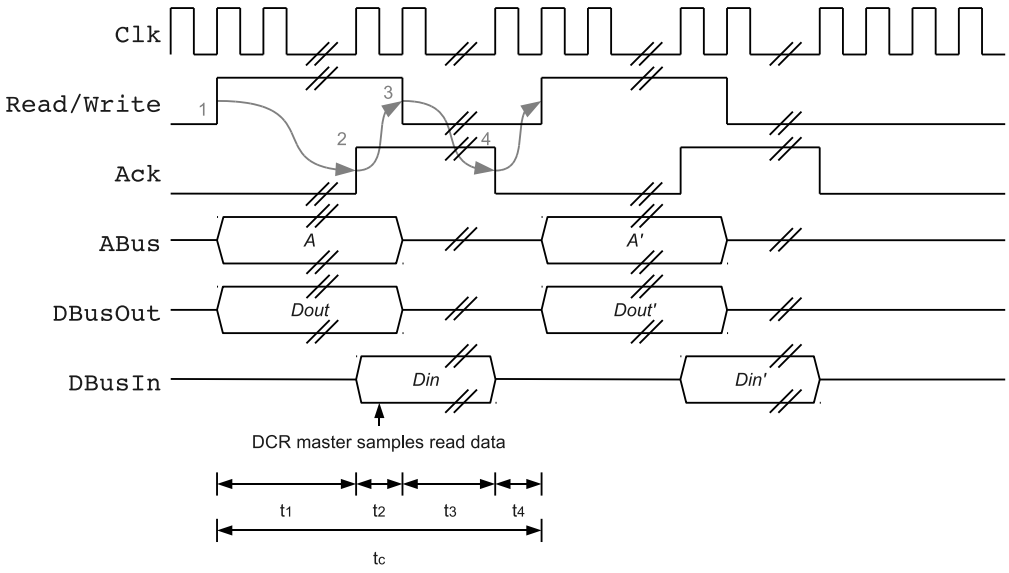

Figure 4.3: Timing diagram for two subsequent DCR bus cycles. The four-phase handshake sequence is shown for the first bus cycle explicitely.

Each DCR bus cycle is governed by a four-phase sequence of handshakes between the master controller and the slave device:

1. On read transaction the DCR master asserts `Read` and deasserts the outbound data bus `DBusOut`. On write transaction the DCR master asserts `Write` and the outbound data bus `DBusOut`. On both read and write transaction the address bus `ABus` is asserted. Data and address bus are valid from assertion to deassertion of `Read` and `Write`. Read and write are mutually exclusive.

2. The DCR transfer acknowledge signal `Ack` is asserted by the DCR slave when data is read from the inbound data bus `DBusIn` on write transaction. On read transaction the DCR slave asserts the outbound data bus `DBusOut` concurrently to `Ack`.

3. The DCR master samples the `Ack` signal and terminates the read or write transaction by deasserting `Read` or `Write`, respectively. On read inbound data is registered from the data bus `DBusIn` the first cycle `Ack` is asserted.

4. The DCR slave deasserts `Ack` when `Read` or `Write` are deasserted. `Read` and `Write` are not re-asserted by the DCR master until `Ack` is deasserted.

The timing diagram for two subsequent DCR bus cycles of master M1 are shown in Fig. 4.3. The DCR bus cycle time t_c is determined by the sum of clock cycles t_i associated with each handshake phase i and addtitional delay cycles t_d in the data path (not shown in the figure),

$$t_c = t_d + \sum_{i=1}^{4} t_i \,. \tag{4.1}$$

The minimal DCR bus cycle time is $t_c = 4$ cycles, which can only be achieved by zero delays in the data path. One DCR clock cycle is required for each handshake phase if the signals are registered in each phase. However, the minimal bus cycle time was not achieved in the QPACE network processor. The delay time introduced by the multi-master arbiter is $t_d = 2$ cycles due to two additional signal

registration stages, see also Sect. 4.3.4. The timings of the phases 2 and 4 are determined by the DCR master implementation. The master M1 introduces fixed timings with $t_2 = t_4 = 1$ cycle. The timings for the phases 1 and 3 depend on the implementation of the slave interface to the DCR bus.

For QPACE unified synchronous and asynchronous DCR slave interfaces were designed that allow for addressing of the register file of slave devices that are running at different clock frequencies. For the synchronous slaves the clock can be same as, or must be derived from, the DCR master clock. If the DCR slave device is driven by the DCR master clock then the DCR bus cycle time in the recent implementation of the logic is $t_c = 9$ cycles. In the asynchronous case the DCR bus cycle time depends on the clock ratio. The synchronous DCR slave interface is described in Sect. 4.3.5. The asynchronous DCR interface supports for clock domain transition (CDT) for arbitrary slave clock source. The CDT-DCR slave interface is described in Ref. [53].

By convention a strobe signal is generated by the unified slave interfaces on read or write request that is passed to the slave device logic. Inbound data must be registered on read strobe by the device, and outbound data must be asserted on write strobe. This convention effectively decouples the internal logic of the slave device from the DCR bus and omits delays of the handshake phase 1 and 3, therefore minimizing the DCR bus cycle time and thus increasing the throughput of DCR transactions.

4.3.3 Master implementation

The DCR master M1 shown in Fig. 4.2 is directly controlled by the Service Processor. It is indirectly accessible via the root-card, cf. Sect. 2.6.1, and therefore allows for external access to the DCR device tree. The *witchlib* library described in Appendix A.3 provides a software interface that allows for access to the DCR device tree from the QPACE master server. The DCR master M1 logic is encoded in the entity `spi_backdoor_dcr_master`. The public ports are listed in Table 4.3.

Signal	Direction	Width	Description
`Clk`	in	1	Master clock
`Reset`	in	1	Master reset
`DBusIn`	in	32	Master data bus in
`Ack`	in	1	Master acknowledge
`Read`	out	1	Master read enable
`Write`	out	1	Master write enable
`ABus`	out	16	Master address bus
`DBusOut`	out	32	Master data bus out

Table 4.3: Public ports declaration for the DCR master M1.

The DCR master logic generates DCR-compliant signals (see Fig. 4.3) and is connected to the DCR arbiter described in the next section. The DCR master uses three registers to define a single DCR transaction: `CTRL[1:0]`, `DATA[31:0]` and `ADDR[DCR_ABUS_WIDTH-1:0]`. Write data must be stored in `DATA` prior to any DCR write transaction. Read data is provided in `DATA` after any DCR read transaction. The corresponding DCR address is stored in `ADDR`. On write the 2-bit control register `CTRL` defines the DCR transaction. `CTRL[1:0] = '11'` starts a DCR read transaction, and `CTRL[1:0] = '01'` starts a DCR write transaction. The control register accepts data only if the master is not busy, i.e., there is no bus cycle in flight. On read `CTRL[0]` returns the master state: `'0'` if idle and `'1'` if busy. Internally the control register `CTRL` is mapped to `master_ctrl_reg` which defines read or write mode. On read of `CTRL` the state of the signal `master_busy` is returned.

The DCR master state machine implements the states `IDLE`, `READ`, and `WRITE` to generate the DCR bus protocol described in Sect. 4.3.2 and supports for the minimal DCR bus cycle time of $t_c = 4$ cycles. The state diagram for the Moore-type state machine is shown in Fig. 4.4. The states directly

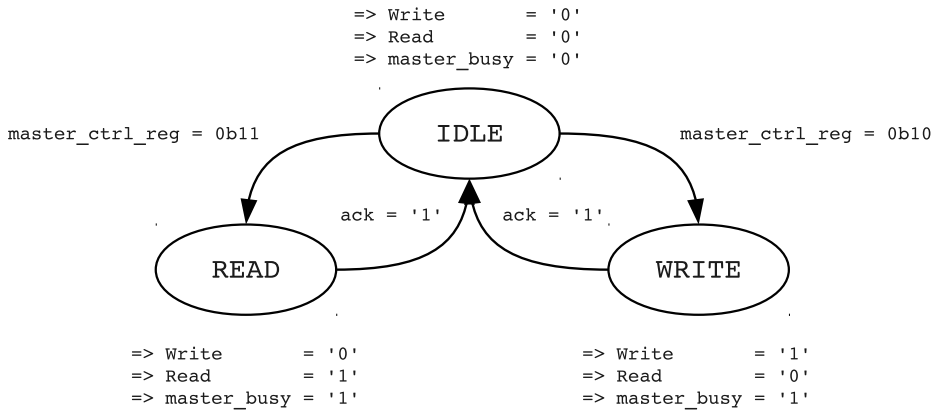

Figure 4.4: State diagram for the DCR master M1.

drive the DCR bus transaction signals `Read` and `Write`. In state `IDLE` the signals `Read`, `Write`, and `master_busy` are deasserted. If `master_ctrl_reg = '11'` then a read transaction is initiated in state `READ`. The DCR bus signal `Read` is asserted until the DCR slave acknowledges the transfer by asserting `Ack`. The `READ` state is followed by `IDLE`. Inbound data is registered in `DATA` only if both `Ack` is asserted and the state is `READ`. If `master_ctrl_reg = '01'` a write transaction is initiated in state `WRITE`. The DCR bus signal `Write` is asserted until the DCR slave acknowledges the transfer by asserting `Ack`. The `WRITE` state is followed by `IDLE`. After each transaction the control register `master_ctrl_reg` is reset to `'00'`. On reset of the DCR master the registers `ADDR`, `DATA` and `CTRL` are deasserted and the state machine is reset to state `IDLE`.

The adress register `ADDR` and the data register `DATA` are instantaneously mapped to the DCR master M1 ports `ABus` and `DBusOut`, respectively. This violates the DCR bus protocol specified in Sect. 4.3.2 if the DCR master M1 is the only master on the bus, because by specification the outbound data bus `DBusOut` has to be deasserted on read transactions (and otherwise does not natively support for the distributed OR topology). However, the multi-master arbiter described in the next section is designed to overcome this violation of the specification. Therefore the DCR master M1 is compliant to the DCR bus protocol only in combination with the multi-master arbiter.

4.3.4 Arbiter implementation

In the following the logic of the DCR multi-master arbiter `dcr_arbiter` shown in Fig. 4.2 is described. The arbiter and the DCR masters M0 and M1 must be clocked synchronous in this implementation. The ports declaration for the arbiter is listed in Table 4.4. Port names are master-centric.

Signal	Direction	Width	Description
`Clk`	in	1	Clock
`Reset`	in	1	Reset
`M0_Read`	in	1	Master 0 read signal
`M0_Write`	in	1	Master 0 write signal
`M0_ABus`	in	16	Master 0 address bus
`M0_DBusOut`	in	32	Master 0 data bus out
`M0_DBusIn`	out	32	Master 0 data bus in
`M0_Ack`	out	1	Master 0 acknowledge
`M1_Read`	in	1	Master 1 read signal

Signal	Direction	Width	Description
M1_Write	in	1	Master 1 write signal
M1_ABus	in	16	Master 1 address bus
M1_DBusOut	in	32	Master 1 data bus out
M1_DBusIn	out	32	Master 1 data bus in
M1_Ack	out	1	Master 1 acknowledge
DBusIn	in	32	Arbiter data bus in
Ack	in	1	Arbiter acknowledge
Read	out	1	Arbiter read signal
Write	out	1	Arbiter write signal
ABus	out	16	Arbiter address bus
DBusOut	out	32	Arbiter data bus out

Table 4.4: Ports declaration for the DCR arbiter.

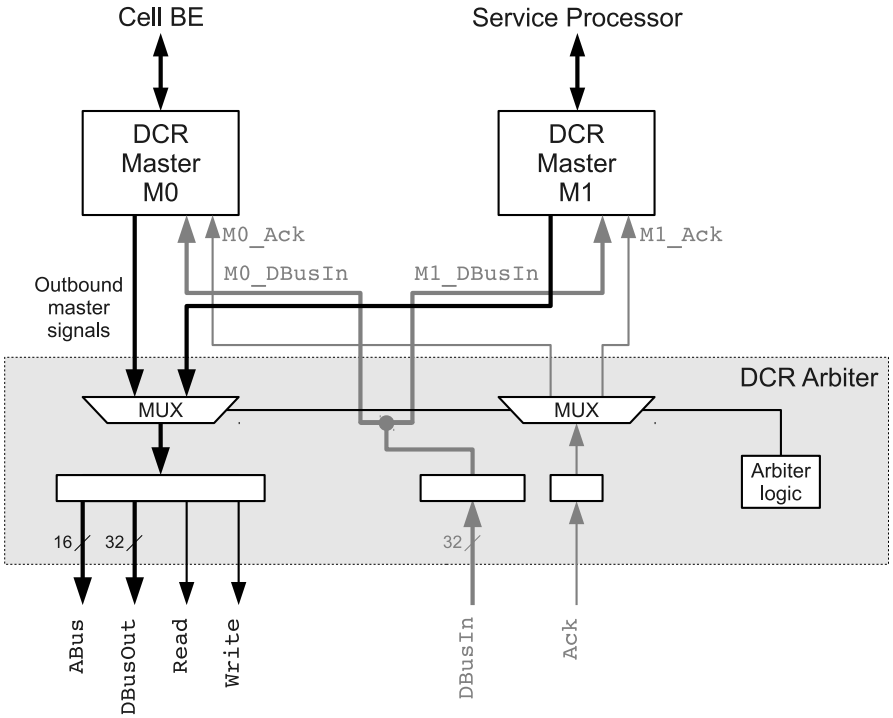

Figure 4.5: Block diagram for the DCR arbiter.

The block diagram for the multi-master arbiter is shown in Fig. 4.5. The arbiter autonomously switches between the DCR masters M0 and M1. The implementation adds one delay cycle in the signal path for both inbound and outbound signals due to signal registration. The inbound and outbound signals are registered on the DCR slave side, therefore minimizing constraints on timing and routing. No skew is introduced between the DCR bus signals.

The outbound signals M*n*_Read, M*n*_Write, M*n*_DBusOut, and M*n*_ABus are multiplexed into unified master registers according to a simple arbitration scheme which prioritizes the DCR master M0 over M1. On read the arbiter outbound data bus is deasserted, DBusOut[31:0] = 0x00000000. Deassertion is required to support for the distributed OR topology of the DCR device tree described in Sect. 4.3.1. The registered inbound data bus DBusIn connects to both DCR masters without

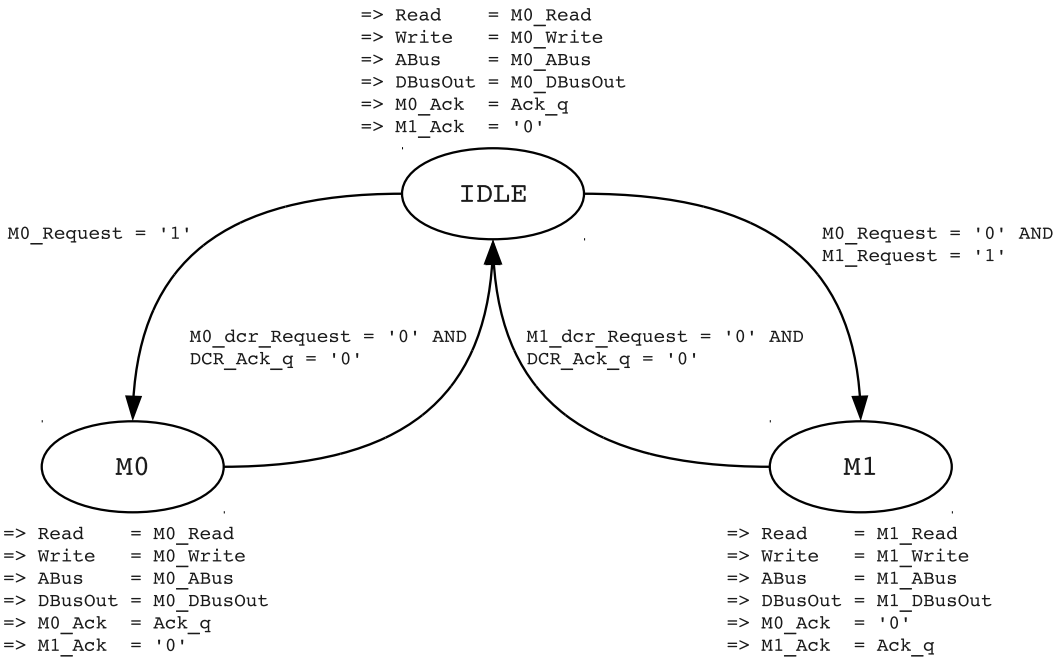

Figure 4.6: State diagram for the DCR arbiter.

further restriction. The DCR acknowledge signal `Ack` is multiplexed to the active master on the DCR device tree.

The state diagram for the DCR arbiter is shown in Fig. 4.6. The arbiter implements a Moore-type state machine. Therefore the outputs are uniquely defined by the actual state. The initial state is `IDLE`. If a DCR transaction is initiated by DCR master Mn either the signal `M`n`_Read` or `M`n`_Write` is asserted, indicating a read or write transaction, respectively. For each DCR master the signals are combined into `M`n`_dcr_Request` by a logical OR reduction. On assertion of `M0_dcr_Request` the state is switched to `M0` regardless of the state of `M1_dcr_Request`, effectively prioritizing DCR master M0 over M1. DCR master M1 may handle the bus only if `M0_dcr_Request` is deasserted. The corresponding state is `M1`.

The outbound master signals `M1_`* are multiplexed to the registered output ports of the arbiter only if the state `M1` is active, otherwise the outbound signals `M0_`* are multiplexed to the registered output ports. The DCR acknowledge signal `Ack` is registered in `DCR_Ack_q` in each clock cycle. In state `M1` the signal `DCR_Ack_q` is forwarded to DCR master M1 via the `M1_Ack` port, otherwise `M1_Ack` is deasserted. The acknowledge signal `M0_Ack` assigned to DCR master M0 is deasserted in state `M1`. In all other states the signal `DCR_Ack_q` is forwarded to DCR master M0 via the `M0_Ack` port. The states `M0` and `M1` switch to `IDLE` state once the DCR transaction has finished, i.e., `M`n`_dcr_Request` and `DCR_Ack_q` are deasserted. On reset the arbiter state is `IDLE`. Any DCR operation is suppressed by deasserting `Read`, `Write` and `DCR_Ack_q` as long as `Reset` is asserted.

4.3.5 Synchronous DCR interface implementation

One of the advantages of the DCR protocol is its simplicity. Any logic device, referred to as “client” in the following, is easily coupled to the DCR device tree by a simple logic interface. Two different types of such interfaces were developed for QPACE. The synchronous interface `dcr_slave_interface` was designed for those client that either use the same clock as the DCR master controller or a derived clock, supporting for integer fractions of the DCR master frequency. The asynchronous

interface `cdt_dcr_slave_interface` described in Ref. [53] supports for arbitrary client clock source and performs the transition of the clock domain within the interface logic. Both interfaces can be exchanged without major effort. On a change of the client logic, i.e., a change in the definition of the clock source, the DCR interface can simply be replaced without major changes in the definition of the interface. In the following the synchronous DCR interface is described. The ports declaration is listed in Table 4.5.

Signal	Direction	Width	Description
`ONETOONEMODE`	generic	boolean	Enable One-To-One mode
`DEVICE_SELECT_WIDTH`	generic	natural	Width of upper address bus
`DEVICE_ADDR`	generic	natural	Device address
`Clk`	in	1	Clock
`Reset`	in	1	Reset
`Read`	in	1	Master read signal
`Write`	in	1	Master write signal
`ABus`	in	16	Master address bus
`DBusOut`	in	32	Master data bus out
`DBusIn`	in	32	Master data bus in
`Ack`	out	1	Master acknowledge
`Reg_DBusIn`	in	32	Client data bus in
`Reg_Read`	out	1	Client read strobe
`Reg_Write`	out	1	Client write strobe
`Reg_ABus`	out	$16 -$ `DEVICE_SELECT_WIDTH`	Client address bus
`Reg_DBusOut`	out	32	Client data bus out

Table 4.5: Ports declaration for the synchronous DCR interface.

The block diagram for the synchronous DCR interface is shown in Fig. 4.7. The master ports interface with the DCR master (arbiter). The client ports are attached to (or embedded into) the client device. The logic of the interface is self-regulating, i.e., the DCR transaction signals `Read` and `Write` drive the operational sequence for read and write transactions and also the four-phase handshake sequence. No state machine is required to operate the interface and the logic count is very small. The DCR address bus `ABus` is split into upper block `Abus[15:`N`]` and lower block `Abus[`$N-1$`:0]` where $N = 16 -$ `DEVICE_SELECT_WIDTH`. Here `DEVICE_SELECT_WIDTH` defines the number of bits reserved for the client device address `DEVICE_ADDR` in the DCR device tree. The address decoder compares the upper address bus block `Abus[15:`N`]` with the device address provided by `DEVICE_ADDR`. Due to the implementation of the address decoder the real client device address must be right-shifted by N. The lower address bus block `Abus[`$N-1$`:0]` is provided to the client device. On address match the master `Read` (`Write`) signal is registered into R1r (R1w). The register R2 is fed by the OR-reduced R1r and R1w, therefore indicating a valid DCR transaction. In the subsequent clock cycle the register R2 is fed into R3 unconditioned and thereby generating the DCR acknowledge signal `Ack`. On reset all registers are deasserted and therefore any DCR transaction is suppressed.

It was mentioned in Sect. 4.3.2 that the DCR interface generates a client read (write) strobe signal. The strobe signal indicates a valid read (write) transaction initiated by the DCR master. The strobe signal is asserted only if R1r (R1w) is asserted, but R2 is still deasserted. Therefore the strobe signals are active exactly for one clock cycle. The client device is allowed to read from the inbound client data bus `Reg_DBusIn` only if the client signal `Reg_Read` is asserted. Data must be registered by the client and passed to the outbound client data bus `Reg_DBusOut` if client `Reg_Write` is asserted. The client outbound data bus `Reg_DBusOut` is multiplexed to the master inbound data

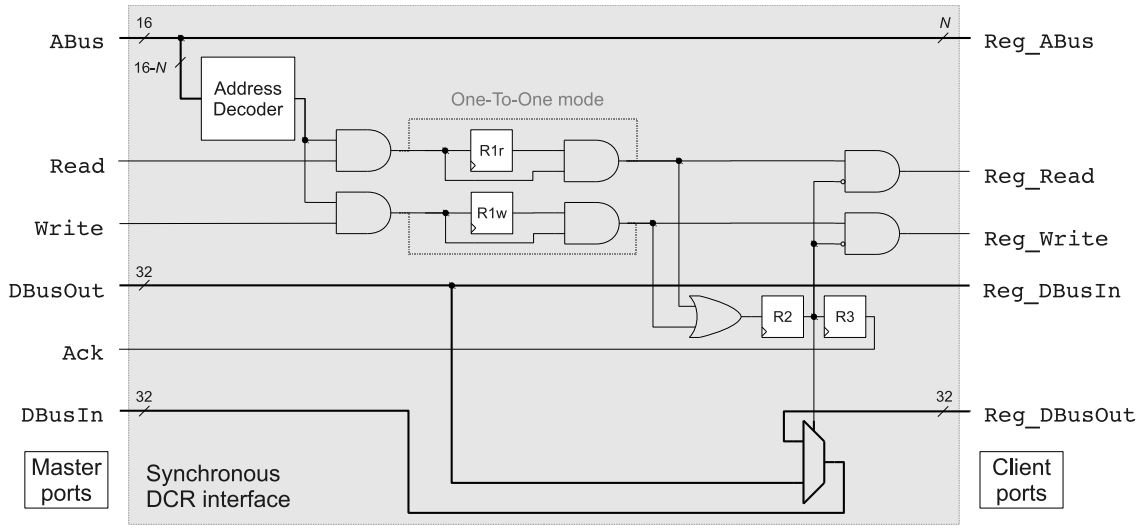

Figure 4.7: Block diagram for the synchronous DCR interface.

bus `DBusIn` only if R2 is asserted. Otherwise the master outbound data bus `DBusOut` is connected through. Multiplexing of the inbound master data bus `DBusIn` is necessary to support for the distributed OR topology of the DCR device tree. By definition the master outbound data bus `DBusOut` is deasserted on read transactions.

The synchronous DCR interface supports for One-To-One mode (`ONETOONEMODE`), cf. Ref. [54]. In this case the registers R1r and R1w are bypassed, effectively reducing the time t_2 within the handshake sequence Eq. (4.1) by one cycle. However, this feature is not used in the QPACE network processor. The timing diagram for subsequent write and read transactions from the DCR master to/from the synchronous DCR interface driven by the DCR master clock is shown in Fig. 4.8. Here One-To-One mode is disabled. The signal changes of the master inbound data bus `DBusIn` are due to the multiplexer that switches between the master outbound data bus `DBusOut` and the client outbound data bus `Reg_DBusOut`. On read transaction the DCR master deasserts the outbound data bus `DBusOut`. The timings for the handshake sequence are $t_2 = t_4 = 1$ cycle, which are determined by the DCR master. The synchronous DCR interface introduces asymmetric timings $t_1 = 3$ cycles and $t_3 = 2$ cycles, because the `Ack` signal is deasserted ahead of schedule. If the DCR client is clocked by the DCR master clock, then the DCR bus cycle time for the QPACE network processor is $t_c = 9$ cycles, because of the additional signal registration stages in the multi-master arbiter. If the DCR client clock is derived from the DCR master clock the bus cycle time varies.

4.4 Inbound Write Controller

The Inbound Write Controller (IWC) shown in Fig. 4.1 is attached to the inbound write GBIF slave interface. This interface grants access to data sent by the SPEs to the network processor. The IWC autonomously handles the proprietary GBIF interface and multiplexes data into one of six torus link transmit buffers, which are organized as FIFOs. As part of the critical data path for remote communication between adjacent Cell BEs the IWC was designed for low latency and high throughput. No flow control is required to sustain high data rates. The design goals for the IWC are summarized in the following:

- GBIF-to-torus transmit buffer latency of $\mathcal{O}(10)$ cycles at data rates up to $\mathcal{O}(6)$ GByte/s.
- Zero-latency back-to-back transfer of 128-byte fixed-size datagrams to the torus interface.

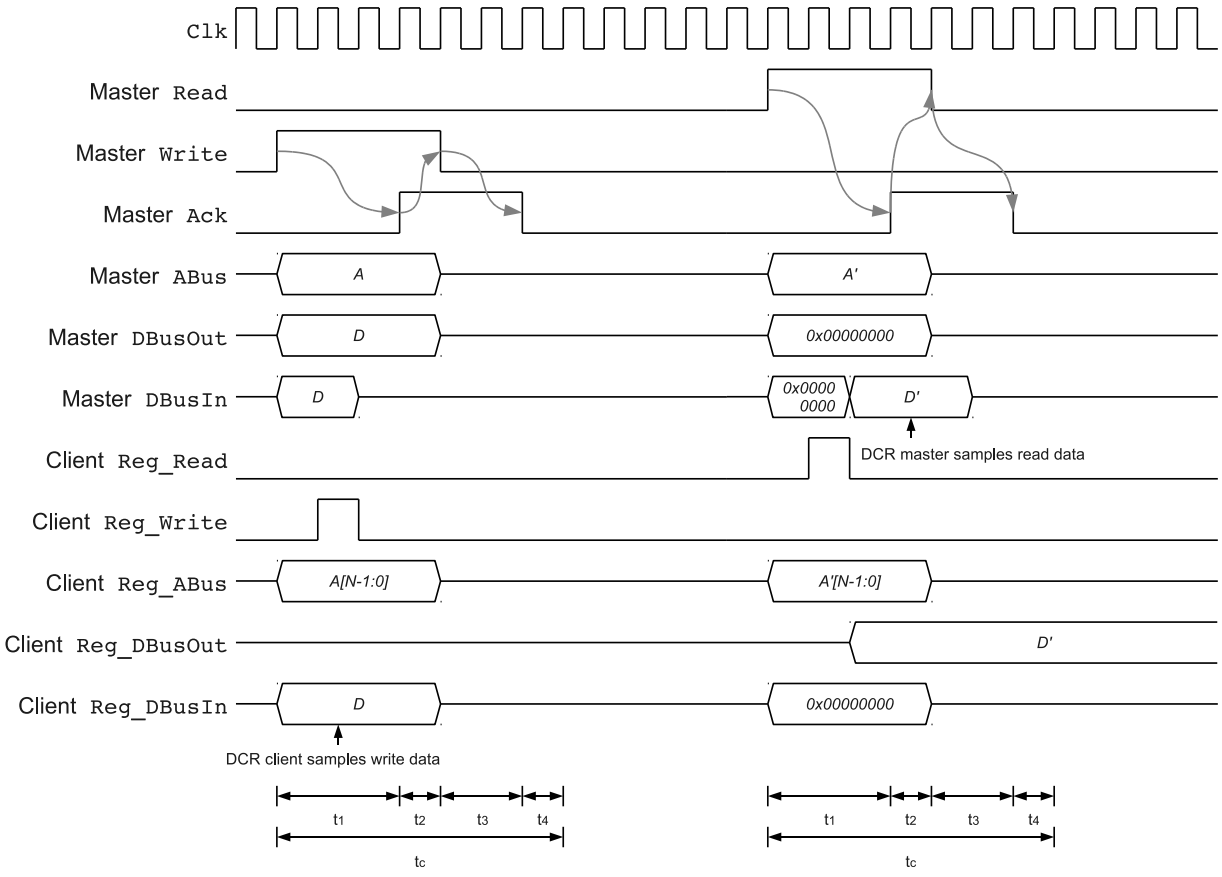

Figure 4.8: Timing diagram for subsequent write and read transaction to/from the synchronous DCR interface. Delays introduced by the DCR arbiter are not shown.

- Support for backpressure handling, in-flight transfer suspendable.
- One-pass address translation.
- Check for critical errors.

In principle the IWC is a simple controller unit that controls the GBIF protocol and passes application data to the torus network logic. However, due to the limited storage capacity of the torus transmit buffers there is no guarantee for sufficient amount of free buffer space for another datagram. Software polling for the buffer status before uploading data from the Cell BE to the network processor introduces prohibitively large latencies and thus is not an option. The IWC is not capable to maintain data. Therefore the strategy to reduce communication overhead and to avoid data loss due to buffer overrun is to suspend in-flight data transfer from the GBIF to the transmit buffers. In this case the inbound write data path is blocked, i.e., no other GBIF transactions can be handled by the IWC during suspension, until the buffer resources are available again and the datagram is transmitted to the torus transmit buffer. The blocking of the inbound write data path is uncritical for the application even if the amount of data uploaded onto the network processor exceeds the limited buffer space of the GBIF. Then the backpressure reaches the Cell BE and temporarily blocks the Cell BE-FPGA interface. In the worst-case scenario the backpressure triggers a checkstop due to timeout (and other limitations imposed by the FlexIO interface). However, such scenario is unlikely to happen in production runs and would indicate a serious flaw within the design of the high-speed

torus network data path. In fact, this critical situation was not observed in the recent design of the QPACE network processor.

4.4.1 Implementation

The IWC consists out of two logic parts. One is the IWC core logic and the other one is the IWC Extension Module (IWCEM). The IWCEM connects to the DCR device tree. It provides status information of the IWC, exceptional states, timeout setting, and supports for an optional instance of a data monitor for inbound write operations. The module is described in detail in Ref. [53]. In the following the IWC core logic is described. The ports of the torus logic and Extension Module of the IWC are listed in Table 4.6.

Signal	Direction	Width	Description
`TORUS_SEGMENT`	generic	5	Torus segment, inbound address [40:36]
`LINK_`n`_ADDR`	generic	6×3	Torus transmit buffer link n, inbound address [23:21]
`Clk`	in	1	Clock
`Reset`	in	1	Reset
`cntTime`	in	3	Timeout counter
`exc`	out	4	Exception vector
`almostFull`	in	6	Torus transmit buffer status
`addr`	out	42	Torus address
`data`	out	128	Torus data
`first`	out	1	Torus first data line
`we`	out	6	Torus write enable

Table 4.6: Ports declaration for the IWC. Only the ports for the torus and Extension Module interfaces are shown.

The block diagram for the IWC is shown in Fig. 4.9. The GBIF, IWC, and the interfaces of the torus network logic are clocked synchronous at 200 MHz in the recent FPGA design. The peak data rate of the 128-bit wide data port `data` of the IWC is 3.2 GByte/s. The IWC fragments data passed from the GBIF into blocks of 128 bytes and assigns a block start address `addr`. Each block consists of eight data lines. The first data line of each block is tagged by `first`. The ports `data`, `addr` and `first` are shared by the six torus interfaces. The status of each of the torus transmit buffers is provided by the `almostFull` vector (assigning link n to the element `almostFull[`n`]`). There is one link assigned to each element of the write enable vector `we`. On assertion of the write enable signal `we[`n`]` the signals `data`, `addr` and `first` are valid and must be handled by the torus logic of link n.

The high throughput of the IWC is achieved by a pipelined design. Pipelining is an efficient method to increase the performance of the logic by overlapping processing tasks. This technique is excessively used in microprocessor architectures to achieve a high level of parallelism, see e.g. Refs. [56, 57, 58] for detailed discussions on that topic. The design methodology is to divide combinational circuits into stages and to insert buffers – in this particular case registers – at proper places. This adoption of the design does not decrease the overall delay time introduced by the processing tasks, however, it significantly increases the throughput. The price to pay for the pipelined design is an increase of resource usage. Pipelined design and coding techniques using VHDL are discussed, e.g., in Refs. [59, 60]. Seven pipeline stages are introduced into the IWC core logic. The stages are shown in Fig. 4.9 and the flow of information follows the sequence

$$\mathrm{R0} \rightarrow \mathrm{R1} \rightarrow \mathrm{R2} \rightarrow \mathrm{C0} \rightarrow \mathrm{C1} \rightarrow \mathrm{P1} \rightarrow \mathrm{P2}\ .$$

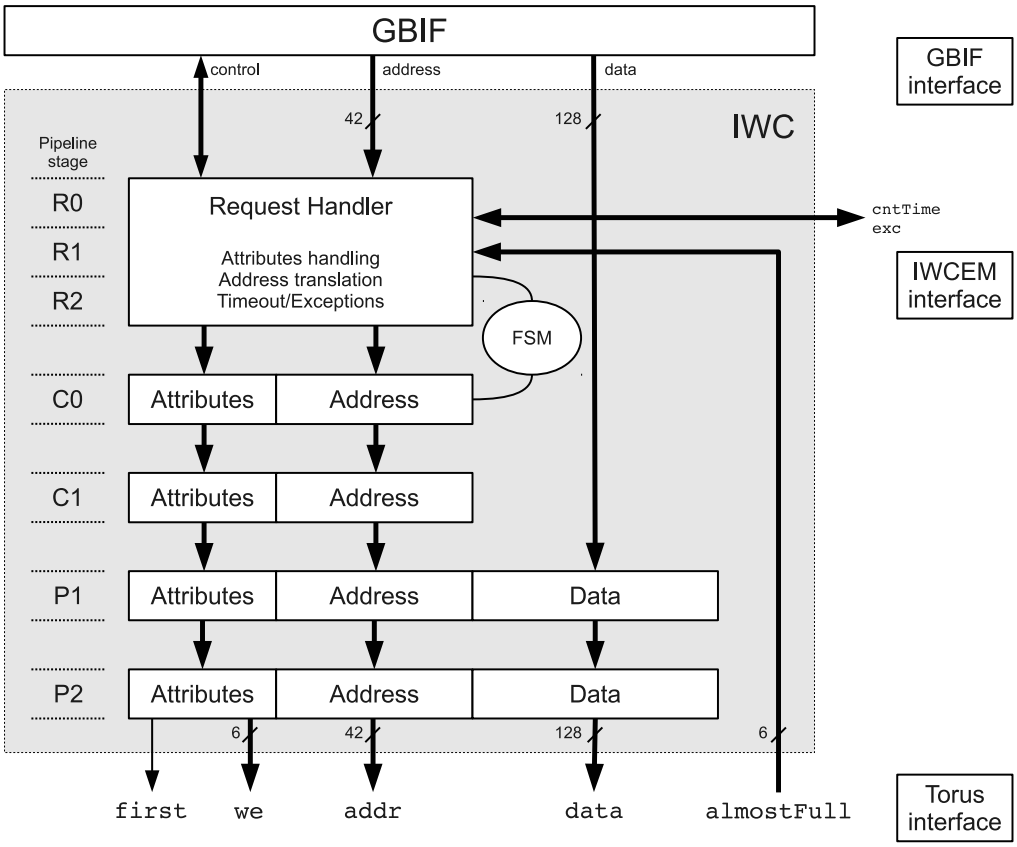

Figure 4.9: Block diagram for the IWC.

The transfer attributes and the address are registered in each pipeline stage S_i. If necessary data is processed at the transitions $S_i \rightarrow S_{i+1}$.

The pipelined design serves two major purposes. One is to achieve high data throughput. The other one is the adaption of the GBIF protocol according to the specifications of the torus transmit interface. The torus interface protocol is not compatible to the GBIF protocol. The torus interface requires the address and data line to be valid at the same clock cycle. This temporal alignment of the data is not supported by the GBIF interface and there is a delay between data and address. The delay is compensated for by the IWC logic by data registration in the pipeline stages P1 and P2 exclusively. The address line, however, is propagated along all sevens stages of the IWC pipeline.

The IWC supports for simultaneous handling of outstanding GBIF requests and data transfer. Outstanding requests are handled by the so-called Request Handler. The Request Handler is realized as a two-pass process in stages R0 and R1. The Request Handler performs several checks for illegal attributes passed by the GBIF prior to any data transfer, e.g., check for invalid data size, invalid address, and invalid control flow. The data size accepted by the Request Handler is restricted to multiples of 128 bytes. The address bits [40:36] must match the segment defined by `TORUS_SEGMENT` and the address bits [23:21] must match one of the six torus link addresses `LINK_`n`_ADDR` ($n = 0 \ldots 5$). Errors detected by the Request Handler are encoded in the exception vector `exc` which is passed to the IWCEM. The IWC exceptions are summarized in Table. 4.7. All IWC exceptions are compute-critical. In case of any exception the IWCEM indicates a critical interrupt to the Cell BE that forces any computation to stop. The `KILL` signal is propagated to all compute nodes within the partition via the global signals network.

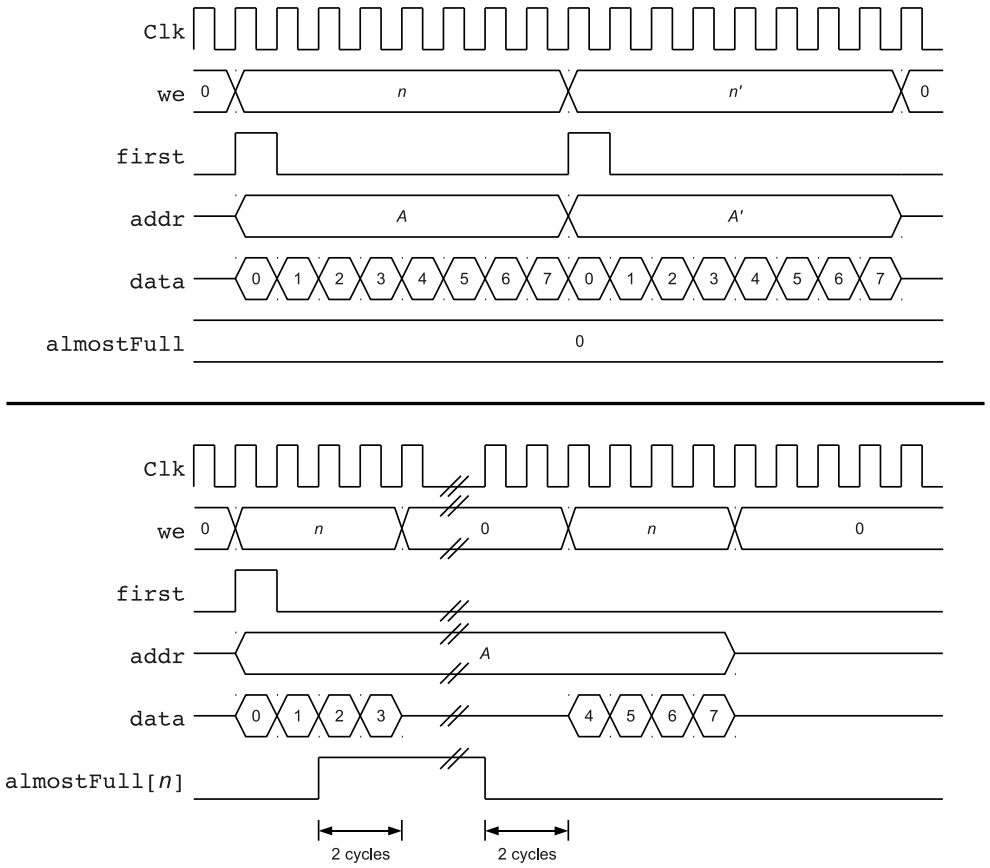

Figure 4.10: The upper panel shows the IWC timing diagram for back-to-back data transfer of two data blocks subsequently addressing the torus transmit buffers n and n'. The lower panel shows the IWC timing diagram for a single data block transfer addressing torus transmit buffer n with backpressure handling (here $N_l = 0$). Only the IWC torus interface is shown.

The Request Handler furthermore defines an one-hot encoded write enable vector that is associated with the torus link to be addressed. An initial design requirement on the IWC was the implementation of a one-pass address translation, supporting for remapping of the torus links during runtime. The address bits [23:21] define the target link. In order to support for address remapping the address bits serve as a pointer into a look-up table (LUT) that comprises six entries. The address bits are then replaced by the corresponding entry in the LUT, effectively remapping the link address. However, the recent implementation of the IWC does not support for dynamic address remapping. Instead, the identity map is implemented. Support for remapping is easily introduced into the logic by replacement of the `LINK_`n`_ADDR` constant declarations by registers accessible from the IWC Extension Module.

The size of the outstanding request, i.e., the number of data lines to be transferred to the torus transmit buffer, the corresponding address, and the write enable vector are permanently registered in pipeline stage R2 if no error has been detected by the Request Handler and no data transfer is yet in flight. The information is kept in the pipeline stage R2 until all data lines associated with the particular request are provided by the GBIF. Each time the pipeline stage R2 is updated a timeout counter started. The timeout counter runs at full IWC clock speed. The width of the timeput counter is adjustable to $24 +$ `cntTime` bits from the IWC Extension Module. If the timeout is reached the exception signal `rExc[0]` is asserted.

Signal	Description
`exc[0]`	Timeout
`exc[1]`	Invalid data size
`exc[2]`	Invalid address
`exc[3]`	GBIF control flow error

Table 4.7: Exceptional states supported by the IWC.

The write enable and address vectors are passed from pipeline stage R2 to C0. For each GBIF request larger than 128 bytes the address is incremented by 128 each 8 data lines – corresponding to 128 bytes – in pipeline stage C0. The first data line within each 128-byte block is tagged. The tag, write enable, and address are passed through the pipeline stages C1, P1, and P2 without further processing. The pipeline stage C1 introduces one delay cycle required for the correct timing of control information, address, and data. Data is registered exclusively in the pipeline stages P1 and P2. Data registration is introduced in order to relax routing constraints on the FPGA. The timing diagram for back-to-back data transfer with subsequently addressed torus transmit buffers n and n' is shown in the upper half of Fig. 4.10.

The IWC supports for backpressure of the torus transmit buffers. If the addressed transmit buffer n asserts the associated `almostFull[`n`]` signal the IWC suspends the data transfer until the signal is deasserted. On suspension the write enable vector is deasserted in pipeline stage C0, effectively de-addressing the torus transmit buffer. Due to the pipelined design of the IWC and the specification of the GBIF protocol it is not possible to suspend the transfer immediately after the assertion of `almostFull`. The number of data lines N transferred to the torus transmit buffer after assertion of `almostFull` is $N = 2+N_l$. Here $N_l = \mathcal{O}(1)$ cycle is a constant latency defined by the implementation of the Request Handler and the proprietary GBIF protocol. The torus transmit buffers were designed for appropriate assertion of `almostFull` in case of limited buffer space. The timing diagram for a data transfer of 128 bytes with backpressure handling and a single torus transmit buffer n addressed is shown in the lower half of Fig. 4.10.

The pipeline stages R0, C1, P1, and P2 are implemented as free-running stages, i.e., the signals are registered unconditioned. Thus no data is retained in any of these stages. This implementation offers a reset strategy for the IWC where only parts of the Request Handler and the pipeline stage R2 need to be reset, effectively reducing the number of reset lines to be distributed amongst the registers. Five clock cycles are required to flush the IWC pipeline.

4.4.2 State machine

The IWC is driven by the Mealy-type state machine shown in Fig. 4.11. The outputs of the state machine are defined by the state transitions. Two states are used: `IDLE` and `TRANS`. Data transfer is handled in state `TRANS` exclusively, otherwise the state machine remains in the wait state `IDLE`. Transitions depend on the signals `r1req` indicating a valid GBIF request, `enable` driving the transfer of a single data line, and on the data line counter `count`. The enable signal and the data line counter are registered outputs. The registers are fed by `nextEnable` and `nextCount`, respectively. Registration relaxes timing constraints on combinational logic at the cost of one delay cycle.

The `enable` signal is driven by the status of the torus transmit buffers. The signal is switched between two sources: (*i*) the signal `s1Full = almostFull[`n`]` associated with the transmit buffer n addressed when a new GBIF request is registered in pipeline stage R2, and (*ii*) the signal `s2Full = almostFull[`n'`]` associated with the transmit buffer n' for the transfer in flight. It is important to distinguish between the two sources, because only this implementation allows for backpressure handling of two link buffers n and n' addressed back-to-back without the introduction of additional

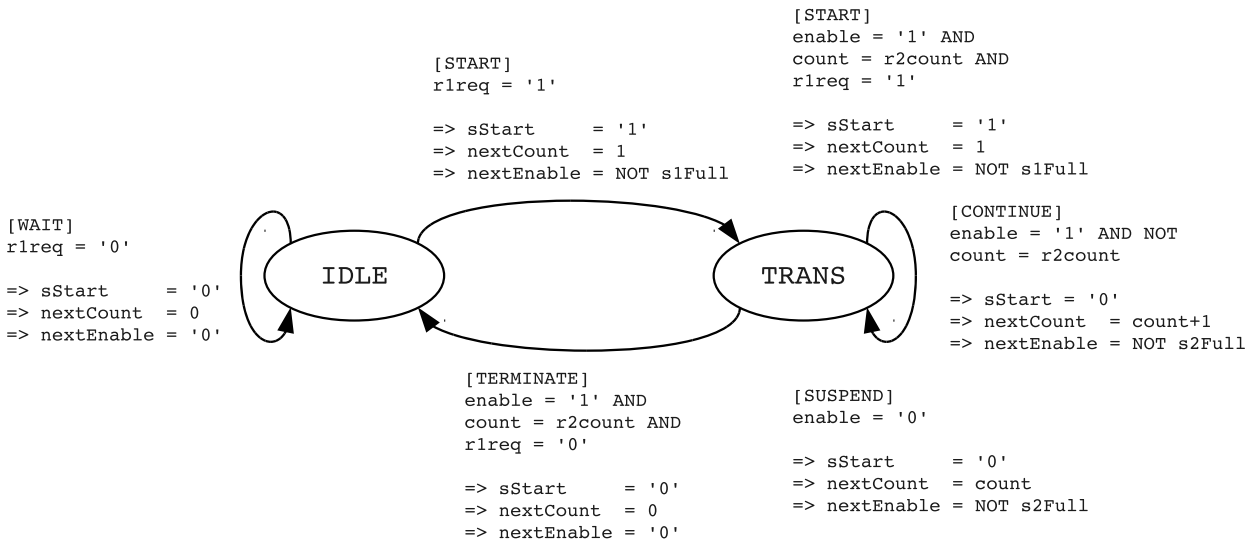

Figure 4.11: State diagram for the IWC.

wait states. The pulse signal `sStart` indicates that a new request from the GBIF is handled. The signal triggers registration of the new transfer attributes in the pipeline stage R2. The number of data lines associated with the new request is registered in `r2count`.

The state is `IDLE` if no GBIF request is outstanding and no data transfer is in flight. The WAIT transition resets all outputs of the state machine to zero, disabling any action of the IWC. If the Request Handler asserts `r1req` the state switches from `IDLE` to `TRANS` via the START transition. The signal `sStart` is asserted and the next-line counter `nextCount` is set to 1. The `nextEnable` signal is set to `NOT s1Full`, driving the `enable` register with the inverse of `almostFull`$[n]$.

Data transfer proceeds by the CONTINUE transition. The data line counter is increased by 1 and `NOT s2Full` is selected as source of the `enable` signal. If the line counter `count` matches `r2count` and no further GBIF request has been registered (`r1req = '0'`) the state machine is switched to `IDLE` state by the TERMINATE transition. If a request is registered (`r1req = '1'`) the state machine continues in state `TRANS` by the START transition. Similar to the `IDLE` state the signal `sStart` is asserted, the next line counter `nextCount` initialized, and `s1Full` is fed into the `enable` register.

The state machine handles backpressure of the torus transmit buffers by the SUSPEND transition in state `TRANS`. On deassertion of `enable`, data transfer is suspended until `s2Full` is deasserted. Data transfer continues by the CONTINUE transition.

4.5 Universal Asynchronous Receiver Transmitter

The Universal Asynchronous Receiver Transmitter (UART) is a standard component in computer technology used for low-speed serial data transfer. The UART is well established especially in the sector of microcontrollers because of its low pin count, low logic count, and high reliability. The UART (de)serializes data transferred along a single line per direction at baud rates typically up to $\mathcal{O}(1)$ MBd. On demand additional handshake signals are used for flow control. The data protocol is simple and the software stack for handling of the data transfer is minimal. The UART designed for the QPACE network processor has minimal compliancy to the PC16550D UART by National Semiconductor Ref. [61]:

- 16-byte receive and transmit buffer each.

- Preassigned sample rate 115200 Bd full duplex.
- Preassigned serial interface characteristics: 8 data bits, no parity, 1 stop bit (mode 8N1).
- Support for hardware handshake.
- Support for software polling and interrupt on receive.

There are two instances of logically identical UARTs embedded into the DCR device tree, see also Sect. 2.6.1 and Figs. 4.1 and 4.2:

1. Connection to the Service Processor, mandatory for controlling the FlexIO training during the boot sequence.
2. Connection to the root-card. Each Cell BE can be addressed either by the root-card microcontroller or by a RS-232 compliant device attached to the root-card's debug connector.

During the bring-up phase of QPACE the second UART was the only available connection to the Cell BE firmware and therefore played an important role before the Ethernet logic and Ethernet driver were completed. Now this UART can be used for monitoring and debugging.

4.5.1 Implementation

The UART top level entity is `uart_16550_dcr_top`. The top level implements other entities split amongst several VHDL sources. All entities are listed in Appendix A.1. The hierarchical schematic overview for the UART is shown in Fig. 4.12. The top level entity `uart_16550_dcr_top` serves as a wrapper for the synchronous DCR slave interface discussed in Sect. 4.3.5 and the UART core logic `uart_16550_top`. The ports of the UART core are listed in Table 4.8.

Signal	Direction	Width	Description
`BAUD_RATE`	generic	—	Baud rate definition
`CLK_FREQUENCY`	generic	—	Input clock frequency in Hertz
`Clk`	in	1	Clock
`Reset`	in	1	Reset
`Addr`	in	4	Input address
`DataIn`	in	8	Input data
`Read`	in	1	Read strobe signal
`Write`	in	1	Write strobe signal
`DataOut`	out	8	Output data
`RXD`	in	1	Received data
`nCTS`	in	1	Clear To Send
`TXD`	out	1	Transmitted data
`nRTS`	out	1	Request To Send
`IRQ`	out	1	Interrupt request

Table 4.8: Ports declaration for the UART.

The basic building blocks of the UART core are the receiver logic `rxunit` and the transmitter logic `txunit`. The input signal of the receiver is `RXD`. The output signal of the transmitter is `TXD`. Both signals are active low. `TXD` and `RXD` are connected crosswise for communication between two UART devices. If no data is transferred the data line is idle, `TXD = '1'`. Both the receiver and transmitter logic are driven by an internal sample rate generator with preassigned frequency `BAUD_RATE` (which generates so-called ticks). Simple state machines generate the UART data protocol. In addition to

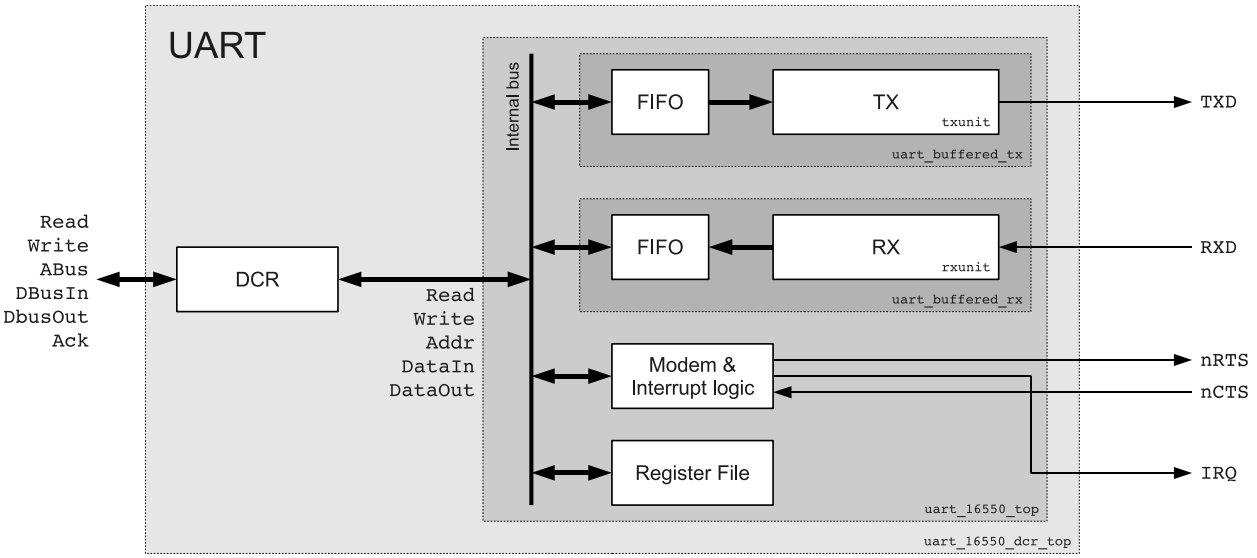

Figure 4.12: Hierarchical schematic diagram for the UART.

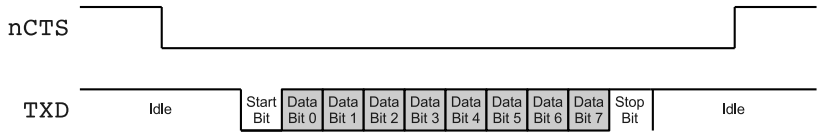

Figure 4.13: UART handshake and serial data transfer protocol in mode 8N1.

the baud rate the UART core clock frequency `CLK_FREQUENCY` must be defined in Hertz. The core clock also defines the synchronous DCR interface clock. The entity `uart_buffered_rx` comprises the receiver logic `rxunit` coupled to an 16-byte FIFO receive buffer. The entity `uart_buffered_tx` comprises the transmitter logic `txunit` coupled to an 16-byte FIFO transmit buffer. Each buffer is instantiated as block RAM FIFO. The implementation into block RAM shifts the registers required for character storage from logic blocks to the FPGA's hard core resources. The modem logic of the UART core autonomously handles the handshake signals Request To Send `nRTS` and Clear To Send `nCTS`. The handshake signals are active low. The signals `nRTS` and `nCTS` have to be connected crosswise for communication between two UART devices. If the receive buffer is not full `nRTS` is asserted (`'0'`), requesting for data transfer. If `nCTS` is asserted (`'0'`) the transmitter is allowed to send data. Support for hardware handling of the handshake signal `nRTS` can be turned off on demand. The interrupt logic is capable of generating an interrupt request on receive asserting the signal `IRQ`.

The UART serial data transfer protocol is shown in Fig. 4.13. Data is transmitted in frames. Each frame is initiated by a leading start bit `TXD = '0'`, followed by eight data bits, and a final stop bit `TXD = '1'`. Parity is not supported by the implementation of the UART. Data is transferred from LSB first to MSB last. The state diagram for the UART transmitter is shown in Fig. 4.14. If no data is available for transfer the `send` signal is deasserted, the state is `IDLE`, and the transmit data line is `TXD = '1'`. If `send` is asserted the transmit operation starts in state `START` transmitting the start bit `TXD = '0'`. Data is transmitted in the states `S`n. The transmit data line is driven by `TXD = data[`n`]`. The states `S`n are successively switched from $n = 0$ to $n = 7$, followed by the state `STOP`. In the state `STOP` the stop bit `TXD = '1'` is transmitted on the transmit data line. If additional data is available a new character is transmitted in state `START`, otherwise the state is

IDLE. The receiver logic is comparably simple: if the receive data line RXD asserts ('0') the start bit, eight data bits, and the final stop bit are sampled. See also Ref. [59] for a textbook description of the UART receiver logic.

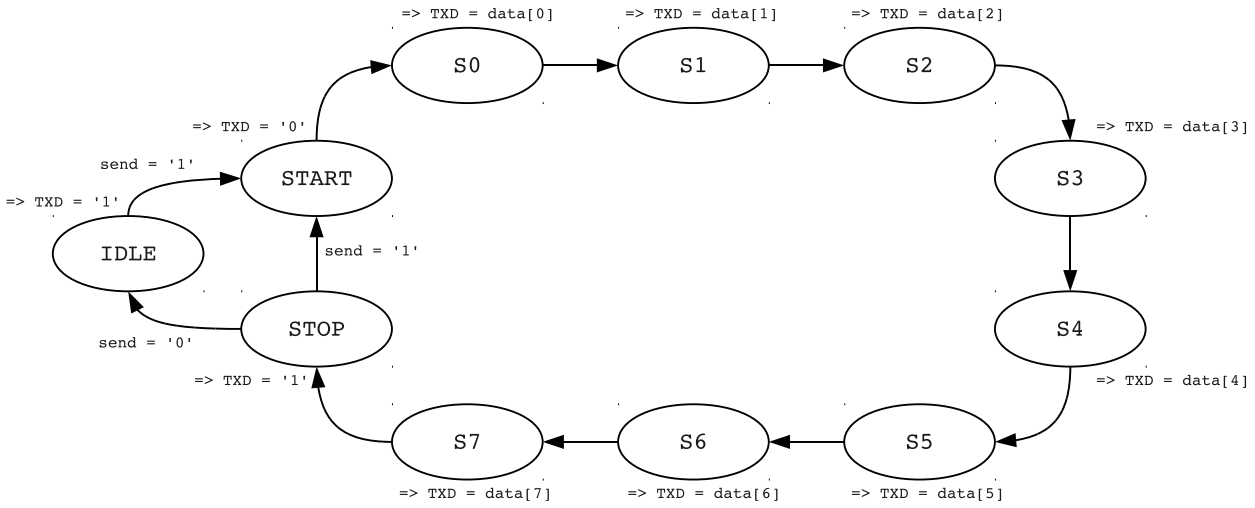

Figure 4.14: State diagram for the UART transmitter logic operating in mode 8N1.

4.5.2 Register file and modes of operation

The UART register file listed in Table 4.9 mimics the register file of the PC16550D UART Ref. [61]. Upon reset all registers are set to '0', except for the Mode Control Register MCR[5] = '1' indicating hardware support for the handshake signal nRTS.

The data ready bit in the Line Status Register LSR[0] is asserted if the 16-byte receive buffer maintains at least one character. Received data is collected at the Receiver Buffer Register RX[7:0]. If LSR[0] = '0' and data is read from RX the data is invalid. Data is lost if the number of bytes received exceeds the capacity of the receive buffer.

Data is transmitted by a write to the Transmitter Buffer Register TX[7:0] as long as the 16-byte transmit buffer offers the capacity to store another character. If the transmit buffer is empty LSR[6] is asserted. If the transmit buffer is full LSR[5] is asserted. Data written to the transmit buffer if LSR[5] is asserted causes data loss. Data transfer is autonomously handles by the UART transmitter if the Clear To Send signal nCTS is asserted. The inverse status of the nCTS signal can be read from the Mode Status Register MSR[4].

The UART supports for interrupt request on receive. If the Interrupt Enable Register is defined IER[0] = '1' an interrupt signal is asserted if the data ready bit LSR[0] toggles from '0' to '1'. On read the Interrupt Identification Register IID[0] returns '0' if no interrupt is pending and '1' if an interrupt is pending. The interrupt is acknowledged by a dummy write operation to the IID.

Software polling of the receive buffer is supported if IER[0] = '0'. The data ready bit LSR[0] can be polled for the receive buffer status. Switching between interrupt support IER[0] = '1' and software polling IER[0] = '0' can be performed at any time.

The hardware handshake is used if the Mode Control Register MCR[5] = '1'. In this case the Ready To Send signal nRTS is automatically handled if the receive buffer has capacity to maintain another character. In manual mode MCR[5] = '0' the nRTS signal is controlled by MCR[1]. Then '1' activates and '0' deactivates the handshake signal.

Register name	Address	Mode	Register description
Receiver Buffer Register (RX)	0x0	R	RX[7:0]: 1 byte received data
Transmitter Buffer Register (TX)	0x0	W	TX[7:0]: 1 byte transmit data
Interrupt Enable Register (IER)	0x1	R/W	IER[0]: interrupt mode
Interrupt Identification Register (IID)	0x2	R	IID[0]: IRQ pending
		W	IRQ acknowledge
Mode Control Register (MCR)	0x4	R/W	MCR[1]: RTS if MCR[5] = '0'
		R/W	MCR[5]: handshake mode
Line Status Register (LSR)	0x5	R	LSR[0]: data ready
		R	LSR[5]: transmit buffer full
		R	LSR[6]: transmit buffer empty
Mode Status Register (MSR)	0x6	R	MSR[4]: CTS

Table 4.9: UART register file.

Chapter 5

System Verification and Software Tools

5.1 System tests

The development of QPACE followed a tight timeline. The project officially started in January 2008. First versions of the node-card, root-card, and backplane became available already in the summer of 2008. A small test system was deployed at the IBM laboratory in Böblingen, Germany, after the bring-up phase. A photograph of this test setup is shown in Fig. 5.1. Intensive hardware integration tests started at the end of 2008 with the goal to identify and remove any flaws in the design before the release for large-scale manufacturing in the first quarter of 2009. With the availability of all components in the final design, large-scale integration and burn-in tests were carried out in the lab before deployment at the destination sites, see also Fig. 5.1. In August 2009 the deployment of QPACE at the Jülich Supercomputing Centre and the University of Wuppertal had been completed and a test operation phase was started.

The development of the hardware and system-relevant software was accompanied by a series of functional tests. The following items give a rough overview on the diversity of the tests applied (without requirement on completeness and no respect to the timeline):

- All components: mechanical and electrical inspections.
- Cooling: temperature monitoring, pressure tests, and check of the water channels inside the coldplate.
- Power management: PSU and PSU adapter-card tests, functional tests of PSU voltage management and load balancing, PSU management via superroot-card.
- Superroot-cards: power, clock, FMod-TCP and CPLD tests; management by software libraries.
- Root-cards: booting tests, flash memory, CPLD, reset lines and clock tree tests; tests of the serial links, including the I^2C, SPI, and RS-232 interfaces; management by software libraries.
- Node-cards: power and temperature, booting tests, FlexIO, and flash memory tests; functional tests of the Ethernet PHY and PMC Sierra PHYs; network processor logic tests; Cell BE and Service Processor communication tests; stress tests of the Cell BE processor and main memory.
- Tests of Ethernet communication, including switches, filesystem, and front-end system; tests of the torus network and global signals network.
- Concurrent tests, e.g., Cell BE and main memory stress tests in combination with torus network and Ethernet network communication.

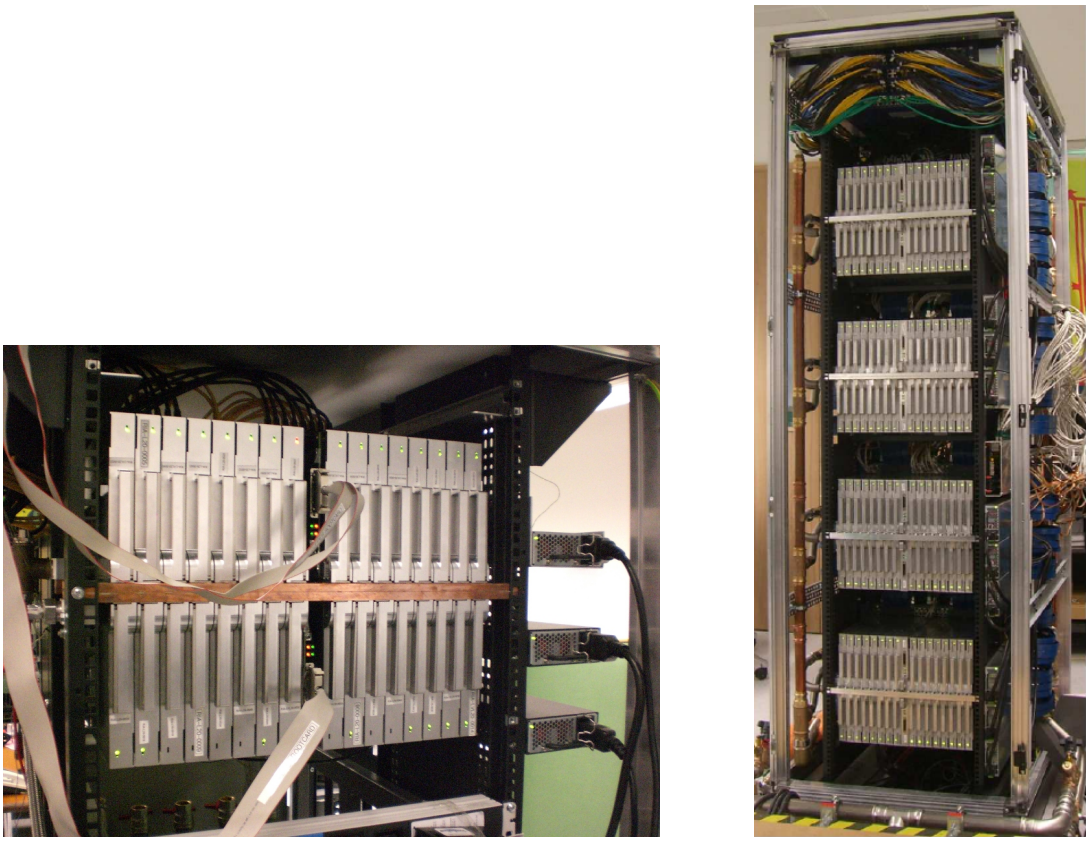

Figure 5.1: The photograph on the left shows the QPACE test setup at the IBM laboratory in Böblingen, Germany. The photograph on the right shows a fully populated and cabled rack tested in the lab.

5.2 Software tests

A tremendous number of software tools were designed for verification of the QPACE design. As a prerequisite to large-scale testing, management libraries for the root-card and superroot-card were developed, see also Appendix A.3. Hardware discovery tools were designed on top of these libraries that allow for monitoring and management of the system status, e.g., voltage control, temperature measurements, components configuration, and post-mortem analysis. These tools served as templates for the design of the QPACE Front-end Client (QFC), a software tool that unifies the access to the machine. The QFC is discussed in more detail in Sect. 5.4.

The diversity of the software tests developed for the system ranges from simple setups such as automated verification of a ping via Ethernet from the test server to the node-card, up to complex stress tests required, e.g., for verification of nearest-neighbour communication via the torus network. The challenge in the design of test software was not only to deploy a stack of specialized programs that probe for a certain kind of system functionality. The goal was also the automatization of the verification of a system that consists of hundreds of components. As an example for the strategy applied consider a simple test setup, e.g., a processor stress test for the Cell BE that is performed by some executable. In this particular case no communication between the node-cards is required and execution of the stress test program is trivially parallelized on an arbitrary number of nodes. A practical approach for test automation in this case is to separate the stress test program and the test execution. A so-called test wrapper script is a useful piece of software that autonomously sets up the stress test program on the nodes, executes the stress test with a given set of parameters, analyzes

the output, and finally summarizes the result. The following items describe the basic structure of such a software test carried out on a single node-card:

1. Start of the test wrapper script on the test server with exact definition of the test setup, i.e., the node-card setup, specific test parameters etc. The wrapper performs a preliminary check of the sub-system, e.g., a test of the boot status of the node-card.

2. The wrapper script copies the test executable from the test server onto the node-card.

3. The wrapper script logs onto the node-card using secure shell, pre-configures the system and collects relevant system information. Then the test program is executed. To minimize the risk of information loss in case of errors the output of the test program is stored locally on the node-card itself, on the test server, and additionally on the network filesystem.

4. Once the test has finished, system information is collected and the text output (logged locally on the test server) is analyzed. The test results are compared with the test definitions, i.e., the test has passed or failed according to the requirements.

5. The test is considered to be successful only if the node-card passed the test, and has failed otherwise. A test summary is displayed.

Test execution on a setup comprising hundreds of nodes requires as many steps as possible to be executed in parallel. For most tests the basic structure of the items stated above were easy to modify and parallelize. Examples for such tests are simple Cell BE floating-point stress tests and main memory stress tests. More complicated test setups that essentially rely on the same structure of the test wrapper are, among many more, test cases designed for verification (and also optimization) of the Ethernet network and the torus network functionality. The test executables were either custom designed or provided by IBM. The test wrapper scripts were written in scripting languages such as Python or Perl. Reference to a subset of tests developed for verification of the node-cards, root-cards, and superroot-cards is provided in Appendix A.1. A closer look on a software tool especially designed for booting of the node-cards is taken in the next section.

5.3 Booting the node-cards: `ncBoot`

A well established command line tool that reliably boots the node-cards into Linux is `ncBoot`. The tool was developed during the bring-up phase of QPACE and is capable of booting an arbitrary number of node-cards into the Linux OS. It was designed primarily for dressing of the nodes prior to large-scale tests and therefore comes with a series of options for configuration. `ncBoot` interacts with the Service Processor of each node-card via the root-card.

5.3.1 Implementation

The `ncBoot` tool is based on a three-phase sequence that is passed sequentially for each node-card during the boot process:

1. Retrieve status

   Check for node-card presence via SPI; read serial number, MAC address, boot status, status events, and clock setting.

2. Define setting

   Contingent hard-power off; set clock source, Service Processor real-time clock; contingent set MAC address; optional reset of error states.

3. Boot into Linux

   Hard-power cycle and monitoring of the boot process if requested.

In phase 1 the backplane slot is checked for presence of a node-card using the SPI interface. If a node-card is present relevant status and settings information is collected. In phase 2 the node-card is hard-powered off if the clock setting has to be adjusted.[1] Subsequently the real-time clock of the Service Processor is synchronized with the master server. The MAC address is adjusted if a faulty entry in the VPD is detected. Optionally the error status is reset. The node-card is ready for booting into Linux if the phases 1 and 2 have been passed successfully. This process is started in phase 3 and the corresponding state machine is described in Sect. 5.3.3.

`ncBoot` relies on the front-end libraries *feenlib* and *witchlib* (see Appendix A.3) and sets up a multi-threaded environment that allows for concurrent access to an arbitrary number of root-cards. One problem that arises in the boot process is the network traffic between the node-cards and the front-end system, which stores the Linux image for the nodes. Transfer of the Linux image from the master server to the node-cards relies on multicast TFTP. The number of root-cards accessed in parallel is limited to 16 in phase 3 exclusively due to the heavy network traffic during the boot process. If the number of root-cards addressed by `ncBoot` exceeds this limit the boot process is partially serialized and blocks of up to 16 root-cards are accessed in parallel.

5.3.2 Command line interface

`ncBoot` is run via

```
ncBoot -r <list> -b <list> -n <list>|-m <mask>|-a [-c <C>|-p <P>|-D <D>|-soeft]
```

Mandatory parameters are the list of racks `-r <list>` (0–3), backplanes `-b <list>` (0–7), and the addressed node-cards `-n <list>` (0–31). The `<list>` parameter supports for single assignments separated by a coma, e.g., `0,1,2`, or range indicated by a dash, e.g., `0-4`. Both kinds of assignments can be combined, e.g., `0,1,10-15,20,25-31`. Instead of a list of node-cards a hexadecimal mask, e.g., `0xff00`, can be supplied using option `-m`. Alternatively all node-cards connected to the backplane are assigned if the option `-a` is supplied.

Optional parameters are

`-c <C>`	set clock source to onboard clock (`<C>` = 0) or global clock (`<C>` = 1)
`-p <P>`	boot into phase `<P>` = 1, 2, 3 (default `<P>` = 1)
`-D <D>`	debug output level `<D>` = 0, 1, 2, 3 (default `<C>` = 0)
`-o`	enforce power off in phase 2
`-e`	invalidate error states in phase 2
`-s`	use relaxed power cycle passed criteria in phase 3
`-f`	enforce power cycle in phase 3
`-t`	probe node-card Ethernet connection via `ssh` login after phase 3

`ncBoot` offers detailed debug information for each of the three phases. Debug level 0 omits any debugging output. Debug level 1 prints out all interactions with the Service Processor. Debug level 2 displays the cycling of the state machines and level 3 additionally prints out the state machine histories.

[1] Hard-power off only affects the Cell BE and FPGA, but not the Service Processor.

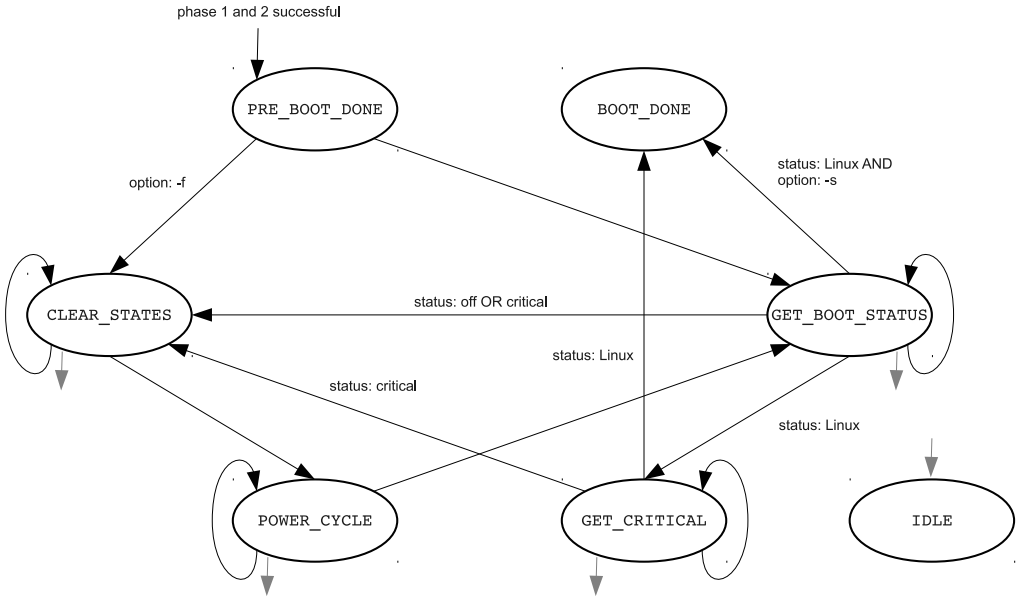

Figure 5.2: State diagram for the boot process run by `ncBoot` in phase 3.

5.3.3 State machine

The control flow of the three phases run by `ncBoot` is defined by state machines associated with each node-card. The state diagram for the third phase is shown in Fig. 5.2 and is described in the following.

If the phases 1 and 2 have been passed successfully the initial state is `PRE_BOOT_DONE` and `IDLE` otherwise. If a prior power cycle is enforced (option `-f`) the state `CLEAR_STATES` is entered. Otherwise the state is `GET_BOOT_STATUS`. In the state `GET_BOOT_STATUS` the node-card boot status is continuously monitored until a change of the boot status. On success, i.e., boot status "Linux", the card has been booted successfully into the OS and the state `GET_CRITICAL` is entered if the option `-s` is supplied. Otherwise the state is switched to `BOOT_DONE`. If the node-card is off or a critical error occured then the next state is `CLEAR_STATES`. In the state `CLEAR_STATES` the error status flag is invalidated. Invalidation allows for the registration of the error status associated with the subsequent power cycle. The state `CLEAR_STATES` is followed by the state `POWER_CYCLE` unconditioned. In the state `POWER_CYCLE` a hard-power cycle is initiated. The following state is `GET_BOOT_STATUS` unconditioned. In the state `GET_CRITICAL` the error status is evaluated. If the status is critical then `CLEAR_STATES` is entered. If the node-card has been booted into Linux successfully the final state is `BOOT_DONE`.

The boot process is considered successful only if the state `BOOT_DONE` is reached. If the boot process fails within five power cycles, or communication with the Service Processor fails subsequently for five times, then the boot process is stopped and the `IDLE` state is entered. No further action is performed if the state is `IDLE` – the boot process has ultimately failed. The limited number of retries for power cycles and Service Processor communication prevents deadlocks.

5.4 QPACE Front-end Client

The QPACE supercomputer comprises unique system components that are interconnected non-trivially. Each component provides its own sub-device structure and data paths, cf. Sect. 2.6.1 and 2.6.2. The addressing of relevant registers and block memory structures is a demanding task,

even for those who have been working for years on the success of the project. The asymmetry of the component structures and data paths, and also the large number of registers and memory blocks, render a unified access to the machine indispensable. The QPACE Front-end Client (QFC) unifies the most common administrative tasks into one single command line tool. It allows for comfortable control over hundreds of node-cards and dozens of power supply units (PSU). The QFC reliably retrieves and manages the machine status without the need for detailed knowledge of the underlying hardware and software architecture. All the details are hidden from the user. The QFC provides the following features:

- Unified access to all node-cards, root-cards, and PSUs.
- Reliable machine management with multi-user support.
- User-friendly command line interface.
- Output displayed with unique identifiers and key-value pairs.

The QFC is installed on the master servers in Wuppertal and Jülich. The client requires the Front-end Utilities for QPACE Daemon (FUQD) to be run in the background.[2]

5.4.1 Implementation

QPACE is supervised by Nagios automation and monitoring services [62]. One issue that arises in the administration of QPACE is the partial lack of multi-user support. Race conditions do occur when multiple users access some of the data paths at the same time. One design goal of the QFC was to disentangle hardware-specific dependencies such that multiple instances of the QFC can be operated concurrently.

Most of the issues with respect to multi-user support are overcome by the design of the utilities daemon FUQD. The utilities daemon is stateless and provides a proprietary interface with support for remote procedure calls (RPC). The RPC functionality includes a set of low-level atomic functions that allow for remote actions on the root- and superroot-cards exclusively. Multi-user support relies on a locking mechanism. Access is granted only if no lock has been acquired yet for the components to be addressed. User-defined operations are performed only if the lock is granted. The lock is released at the end of the operation, which may consist of multiple RPC requests. Deadlocks of the locking mechanism are prevented by a timeout mechanism.

The QFC implements a set of high-level functions called “actions” that are requested to the QFC by the user. Examples for such actions are generic status requests, FPGA bitstream updates, and power management. The QFC breaks down the action into stacks of low-level atomic operations supported by the FUQD. Each stack is guaranteed to avoid read-after-write and write-after-write dependencies. The stacks are communicated sequentially to the FUQD via the RPC interface. Almost all actions supported by the QFC initially lock the components, then perform all the necessary operations and finally unlock the components. However, this procedure is inadequate for flash processes of the node-card via the root-card. The update of the node-card’s flash memory is a rather slow operation that potentially annules concurrent requests to other node-cards attached to the same root-card. Therefore the QFC relies on a dedicated flash engine that avoids this issue. The root-card is locked only to perform an update of a single sector in the flash memory (64 kB). After each update the root-card is unlocked. The procedure is repeated until all sectors have been updated. This method allows other instances of the QFC, e.g., scheduled by the automation software, to access the root-card during the flash process.

[2] FUQD was written by S. Solbrig.

By far not all the functionality necessary for administration of QPACE is provided by the utilities daemon, e.g., several actions require access to the Linux shell on the node-card. Support for such operations is integrated into the QFC by a dedicated multi-threading engine. The QFC actions based on this engine do not support the locking mechanism provided by the daemon. There was, however, no flaw observed in this choice for the design of the QFC.

5.4.2 Command line interface

All QFC actions are performed via the command line interface using

```
qc [options] [<mode>] <target> <action> [parameters]
```

Each action is defined by the 3-tuple `mode`, `target` and `action`. The command line interface of the QFC supports for command completion, i.e., any action uniquely identified by `target` and `action` does not require the `mode` identifier. Nevertheless it is accepted. Depending on the action to be performed additional parameters are required.

The QFC supports for the following options:

`-h`, `--help`	displays the QFC quick reference
`-g`, `--grep`	the output is optimized for regular expressions parsing
`-s`, `--short`	the output is shortened at the cost of readability, implies option `--grep`

The `mode` identifier defines the mode of operation and accepts the following key words:

`get`	retrieve (status) information from the node-cards and PSUs
`set`	write data to the node-cards
`power`	all power-related actions on the node-cards and PSUs
`clear`	clear error states on the node-cards
`flash`	flash actions performed on the node-cards

The `target` identifier defines the target device of the action. Allowed targets are `node` and `psu`, addressing node-cards and PSUs, respectively. The target identifier has to be followed by a `<list>` parameter which determines the components to be selected. The format of the list supports for backplane-centric, rack-centric and combined assignments of the targets. In the backplane-centric metric each component of the 4-rack installation is addressed relative to one of 32 backplanes (BP), labeled 0–31. In the rack-centric metric each component is addressed by the rack number (0–3) and the backplane in the rack (0–7). In both metrics there are 32 node-cards (0–31) and three PSUs (0–2) assigned to each backplane. The `<list>` parameter supports for the following forms of addressing:

- Backplane-centric assignment (2 forms):

  1. `<list>` = [list of BPs]
  2. `<list>` = [list of BPs]`:`[list of nodes/PSUs]

  Form 1 addresses all nodes/PSUs assigned to the backplanes. Specific backplanes are selected by separation of the backplane label by a coma, e.g., `0,1,5,30`, or as a range defined by a dash, e.g., `0-4`. Both variants can be combined, e.g., `0,1,10-15,20,25-31`. Form 2 allows to address specific nodes/PSUs attached to the backplanes, e.g., `5-7:0,1` selects the nodes/PSUs 0 and 1 assigned to backplanes 5, 6 and 7.

- Rack-centric assignment of targets (3 forms):

1. `<list>` = `r`[list of racks]
2. `<list>` = `r`[list of racks]:[list of BPs]
3. `<list>` = `r`[list of racks]:[list of BPs]:[list of nodes/PSUs]

Form 1 addresses all backplanes and nodes/PSUs assigned to the racks, e.g., `r0,2-3` addresses all components of the racks 0, 2 and 3. Form 2 allows for selection of all components attached to the backplanes in the racks, while the third form furthermore allows to address specific nodes/PSUs.

- Combined assignment of targets: `<list>` = [list]`+`[list]`+`...

  Multiple backplane- and rack-centric assignments are combined into a single list using `+` as a list separator, e.g., `r0,1:0-3,7:0-2+8+r3`.

Table 5.1 provides a quick reference to all actions supported by the QFC. The number of mandatory parameters is indicated in the column P: (–) no parameters, (n) n parameters and (n+) at least n parameters. The full reference to the QFC is provided in Appendix A.4.

Note: The QFC is not aware of any partitioning of the system amongst the users. Mal-operation or mal-addressing of nodes within partitions of different users will not be recognized.

5.4.3 Error handling

By design the QFC does not quietly terminate if operations fail to succeed. Such failures include, e.g., runtime errors and specific errors propagated by the utilities daemon. Two kinds of errors are being intercepted:

1. *Global errors*, e.g., errors that prevent the startup of any operation of the engines, no connection to FUQD, file errors and unknown actions.

2. *Local errors*, e.g., no connection to node-card, root-card, superroot-card or PSU; error during execution of some stack of low-level commands.

All errors are displayed with a message string. Error messages are displayed with a leading identifier `GLOBAL ERRORS` if a global error was detected. Global errors are critical and force the QFC to terminate operation. On local errors the QFC action is continued exclusively on those targets not affected by the error. Local errors are displayed with a leading identifier `LOCAL ERRORS`.

The QFC allows for access to an arbitrary number of system components, in the QFC nomenclature referred to as "targets". After completion of any QFC action the following output displayed:

`success> [`N/M`]`	action completed on N out of M targets
`succlist> <list>`	action completed on list of targets (only if $N > 0$)
`failures> [`N/M`]`	action failed on N out of M targets
`faillist> <list>`	action failed on list of targets (only if $N > 0$)

The targets are summarized in the lists `succlist` (success) and `faillist` (failure). The list format allows for direct reuse of the string by a new instance of the QFC.

<mode>	<target>		<action>	P	Description
get	node	<list>	status	–	retrieve status
			config	–	retrieve configuration
			version	–	retrieve version of images
			serial	–	retrieve serial number
			temp	–	retrieve temperature
			mac	–	retrieve MAC address
			dump	–	retrieve FIR dump
			vpd	2	read from VPD
	psu	<list>	status	–	retrieve status
set	node	<list>	clock	1	set local or global clock
			vpd	2+	write to VPD
			mac	–/6	set MAC address
power	node	<list>	off	–	hard-power off via Service Processor
			on	–	hard-power on via Service Processor
			cycle	–	hard-power cycle via Service Processor
			sp	–	hard-power cycle the Service Processor
			noboot	–	hard-power on w/o booting via Service Proc.
			reboot	–	soft-power cycle via Linux `reboot`
			halt	–	soft-power down via Linux `halt`
			reset	–	hard-reset node-card via reset lines
			probe	–	test login via secure shell
			magic	–	try to boot into Linux automatically
			woof	–	like `magic`, but do a link stress, too
	psu	<list>	off	–	power off
			on	–	power on
clear	node	<list>	states	–	clear error states
flash	node	<list>	linux	1	flash FPGA/SLOF image via secure copy
			fpga	1	flash FPGA image via root-card
			slof	1	flash SLOF image via root-card
			sp	1	flash Service Processor image via root-card

Table 5.1: QFC quick reference.

Part II

Applications of Random Matrix Theory in Two-Colour Quantum Chromodynamics

Chapter 6

Quantum Chromodynamics

6.1 Introduction

Quantum Chromodynamics (QCD) is the theory of the strong interaction. QCD pictures the world of six massive particles, the quarks, interacting with each other by the mediators of the strong force, the gluons. Quarks and gluons carry so-called colour charge, and the picture drawn by ordinary QCD is composed of three colours. However, due to confinement isolated quarks have never been observed in nature, and all snapshots of strongly interacting particles taken in experiments appear colour-neutral. In low-energy reactions quarks and gluons interact strongly, and the QCD coupling constant becomes large at small momentum transfer. Therefore perturbative methods cannot be applied in this domain. However, non-perturbative phenomena of QCD have become accessible by computer simulations of the theory on a space-time lattice. Recently the mass spectrum of light hadrons has been successfully predicted by ab-initio simulations of QCD on the lattice [63].

Pions are the lightest particles of the hadron spectrum. These particles are the pseudo-Goldstone bosons of the theory of the strong interaction which arise from the spontaneous breaking of chiral symmetry. Chiral symmetry is only an approximate symmetry of QCD because the lightest quarks have non-zero mass, and therefore the pions are massive. The dynamics of these particles is described by a low-energy effective theory. In a specific finite-volume regime, the so-called ε-regime, a precise mapping can be made between observables calculated on the lattice, the partition function of pion effective theory, and chiral random matrix theory (chRMT).

Chiral random matrix theory was formulated in the 90's. In the Hermitian formulation of chRMT the matrix elements of the (anti-)Hermitian Dirac operator are modelled by independently distributed random variables and only global symmetries of QCD are taken into account. The random matrix approach predicts universal correlations in the eigenvalues within the deep infrared spectrum of the Dirac operator in the phase of broken chiral symmetry. Moreover, chRMT gives an exact expression for the microscopic limit of the spectral density of the Dirac operator. The microscopic spectral density is a universal function that allows to determine the order parameter of chiral symmetry breaking, the chiral condensate, from simulations of QCD on the lattice. The applicability of chRMT is not limited to ordinary QCD with gauge group SU(3). In fact, three random matrix ensembles exist which apply to different formulations of QCD [64]: the chiral orthogonal ensemble applies to QCD with gauge group SU(2) and fermions in the fundamental representation [65], the chiral unitary ensemble applies to gauge groups SU(N_c) with $N_c \geq 3$ colours and fermions in the fundamental representation [66], and the chiral symplectic ensemble applies to gauge groups for all N_c in the adjoint representation [67]. The applicability of Hermitian chRMT has been successfully verified by various studies of QCD and QCD-like theories on the lattice, e.g., [68, 69, 70].

In this study the focus is on chiral random matrix theory and its applications in QCD with gauge group SU(2) and fermions in the fundamental representation. Recently the non-Hermitian

extension to the chiral orthogonal ensemble has been solved, and an expression for the associated microscopic spectral density has been derived [71, 72]. A single symmetry-breaking parameter is introduced into the random matrix model that reflects the "strength" of the breaking of Hermiticity. The non-Hermitian chiral random matrix model can be mapped to the partition function of pion effective theory associated with two-colour QCD and non-zero baryon chemical potential in the ε-regime [73]. If the microscopic spectral density correctly reproduces the eigenvalue correlations of the Dirac operator in the deep infrared, then both the chiral condensate and the pion decay constant can be obtained from simulations of two-colour QCD on the lattice. The baryon chemical potential renders the Dirac operator non-Hermitian and its spectrum complex-valued. By a formal symmetry analysis it can be shown that the spectrum of the Dirac operator at non-zero baryon chemical potential exhibits pairs of purely imaginary eigenvalues (as in the Hermitian case), pairs of real eigenvalues, and also quadruplets of complex eigenvalues. The microscopic spectral density derived from non-Hermitian chRMT gives a prediction of *how* the eigenvalues of the Dirac operator near zero are distributed in the complex plane.

In this study lattice simulations are carried for the gauge group SU(2) in the quenched approximation. The spectral properties of the lowest-lying eigenvalues of the overlap operator are analyzed and compared to predictions of Hermitian and non-Hermitian chRMT of the orthogonal ensemble. The study is structured as follows. In this chapter the formalities of QCD, chiral symmetry breaking, and pion effective theory associated with gauge group SU(2) are introduced. The essential ingredients for simulations of QCD on the lattice and the properties of the overlap operator are reviewed. The baryon chemical potential is introduced into the overlap operator along the lines of Ref. [74]. In Chap. 7 the basic ideas of chiral random matrix theory are sketched and the microscopic spectral densities derived from the Hermitian and non-Hermitian formulations of chRMT are introduced. For the Hermitian case also the distribution of the lowest-lying eigenvalue is available. In Chap. 8 the results of the lattice simulations are discussed. The low-lying spectrum of the overlap operator obtained from simulations of 8^4, 10^4, and 12^4 lattices with different choices for the Wilson mass parameter is analyzed and the spectral density compared to the microscopic spectral density provided by Hermitian chRMT. The flow of the lowest-lying eigenvalues of the overlap operator with the chemical potential is evaluated on a 4^4 lattice. The spectral density of the overlap operator is evaluated on a 8^4 lattice for several strength of the chemical potential and compared to the microscopic spectral density provided by non-Hermitian chRMT. The results are summarized in Chap. 9.

6.2 Non-Abelian gauge theory

The QCD action is defined by the 4-dimensional integral over the QCD Lagrangian density $\mathcal{L}[\psi, \overline{\psi}, A]$, which is a functional of the fermion fields $\psi, \overline{\psi}$ and the gauge field A, over all space-time. In Euclidian space-time one has

$$S_{\mathrm{QCD}}[\psi, \overline{\psi}, A] = \int \mathrm{d}^4x \; \mathcal{L}[\psi, \overline{\psi}, A] = S_F[\psi, \overline{\psi}, A] + S_G[A] \; . \tag{6.1}$$

For a single quark flavour with mass m_f the fermion part of the QCD action has the form

$$S_F[\psi, \overline{\psi}, A] = \int \mathrm{d}^4x \; \overline{\psi}(x) \left[\gamma_\mu D_\mu + m_f\right] \psi(x) \; . \tag{6.2}$$

In Euclidian representation the traceless Dirac matrices γ_μ are Hermitian and fulfill the following relations

$$\{\gamma_\mu, \gamma_\nu\} = 2\delta_{\mu\nu} \ , \tag{6.3}$$
$$\{\gamma_5, \gamma_\mu\} = 0 \ , \tag{6.4}$$
$$\gamma_5 = \gamma_1\gamma_2\gamma_3\gamma_4 \ . \tag{6.5}$$

In the chiral representation γ_5 is diagonal and one has

$$\gamma_{1,2,3} = \begin{pmatrix} 0 & -\mathrm{i}\sigma_{1,2,3} \\ \mathrm{i}\sigma_{1,2,3} & 0 \end{pmatrix} \ , \quad \gamma_4 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \ , \quad \gamma_5 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \ , \tag{6.6}$$

with Pauli matrices σ_i defined below. The covariant derivative at space-time x in direction $\mu = 1, 2, 3, 4$ is

$$D_\mu(x) = \partial_\mu + \mathrm{i}gA_\mu(x) \ . \tag{6.7}$$

Here $A_\mu(x)$ is the gauge field, connected to the theory by the coupling constant g. One can identify

$$A_\mu(x) = \sum_{a=1}^{N_c^2-1} A_\mu^a(x)\frac{\lambda^a}{2} \tag{6.8}$$

with real-valued colour components $A_\mu^a(x)$. The $N_c^2 - 1$ matrices $\lambda^a/2$ are the generators of the gauge group SU(N_c), with N_c being the number of colours described by the theory. The generators are chosen traceless, complex and Hermitian $N_c \times N_c$ matrices. Ordinary QCD corresponds to gauge group SU(3), and the standard representation of the generators are the Gell-Mann matrices. However, in this study the focus is on gauge group SU(2), and the standard representation of its generators $\sigma^a/2$ are the Pauli matrices

$$\sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \ , \quad \sigma_2 = \begin{pmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{pmatrix} \ , \quad \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \ . \tag{6.9}$$

For any number of colours the gauge action part of the QCD action reads

$$S_G[A] = \frac{1}{2}\int \mathrm{d}^4x \ \mathrm{Tr}_C\left[F_{\mu\nu}(x)F_{\mu\nu}(x)\right] = \frac{1}{4}\sum_{a=1}^{N_c^2-1}\int \mathrm{d}^4x \ F_{\mu\nu}^a(x)F_{\mu\nu}^a(x) \ , \tag{6.10}$$

with the trace taken over the colour index. Here $F_{\mu\nu}$ is the field strength tensor for non-Abelian gauge theories, defined as

$$F_{\mu\nu}(x) = -\frac{\mathrm{i}}{g}[D_\mu(x), D_\nu(x)] = \partial_\mu A_\nu(x) - \partial_\nu A_\mu(x) + \mathrm{i}g[A_\nu(x), A_\mu(x)] \ , \tag{6.11}$$
$$F_{\mu\nu}^a(x) = \partial_\mu A_\nu^a(x) - \partial_\nu A_\mu^a(x) - gf^{abc}A_\mu^b(x)A_\nu^c(x) \ , \tag{6.12}$$

with structure constants f^{abc}. Some observable Ω can be determined by evaluation of the Euclidian path integral, given by

$$\langle\Omega\rangle = \frac{1}{Z}\int \mathcal{D}A\mathcal{D}\psi\mathcal{D}\overline{\psi} \ \mathrm{e}^{-S_{\mathrm{QCD}}[\psi,\overline{\psi},A]} \ \Omega[\psi, \overline{\psi}, A] \ . \tag{6.13}$$

Here the integration measures of the integral are defined by the degrees of freedom of the system,

$$\mathcal{D}\psi \equiv \prod_{x,\alpha,c} \mathrm{d}\psi_{\alpha,c}(x) \;, \tag{6.14}$$

$$\mathcal{D}\overline{\psi} \equiv \prod_{x,\alpha,c} \mathrm{d}\overline{\psi}_{\alpha,c}(x) \;, \tag{6.15}$$

$$\mathcal{D}A \equiv \prod_{x,a,\mu} \mathrm{d}A^a_\mu(x) \;, \tag{6.16}$$

with Dirac index α and colour index c. The path integral Eq. (6.13) is normalized by the QCD partition function

$$Z = \int \mathcal{D}A\mathcal{D}\psi\mathcal{D}\overline{\psi}\; \mathrm{e}^{-S_{\mathrm{QCD}}[\psi,\overline{\psi},A]} = \int \mathcal{D}A\mathcal{D}\psi\mathcal{D}\overline{\psi}\; \mathrm{e}^{-S_F[\psi,\overline{\psi},A]-S_G[A]} \;. \tag{6.17}$$

The fermion fields ψ and $\overline{\psi}$ are independent Grassmann numbers and, according to the Matthews-Salam formula, the fermion part of the partition function can be rewritten in terms of a fermion determinant [75]. Finally one has

$$Z = \int \mathcal{D}A\; \mathrm{e}^{-S_G[A]} \det(D + m_f) \;, \tag{6.18}$$

with the Dirac operator $D = \gamma_\mu D_\mu$ being a functional of the gauge fields.

In practice the integration over an infinite number of degrees of freedom turns out to be impossible. However, the definition of the path integral invites to use methods of statistical mechanics. If the QCD action Eq. (6.1) is real then the exponential of the action can be interpreted as a Boltzmann weight, suitable for computer simulations of QCD applying the techniques developed for Monte-Carlo simulations.

6.3 Chiral symmetry

6.3.1 Introduction

Chiral symmetry and its spontaneous breaking are central aspects of the theory of the strong interaction and will be briefly touched in the following. Consider the Lagrangian with $N_f \times N_f$ mass matrix $M = \mathrm{diag}(m_u, m_d, \ldots)$ associated with N_f flavours of quarks,

$$\mathcal{L}[\psi,\overline{\psi},A] = \overline{\psi}(x)(D+M)\psi(x) = \overline{\psi}(x)\left[\gamma_\mu(\partial_\mu + \mathrm{i}gA_\mu) + M\right]\psi(x) \tag{6.19}$$

under chiral rotation

$$\psi(x) \to \psi'(x) = \mathrm{e}^{\mathrm{i}\alpha\gamma_5 1_{N_f}}\psi(x) \;, \quad \overline{\psi}(x) \to \overline{\psi}'(x) = \overline{\psi}(x)\mathrm{e}^{\mathrm{i}\alpha\gamma_5 1_{N_f}} \;. \tag{6.20}$$

In the chiral limit, i.e, at zero quark mass, the Lagrangian is invariant under chiral rotations, and thus $\mathcal{L}[\psi',\overline{\psi}',A] = \mathcal{L}[\psi,\overline{\psi},A]$. However, if one introduces any mass term the invariance is explicitely broken and one has

$$M\overline{\psi}(x)\psi(x) \to M\overline{\psi}'(x)\psi'(x) = M\overline{\psi}(x)\mathrm{e}^{\mathrm{i}2\alpha\gamma_5 1_{N_f}}\psi(x) \;. \tag{6.21}$$

Upon introduction of chiral projection operators $P_{R/L} = (1 \pm \gamma_5)/2$, each Dirac spinor can be decomposed into right-handed and left-handed fields $\psi_{R/L}$ and $\overline{\psi}_{R/L}$,

$$\psi(x) = P_R\psi(x) + P_L\psi(x) \equiv \psi_R(x) + \psi_L(x) \tag{6.22}$$

$$\overline{\psi}(x) = \overline{\psi}(x)P_L + \overline{\psi}(x)P_R \equiv \overline{\psi}_R(x) + \overline{\psi}_L(x)\ . \tag{6.23}$$

The Lagrangian Eq. (6.19) can be rewritten with respect to the handedness of the fermion fields. Then one has

$$\mathcal{L}[\psi, \overline{\psi}, A] = \overline{\psi}_R(x)D\psi_R(x) + \overline{\psi}_L(x)D\psi_L(x) + M\left[\overline{\psi}_R(x)\psi_L(x) + \overline{\psi}_L(x)\psi_R(x)\right]\ . \tag{6.24}$$

The mass term mixes right- and left-handed fermions explicitely. In the chiral limit there is no mixing of the spinor components and thus right- und left-handed particles coexist independently. This description is valid for QCD with arbitrary number of colours. However, the symmetry breaking pattern of the gauge groups SU(3) and SU(2) is different and exhibit different phenomena, as will be sketched in the next sections.

6.3.2 Gauge group SU(3)

In the massless theory with gauge group SU(3) the symmetries of the classical QCD action are given by [75, 76]

$$\mathrm{SU}(N_f)_L \times \mathrm{SU}(N_f)_R \times \mathrm{U}(1)_B \times \mathrm{U}(1)_A\ , \tag{6.25}$$

where the $\mathrm{SU}(N_f)_{L/R}$ symmetries originate from the invariance of the action under independent flavour transformations of the N_f left- and right-handed fields. The $\mathrm{U}(1)_B \equiv \mathrm{U}(1)_{L=R}$ symmetry represents baryon charge conservation. The $\mathrm{U}(1)_A$ axial symmetry comes from the invariance of the action under chiral rotations Eq. (6.20). However, in the quantized theory the fermion integration measure is not invariant under chiral rotations and the $\mathrm{U}(1)_A$ axial symmetry is broken anomalously. Taking into account this so-called axial anomaly the remaining symmetries of the massless theory are

$$\mathrm{SU}(N_f)_L \times \mathrm{SU}(N_f)_R \times \mathrm{U}(1)_B\ . \tag{6.26}$$

Introduction of a mass term $M = \mathrm{diag}(m, m, \ldots)$ with N_f degenerate masses further reduces the symmetries to

$$\mathrm{SU}(N_f)_V \times \mathrm{U}(1)_B\ , \tag{6.27}$$

and the action is invariant under simultaneous transformations $\mathrm{SU}(N_f)_V \equiv \mathrm{SU}(N_f)_{L=R}$ of the left- and right-fields. If the mass degeneracy is lifted the flavour symmetry is broken down even further.

For the gauge group SU(3), the physical theory of the strong interaction, one can expect the remaining symmetries (6.27) to be a very good approximation to nature in the case $N_f = 2$, because of the small masses of the lightest generation of quarks – the up and down quarks with corresponding masses $m_u \approx m_d \approx 5$ MeV – compared to the QCD scale $\Lambda_{\mathrm{QCD}} \approx 1$ GeV. A consequence of the broken chiral symmetry are $N_f^2 - 1$ almost massless bosonic excitations, the pseudo-Goldstone modes. The lightest Goldstone modes in QCD are the pions. Due to the small quark masses the flavour symmetry (6.27) is only approximate and the pions acquire a mass $m_\pi \approx 140$ MeV that is small compared to the QCD scale. Chiral symmetry is also broken in the massless formulation of QCD. The QCD ground state is not invariant under chiral rotations, giving rise to massless pions.

6.3.3 Gauge group SU(2)

In quantized QCD with two-colours, i.e., with gauge group SU(2), the $\mathrm{U}(1)_A$ axial symmetry is also broken anomalously, while the $\mathrm{U}(1)_B$ symmetry remains and thus baryon charge is conserved.

However, the flavour symmetry breaking pattern is different from gauge group SU(3) and will be discussed in the following.

The massless QCD Lagrangian that describes N_f fermion flavours in the fundamental representation of gauge group SU(2) is [77]

$$\mathcal{L} = \overline{\psi}\gamma_\nu D_\nu \psi = \mathrm{i} \begin{pmatrix} \psi_L^* \\ \psi_R^* \end{pmatrix}^T \begin{pmatrix} \sigma_\nu D_\nu & 0 \\ 0 & -\sigma_\nu^\dagger D_\nu \end{pmatrix} \begin{pmatrix} \psi_L \\ \psi_R \end{pmatrix} , \tag{6.28}$$

with $\overline{\psi} = \psi^\dagger \gamma_4$. In this notation the flavour index and the space-time dependency are suppressed. The matrices $\sigma_{1,2,3}$ are the Pauli matrices defined in Eq. (6.9), and the anti-Hermitian matrix $\sigma_4 = \mathrm{diag}(-\mathrm{i}, -\mathrm{i})$ is introduced for convenience. The covariant derivative is $D_\nu = \partial_\nu + \mathrm{i}gA_\nu^a \tau_a/2$. The Pauli matrices τ_a are the generators of the gauge group SU(2). One can define a conjugate quark field

$$\widetilde{\psi}_R \equiv \sigma_2 \tau_2 \psi_R^* , \tag{6.29}$$

introduce new spinors of dimension $2N_f$,

$$\Psi \equiv \begin{pmatrix} \psi_L \\ \widetilde{\psi}_R \end{pmatrix} \quad \text{and} \quad \Psi^\dagger \equiv \begin{pmatrix} \psi_L^* \\ \widetilde{\psi}_R^* \end{pmatrix}^T , \tag{6.30}$$

and rearrange the massless Lagrangian Eq. (6.28) into

$$\mathcal{L} = \mathrm{i} \begin{pmatrix} \psi_L^* \\ \widetilde{\psi}_R^* \end{pmatrix}^T \begin{pmatrix} \sigma_\nu D_\nu & 0 \\ 0 & \sigma_\nu D_\nu \end{pmatrix} \begin{pmatrix} \psi_L \\ \widetilde{\psi}_R \end{pmatrix} = \mathrm{i}\Psi^\dagger \sigma_\nu D_\nu \Psi . \tag{6.31}$$

Apparently the Lagrangian is invariant under flavour transformations

$$\Psi \to V\Psi , \quad \Psi^\dagger \to \Psi^\dagger V^\dagger , \tag{6.32}$$

with $V \in \mathrm{SU}(2N_f)$. Thus the remaining symmetries of the quantized and massless theory with gauge group SU(2) are

$$\mathrm{SU}(2N_f) \times \mathrm{U}(1)_B . \tag{6.33}$$

Note the difference to QCD with gauge group SU(3), which exhibits invariance under flavour transformations $\mathrm{SU}(N_f)_L \times \mathrm{SU}(N_f)_R$ in the massless case. In the following the symmetry breaking pattern of QCD with gauge group SU(2) is further characterized.

6.3.4 Chiral condensate

The breaking of chiral symmetry is characterized by the order parameter

$$\langle \overline{\psi}\psi \rangle = \langle \overline{\psi}_R \psi_L + \overline{\psi}_L \psi_R \rangle , \tag{6.34}$$

known as the chiral condensate. The condensate is non-zero if the system is in the broken phase, and vanishes if chiral symmetry is restored. For the two-colour theory one has

$$\overline{\psi}\psi = \psi^\dagger \gamma_4 \psi = \begin{pmatrix} \psi_L^* \\ \psi_R^* \end{pmatrix}^T \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \psi_L \\ \psi_R \end{pmatrix} = \frac{1}{2} \Psi^T \sigma_2 \tau_2 \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \Psi + \mathrm{h.c.} . \tag{6.35}$$

Here the Pauli matrices σ_2 and τ_2 ensure anti-symmetrization in spin and colour indices, respectively, and a colour singlet is produced. The notation can be simplified by introduction of the $2N_f \times 2N_f$

matrix

$$M = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} . \tag{6.36}$$

Then the notation becomes

$$\overline{\psi}\psi = -\frac{1}{2}\Psi^T \sigma_2 \tau_2 M \Psi + \text{h.c.} . \tag{6.37}$$

Note that the condensate is not invariant under transformations of flavour group $\mathrm{SU}(2N_f)$. The subgroup which leaves Eq. (6.37) invariant is the symplectic group $\mathrm{Sp}(2N_f)$. Thus the symmetry breaking pattern for the gauge group SU(2) and fermions in the fundamental representation is[1]

$$\mathrm{SU}(2N_f) \to \mathrm{Sp}(2N_f) . \tag{6.38}$$

The massless Lagrangian Eq. (6.31) of the two-colour theory is invariant under transformations of flavour group $\mathrm{SU}(2N_f)$. This group is parameterized by $(2N_f)^2 - 1$ generators. However, the condensate is only invariant under transformations of flavour group $\mathrm{Sp}(2N_f)$, which is parameterized by $N_f(2N_f+1)$ generators. Thus the Goldstone manifold is the coset space $\mathrm{SU}(2N_f)/\mathrm{Sp}(2N_f)$. The manifold is parameterized by $N_f(2N_f-1)-1$ generators. The (pseudo-)Goldstone particles of the two-colour theory are not only represented by pseudo-scalar mesons, but also by diquark states that come with non-zero baryon number.

6.4 Low-energy effective theory

6.4.1 Motivation

The dynamics of the Goldstone modes is governed by the deep infrared spectrum of the Dirac operator. A low-energy effective Lagrangian that applies to the phase of broken chiral symmetry can be constructed based on symmetry considerations. To lowest order in momentum expansion the effective theory comes with two low-energy constants. One is the chiral condensate and the other one is the pion decay constant. Both constants can be determined by simulation of the two-colour QCD partition function on the lattice.

In the so-called ε-regime the zero-momentum modes of the Goldstone fields dominate the partition function. Other hadronic excitations are suppressed. This is the case at length scales $m_\pi \ll 1/L \ll \Lambda_{\mathrm{QCD}}$, where the box length L of the simulated finite four-volume $V = L^4$ is much smaller than the pion correlation length $1/m_\pi$, but still larger than other non-Goldstone excitations that occur at some scale Λ_{QCD}. A correspondence between the partition function associated with this static limit of the effective Lagrangian and chiral random matrix theory allows to determine the chiral condensate from the spectra of a lattice Dirac operator that respects chiral symmetry. Extraction of the pion decay constant from lattice simulations is also possible, but somewhat more elaborate. It requires the introduction of a baryon chemical potential into the effective theory, into the lattice simulations, and also into the chiral random matrix models.

In the next two sections the low-energy effective theory for two-colour QCD is introduced. Although in this study the limiting case $N_f \to 0$ is of interest, one has to start from a description of a system with N_f quark flavours (with masses). Afterwards the basic ingredients for simulations of quenched QCD on the lattice are introduced. The random matrix models that map to the static limit of the partition functions associated with the effective theory are sketched in Chap. 7 and finally compared to the lattice simulations in Chap. 8.

[1] The symmetry breaking pattern in SU(3) gauge theory is $\mathrm{SU}(N_f) \times \mathrm{SU}(N_f) \to \mathrm{SU}(N_f)$.

6.4.2 Lagrangian of the effective theory

The two-colour QCD Lagrangian for N_f fermions with degenerate masses $m \equiv m_f$ is [77]

$$\mathcal{L} = \overline{\psi}\gamma_\nu D_\nu \psi + m\overline{\psi}\psi = \mathrm{i}\Psi^\dagger \sigma_\nu D_\nu \Psi + \left(-\frac{1}{2}\Psi^T \sigma_2 m M \Psi + \mathrm{h.c.}\right) . \tag{6.39}$$

Here mM is the mass matrix, with M already defined by Eq. (6.36). Similar to the chiral condensate, the mass term is only invariant under the subgroup $\mathrm{Sp}(2N_f)$. However, the invariance of the Lagrangian under any group element $V \in \mathrm{SU}(2N_f)$ is restored by introduction of the transformations

$$\Psi \to V\Psi , \quad \Psi^\dagger \to \Psi^\dagger V^\dagger , \quad M \to V^* M V^\dagger . \tag{6.40}$$

The symmetry is also imposed on the effective theory. At lowest order in momentum expansion the low-energy effective Lagrangian reads

$$\mathcal{L}_{\mathrm{eff}} = \frac{F^2}{2}\,\mathrm{Tr}(\partial_\nu \mathcal{U} \partial_\nu \mathcal{U}^\dagger) - mG\,\mathrm{Re}\,\mathrm{Tr}(M\mathcal{U}) . \tag{6.41}$$

The effective Lagrangian comes with two low-energy constants. One is the pion decay constant F. The other coefficient $G = \Sigma/2N_f$ is the derivative of the vacuum energy with respect to the quark mass. It is proportional to the chiral condensate $\Sigma \equiv \langle\overline{\psi}\psi(m \to 0)\rangle$. The Goldstone manifold is described by the matrix $\mathcal{U} \in \mathrm{SU}(2N_f)/\mathrm{Sp}(2N_f)$. The matrix is unitary and anti-symmetric, and is defined by $N_f(2N_f - 1) - 1$ independent components, thus reflecting the phase of broken chiral symmetry. This matrix transforms as

$$\mathcal{U} \to V\mathcal{U}V^T . \tag{6.42}$$

The Goldstone modes fluctuate around some equilibrium orientation chosen to minimize the effective Lagrangian. One can decompose $\mathcal{U}$ into

$$\mathcal{U} = UIU^T , \quad U = \exp\left(\frac{\mathrm{i}\Pi}{2F}\right) . \tag{6.43}$$

Here Π is an $2N_f \times 2N_f$ matrix that picks up the Goldstone modes living in the coset space $\mathrm{SU}(2N_f)/\mathrm{Sp}(2N_f)$, see Ref. [77] for its explicit definition. The matrix I denotes the equilibrium orientation. The natural choice for the orientation is $I = M^{-1} = M^\dagger$. Then the effective Lagrangian Eq. (6.41) is minimized.

One obtains the partition function associated with pion effective theory by integration over the effective Lagrangian Eq. (6.41),

$$Z \sim \int \mathrm{d}\mathcal{U}\, \exp\left[-\int \mathrm{d}^4x\, \mathcal{L}_{\mathrm{eff}}\right] . \tag{6.44}$$

In this study the interesting case is when the effective Lagrangian is squeezed into a finite space-time volume $V = L^4$. On the length scale much smaller than the pion correlation length, $L \ll 1/m_\pi$, but still larger than other hadronic excitations that occur at some scale Λ_{QCD}, only the zero-momentum mode U_0 of the field $U(x)$ is relevant [78]. The constant mode U_0 is independent of the space-time coordinate and one can show that in the sector of topological charge ν the partition function reads [73]

$$Z_\nu \sim \int \mathrm{d}\theta\, \mathrm{e}^{\mathrm{i}\nu\theta} \int\limits_{\mathrm{SU}(2N_f)} \mathrm{d}U_0\, \exp\left[mGV\,\mathrm{Re}\,\mathrm{Tr}\left(\mathrm{e}^{\mathrm{i}\theta/N_f} M U_0 I U_0^T\right)\right] . \tag{6.45}$$

The chiral random matrix model that maps to this static limit is introduced in Sect. 7.2.

6.4.3 Non-zero baryon chemical potential

The two-colour QCD Lagrangian Eq. (6.39) can be extended to non-zero baryon density. Inclusion of the baryon chemical potential $\mu \in \mathbb{R}$ yields [77]

$$\begin{aligned}\mathcal{L} &= \overline{\psi}\gamma_\nu D_\nu \psi - \mu\overline{\psi}\gamma_4\psi + m\overline{\psi}\psi \\ &= \mathrm{i}\Psi^\dagger\sigma_\nu D_\nu\Psi - \mu\Psi^\dagger B\Psi + \left(-\frac{1}{2}\Psi^T\sigma_2 mM\Psi + \text{h.c.}\right) \\ &= \mathrm{i}\Psi^\dagger(\sigma_\nu D_\nu - \delta_{\nu 4}\mu B)\Psi + \left(-\frac{1}{2}\Psi^T\sigma_2 mM\Psi + \text{h.c.}\right) .\end{aligned} \tag{6.46}$$

Here the $2N_f \times 2N_f$ baryon charge matrix

$$B = \begin{pmatrix} +1 & 0 \\ 0 & -1 \end{pmatrix} \tag{6.47}$$

was introduced. Its entries are the baryon charges associated with the N_f quarks ψ_L and the N_f conjugate quarks $\widetilde{\psi}_R$. The chemical potential introduces an asymmetry between the number of quarks and anti-quarks and violates the $\mathrm{SU}(2N_f)$ flavour symmetry. For sufficiently small strength of the chemical potential the baryon charge is conserved by the $\mathrm{U}(1)_B$ symmetry, but flavour symmetry is broken down to $\mathrm{SU}(N_f)_L \times \mathrm{SU}(N_f)_R$ in the massless case. In the massive case only the vector symmetry $\mathrm{SU}(N_f)_V$ remains. However, the full $\mathrm{SU}(2N_f)$ flavour symmetry can be restored if one introduces the transformations

$$\Psi \to V\Psi\ , \quad \Psi^\dagger \to \Psi V^\dagger\ , \quad M \to V^* M V^\dagger\ , \quad B \to VBV^\dagger\ , \tag{6.48}$$

with $V \in \mathrm{SU}(2N_f)$. This extended symmetry is also imposed on the effective theory. To lowest order in momentum expansion and with lowest order non-derivative terms in the baryon chemical potential μ the effective Lagrangian reads

$$\begin{aligned}\mathcal{L}_{\text{eff}} = \frac{F^2}{2}\,\mathrm{Tr}(\partial_\nu\mathcal{U}\partial_\nu\mathcal{U}^\dagger) + 2\mu F^2\,\mathrm{Tr}(B\mathcal{U}^\dagger\partial_4\mathcal{U}) \\ - \mu^2 F^2\,\mathrm{Tr}(\mathcal{U}B^T\mathcal{U}^\dagger B + BB) - mG\,\mathrm{Re}\,\mathrm{Tr}(M\mathcal{U})\ .\end{aligned} \tag{6.49}$$

Only the static part of the effective Lagrangian is of interest, and one has

$$\begin{aligned}\mathcal{L}_{\text{st}} &= -\mu^2 F^2\,\mathrm{Tr}(\mathcal{U}B^T\mathcal{U}^\dagger B + BB) - mG\,\mathrm{Re}\,\mathrm{Tr}(M\mathcal{U}) \\ &= -\mu^2 F^2\,\mathrm{Tr}(UIU^T B^T (UIU^T)^\dagger B + BB) - mG\,\mathrm{Re}\,\mathrm{Tr}(MUIU^T)\ .\end{aligned} \tag{6.50}$$

Note that in this limit there is a competition for the equilibrium orientation of $\mathcal{U}$. One can apply the Gell-Mann-Oakes-Renner relation and trade the coefficient G against $F^2 m_\pi^2/m$. Here m_π is the mass of the lightest baryon (which in two-colour QCD is a diquark). One observes that the static limit of the effective Lagrangian is dominated by the mass term as long as $\mu < m_\pi/2$. In this case the equilibrium orientation of $\mathcal{U}$ that minimizes the Lagrangian is $I = M^\dagger$. However, the minimum shifts away from $M^\dagger$ non-trivially if $\mu > m_\pi/2$, with the effect of diquark condensation and violation of baryon number conservation as discussed in more detail in Ref. [77].

In this study the domain of interest is $\mu < m_\pi/2$. The finite-volume partition function associated

with the ε-regime in the sector of topological charge ν can be formulated as [73]

$$Z_\nu \sim \int \mathrm{d}\theta\, \mathrm{e}^{\mathrm{i}\nu\theta} \int\limits_{\mathrm{SU}(2N_f)} \mathrm{d}U_0\, \exp\Big[\mu^2 F^2 V\, \mathrm{Tr}(U_0 I U_0^T B^T (U_0 I U_0^T)^\dagger B + BB) + mGV\, \mathrm{Re}\,\mathrm{Tr}\left(\mathrm{e}^{\mathrm{i}\theta/N_f} M U_0 I U_0^T\right)\Big] . \quad (6.51)$$

The partition function is valid in the domain $m_\pi, \mu \ll 1/L \ll \Lambda_{\mathrm{QCD}}$, with $V = L^4$ the space-time volume. The chiral random matrix model that maps to this static limit is introduced in Sect. 7.3.

6.5 Lattice QCD

A computer-friendly version of QCD is obtained by discretization of space-time on a 4-dimensional toroidal lattice [75]

$$\Lambda = \{n = (n_1, n_2, n_3, n_4) | n_1, n_2, n_3 = 0, \ldots, N_S - 1\ ;\ n_4 = 0, \ldots, N_T - 1\} . \quad (6.52)$$

The vectors n represent points in space-time separated by the lattice spacing a, and some space-time vector x in the continuum is therefore replaced by the vector $n = x/a$ on the lattice. The number of lattice points in the spatial direction is labeled N_S, and the number of lattice points in the Euclidian time direction is N_T. The physical volume described by the space-time lattice is Va^4, with $V = N_S^3 \times N_T$ the 4-dimensional lattice volume. In this study only isotropic lattices were evaluated. In this case one has $N_S = N_T$.

Fermions on the lattice are described by their continuum analogues

$$\psi(x) \to \psi(n) , \quad \overline{\psi}(x) \to \overline{\psi}(n) , \quad (6.53)$$

however they live exclusively on the lattice sites $n \in \Lambda$. For the free case the massless continuum Dirac operator is replaced by a central difference term that mirrors the granularity of space-time on the lattice in a naive approach to discretization. One has

$$\gamma_\mu \partial_\mu \psi(x) \to \sum_{\mu=1}^{4} \gamma_\mu \frac{\psi(n + \hat{\mu}) - \psi(n - \hat{\mu})}{2a} , \quad (6.54)$$

with $\hat{\mu}$ being the unit vector in the μ-direction. The fermion action that represents a single quark flavour with mass m_f reads

$$S_F[\psi, \overline{\psi}] = a^4 \sum_{n \in \Lambda} \overline{\psi}(n) \left[\sum_{\mu=1}^{4} \left(\gamma_\mu \frac{\psi(n + \hat{\mu}) - \psi(n - \hat{\mu})}{2a}\right) + m_f \psi(n)\right] . \quad (6.55)$$

The fermion action has the correct behaviour in the continuum limit $a \to 0$, but by definition suffers from dependency of the gauge, i.e., S_F is not yet invariant under local rotations $\Omega(n) \in \mathrm{SU}(N_c)$ in colour space,

$$\psi(n) \to \psi'(n) = \Omega(n)\psi(n) , \quad (6.56)$$

$$\overline{\psi}(n) \to \overline{\psi}'(n) = \overline{\psi}(n)\Omega(n)^\dagger , \quad (6.57)$$

$$\overline{\psi}(n)\psi(n \pm \hat{\mu}) \to \overline{\psi}'(n)\psi'(n \pm \hat{\mu}) = \overline{\psi}(n)\Omega(n)\Omega(n \pm \hat{\mu})^\dagger \psi(n \pm \hat{\mu}) . \quad (6.58)$$

The mass term of the fermion action is invariant under local rotations of the gauge group, but the

displacement introduced by the partial derivative explicitely prohibits gauge invariance. However, this issue can be fixed by introduction of new directed fields $U_\mu(n)$ that are associated with the links between two adjacent lattice sites n and $n+\hat{\mu}$. The so-called link variables $U_\mu(n)$ are elements of the gauge group SU(N_c). If one introduces the transformation rule

$$U_{\pm\mu}(n) \to U'_{\pm\mu}(n) = \Omega(n) U_{\pm\mu}(n) \Omega(n \pm \hat{\mu})^\dagger \;, \tag{6.59}$$

and furthermore defines $U_{-\mu}(n) \equiv U_\mu(n-\hat{\mu})^\dagger$, then

$$\overline{\psi}(n) U_{\pm\mu}(n) \psi(n) \to \overline{\psi}'(n) U'_{\pm\mu}(n) \psi'(n) \tag{6.60}$$

is invariant under local rotations in colour space. The massless Dirac operator then becomes

$$\gamma_\mu \partial_\mu \psi(x) \to \sum_{\mu=1}^{4} \gamma_\mu \frac{U_\mu(n)\psi(n+\hat{\mu}) - U_{-\mu}(n)\psi(n-\hat{\mu})}{2a} \;, \tag{6.61}$$

and the fermion action now is a functional of both the fermion fields and gauge fields on the lattice,

$$S_F[\psi, \overline{\psi}, U] = a^4 \sum_{n\in\Lambda} \overline{\psi}(n) \left[\sum_{\mu=1}^{4} \left(\gamma_\mu \frac{U_\mu(n)\psi(n+\hat{\mu}) - U_{-\mu}(n)\psi(n-\hat{\mu})}{2a} \right) + m_f \psi(n) \right] \;. \tag{6.62}$$

Each link variable $U_\mu(n)$ acts as gauge transporter between two adjacent lattice sites n and $n+\hat{\mu}$. In the continuum theory the gauge transporter that connects two points x and y in space-time along the path C is defined by

$$G(x,y) = P \exp\left(\mathrm{i}g \int_C \mathrm{d}s_\mu \, A_\mu \right) \;, \tag{6.63}$$

where the operator P ensures path ordering. On the lattice the smallest path C is between two adjacent lattice sites, and the corresponding gauge transporter is defined as

$$U_\mu(n) = \exp\left[\mathrm{i}agA_\mu(n) \right] \;. \tag{6.64}$$

On the lattice the gauge action can be constructed from the shortest non-trivial closed loops of link variables, so-called plaquettes, that are defined as

$$\begin{aligned} U_{\mu\nu}(n) &= U_\mu(n) U_\nu(n+\hat{\mu}) U_{-\mu}(n+\hat{\mu}+\hat{\nu}) U_{-\nu}(n+\hat{\nu}) \\ &= U_\mu(n) U_\nu(n+\hat{\mu}) U_\mu(n+\hat{\mu})^\dagger U_\nu(n)^\dagger \;, \end{aligned} \tag{6.65}$$

where in the second line the identity $U_{-\mu}(n) = U_\mu(n-\hat{\mu})^\dagger$ is applied. The gluon gauge action [79]

$$S_G[U] = \frac{\beta}{N_c} \sum_{n\in\Lambda} \sum_{\mu<\nu} \mathrm{Re}\,\mathrm{Tr}\left[1 - U_{\mu\nu}(n) \right] \;, \tag{6.66}$$

is called the Wilson action. The trace over the plaquettes ensures gauge invariance and one can show that in the limit $a \to 0$ the Wilson action approaches the correct continuum limit, i.e., the continuum gauge action Eq. (6.10) is recovered. The parameter $\beta = 2N_c/g^2$ defines the coupling strength.

Observables are evaluated on the lattice by the path integral approach. The equivalent formula-

tion of Eq. (6.13) for the evaluation of some observable Ω on the lattice is

$$\langle\Omega\rangle = \frac{1}{Z}\int \mathcal{D}U\mathcal{D}\psi\mathcal{D}\overline{\psi}\ \mathrm{e}^{-S_{\mathrm{QCD}}[\psi,\overline{\psi},U]}\ \Omega[\psi,\overline{\psi},U] \tag{6.67}$$

$$= \frac{1}{Z}\int \mathcal{D}U\mathcal{D}\psi\mathcal{D}\overline{\psi}\ \mathrm{e}^{-S_F[\psi,\overline{\psi},U]-S_G[U]}\ \Omega[\psi,\overline{\psi},U]\ . \tag{6.68}$$

Here the lattice action S_{QCD} is the equivalent to the continuum action Eq. (6.1) with lattice Dirac spinors $\psi, \overline{\psi}$ and link variables U_μ. The QCD partition function on the lattice is

$$Z = \int \mathcal{D}U\mathcal{D}\psi\mathcal{D}\overline{\psi}\ \mathrm{e}^{-S_F[\psi,\overline{\psi},U]-S_G[U]} = \int \mathcal{D}U\ \det(D+m_f)\mathrm{e}^{-S_G[U]}\ , \tag{6.69}$$

where D is some lattice Dirac operator. The integration measures are given by

$$\mathcal{D}\psi \equiv \prod_{n\in\Lambda}\prod_{\alpha,c}\mathrm{d}\psi_{\alpha,c}(n)\ , \tag{6.70}$$

$$\mathcal{D}\overline{\psi} \equiv \prod_{n\in\Lambda}\prod_{\alpha,c}\mathrm{d}\overline{\psi}_{\alpha,c}(n)\ , \tag{6.71}$$

$$\mathcal{D}U \equiv \prod_{n\in\Lambda}\prod_{\mu}\mathrm{d}U_\mu(n)\ . \tag{6.72}$$

Here $\mathcal{D}U$ defines the product measure which integrates the link variables over the whole group manifold of SU(N_c) [75].

In simulations of Lattice QCD the exponential of the lattice gauge action is interpreted as Boltzmann weight and allows to apply methods developed for Monte-Carlo simulations. In practice the generation of gauge field configurations is necessary for the determination of some observable. The evaluation of the fermion determinant $\det(D+m_f)$ is quite a compute-intensive task. A computationally milder approach to simulations of quantum field theories on the lattice is to neglect the contributions of the sea quarks. In the quenched approximation, which corresponds to lattice simulations in the limit $N_f \to 0$, the fermion determinant is set to unity. Then the evaluation of some observable Ω simplifies to

$$\langle\Omega\rangle_{\mathrm{G}} = \frac{1}{Z}\int \mathcal{D}U\ \mathrm{e}^{-S_{\mathrm{G}}[U]}\ \Omega[U] = \frac{\displaystyle\int \mathcal{D}U\ \mathrm{e}^{-S_{\mathrm{G}}[U]}\ \Omega[U]}{\displaystyle\int \mathcal{D}U\ \mathrm{e}^{-S_{\mathrm{G}}[U]}}\ . \tag{6.73}$$

6.6 Fermions on the lattice

6.6.1 Wilson-Dirac operator

The lattice Dirac operator with naive discretization Eq. (6.61) does not come without a price. Instead of a single fermion species the discretized Dirac operator describes 16, the so-called fermion doublers. To see this, one can use the quark propagator in momentum space on the lattice. This propagator is obtained by the discrete Fourier-transform of the Dirac operator in coordinate space and subsequent inversion.

Consider the lattice Dirac operator Eq. (6.61) for a single fermion flavour with mass m_f. In the free case the link variables are replaced by the identity, $U_\mu(n) \equiv 1\ \forall\ n,\mu$. The matrix elements of

this operator can be compactly quoted by

$$D(n|m) = m_f + \sum_{\mu=1}^{4} \gamma_\mu \frac{\delta_{n+\hat{\mu},m} - \delta_{n-\hat{\mu},m}}{2a} \ . \tag{6.74}$$

The Fourier-transform of this operator is [75]

$$\widetilde{D}(p|q) = \frac{1}{V} \sum_{n,m\in\Lambda} \mathrm{e}^{-\mathrm{i}p_\mu n_\mu a} D(n|m) \mathrm{e}^{\mathrm{i}q_\mu m_\mu a} \tag{6.75}$$

$$= \frac{1}{V} \sum_{n\in\Lambda} \mathrm{e}^{-\mathrm{i}(p_\mu - q_\mu) n_\mu a} \left(m_f + \sum_{\mu=1}^{4} \gamma_\mu \frac{\mathrm{e}^{\mathrm{i}q_\mu a} - \mathrm{e}^{-\mathrm{i}q_\mu a}}{2a} \right) \tag{6.76}$$

The strong locality of the operator in coordinate space cancels the sum over m and projects out the μ-component of the momentum vector q,

$$\sum_{m\in\Lambda} \delta_{n\pm\hat{\mu},m} \mathrm{e}^{\mathrm{i}q_\mu m_\mu a} = \mathrm{e}^{\mathrm{i}q_\mu (n_\mu \pm \hat{\mu})a} = \mathrm{e}^{\mathrm{i}q_\mu n_\mu a} \mathrm{e}^{\pm \mathrm{i}q_\mu a} \ . \tag{6.77}$$

The exponentials in Eq. (6.76) can be rewritten in terms of the sine function and one obtains the lattice Dirac operator in momentum space

$$\widetilde{D}(q) = m_f + \frac{\mathrm{i}}{a} \sum_{\mu=1}^{4} \gamma_\mu \sin(q_\mu a) \ . \tag{6.78}$$

The lattice propagator is the inverse of the operator. One finds

$$\widetilde{D}(q)^{-1} = \frac{m_f - \frac{\mathrm{i}}{a} \sum_\mu \gamma_\mu \sin(q_\mu a)}{m_f^2 + \frac{1}{a^2} \sum_\mu \sin^2(q_\mu a)} \ . \tag{6.79}$$

Expansion of the sine function around the lattice spacing at small momentum, $\sin(q_\mu a) = q_\mu a + \mathcal{O}(a^2)$, and taking the continuum limit $a \to 0$ yields the free fermion propagator in the continuum,

$$\lim_{a\to 0} \widetilde{D}(q)^{-1} = \frac{m_f - \mathrm{i} \sum_\mu \gamma_\mu q_\mu}{m_f^2 + \sum_\mu q_\mu^2} \ . \tag{6.80}$$

However, not only the momentum vector $q = (0,0,0,0)$ contributes in the continuum limit, but also those momenta with at least one component at the corner of the first Brillouin zone, i.e., $q_\mu = \pi/a$. In total there are fifteen such contributions to the fermion propagator and therefore in the continuum limit the Dirac operator with naive discretization describes sixteen fermion species instead of just a single one.

A modification of the lattice Dirac operator in momentum space operator Eq. (6.78) gives unwanted fermion species a higher mass. The modification does not remove the doubler modes completely, however, they decouple from the theory in the continuum limit. The proposed modification is

$$\widetilde{D}(q) \to \widetilde{D}_{\mathrm{W}}(q) = m_f + \frac{\mathrm{i}}{a} \sum_{\mu=1}^{4} \gamma_\mu \sin(q_\mu a) + \frac{1}{a} \sum_{\mu=1}^{4} \left[1 - \cos(q_\mu a)\right] \ . \tag{6.81}$$

A fermion with zero momentum is not affected by the additional term. But fermions with momentum

components $q_\mu = \pi/a$ acquire an additional mass term

$$m_f + \frac{2l}{a} \,, \tag{6.82}$$

where l is the number of components with $q_\mu = \pi/a$. The additional mass obviously diverges in the limit $a \to 0$. In coordinate space the additional term acts like a Laplacian extension to the operator that vanishes in the continuum limit. The modified lattice Dirac operator (with non-trivial link variables) is known as the Wilson-Dirac operator. In coordinate space it reads

$$\begin{aligned} D_{\mathrm{W}}(n|m) = m_f + \sum_{\mu=1}^{4} \gamma_\mu \frac{U_\mu(n)\delta_{n+\hat{\mu},m} - U_{-\mu}(n)\delta_{n-\hat{\mu},m}}{2a} \\ - a \sum_{\mu=1}^{4} \frac{U_\mu(n)\delta_{n+\hat{\mu},m} + U_{-\mu}(n)\delta_{n-\hat{\mu},m} - 2\delta_{n,m}}{2a^2} \end{aligned} \tag{6.83}$$

$$= m_f + \frac{4}{a} - \frac{1}{2a} \sum_{\mu=1}^{4} \Big(U_\mu(n)(1+\gamma_\mu)\delta_{n+\hat{\mu},m} + U_{-\mu}(n)(1-\gamma_\mu)\delta_{n-\hat{\mu},m} \Big) \,, \tag{6.84}$$

again with $U_{-\mu}(n) = U_\mu(n-\hat{\mu})^\dagger$. A slightly more compact notation is[2]

$$D_{\mathrm{W}} = 1 - \kappa \sum_{\mu=1}^{4} (T_\mu^+ + T_\mu^-) \,. \tag{6.85}$$

Here

$$T_\mu^\pm(n|m) \equiv (1 \pm \gamma_\mu) U_{\pm\mu}(n) \delta_{n\pm\hat{\mu},m} \,, \tag{6.86}$$

and the mass term has been absorbed into the so-called hopping parameter $\kappa = 1/(8 + 2m_f a)$.

6.6.2 Lattice chiral symmetry

In the massless limit the continuum Dirac operator anti-commutes with γ_5,

$$\{D, \gamma_5\} = D\gamma_5 + \gamma_5 D = 0 \,. \tag{6.87}$$

The massless Dirac operator is anti-Hermitian, and due to chiral symmetry the non-zero eigenvalues occur in complex conjugate pairs. The Dirac operator also exhibits exact zero modes $D\phi_0 = 0$, which can be chosen as eigenstates of γ_5. The zero modes of the Dirac operator $D = \gamma_\mu(\partial_\mu + \mathrm{i}gA_\mu)$ are intimately related to the topological structure of the gauge field A. A winding number, also called the topological charge, can be associated with each gauge field. This number has integer value and is defined by [75]

$$Q(A) = \frac{1}{32\pi^2} \int \mathrm{d}^4x \; \varepsilon_{\mu\nu\delta\rho} F^a_{\mu\nu}(x) F^a_{\delta\rho}(x) \in \mathbb{Z} \,, \tag{6.88}$$

where $F^a_{\mu\nu}(x)$ is a colour component of the field strength tensor defined in Eq. (6.12). The Atiyah-Singer index theorem [80] relates the topological charge to the index of the continuum Dirac operator,

$$\operatorname{index} D[A] = n_- - n_+ = Q(A) \,. \tag{6.89}$$

[2] In this notation an irrelevant constant prefactor is dropped which can be absorbed into a redefinition of the quark fields.

Here n_+ (n_-) is the number of eigenmodes of the Dirac operator which come with definite positive (negative) chirality.

The lattice Dirac operator with naive discretization introduced in Sect. 6.5 also respects chiral symmetry in the massless limit. Unfortunately the operator suffers from unphysical fermion doubler modes. These modes can be given some additional mass diverging in the continuum limit. However, at finite lattice spacing the Wilson-Dirac operator introduced in Sect. 6.6.1 violates chiral symmetry even in the massless case, and generically one has $\{D_{\mathrm{W}}, \gamma_5\} \neq 0$. The eigenvalues occur in complex conjugate pairs, however, the Wilson-Dirac operator is not anti-Hermitian [81]. In fact, according to the Nielsen-Ninomiya theorem [82], it is not possible at all to construct a lattice version of the Dirac operator that at the same time fulfills the following requirements:

(i) Restoration of chiral symmetry such that $\{D, \gamma_5\} = 0$, in analogy to Eq. (6.87).

(ii) Locality of the eigenmodes.

(iii) The operator has the correct continuum limit.

(iv) No fermion doubler modes.

Satisfaction of the criteria (ii) and (iii) is a necessity for the lattice simulations to describe physics, however, criterium (iv) is acceptable if the doubler modes do not give a contribution in the continuum limit. The Ginsparg-Wilson relation [83]

$$D\gamma_5 + \gamma_5 D = aD\gamma_5 D \tag{6.90}$$

also proposes a cure to criterium (i). The Ginsparg-Wilson relation represents the lattice analogue to chiral symmetry in the continuum. The lattice Dirac operator D analyzed in this study will be defined explicitely in the next section.

The Ginsparg-Wilson relation has important consequences for the properties of the (approximately) chiral lattice Dirac operator. These are briefly discussed in the following. The lattice spacing a is set to unity from now on. By multiplication of the Ginsparg-Wilson relation Eq. (6.90) with γ_5 once from the left and once from the right, and furthermore imposing the condition of γ_5-Hermiticity on the operator such that $D = \gamma_5 D^\dagger \gamma_5$, then one has

$$D^\dagger + D = D^\dagger D \,, \tag{6.91}$$

$$D + D^\dagger = D D^\dagger \,, \tag{6.92}$$

and thus $DD^\dagger = D^\dagger D$. If the operator is γ_5-Hermitian then it is automatically normal. Normality guarantees that D is diagonalizable by some unitary transformation $D = U\Lambda U^\dagger$ with unitary matrix U and diagonal matrix $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$, $N = \dim(U)$. The characteristic polynomial $P(\lambda)$ of the operator is

$$P(\lambda) = \det[D - \lambda] = \det[D^\dagger - \lambda] = \det[D - \lambda^*]^* = P(\lambda^*)^* \,, \tag{6.93}$$

where the second step follows from γ_5-Hermiticity. Consequently the eigenvalues λ_i of D are either real or come in complex conjugate pairs if the operator is γ_5-Hermitian. Now consider the eigensystem

$$D\phi_\lambda = \lambda\phi_\lambda \tag{6.94}$$

with ϕ_λ being normalized eigenvectors of D with corresponding eigenvalue $\lambda = x + \mathrm{i}y$. If one sandwiches Eq. (6.91) with (normalized) eigenvectors $\phi_\lambda^\dagger$ from the left and ϕ_λ from the right, then

one finds that the eigenvalues of D are restricted to lie on a unit circle in the complex plane centered on the real axis at 1,

$$\lambda^* + \lambda = \lambda^*\lambda \quad \Leftrightarrow \quad (\lambda - 1)\,(\lambda^* - 1) = 0 \quad \Leftrightarrow \quad (x-1)^2 - y^2 = 1\ . \tag{6.95}$$

The circle is known as the Ginsparg-Wilson circle. The real eigenvalues are thus restricted to $\lambda \in \{0, 2\}$, while each complex eigenvalue is accompanied by a partner $\lambda^* = \lambda/(\lambda - 1)$. Exploiting the γ_5-Hermiticity once more allows to investigate chirality. One has

$$\phi_\lambda^\dagger \gamma_5 D \phi_\lambda = \phi_\lambda^\dagger D^\dagger \gamma_5 \phi_\lambda \quad \Rightarrow \quad \lambda \phi_\lambda^\dagger \gamma_5 \phi_\lambda = \lambda^* \phi_\lambda^\dagger \gamma_5 \phi_\lambda \quad \Rightarrow \quad (\mathrm{Im}\,\lambda)\phi_\lambda^\dagger \gamma_5 \phi_\lambda = 0\ . \tag{6.96}$$

The equation on the right implies that each eigenvector ϕ_λ has vanishing chirality, $\phi_\lambda^\dagger \gamma_5 \phi_\lambda = 0$, if the imaginary part of the corresponding eigenvalue $\lambda \in \mathbb{C}$ does not vanish, $\mathrm{Im}\,\lambda \neq 0$. Only those eigenvectors with corresponding real eigenvalues $\lambda \in \mathbb{R}$ may have non-vanishing chirality, and thus $\phi_\lambda^\dagger \gamma_5 \phi_\lambda \neq 0$.

In fact, on the subspace of zero modes ϕ_0, the operator D commutes with γ_5. These states can be chosen as eigenstates of γ_5. The same applies to the non-zero modes ϕ_2 with corresponding eigenvalues $\lambda = 2$. Let's consider all states ϕ_0 and ϕ_2 to be chosen as eigenstates of γ_5. Since $\gamma_5^2 = 1$ it follows that $\phi_\lambda^\dagger \gamma_5 \phi_\lambda = \pm 1$ if $\lambda \in \{0, 2\}$. Exact chirality allows to evaluate the topological charge associated with the gauge field background from the zero modes. One has [75, 84, 85]

$$\begin{aligned}
\frac{1}{2}\,\mathrm{Tr}[\gamma_5 D] &= -\frac{1}{2}\,\mathrm{Tr}[\gamma_5 (2 - D)] \\
&= -\frac{1}{2}\sum_\lambda (2 - \lambda)\phi_\lambda^\dagger \gamma_5 \phi_\lambda \\
&= -\sum_{\lambda = 0} \phi_\lambda^\dagger \gamma_5 \phi_\lambda \\
&= n_- - n_+ \equiv \mathrm{index}\, D \in \mathbb{Z}\ .
\end{aligned} \tag{6.97}$$

All eigenstates with complex eigenvalues have vanishing chirality and therefore only the sum over the states ϕ_0 and ϕ_2 remain. The latter get cancelled by sneaking in the extra term $\mathrm{Tr}(2\gamma_5) = 0$. Here n_+ (n_-) is the number of zero modes which come with definite positive (negative) chirality. Alternatively the definite chirality of the eigenmodes ϕ_2 can be used to evaluate the topological charge. One finds

$$\frac{1}{2}\,\mathrm{Tr}[\gamma_5 D] = \frac{1}{2}\sum_\lambda \lambda \phi_\lambda^\dagger \gamma_5 \phi_\lambda = \sum_{\lambda = 2} \phi_\lambda^\dagger \gamma_5 \phi_\lambda = n'_+ - n'_- = \mathrm{index}\, D \in \mathbb{Z}\ , \tag{6.98}$$

where n'_+ (n'_-) is the number of eigenstates ϕ_2 with positive (negative) chirality. Both assignments Eq. (6.97) and Eq. (6.98) yield the same result, and therefore it follows that

$$n_+ - n_- = -(n'_+ - n'_-)\ . \tag{6.99}$$

It is stated in Ref. [75] that in the absence of fine-tuning one only finds eigenstates of either chirality, i.e., either $n_+ \neq 0$ or $n_- \neq 0$ (or $n_+ = n_- = 0$). This behaviour was also observed in this study without any exception. One furthermore expects that for sufficiently smooth lattice gauge field configurations U the continuum index theorem Eq. (6.89) is also applicable to lattice simulations. Then one has $\mathrm{index}\, D[U] = Q(U)$. However, in simulations of Lattice QCD violations do occur. In the following two sections an explicit definition for the lattice Dirac operator that respects the lattice chiral symmetry will be given.

6.6.3 Overlap operator

An explicit definition of a Dirac operator that respects chiral symmetry on the lattice was found by Neuberger and Narayanan [86, 87]. The so-called overlap operator satisfies the Ginsparg-Wilson relation Eq. (6.90) and describes (massive) quarks on the lattice. In this study the massless overlap operator is of interest, which can be written as

$$D_{\mathrm{ov}} = 1 + \gamma_5 \varepsilon(H) \; . \tag{6.100}$$

Here $\varepsilon(H)$ is the matrix sign function and the kernel H is an Hermitian lattice Dirac operator. The overlap operator has the correct continuum limit, is free of doubler modes, and obeys a lattice version of chiral symmetry.

Before introducing the kernel operator let's stick to the matrix sign function. A convenient expression for this function is obtained if its kernel is diagonalizable [88]. Then one can apply the transformation $H = U\Lambda U^{-1}$. Here $U \in \mathrm{Gl}(N, \mathbb{C})$, $N = \dim(H)$, and $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$ is the diagonal matrix of eigenvalues. For any diagonalizable kernel the matrix sign function can then be defined by

$$\varepsilon(H) = U \,\mathrm{sgn}(\mathrm{Re}\,\Lambda) U^{-1} \; . \tag{6.101}$$

This definition is valid both for Hermitian kernel (in this case $\lambda_i \in \mathbb{R}$) and also non-Hermitian kernel (then $\lambda_i \in \mathbb{C}$). The numerical computation of the overlap operator is quite demanding and an efficient algorithm that diagonalizes the operator is introduced in Appendix B.1.

Given the definition of the matrix sign function it is straight forward to show that the overlap operator indeed satisfies the Ginsparg-Wilson relation. Exploiting $\gamma_5^2 = 1$ and $\varepsilon(H)^2 = 1$ one has (here for simplicity $\varepsilon \equiv \varepsilon(H)$)

$$\begin{aligned} \gamma_5 D_{\mathrm{ov}} + D_{\mathrm{ov}} \gamma_5 &= \gamma_5(1 + \gamma_5\varepsilon) + (1 + \gamma_5\varepsilon)\gamma_5 \\ &= \gamma_5 + \varepsilon + \gamma_5\varepsilon\gamma_5 + \gamma_5\varepsilon^2 \\ &= (1 + \gamma_5\varepsilon)\gamma_5(1 + \gamma_5\varepsilon) \\ &= D_{\mathrm{ov}} \gamma_5 D_{\mathrm{ov}} \; . \end{aligned} \tag{6.102}$$

The matrix sign function inherits the Hermiticity property of the kernel lattice Dirac operator. If the kernel is Hermitian then $\varepsilon(H) = \varepsilon(H)^\dagger$. Note that Hermiticity is not a necessity for satisfaction of the Ginsparg-Wilson relation, because in either case one has $\varepsilon(H)^2 = 1$. However, if the kernel is an γ_5-Hermitian operator, then the overlap operator is also γ_5-Hermitian,

$$D_{\mathrm{ov}} = \gamma_5 D_{\mathrm{ov}}^\dagger \gamma_5 \; . \tag{6.103}$$

As derived in the last section, γ_5-Hermiticity implies that D_{ov} is normal, thus $D_{\mathrm{ov}} D_{\mathrm{ov}}^\dagger = D_{\mathrm{ov}}^\dagger D_{\mathrm{ov}}$. Its eigenvalues are either real or come in complex conjugate pairs and are restricted to the Ginsparg-Wilson circle Eq. (6.95). The overlap operator also exhibits exact zero modes (and also modes with corresponding eigenvalues $\lambda = 2$). These modes can be chosen as eigenstates of γ_5.

The common choice for the kernel lattice Dirac operator is the Hermitian Wilson-Dirac operator Eq. (6.85),

$$H \equiv \gamma_5 D_{\mathrm{W}}(m_{\mathrm{W}} \equiv -m_f) \; , \tag{6.104}$$

which is also used in this study. The overlap operator acts as a projection of the non-chiral lattice Wilson-Dirac operator onto the solution of the Ginsparg-Wilson relation. The operator describes exactly massless quarks and is free of doubler modes if the so-called Wilson mass $m_{\mathrm{W}} \equiv -m_f$ is chosen in the range $m_{\mathrm{W}} \in (0, 2)$ [89]. The Wilson mass also has some impact on the locality properties of the overlap operator as discussed in [90].

The eigenvalues of the overlap operator are pinned to the Ginsparg-Wilson circle. However, the continuum Dirac operator is anti-Hermitian, thus its non-zero eigenvalues are purely imaginary. Another lattice Dirac operator obeys this symmetry property of the continuum operator. The projected overlap operator is defined as [91]

$$D_p = \frac{2D_{\rm ov}}{2 - D_{\rm ov}} = 2\,\frac{1+\varepsilon(H)}{1-\varepsilon(H)}\ . \tag{6.105}$$

Its spectrum has the desired properties, i.e., the non-zero eigenvalues λ_p of D_p lie on the imaginary axis (and come in complex conjugate pairs). Given the eigenvalues $\lambda_{\rm ov}$ of the (original) overlap operator Eq. (6.100) the spectrum of the projected overlap operator is obtained by application of the stereographic projection

$$\lambda_{\rm ov} \to \lambda_p = \frac{2\lambda_{\rm ov}}{2 - \lambda_{\rm ov}}\ . \tag{6.106}$$

The stereographic projection leaves the eigenvalues $\lambda_{\rm ov} = 0$ unchanged (those eigenvalues living on the opposite side of the Ginsparg-Wilson circle are mapped to infinity), whereas pairs of complex eigenvalues $\lambda_{\rm ov}$ and $\lambda_{\rm ov}/(\lambda_{\rm ov}-1)$ on the Ginsparg-Wilson circle are mapped to pairs λ_p and λ_p^* on the imaginary axis. For the comparison between the lattice data and random matrix theory performed in this study especially the projected spectrum of the overlap operator is of interest.

6.7 Chemical potential on the lattice

In the continuum theory the Dirac operator with baryon chemical potential $\mu \in \mathbb{R}$ reads

$$D(\mu) = \gamma_\nu(\partial_\nu + {\rm i}gA_\nu) + \mu\gamma_4\ . \tag{6.107}$$

This operator is imbalanced with respect to the number of quarks and anti-quarks and therefore introduces a non-zero baryon density into the field theory. The combination $\mu\gamma_4$ renders the operator non-Hermitian and one has $D(\mu) = -D(-\mu)^\dagger$. The breaking of Hermiticity is discussed in more detail in Chap. 7 especially for the gauge group SU(2) and fermions in the fundamental representation.

The chemical potential is introduced on the lattice by a modification of the 4-direction of the Wilson-Dirac operator Eq. (6.85) [92],

$$D_{\rm W} \to D_{\rm W}(\mu) = 1 - \kappa\sum_{\nu=1}^{3}(T_\nu^+ + T_\nu^-) - \kappa({\rm e}^{+\mu}T_4^+ + {\rm e}^{-\mu}T_4^-)\ . \tag{6.108}$$

Here again

$$T_\nu^\pm(n|m) \equiv (1 \pm \gamma_\nu)U_{\pm\nu}(n)\delta_{n\pm\hat{\nu},m} \tag{6.109}$$

and $\kappa = 1/(8 + 2m_f)$ is the hopping parameter. For real chemical potential $\mu \in \mathbb{R}$ the exponential weight factor ${\rm e}^{\pm\mu}$ favours propagation in positive Euclidian time direction (+) and disfavours propagation in the negative time direction (−). In the limit $\mu \to 0$ the Wilson-Dirac operator at zero chemical potential Eq. (6.85) is recovered. However, at non-zero μ the Wilson-Dirac operator is no longer γ_5-Hermitian. From

$$\gamma_5{\rm e}^{\pm\mu}T_4^\pm = {\rm e}^{\pm\mu}T_4^\mp\gamma_5 \tag{6.110}$$

it follows that

$$D_{\rm W}(\mu) = \gamma_5 D_{\rm W}(-\mu)^\dagger\gamma_5\ . \tag{6.111}$$

An intuitive proposal for the overlap operator with chemical potential is [74, 93]

$$D_{\mathrm{ov}} \to D_{\mathrm{ov}}(\mu) = 1 + \gamma_5 \varepsilon[H(\mu)] \tag{6.112}$$

with the Wilson-Dirac kernel Eq. (6.108) at non-zero chemical potential μ,

$$H(\mu) \equiv \gamma_5 D_{\mathrm{W}}(\mu; m_{\mathrm{W}} \equiv -m_f) \; . \tag{6.113}$$

As mentioned above, this kernel is not Hermitian. Thus the sign function, readily defined by Eq. (6.101), is also non-Hermitian and generically one has $\varepsilon(H) \neq \varepsilon(H)^\dagger$ in the case $\mu \neq 0$. However, in any case $\varepsilon(H)^2 = 1$ holds true, and thus the overlap operator satisfies the Ginsparg-Wilson relation Eq. (6.102) also in the case $\mu \neq 0$. Therefore a lattice chiral symmetry is respected and the operator exhibits exact zero modes (which can be chosen as eigenstates of γ_5). The index of the operator is readily defined by Eq. (6.97) and thus can be determined from the chirality of the zero modes, completely analogous to the case $\mu = 0$.

The chemical potential renders the matrix sign function non-Hermitian. Therefore the overlap operator is also non-Hermitian and one finds

$$D_{\mathrm{ov}}(\mu) = \gamma_5 D_{\mathrm{ov}}(-\mu)^\dagger \gamma_5 \; . \tag{6.114}$$

Consequently the operator is not normal,

$$D_{\mathrm{ov}}(\mu) D_{\mathrm{ov}}(\mu)^\dagger \neq D_{\mathrm{ov}}(\mu)^\dagger D_{\mathrm{ov}}(\mu) \; . \tag{6.115}$$

With the loss of Hermiticity the spectrum of the overlap operator at non-zero chemical potential has different properties compared to the zero case. The lattice chiral symmetry forces the non-zero eigenvalues to appear in pairs, however, non-trivially distorted from the Ginsparg-Wilson circle. Lattice simulations of the overlap operator at non-zero chemical potential are discussed in Chap. 8.

Chapter 7

Random Matrix Theory

7.1 Introduction

Since its first application to nuclear spectra by Wigner, random matrix theory (RMT) has found a lot of applications in, e.g., nuclear, atomic, molecular physics, disordered mesoscopic systems, and also QCD, see Ref. [94] for a review. In the theory of the strong interaction RMT especially applies to pion effective field theory, cf. Sect. 6.4. At length scales $1/m_\pi \gg L \gg 1/\Lambda_{\rm QCD}$ hadronic excitations are exponentially suppressed and the QCD partition function is governed by the fluctuations of the zero momentum modes of Goldstone bosons. The partition function of this system can be modelled by random matrices. The random matrix approach allows to determine universal quantities of the physical system analytically, typically in terms of correlation functions. Of particular interest in this study are the so-called microscopic spectral densities. These densities correspond to 1-point correlation functions that describe the lower edge of the spectrum of the Dirac operator. On the scale of the average level spacing the correlations in the spectrum of the Dirac operator near zero are considered to be universal, i.e., they do not change despite substantial variations of the average spectral density [95]. Universality was also found in the spectra of chiral lattice Dirac operators such as the overlap operator. A match of the Dirac spectrum obtained from lattice simulations to the RMT predictions allows to determine low-energy constants of QCD. In the following the essential ingredients for a model of the QCD partition function are introduced.

In the continuum the massless Dirac operator respects chiral symmetry,

$$\{D, \gamma_5\} = 0 \; . \tag{7.1}$$

Furthermore the Dirac operator is anti-Hermitian. Its non-zero eigenvalues come in complex conjugate pairs,

$$D\psi_n = \mathrm{i}\lambda_n \psi_n \; , \tag{7.2}$$

with corresponding eigenvectors ψ_n and $\gamma_5\psi_n$ which can be chosen with exact chirality. In this case one has $\gamma_5\psi_n^R = +\psi_n^R$ and $\gamma_5\psi_n^L = -\psi_n^L$. With these properties the matrix representation of the massless Dirac operator in the chiral basis reads

$$D = \begin{pmatrix} 0 & \mathrm{i}W \\ \mathrm{i}W^\dagger & 0 \end{pmatrix} \tag{7.3}$$

with matrix elements $W_{mn} = \langle \psi_m^R | D[A] | \psi_n^L \rangle$ and A the gauge field. Note that the matrix W is quadratic only if the spectrum of the Dirac operator is free of zero modes. Otherwise W is rectangular. Besides chiral symmetry the continuum Dirac operator can also obey additional anti-unitary symmetries. There are three symmetry classes which impose additional constraints on the

matrix elements of the operator. One has [64]:

- For the gauge group SU(N_c) with $N_c \geq 3$ and fermions in the fundamental representation the continuum Dirac operator $D = \gamma_\mu(\partial_\mu + \mathrm{i}g\lambda^a A^a_\mu/2)$, with λ^a the generators of the gauge group, obeys no additional symmetry. The matrix elements W_{mn} are complex.

- For the gauge group SU(2) and fermions in the fundamental representation the continuum Dirac operator $D = \gamma_\mu(\partial_\mu + \mathrm{i}g\sigma^a A^a_\mu/2)$, with Pauli matrices σ^a as the generators of the gauge group, obeys the additional symmetry $[D, C\sigma_2 K] = 0$. Here $C = \gamma_4\gamma_2$ is charge conjugation and K denotes complex conjugation. One has $(C\sigma_2 K)^2 = 1$, and the matrix elements W_{mn} can be chosen real.

- For the gauge group SU(N_c) with $N_c \geq 2$ and fermions in the adjoint representation the continuum Dirac operator $D(b|c) = \gamma_\mu(\partial_\mu + gf^{abc}A^a_\mu)$, with structure constants f^{abc}, obeys the additional symmetry $[D, CK] = 0$. Here $(CK)^2 = -1$, and the matrix elements A_{ij} can be represented by real quaternions.

For all symmetry classes and N_f quarks with masses m_f the QCD partition function reads [64, 95]

$$Z = \int \mathcal{D}A\, \mathrm{e}^{-S_\mathrm{G}[A]} \prod_{f=1}^{N_f} \det(D + m_f) = \int \mathcal{D}A\, \mathrm{e}^{-S_\mathrm{G}[A]} \prod_{f=1}^{N_f} \det \begin{pmatrix} m_f & \mathrm{i}W \\ \mathrm{i}W^\dagger & m_f \end{pmatrix} . \tag{7.4}$$

Here $S_\mathrm{G}[A]$ is the gauge action and the integral is carried out over all gauge field configurations A. This partition function serves as a template for Hermitian chiral gaussian random matrix theory introduced in the next section.

7.2 Hermitian chiral random matrix theory

Pioneering work was done in the 90's by Verbaarschot who introduced chiral extensions to the gaussian random matrix models, opening the door to applications of random matrix theory in non-Abelian field theory [64, 65, 96]. In random matrix theory the matrix elements of the Dirac operator are replaced by independent and identically distributed random variables. The gauge action is replaced by a gaussian distribution of the matrix elements. For each symmetry class of the Dirac operator there exists an associated random matrix model, typically classified by the Dyson index β. The three random matrix ensembles are: the chiral gaussian orthogonal ensemble (chGOE) for real matrices ($\beta = 1$), the chiral gaussian unitary ensemble (chGUE) for Hermitian complex matrices ($\beta = 2$) and the chiral gaussian symplectic ensemble (chGSE) for quaternion matrices ($\beta = 4$). The random matrix model derived from the QCD partition function Eq. (7.4) is

$$Z_\nu = \int \mathcal{D}A\, \exp\left(-\frac{N\beta}{4}\,\mathrm{Tr}\, A^\dagger A\right) \prod_{f=1}^{N_f} \det \begin{pmatrix} m_f & \mathrm{i}A \\ \mathrm{i}A^\dagger & m_f \end{pmatrix} . \tag{7.5}$$

In this model A is a $N \times (N + \nu)$ quadratic ($\nu = 0$) or rectangular ($\nu > 0$) matrix of the same symmetry class as the Dirac operator but chosen with random entries. The integral is over all matrix elements. The (massless) random matrix model reproduces the following properties of QCD in the continuum [95]:

- The eigenvalue spectrum of the random matrix Dirac operator exhibits pairs of purely imaginary eigenvalues $\pm\mathrm{i}x_n$ and exhibits exactly ν zero modes.

- The fermion determinant is invariant under the same flavour transformations as in the continuum. In the case $\beta = 1$ the determinant is invariant under flavour SU($2N_f$), cf. Sect. 6.3.

- The flavour symmetry breaking pattern is the same as in the continuum theory. In the case $\beta = 1$ the flavour symmetry is broken down to Sp($2N_f$).

The random matrix model of the QCD partition function can be mapped to the partition function Eq. (6.45) of the static limit of the chiral Lagrangian Eq. (6.41) in the sector of topological charge ν. Thus, in principle, correlation functions derived from the random matrix model also apply to pion effective theory in the ε-regime (up to rescaling factors). In the following the basic ingredients for the 1-point correlation function of eigenvalues derived from the random matrix model are reviewed. Finally the microscopic spectral density is introduced. This universal function describes the fluctuations of the eigenvalues around the average spectral density at the spectrum near zero.

The initial step towards the microscopic spectral density is a transformation of variables. Instead of integration over matrix elements of the random matrix A the partition function Eq.(7.5) is integrated over the $N + \nu$ eigenvalues ix_n ($x_n \geq 0$) of the model Dirac operator. The Jacobian of the transformation is [95]

$$J = |\Delta(x^2)|^\beta \prod_{k=1}^{N} x_k^{\beta|\nu|+\beta-1} \tag{7.6}$$

with the Vandermonde determinant

$$\Delta(x^2) = \prod_{1\leq k<l\leq N} (x_k^2 - x_l^2) \, . \tag{7.7}$$

Up to a normalization constant the partition function reads

$$Z_\nu \sim \int |\Delta(x^2)|^\beta \prod_{k=1}^{N} \mathrm{d}x_k \, x_k^{\beta|\nu|+\beta-1} \exp\left(-\frac{N\beta}{4} x_k^2\right) \prod_{f=1}^{N_f} (x_k^2 + m_f^2) m_f^{|\nu|} \, . \tag{7.8}$$

One can immediately derive the joint probability density function (jpdf) from the partition function. The jpdf contains all the information about the correlations in the spectrum. For the limiting case $m_f \to 0$ one finds

$$\rho(x_1, \ldots, x_N) \sim \prod_{1\leq k<l\leq N} |x_k^2 - x_l^2|^\beta \prod_{k=1}^{N} x_k^{2N_f+\beta|\nu|+\beta-1} \exp\left(-\frac{N\beta}{4} x_k^2\right) \, , \tag{7.9}$$

which is valid for all three chiral random matrix ensembles.

At this stage it is interesting to take a closer look at the jpdf. In the limit $x_i \to 0$ the density is suppressed. This property holds true for any combination of the Dyson index β, the number of quark flavours N_f and the number of zero modes ν, except for the case $\beta = 1$, $N_f = \nu = 0$. Only for this combination of parameters the jpdf does not vanish if one of the eigenvalues is near zero. The spectral density, which is the 1-point correlation function derived from the jpdf, inherits this property. One arrives at the spectral density by integrating out $N - 1$ degrees of freedom,

$$\rho(x) \sim \int \prod_{k=2}^{N} \mathrm{d}x_k \, \rho(x, x_2, \ldots, x_N) \, . \tag{7.10}$$

The microscopic spectral density is then obtained by rescaling of the eigenvalues to $z = x/N$ and taking the thermodynamic limit $N \to \infty$,

$$\rho_{\mathrm{S}}(z) = \lim_{N\to\infty} \frac{1}{N} \rho\left(\frac{z}{N}\right) . \tag{7.11}$$

The microscopic spectral density describes the fluctuations of the eigenvalues around the average spectral density at the spectrum near zero. An explicit expression for this density is evaluated in Ref. [65] for the orthogonal ensemble (but symmetric Dirac operator). In the case $N_f = 0$ it is given by

$$\rho_{\mathrm{S}}(z) = \frac{1}{2} \int_0^z \mathrm{d}u \int_0^1 \mathrm{d}s\, s^2 \left[z J_{\nu+1}(sz)\, J_\nu(su) - u J_{\nu+1}(su)\, J_\nu(sz) \right] + \frac{1}{2} J_\nu(z) . \tag{7.12}$$

Here the J_ν denote Bessel functions of the first kind of integer order ν. There is, however, no analytic expression for the density available and the function has to be evaluated numerically. In the case $N_f = 0$ additional information is available in terms of the distribution of the lowest-lying eigenvalue in the microscopic limit. The predictions are however restricted to $\nu = 0$ and 1. The distribution of the lowest-lying eigenvalue is [97]

$$\rho_{\min}(z) = \begin{cases} \dfrac{2+z}{4} \exp\left(-\dfrac{z}{2} - \dfrac{z^2}{8}\right) , & \nu = 0 \\ \dfrac{z}{4} \exp\left(-\dfrac{z^2}{8}\right) , & \nu = 1 . \end{cases} \tag{7.13}$$

The distribution of the lowest-lying eigenvalue $\rho_{\min}(z)$ and the microscopic spectral density $\rho_{\mathrm{S}}(z)$ are shown in Fig. 7.1.[1] Note that both the microscopic spectral density Eq. (7.12) and the distribution of the lowest-lying eigenvalue Eq. (7.13) take real arguments and apply to a symmetric Dirac operator (with real eigenvalues). The (projected) overlap operator introduced in Sect. 6.6.3 obeys the same symmetry class as the Dirac operator in the continuum [68, 69, 101, 102]. However, the (projected) spectrum of the overlap operator obtained from lattice simulations is purely imaginary. Therefore, a comparison between the lattice results and RMT requires the eigenvalues of the overlap operator to be rotated onto the real axis. The projection and the rotation are implicity assumed in the following.

Yet another step has to be made for the comparison. The Banks-Casher relation connects the chiral condensate with the spectral density $\rho(\lambda)$ of the Dirac operator in the infinite volume limit. One has [103]

$$\Sigma = \lim_{\lambda\to 0} \lim_{m_f\to 0} \lim_{V\to\infty} \pi\rho(\lambda) . \tag{7.14}$$

In a finite volume V the spectrum of the Dirac operator is discrete and the lowest-lying eigenvalues are inversely proportional to the volume, $\lambda_i \sim 1/V$ [104]. The mean level spacing at the lower edge of the spectrum is $\Delta\lambda \simeq 1/V\rho(\lambda \to 0) = \pi/\Sigma V$, and the distribution of the eigenvalues is sensitive to the topological charge ν associated with the gauge field background. The microscopic spectral density $\rho_{\mathrm{S}}(z)$ derived from RMT corresponds to a "magnification" of $\rho(\lambda \to 0)$ by a factor ΣV and performing the infinite volume limit (the same applies to the distribution of the lowest-lying eigenvalue $\rho_{\min}$),

$$\rho_{\mathrm{S/min},\nu}(z) = \lim_{V\to\infty} \frac{1}{\Sigma V} \rho_\nu\left(\frac{z}{\Sigma V}\right) . \tag{7.15}$$

On the lattice one is restricted to simulations of a finite volume V. If one introduces the dimensionless rescaled variable $z = \lambda\Sigma V$ then a fit of the density $\rho_{\mathrm{ov}}(\lambda)$ of the (few) lowest-lying eigenvalue(s) of

[1] All functions were evaluated numerically using Scientific Python `SCIPY` [98], the GNU Scientific Library `GSL` [99], and `CUBA` [100] for multi-dimensional integration based on the Cuhre algorithm.

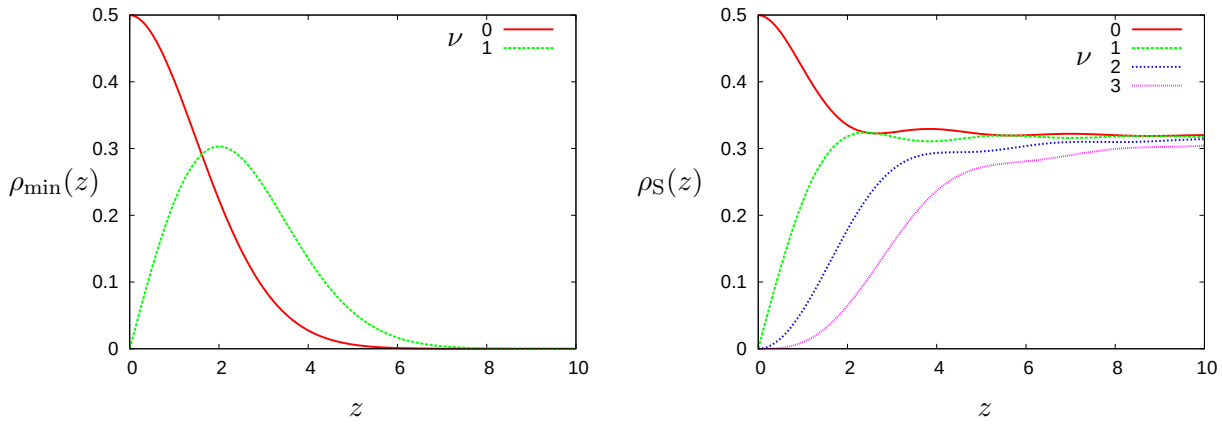

Figure 7.1: Distribution of the lowest-lying eigenvalue and microscopic spectral density associated with the orthogonal ensemble ($\beta = 1$). The left panel shows the distribution of the lowest-lying eigenvalue Eq. (7.13) for the number of zero modes $\nu = 0$ and 1 in the spectrum of the Dirac operator. The right panel shows the microscopic spectral density Eq. (7.12) for $\nu = 0$, 1, 2, and 3. In the limit $z \to 0$ the density is non-vanishing if $\nu = 0$ and suppressed otherwise.

the overlap operator to RMT subject to

$$\rho_{\rm S/min,\nu}(z) = \frac{1}{\Sigma V}\rho_{\rm ov,\nu}\left(\frac{z}{\Sigma V}\right) \tag{7.16}$$

allows for (i) test of universality in the spectrum of the overlap operator and (ii) determination of the chiral condensate obtained from simulations of the ε-regime at finite lattice spacing. The (rescaled) eigenvalue density of the overlap operator should be described by chiral random matrix theory up to some maximal scale $z_{\rm max} \sim \sqrt{V}$ where the non-zero momentum modes of the pion fields contribute to the QCD partition function [95]. Lattice simulations of the overlap operator were carried out for several simulation parameters in this study. The results are presented in Chap. 8.

7.3 Non-Hermitian chiral random matrix theory

In the continuum theory the massless Dirac operator with baryon chemical potential $\mu_{\rm phys}$ reads

$$D(\mu_{\rm phys}) = \gamma_\nu(\partial_\nu + {\rm i}gA_\nu) + \mu_{\rm phys}\gamma_4 \ . \tag{7.17}$$

In the chiral representation, cf. Eq. (7.3), the operator reads

$$D = \begin{pmatrix} 0 & {\rm i}W + \mu_{\rm phys} \\ {\rm i}W^\dagger + \mu_{\rm phys} & 0 \end{pmatrix} \tag{7.18}$$

with matrix elements W_{mn}. In the limit $\mu_{\rm phys} \to 0$ this operator is anti-Hermitian. However, the combination $\mu_{\rm phys}\gamma_4$ is Hermitian for real chemical potential $\mu_{\rm phys} \in \mathbb{R}$. The chemical potential renders the Dirac operator non-Hermitian, one has $D(\mu_{\rm phys}) = -D(-\mu_{\rm phys})^\dagger \neq -D(\mu_{\rm phys})^\dagger$, and therefore its spectrum gets non-trivially distorted into the complex plane.

The non-Hermitian random matrix model associated with SU(2) gauge theory and fermions in the fundamental representation and baryon chemical potential has been solved in Refs. [71, 72] (for an even number of quark flavours). The proposed random matrix formulation for the massless Dirac

operator in the chiral representation is[2]

$$D = \begin{pmatrix} 0 & P+\mu Q \\ -P^T+\mu Q^T & 0 \end{pmatrix} = \begin{pmatrix} 0 & P+\mu Q \\ -(P^T-\mu Q^T) & 0 \end{pmatrix} \equiv \begin{pmatrix} 0 & A \\ -B^T & 0 \end{pmatrix} , \tag{7.19}$$

where $P, Q \in \mathbb{R}^{N\times(N+\nu)}$ are independent rectangular real-valued matrices without further symmetry restrictions. Here the symmetry breaking parameter μ is associated with the baryon chemical potential in QCD. Note that this model for the Dirac operator is anti-symmetric (anti-Hermitian) in the limit $\mu \to 0$. Otherwise one has $D(\mu) = -D(-\mu)^\dagger$, which mimics the symmetry of the Dirac operator in the continuum.

The eigenvalues Λ of the matrix D are determined by the characteristic equation

$$0 = \det(\Lambda 1_{2N+\nu} - D) = \Lambda^\nu \det[\Lambda^2 1_N + AB^T] = \Lambda^\nu \prod_{i=1}^{N} (\Lambda^2 + \Lambda_i^2) . \tag{7.20}$$

The matrix elements of the $N \times N$ matrix AB^T are real. The eigenvalues Λ^2 of AB^T are either real or come in complex conjugate pairs [71]. The matrix D itself has ν zero-valued eigenvalues $\Lambda_i = 0$ and $2N$ eigenvalues coming in pairs $\Lambda = \pm \mathrm{i}\Lambda_i$. Thus three categories of eigenvalues are classified:

(*i*) Real pairs $\Lambda = \pm \mathrm{i}\Lambda_i \in \mathbb{R}$ for $\Lambda_i^2 < 0$.

(*ii*) Imaginary pairs $\Lambda = \pm \mathrm{i}\Lambda_i \in \mathrm{i}\mathbb{R}$ for $\Lambda_i^2 > 0$.

(*iii*) Complex quadruplets $\Lambda = \pm \mathrm{i}\Lambda_i, \pm \mathrm{i}\Lambda_i^* \in \mathbb{C}\backslash\{\mathbb{R} \cup \mathrm{i}\mathbb{R}\}$ for pairs $\Lambda_i^2, \Lambda_i^{*2}$.

Given the model for the Dirac operator with chemical potential, the partition function associated with the non-Hermitian random matrix model and N_f quark flavours is

$$Z_\nu \sim \int \mathrm{d}P \int \mathrm{d}Q \exp\left[-\frac{N}{2}\,\mathrm{Tr}(PP^T + QQ^T)\right] \prod_{f=1}^{N_f} \det \begin{pmatrix} m_f & P+\mu_f Q \\ -P^T+\mu_f Q^T & m_f \end{pmatrix} , \tag{7.21}$$

with the integration performed over all matrix elements. For each quark flavour the parameter μ_f denotes the associated quark chemical potential and m_f is the quark mass. In the limit $\mu_f \to 0$ the matrix Q decouples and the Hermitian random matrix model introduced in Sect. 7.2 is obtained.

In the following all quark chemical potentials are set equal, $\mu \equiv \mu_f$. In the thermodynamic limit $N \to \infty$ the chemical potential μ and the quark masses m_f are taken to zero such that the products $N\mu^2$ and Nm_f remain finite. If one introduces the transformations

$$\hat{\mu}^2 \equiv 2N\mu^2 \ = 4\mu_{\mathrm{phys}}^2 F^2 V \tag{7.22}$$

$$\hat{m}_f \equiv 2Nm_f = 2m_{f,\mathrm{phys}} GV , \tag{7.23}$$

then the non-trivial partition function Eq. (7.21) indeed maps to the static part of the chiral Lagrangian Eq. (6.50), see Ref. [73]. The parameters μ_{phys} and $m_{f,\mathrm{phys}}$ refer to the physical values of the chemical potential and the quark masses, respectively. The physical volume is denoted by V, while F and G are low-energy constants.

It is possible to derive expressions for the microscopic spectral density from the partition function Eq. (7.21). The eigenvalues of the model Dirac operator with chemical potential Eq. (7.20) scatter non-trivially on the real and imaginary axis, and also in the complex plane. Consequently, for each category of eigenvalues a spectral density is associated. In the thermodynamic limit the eigenvalues Λ

[2] Note that the model for the Dirac operator provided in Refs. [71, 72] is chosen symmetric.

of the model Dirac operator are rescaled by $\xi \equiv 2N\Lambda$. The microscopic spectral densities associated with the real and imaginary eigenvalues $\xi \in (\mathrm{i})\mathbb{R}$ in the case $N_f = 0$ are given by [72]

$$\rho_{\mathrm{S}}^{(\mathrm{i})\mathbb{R}}(\xi) = -2|\xi| G_w(-\xi^2, -\xi^2) \,, \tag{7.24}$$

with the non-analytic expression

$$\begin{aligned} G_w(x,x') = &- \frac{\hat{h}_w(x')}{[\mathrm{sgn}(x')]^{\nu/2}} \Bigg\{ \left((-\mathrm{i})^\nu \int_{-\infty}^{0} \mathrm{d}y + \frac{2}{[\mathrm{sgn}(x')]^{\nu/2}} \int_0^{x'} \mathrm{d}y \right) \mathcal{K}_w(x,y) \hat{h}_w(y) \\ &- \frac{1}{32\sqrt{\pi}} \Bigg[- \frac{\mathrm{e}^{-\hat{\mu}^2}}{\hat{\mu}} J_\nu\left(\sqrt{x}\right) + \frac{2\hat{\mu}^\nu}{\Gamma\left(\frac{\nu+1}{2}\right)} \int_0^1 \mathrm{d}s \, \mathrm{e}^{-\hat{\mu}^2 s^2} s^{\nu+2} \\ &\times \left(\frac{\sqrt{x}}{2} E_{\frac{1-\nu}{2}}(\hat{\mu}^2 s^2) J_{\nu+1}\left(s\sqrt{x}\right) - \hat{\mu}^2 s \left(E_{\frac{-1-\nu}{2}}(\hat{\mu}^2 s^2) - E_{\frac{1-\nu}{2}}(\hat{\mu}^2 s^2) \right) J_\nu\left(s\sqrt{x}\right) \right) \Bigg] \Bigg\} \,. \end{aligned} \tag{7.25}$$

Here the kernel is given by

$$\mathcal{K}_w(x,y) = \frac{1}{256\pi\hat{\mu}^2} \int_0^1 \mathrm{d}s \; s^2 \mathrm{e}^{-2\hat{\mu}^2 s^2} \left[\sqrt{x} J_{\nu+1}\left(s\sqrt{x}\right) J_\nu\left(s\sqrt{y}\right) - \sqrt{y} J_{\nu+1}\left(s\sqrt{y}\right) J_\nu\left(s\sqrt{x}\right) \right] \,, \tag{7.26}$$

with Bessel functions J_ν of the first kind of integer order. The weight function

$$\hat{h}_w(x) = \mathrm{e}^{x/8\hat{\mu}^2} 2 K_{\frac{\nu}{2}}\left(\frac{|x|}{8\hat{\mu}^2} \right) \tag{7.27}$$

comes with Basset function $K_{\frac{\nu}{2}}$ of half-integer order and

$$\mathrm{E}_n(x) = \int_1^\infty \mathrm{d}t \; t^{-n} \mathrm{e}^{-xt} \tag{7.28}$$

denotes the exponential integral. The densities associated with the real and imaginary Dirac eigenvalues differ, one has $\rho_{\mathrm{S}}^{\mathbb{R}}(\xi) \neq \rho_{\mathrm{S}}^{\mathrm{i}\mathbb{R}}(\mathrm{i}\xi)$. Due to the pairing of real and imaginary eigenvalues the densities are symmetric, i.e., $\rho_{\mathrm{S}}^{\mathbb{R}}(\xi) = \rho_{\mathrm{S}}^{\mathbb{R}}(-\xi)$ and $\rho_{\mathrm{S}}^{\mathrm{i}\mathbb{R}}(\mathrm{i}\xi) = \rho_{\mathrm{S}}^{\mathrm{i}\mathbb{R}}(-\mathrm{i}\xi)$.[3] The microscopic spectral density is shown for $\nu = 0$ and 1 zero modes of the Dirac operator in Fig. 7.2 as function of the rescaled symmetry breaking parameter $\hat{\mu}$. Note that in the limit $\hat{\mu} \to 0$ all Dirac eigenvalues are purely imaginary and the spectrum is described by Hermitian RMT introduced in Sect. 7.3.

The microscopic spectral density associated with the complex Dirac eigenvalues $\xi \in \mathbb{C} \backslash \{\mathbb{R} \cup \mathrm{i}\mathbb{R}\}$ is given by [71, 72]

$$\rho_{\mathrm{S}}^{\mathbb{C}}(\xi) = 4|\xi|^2 \hat{g}_w(-\xi^{*2}, -\xi^2) \mathcal{K}_w(-\xi^2, -\xi^{*2}) \,, \tag{7.29}$$

with the kernel $\mathcal{K}_w$ defined in Eq. (7.26) and weight function

$$\begin{aligned} \hat{g}_w(z,z^*) = &- 4\mathrm{i} \, \mathrm{sgn}(\mathrm{Im}\, z) \exp\left(\frac{\mathrm{Re}\, z}{4\hat{\mu}^2} \right) \\ &\times \int_0^\infty \frac{\mathrm{d}t}{t} \exp\left[-\frac{t(z^2 + z^{*2})}{64\hat{\mu}^2} - \frac{1}{4t} \right] K_{\frac{\nu}{2}}\left(\frac{t|z|^2}{32\hat{\mu}^4} \right) \mathrm{erfc}\left(\frac{\sqrt{t}|\,\mathrm{Im}\, z|}{4\hat{\mu}^2} \right) \,. \end{aligned} \tag{7.30}$$

The microscopic spectral density describes the distribution of the eigenvalues in the complex plane and is indeed symmetric for the arguments $\pm\xi$, $\pm\xi^*$. Plots of the density for $\nu = 0$ and 1 zero

[3] It is pointed out here that Eq. (7.25) may be cast into a representation with Bessel functions J_ν and $K_{\frac{\nu}{2}}$ for real arguments only. This representation is well suited for numerical integration.

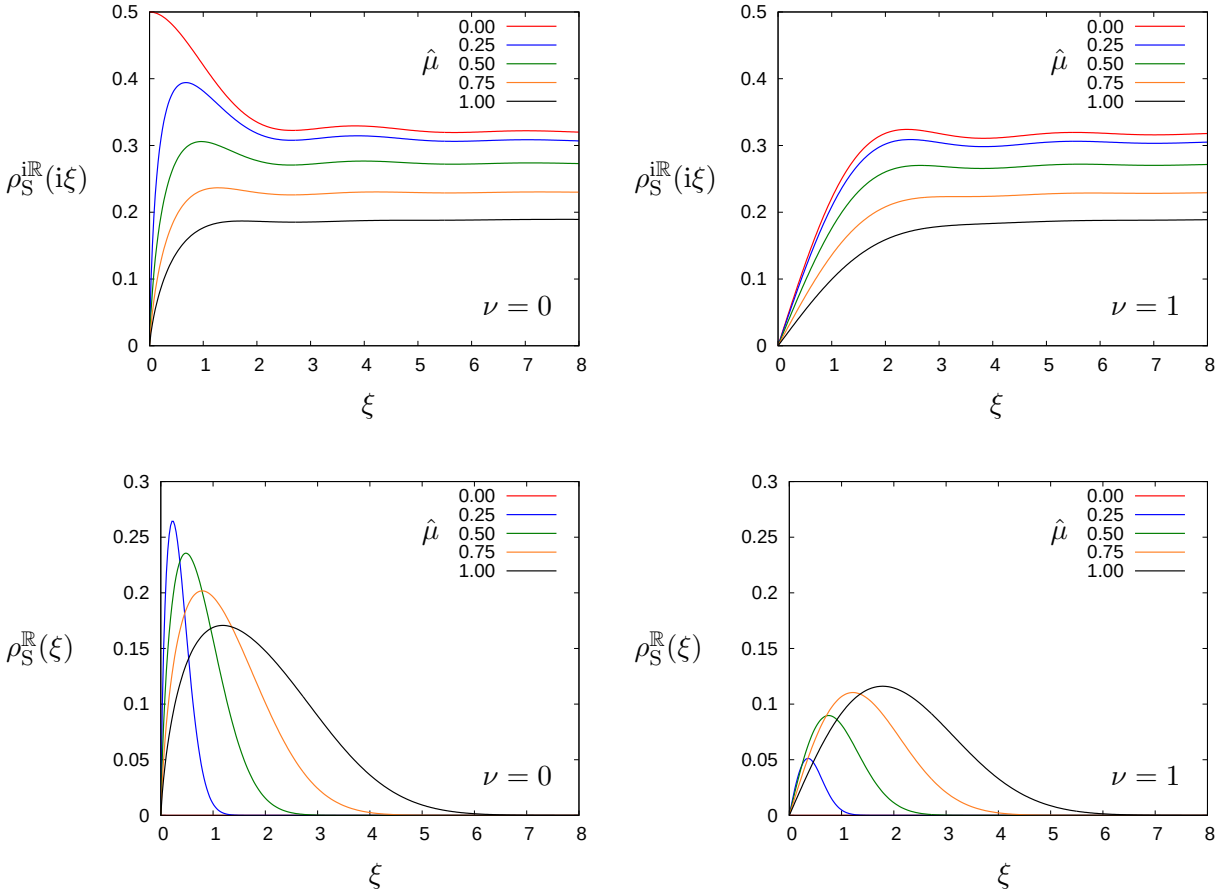

Figure 7.2: Microscopic spectral density associated with real and imaginary eigenvalues. The symmetry-breaking parameter is chosen $\hat{\mu} = 0.0$, 0.25, 0.5, 0.75, and 1.0. The left column shows the microscopic spectral density Eq. (7.24) for purely imaginary eigenvalues on top and for real eigenvalues on bottom for $\nu = 0$ zero modes of the Dirac operator. The right column shows the densities for $\nu = 1$.

modes of the Dirac operator as function of the rescaled symmetry-breaking parameter $\hat{\mu}$ are shown in Fig. 7.3.[4]

The effect of the parameter $\hat{\mu}$ on the spectrum is clearly visible in Figs. 7.2 and 7.3. At $\hat{\mu} = 0$ the Dirac operator is anti-Hermitian and its paired eigenvalues are purely imaginary. If $\hat{\mu}$ is switched on, the eigenvalues scatter (as pairs) onto the real axis and (as quadruplets) into the complex plane. This effect is continuous and increases with $\hat{\mu}$, such that an increasing number of eigenvalues accumulate on the real axis and in the complex plane. The (projected) overlap operator with chemical potential μ_{phys} introduced in Sect. 6.7 obeys the same symmetry class as the continuum Dirac operator and is therefore associated with the orthogonal emsemble [102]. For lattice simulations within the domain $m_\pi, \mu \ll 1/L \ll \Lambda_{\text{QCD}}$, the spectrum of the chiral lattice Dirac operator near zero is expected to be adequately described by the non-Hermitian random matrix model. A fit subject to

$$\rho_{S,\nu}^{(i)\mathbb{R}}(\xi) = \frac{1}{\Sigma V}\rho_{ov,\nu}^{(i)\mathbb{R}}\left(\frac{\xi}{\Sigma V}\right), \quad \rho_{S,\nu}^{\mathbb{C}}(\xi) = \frac{1}{(\Sigma V)^2}\rho_{ov,\nu}^{\mathbb{C}}\left(\frac{\xi}{\Sigma V}\right), \tag{7.31}$$

[4] Bessel functions with complex arguments were evaluated with the `SLATEC` Fortran library [105].

where $\hat{\mu} = 2\mu_{\rm phys}F\sqrt{V}$, allows for (i) test of universality in the spectrum of the overlap operator and (ii) determination of the chiral condensate and the pion decay constant obtained from simulations of the ε-regime at finite lattice spacing. The lattice simulations are presented in the next chapter.

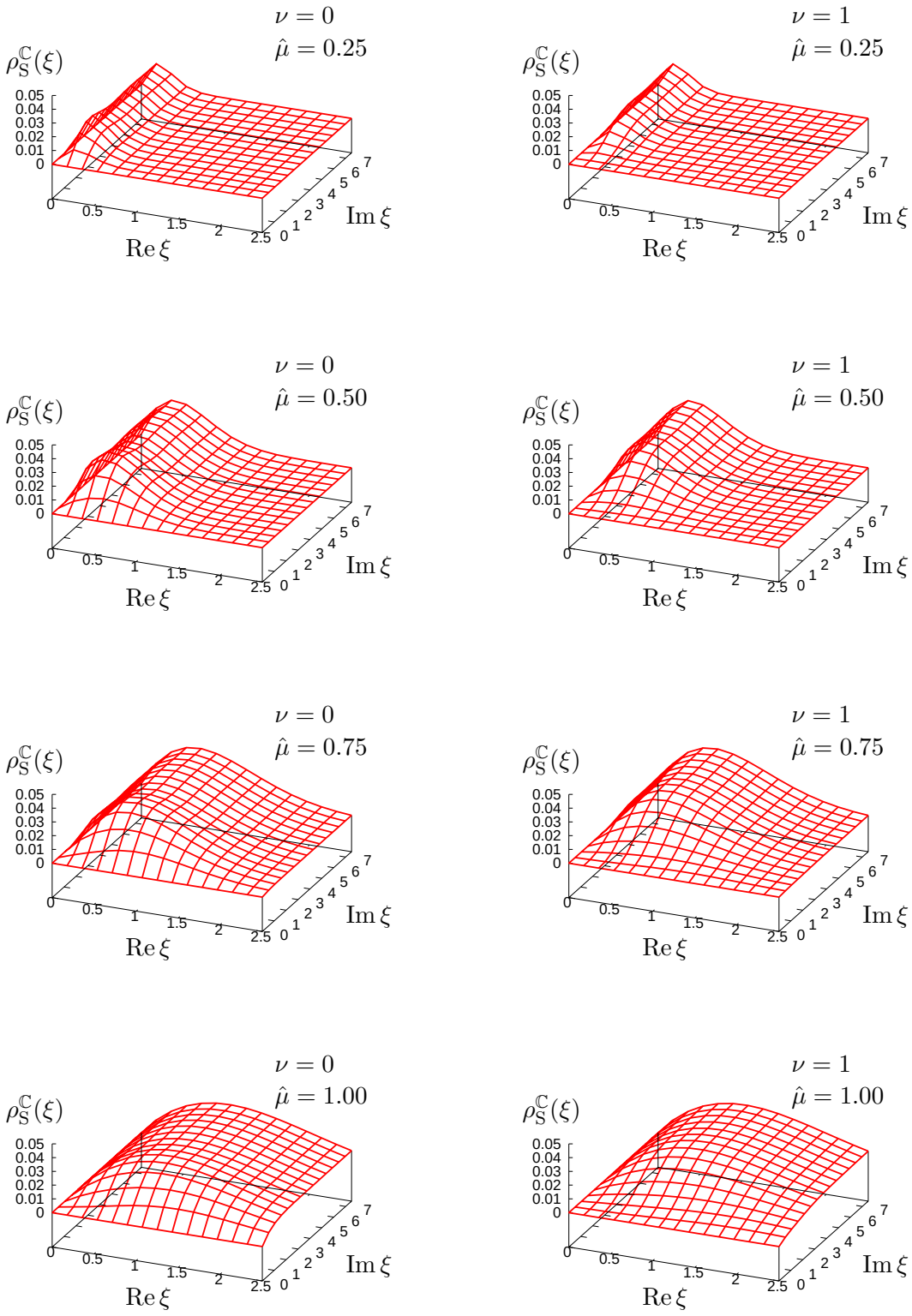

Figure 7.3: Microscopic spectral density associated with complex eigenvalues. The symmetry-breaking parameter is chosen $\hat{\mu} = 0.25$, 0.5, 0.75, and 1.0. The left column shows the microscopic spectral density Eq. (7.29) for $\nu = 0$ zero modes of the Dirac operator. The right column shows the density for $\nu = 1$.

Chapter 8

Evaluation and Results

8.1 Two-colour QCD at zero chemical potential

In this section the spectrum of the overlap operator at zero chemical potential is discussed. The spectral densities obtained from lattice simulations are compared to the universal predictions provided by Hermitian random matrix theory introduced in Sect 7.2. Simulations of quenched two-colour QCD were carried out on lattice volumes $V = 8^4, 10^4$, and 12^4. The coupling strength β of the gauge fields, cf. Eq. (6.66), was chosen such that the associated physical volume $V_{\text{phys}} = Va^4$ (here a is the lattice spacing) is approximately the same for all lattice volumes. Estimates for the values of β were obtained by polynomial interpolation of the string tension σa^2 as function of the coupling strength. The data was taken from Ref. [106] and is shown in Fig. 8.1. For each lattice volume the gauge field configurations were computed only once using `CHROMA` [107], and then (re-)used for evaluation of the spectrum of the overlap operator with different choices for the Wilson mass. The simulation parameters are summarized in Table 8.1. Only the lowest-lying eigenvalues of the overlap operator were of interest for this study. The spectra were evaluated using the Krylov-Ritz method described in Appendix B.1.

8.1.1 Choice of the Wilson mass

With the choice for the lattice volume V and the coupling strength β no other simulation parameters are required for creation of the gauge fields in the quenched approximation. However, for the evaluation of the spectrum of the overlap operator

$$D_{\text{ov}} = 1 + \gamma_5 \varepsilon[\gamma_5 D_{\text{W}}(m_{\text{W}})] \tag{8.1}$$

with Hermitian Wilson-Dirac kernel a choice for the Wilson mass m_{W} has to be made. There is no a-priori best choice for this mass parameter. As will be shown in the following, the choice for the mass has some impact on the spectra of the lattice Dirac operators.

The impact of the Wilson mass on the spectrum of the Hermitian Wilson-Dirac operator Eq. (6.85) and the overlap operator Eq. (8.1) is illustrated in Fig. 8.2. The plots shows the flow of the eigenvalues of the operators as function of the mass. The spectra were evaluated on a sample gauge field configuration on the 8^4 lattice. The eigenvalues of the overlap operator are mapped from the Ginsparg-Wilson circle to the imaginary axis by application of the stereographic projection Eq. (6.106). Below some critical value of the Wilson mass both operators exhibit a spectral gap near zero. As the Wilson mass is increased, the magnitude of the lowest-lying eigenvalues of the Wilson-Dirac operator continuously decrease and the eigenvalues start to fluctuate around the real axis. Whilst the magnitude of the eigenvalues of the Hermitian Wilson-Dirac operator change smoothly as function of the mass, this does not apply to the eigenvalues of the overlap operator. In fact, each

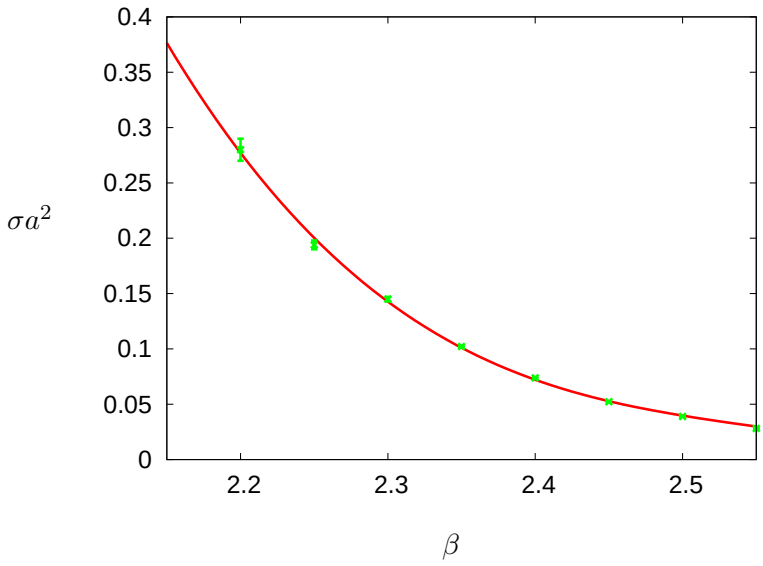

Figure 8.1: String tension σa^2 as function of the coupling strength β (here a is the lattice spacing). The solid line corresponds to a fit of the data points taken from Ref. [106] to a low-order polynomial.

time one of the eigenvalues of the Hermitian Wilson-Dirac operator crosses the real axis, the index of the overlap operator changes by ± 1 and its spectrum gets rearranged.[1] However, the index is intimately related to the number of eigenvalues of the kernel operator with negative sign (n_-^{W}) and positive sign (n_+^{W}). One has

$$\mathrm{Tr}(\gamma_5 D_{\mathrm{ov}}) = \mathrm{Tr}\left[\gamma_5 + \varepsilon(\gamma_5 D_{\mathrm{W}})\right] = \mathrm{Tr}\left[\varepsilon(\gamma_5 D_{\mathrm{W}})\right] = n_+^{\mathrm{W}} - n_-^{\mathrm{W}} = 2\,\mathrm{index}\,D_{\mathrm{ov}}\ , \tag{8.2}$$

where the matrix sign function $\varepsilon(\gamma_5 D_{\mathrm{W}})$ only takes the values ± 1, and is ill-defined if one or more of the eigenvalues are exactly zero (the latter situation was not observed in this study). A sign change of one of the eigenvalues of the kernel operator results in a net change $n_+^{\mathrm{W}} - n_-^{\mathrm{W}} = \pm 2$. This explains the changes in the index of the overlap operator by exactly ± 1 as the Wilson mass is increased. Table 8.1 summarizes the ensemble average of the relative frequency $P(\nu)$ of the number ν of zero modes of the overlap operator obtained for different choices of the Wilson mass and different lattice volumes. Roughly 90% of the spectra of the overlap operator exhibit 0 or 1 zero mode on the 8^4 lattice for $m_{\mathrm{W}} = 1.2$. The frequency of zero modes increases as the mass is increased and the corresponding distribution freezes out in the limit $m_{\mathrm{W}} \to 2$.[2] One furthermore observes that the variations in $P(\nu)$ tend to be milder for larger lattice volumes (and thus smaller lattice spacings). This hints for the variations to be an artifact on the lattice that disappears in the continuum limit.

The Wilson mass also influences the distribution of the eigenvalues of the lattice Dirac operators. Formally the normalized distribution of the i-th eigenvalue λ_i is given as ensemble average over the gauge field configurations by

$$\rho^{(i)}(\lambda) = \langle \delta(\lambda - \lambda_i) \rangle_{\mathrm{G}}\ , \quad \int \mathrm{d}\lambda\, \rho^{(i)}(\lambda) = 1\ . \tag{8.3}$$

[1] The index of the overlap operator was obtained by evaluation of the zero modes, as discussed in Sect. 6.6.3.

[2] An analogous observation has been made in simulations of quenched QCD with gauge group SU(3), see Ref. [108]. There it was observed that the topological susceptibility $\chi_{\mathrm{top}} = \langle \nu^2 \rangle_{\mathrm{G}} / V$ is approximately constant in this mass limit.

V	β	No. config.	m_{W}	$P(\nu=0)$	$P(\nu=1)$	$P(\nu=2)$	$P(\nu=3)$	$P(\nu\geq 4)$
8^4	2.2	3072	1.2	0.405	0.458	0.117	0.018	0.001
		8976	1.4	0.217	0.373	0.241	0.111	0.059
		8192	1.5	0.191	0.326	0.249	0.135	0.100
		8192	1.6	0.169	0.307	0.241	0.149	0.134
		9216	1.7	0.155	0.295	0.233	0.153	0.163
		8444	1.8	0.153	0.283	0.225	0.163	0.175
		6144	1.9	0.142	0.287	0.223	0.163	0.185
		6961	2.0	0.145	0.283	0.224	0.163	0.185
10^4	2.266	2048	1.4	0.146	0.281	0.220	0.161	0.193
		2048	1.6	0.134	0.229	0.206	0.167	0.265
		2048	1.8	0.113	0.236	0.196	0.163	0.292
		1918	2.0	0.112	0.230	0.198	0.164	0.296
12^4	2.318	2048	1.4	0.117	0.227	0.188	0.155	0.314
		2048	1.6	0.116	0.204	0.179	0.157	0.345
		799	1.8	0.106	0.189	0.163	0.171	0.370

Table 8.1: Parameters for the simulations of quenched two-colour QCD at zero chemical potential: lattice volume V, coupling strength β and number of gauge field configurations evaluated at fixed Wilson mass m_{W}. Here $P(\nu)$ is the relative frequency of the number ν of zero modes found in the spectra of the overlap operator.

The normalized spectral density is given by

$$\rho(\lambda) = \langle \sum_k \delta(\lambda - \lambda_k) \rangle_{\mathrm{G}} \ , \quad \int \mathrm{d}\lambda \ \rho(\lambda) = 4N_c V \ , \tag{8.4}$$

with lattice volume V and the number of colours $N_c = 2$ in this study. These (continuous) distributions were approximated by binning of the lattice data in form of a histogram (and proper normalization). The unnormalized statistical error on the i-th bin is $\mathcal{O}(\sqrt{N_i})$. Here N_i is the number of eigenvalues that fall into bin number i. Minimization of statistical fluctuations requires the evaluation of a large number of spectra, as the relative error on the i-th bin is $\mathcal{O}(1/\sqrt{N_i})$.

The spectral densities derived from RMT, cf. Sect. 7.2, are valid for the Dirac operator being Hermitian. Therefore the densities apply to real eigenvalues. However, the spectrum of the overlap operator is purely imaginary after stereographic projection. To avoid any confusion with the choice for the Dirac operator being Hermitian or anti-Hermitian in the following it is implicitely assumed that the eigenvalues of the overlap operator are rotated onto the real axis. In Fig. 8.3 the lower edge of the spectral density and the distributions of the few lowest-lying eigenvalues of both the Hermitian Wilson-Dirac kernel and the overlap operator are shown. The spectra are selected by the number of zero modes of the overlap operator. Note that the non-zero eigenvalues of the overlap operator occur in (originally complex conjugate) pairs, thus the spectrum is symmetric. The Hermitian Wilson-Dirac operator does not obey a lattice chiral symmetry, and its eigenvalues do not occur in pairs. With increasing Wilson mass the average magnitude of the eigenvalues of both operators decrease. However, the spectral density of the overlap operator evolves from a non-uniform distribution to a uniform one where the spectral density (and thus the level density) near zero is approximately constant. Although at $m_{\mathrm{W}} = 1.2$ the spectral density does not vanish near zero, the spectrum does not mimic the predictions of random matrix theory introduced in Sect. 7.2. However, if the Wilson mass is chosen in the range $1.4 \leq m_{\mathrm{W}} \leq 2$ the spectrum of the overlap operator near zero is uniform

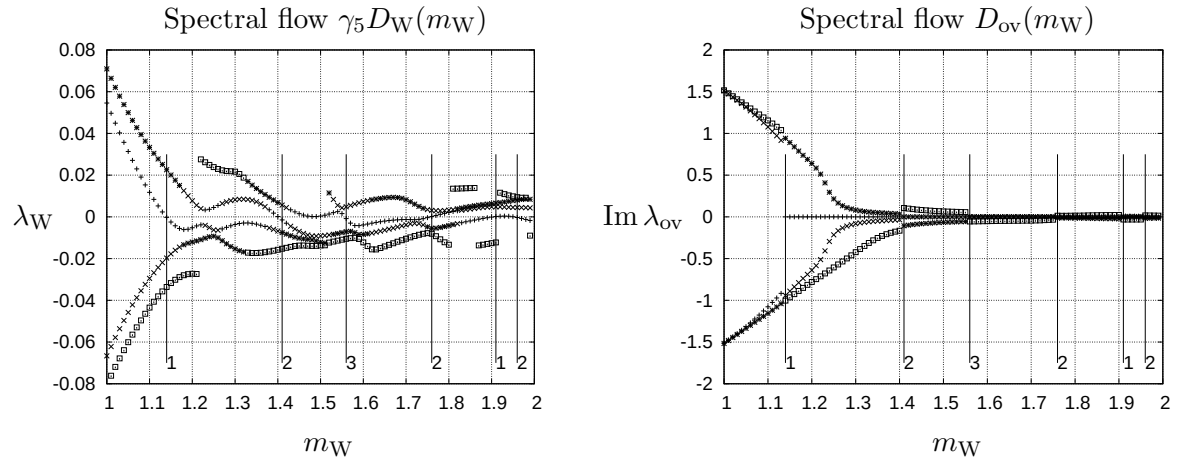

Figure 8.2: Spectral flow of the four lowest-lying eigenvalues of the Hermitian Wilson-Dirac and overlap operator as function of the Wilson mass m_{W} evaluated on a sample gauge field configuration on the 8^4 lattice. The left panel shows the flow of the (real) eigenvalues λ_{W} of the Hermitian Wilson-Dirac operator. The discontinuities in the plot only arise from the interchange of the magnitude of the eigenvalues. In fact the flow is continuous. The right panel shows the corresponding flow of the (purely imaginary) eigenvalues λ_{ov} of the overlap operator after stereographic projection. Each time one eigenvalue of the Wilson-Dirac operator crosses the real axis the index of the overlap operator changes by ± 1, which furthermore results in a discontinuous change of the spectrum. The change of the index is indicated by vertical bars labeled with the number of zero modes of the overlap operator.

and does match with RMT.

8.1.2 Analysis of the distribution of the lowest-lying eigenvalue

The distribution of the (projected and rotated) lowest-lying non-zero eigenvalue λ_{min} of the overlap operator is expected to be a universal quantity described by RMT (cf. Sect. 7.2). In this study universality was found in the spectra for the choices of the Wilson mass $1.4 \leq m_{\mathrm{W}} \leq 2$. A fit of the distribution of the lowest-lying eigenvalue to the prediction allowed to determine the chiral condensate Σ from the lattice simulations. The chiral condensate was obtained by a fit subject to (limited to $\nu = 0, 1$)

$$\rho_{\mathrm{min}}(z) = \frac{1}{\Sigma V} \rho_{\mathrm{min,ov}} \left(\frac{z}{\Sigma V} \right) \ . \tag{8.5}$$

Here V is the lattice volume, ρ_{min} is the continuous eigenvalue distribution Eq. (7.13) predicted by RMT, and $\rho_{\mathrm{min,ov}}$ is the distribution obtained by binning of the lattice data as described in Sect. 8.1.1.

The results of the fits are summarized in Table 8.2 and shown in Figs. 8.4 and 8.5 for all lattice volumes investigated. The Wilson mass was chosen in the range $1.4 \leq m_{\mathrm{W}} \leq 2$. The chiral condensate was determined by a simultaneous fit to both the distributions for $\nu = 0, 1$ using a single fit parameter. Discretization errors introduced by binning of the lattice data were compensated for by fitting of the densities integrated over the bin size.[3] The statistical errors on the chiral condensate were estimated by the bootstrap method sketched in Appendix B.2. In general, the lattice data is well described by the universal RMT results, although minor deviations from the predictions do

[3] It turned out that the results of the fits are stable against variations in the number of bins chosen if $\mathcal{O}(300)$ eigenvalues are available.

V	m_{W}	ν	Σ	χ^2/dof
8^4	1.4	$0+1$	0.00543(5)	1.0(6)
	1.5	$0+1$	0.00876(9)	2.1(7)
	1.6	$0+1$	0.01307(15)	3.5(8)
	1.7	$0+1$	0.01786(21)	3.3(8)
	1.8	$0+1$	0.02292(29)	2.8(8)
	1.9	$0+1$	0.0283(4)	2.8(8)
	2.0	$0+1$	0.0340(5)	1.6(7)
10^4	1.4	$0+1$	0.00655(16)	2.0(1.0)
	1.6	$0+1$	0.0129(4)	4.0(1.7)
	1.8	$0+1$	0.0218(6)	1.5(8)
	2.0	$0+1$	0.0306(9)	1.2(7)
12^4	1.4	$0+1$	0.00694(20)	1.5(2.0)
	1.6	$0+1$	0.0127(5)	2.5(1.0)
	1.8	$0+1$	0.0210(17)	4.5(2.2)

Table 8.2: Results for the chiral condensate Σ obtained from simultaneous fits to the distribution of the lowest-lying eigenvalue Eq. (7.13) for $\nu = 0, 1$. Here V is the lattice volume and m_{W} is the Wilson mass. The statistical errors were estimated by the bootstrap method sketched in Appendix B.2.

occur (for some particular values of the Wilson mass). These deviations are also reflected in the values of χ^2/dof. Note that the statistics of only 79 (151) eigenvalues in the case $\nu = 0$ (1) obtained on the 12^4 lattice and the mass chosen $m_{\mathrm{W}} = 1.8$ is clearly insufficient to approximate the eigenvalue distribution.

At first glance it is intriguing that the chiral condensate shows such a strong dependence on the Wilson mass. Clearly the dependence gets milder for simulations towards the continuum limit. On a qualitative level the variations in the chiral condensate originate from the average magnitude of the lowest-lying eigenvalue, which decreases as the mass is increased. However, the variations become milder for smaller lattice spacings and therefore are considered to be some lattice artefact.

8.1.3 Analysis of the spectral density

For this study a large number of (projected) spectra of the overlap operator were analyzed. The statistics was sufficient to compare those spectra which exhibit $\nu = 0 \ldots 3$ zero modes to the microscopic spectral density, which is another universal quantity derived from RMT (cf. Sect. 7.2). Again, universality was found in the spectra for the choices of the Wilson mass $1.4 \leq m_{\mathrm{W}} \leq 2$ and allowed to determine the chiral condensate by a fit subject to (for a given ν)

$$\rho_{\mathrm{S}}(z) = \frac{1}{\Sigma V} \rho_{\mathrm{ov}} \left(\frac{z}{\Sigma V} \right) \ . \tag{8.6}$$

Here ρ_{S} is the microscopic spectral density Eq. (7.12) and ρ_{ov} is the lattice data binned into a histogram according to Sect. 8.1.1.[4] Multiple densities (with different ν) were fitted simultaneously using a single fit parameter. Discretization errors were compensated for by integration over the bin width. The upper limit of the fit, $z_{\max} = \lambda_{\max} \Sigma V$, was chosen such that $\chi^2/\mathrm{dof} \approx 1$. However, this choice is not unique and other methods can lead to different results. A somewhat related quantity is the expectation value of the number of eigenvalues of the overlap operator that support the universal

[4] The argument of the microscopic spectral density is real and thus the (purely imaginary) eigenvalues $\mathrm{i}\lambda$ of the overlap operator need to be rotated in the complex plane for the comparison.

V	m_{W}	ν	Σ	χ^2/dof	z_{max}	N_{ev}
8^4	1.4	0 – 3	0.00550(4)	1.2(4)	4.76	1.74
	1.5	0 – 3	0.00904(6)	1.2(3)	7.64	2.67
	1.6	0 – 3	0.01298(7)	1.1(3)	10.3	3.52
	1.7	0 – 3	0.01814(17)	1.2(4)	3.86	1.45
	1.8	0 – 3	0.02303(19)	1.2(3)	5.34	1.93
	1.9	0 – 3	0.0290(3)	1.3(4)	4.21	1.56
	2.0	0 – 3	0.0344(3)	1.1(4)	4.77	1.75
10^4	1.4	0 – 3	0.00651(4)	1.3(4)	18.6	6.18
	1.6	0 – 3	0.01242(15)	1.3(4)	9.78	3.35
	1.8	0 – 3	0.02060(28)	1.3(4)	7.72	2.69
	2.0	0 – 3	0.0287(4)	1.3(5)	7.54	2.63
12^4	1.4	0 – 3	0.00639(8)	1.3(4)	9.00	3.10
	1.6	0 – 3	0.01170(18)	1.4(4)	7.37	2.58
	1.8	0 – 1	0.0173(7)	1.8(1.0)	7.65	2.67

Table 8.3: Results for the chiral condensate Σ obtained from simultaneous fits to the microscopic spectral density Eq. (7.12) for various ν. Here V is the lattice volume and m_{W} is the Wilson mass. The rescaled variable $z_{\mathrm{max}} = \lambda_{\mathrm{max}}\Sigma V$ indicates the fit interval, and N_{ev} is the expectation value of the number of eigenvalues that contribute to the fit for $\nu = 0$. The statistical errors were estimated by the bootstrap method (cf. Appendix B.2).

distributions. One can estimate[5]

$$N_{\mathrm{ev}} = \int_0^{z_{\mathrm{max}}} \mathrm{d}z \; \rho_{\mathrm{S};\nu=0}(z) \; . \tag{8.7}$$

The results of the fits to the data obtained from the lattice simulations are summarized in Table 8.3 and plots are shown in Figs. 8.6 and 8.7. The limit z_{max} was determined independently for each choice of simulation parameters. The fit results are in reasonable agreement with the results obtained from fits to the distribution of the lowest-lying eigenvalues, cf. Table 8.2. As observed in the previous section, the chiral condensate shows a strong dependence on the choice of the Wilson mass. However, as shown in Fig. 8.8, the differences in the fitted values of the condensate diminish as one approaches the continuum limit. There is also some impact of the simulation parameters on the limit z_{max} (or, equivalently, N_{ev}). One observation is that the limit tends to increase with the lattice volume. This scaling with the volume is somewhat subtle, because the impact of the choice for the Wilson mass on z_{max} is drastic. At least on a qualitative level the explanation for this impact is rather simple: RMT predicts the fluctuations of the eigenvalues of the chiral lattice Dirac operator around the average spectral density. It applies if the spectral density near zero is uniform, i.e., the level spacing is constant. Obviously the spectral support for this portion of the density is sensitive to the choice of simulation parameters.

[5]Here $\nu = 0$ was chosen for convenience. However, this choice is not unique and one could think about other choices for this quantity.

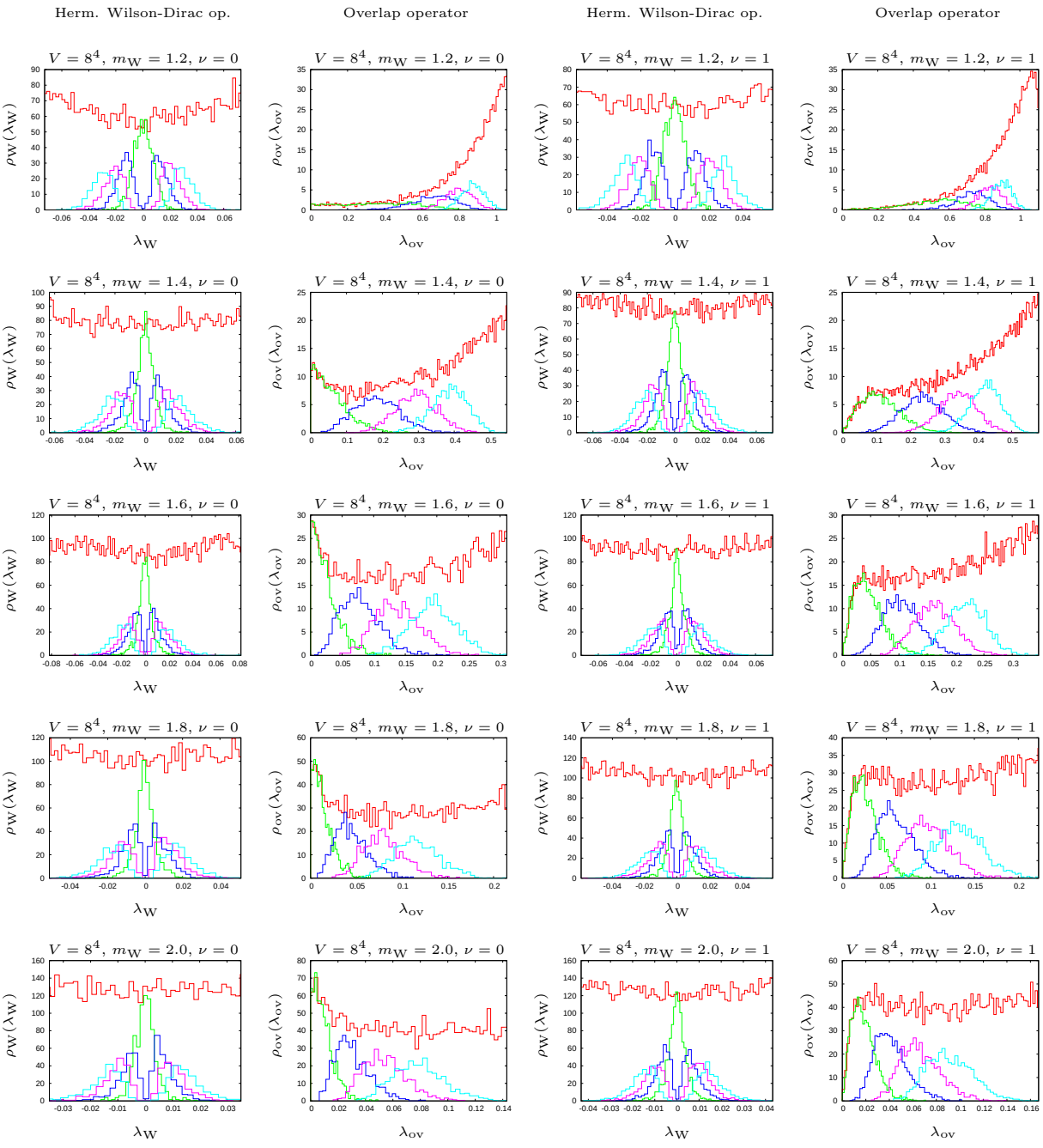

Figure 8.3: Spectral density and distribution of the eight lowest-lying eigenvalues λ_W of the Hermitian Wilson-Dirac kernel and the corresponding four lowest-lying non-zero eigenvalues λ_{ov} of the overlap operator obtained from simulations of the 8^4 lattice (after stereographic projection and rotation onto the real axis). The Wilson mass m_W is increased from 1.2 in the first row up to 2.0 in the last row in steps $\Delta m_W = 0.2$. The left two columns show the densities for $\nu = 0$, the right two columns show the densities for $\nu = 1$ zero modes of the overlap operator.

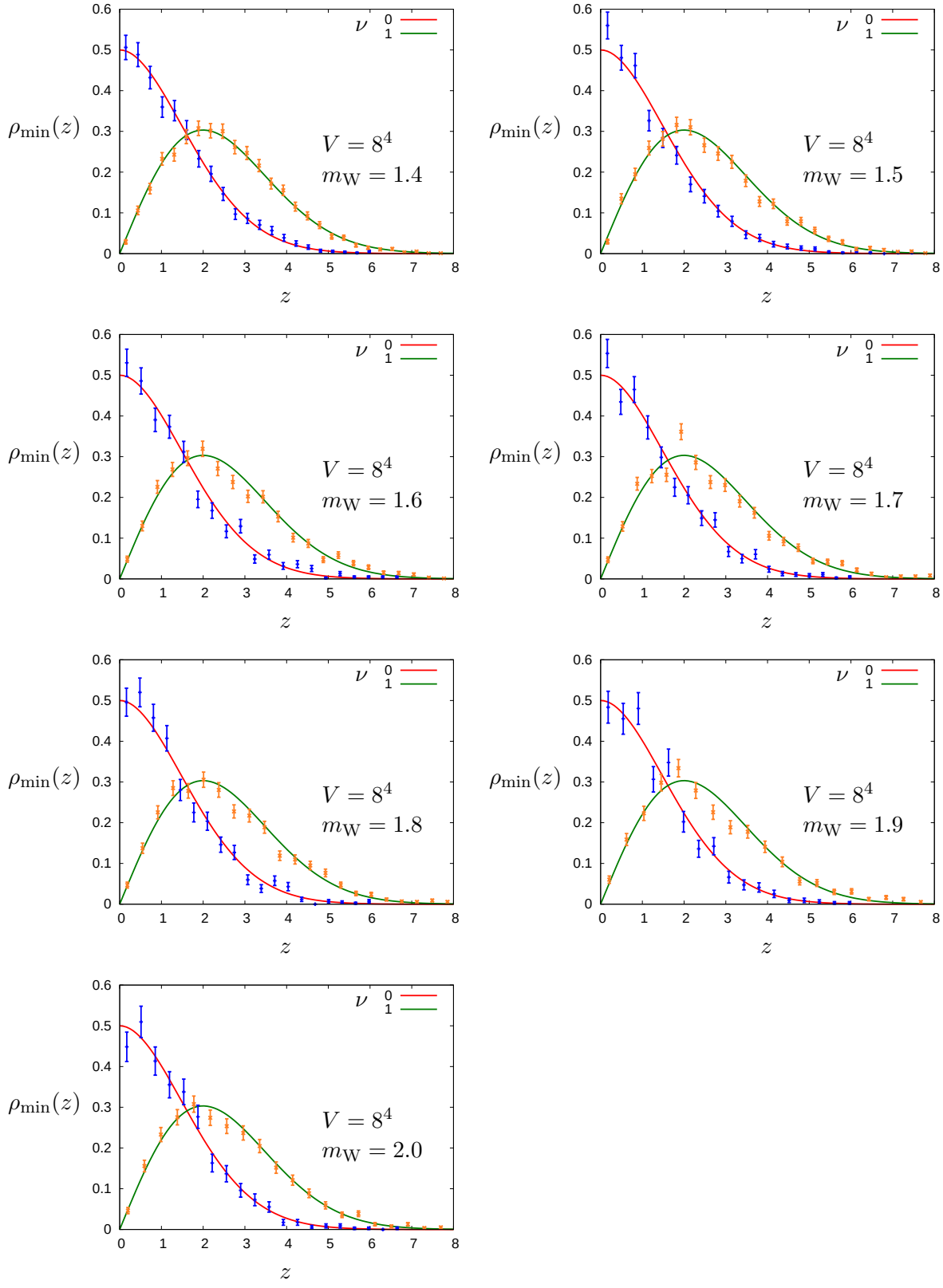

Figure 8.4: Distribution of the lowest-lying eigenvalue $\lambda_{\min}$ of the overlap operator for $\nu = 0, 1$ obtained from simulations of the 8^4 lattice for various values of the Wilson mass m_W. The eigenvalues of the overlap operator are rescaled to $z = \lambda_{\min}\Sigma V$ to match the RMT prediction Eq. (7.13), indicated by solid lines.

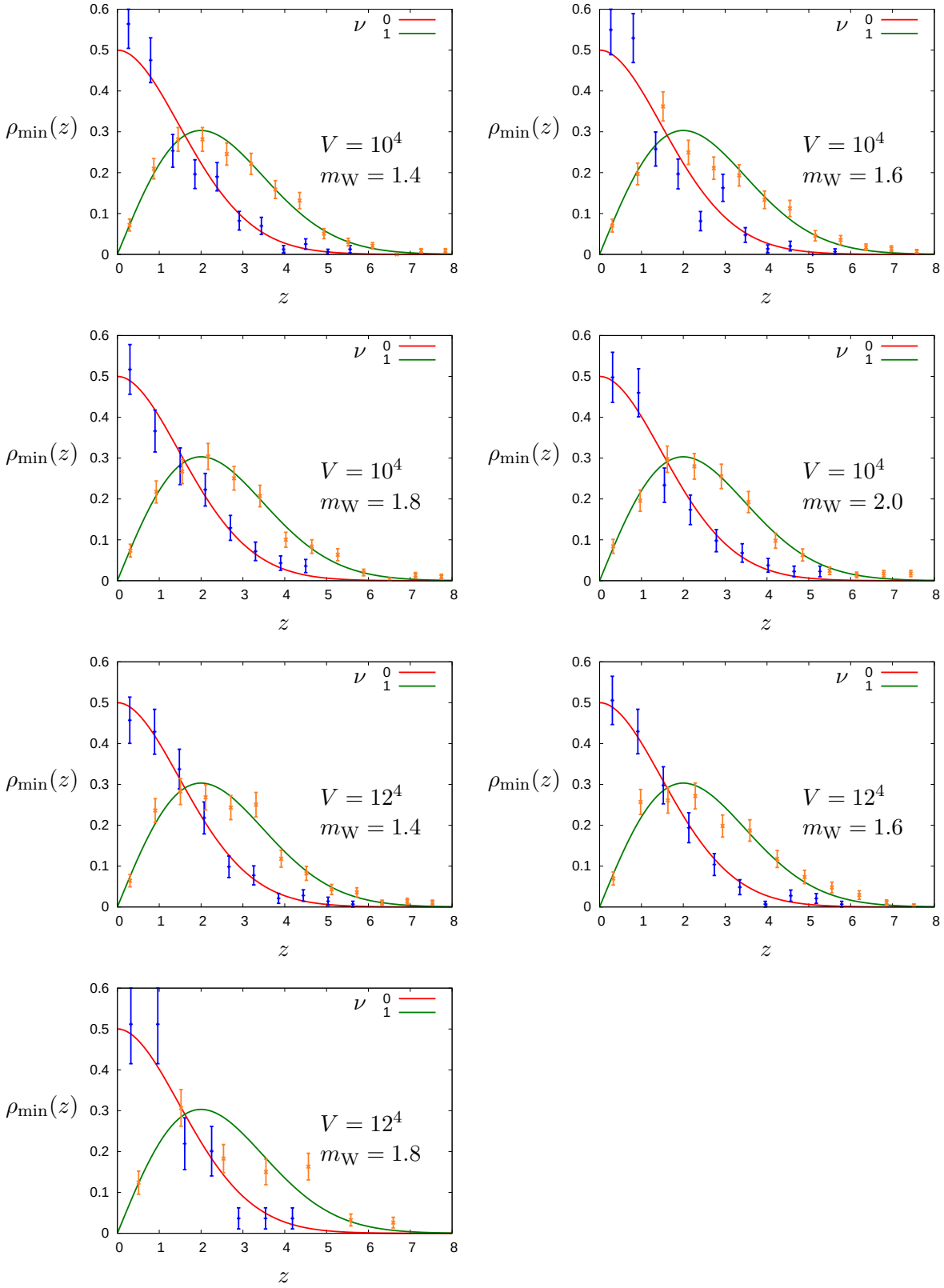

Figure 8.5: Distribution of the lowest-lying eigenvalue $\lambda_{\min}$ of the overlap operator for $\nu = 0, 1$ obtained from simulations of the 10^4 and 12^4 lattices for various values of the Wilson mass m_W. The eigenvalues of the overlap operator are rescaled to $z = \lambda_{\min}\Sigma V$ to match the RMT prediction Eq. (7.13), indicated by solid lines.

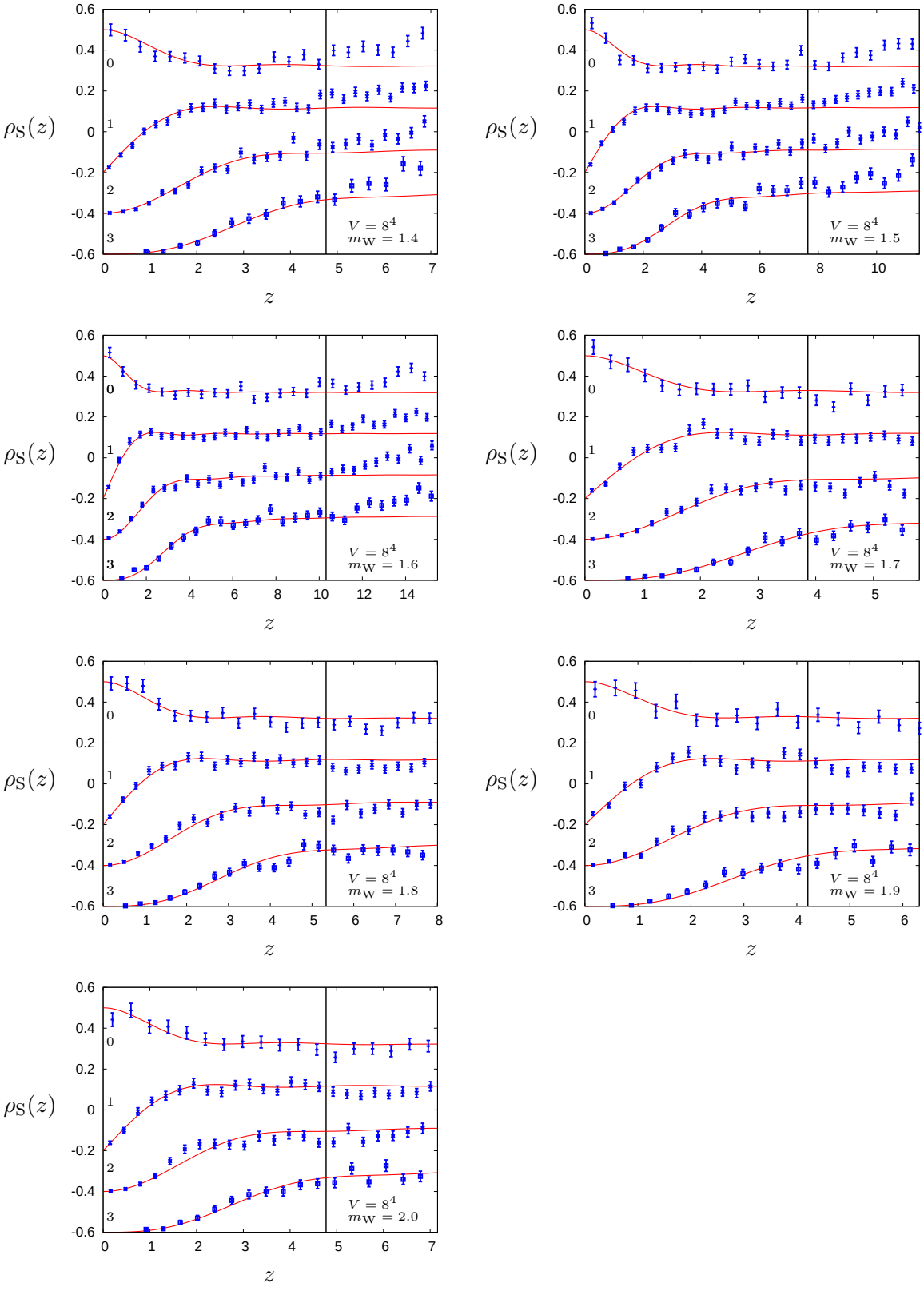

Figure 8.6: Density of the lowest-lying eigenvalues λ_{ov} of the overlap operator for $\nu = 0 \ldots 3$ obtained from simulations of the 8^4 lattice for various values of the Wilson mass m_W. The eigenvalues of the overlap operator are rescaled to $z = \lambda_{ov}\Sigma V$ to match the RMT prediction for the microscopic spectral density Eq. (7.12), indicated by solid lines. For visualization the densities are shifted by a negative offset of 0.2ν. Vertical lines indicate the limit z_{max}.

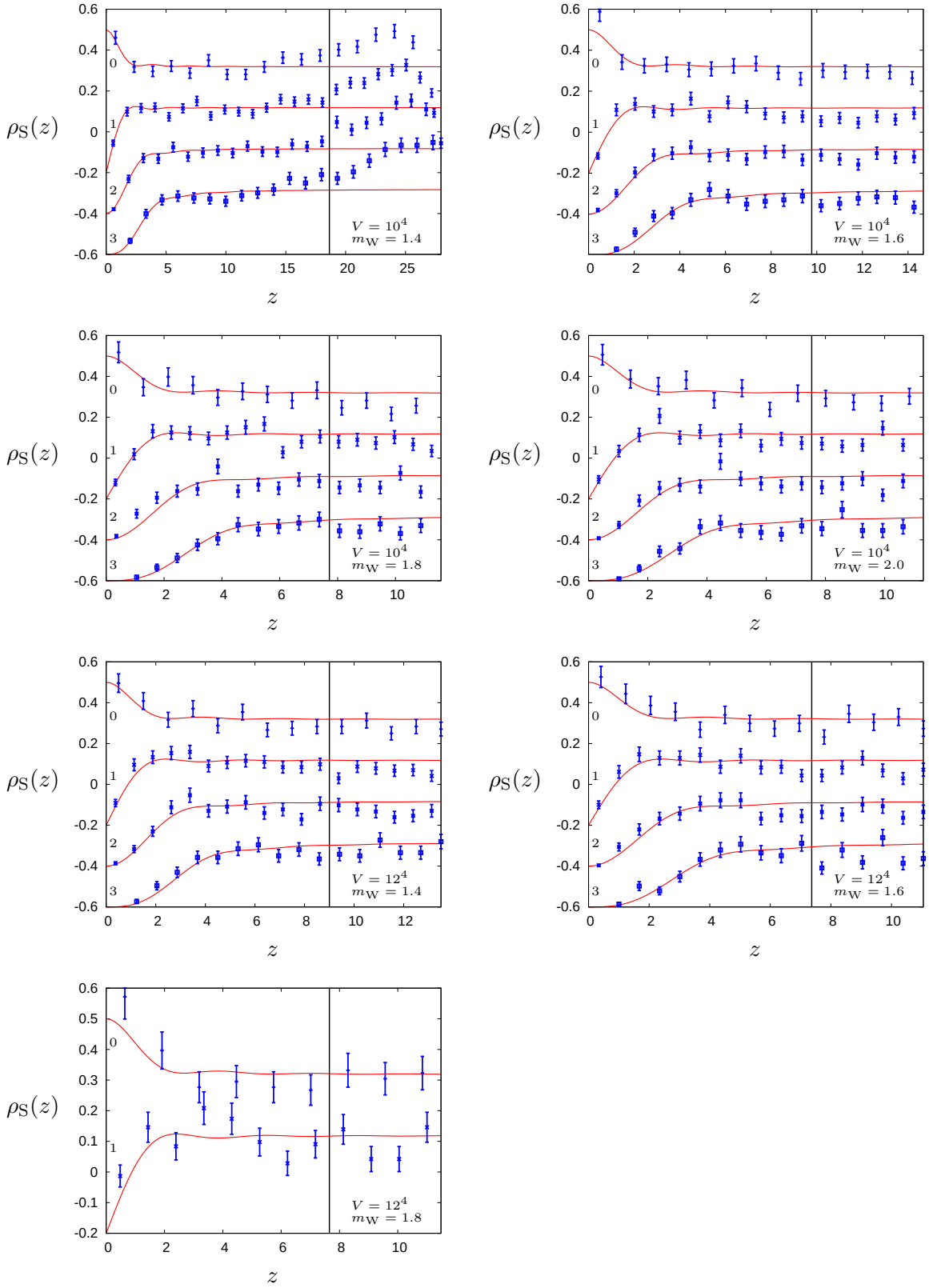

Figure 8.7: Density of the lowest-lying eigenvalues λ_{ov} of the overlap operator for $\nu = 0 \ldots 3$ obtained from simulations of the 10^4 and 12^4 lattices for various values of the Wilson mass m_W. The eigenvalues of the overlap operator are rescaled to $z = \lambda_{ov} \Sigma V$ to match the RMT prediction for the microscopic spectral density Eq. (7.12), indicated by solid lines. For visualization the densities are shifted by a negative offset of 0.2ν. Vertical lines indicate the limit z_{max}.

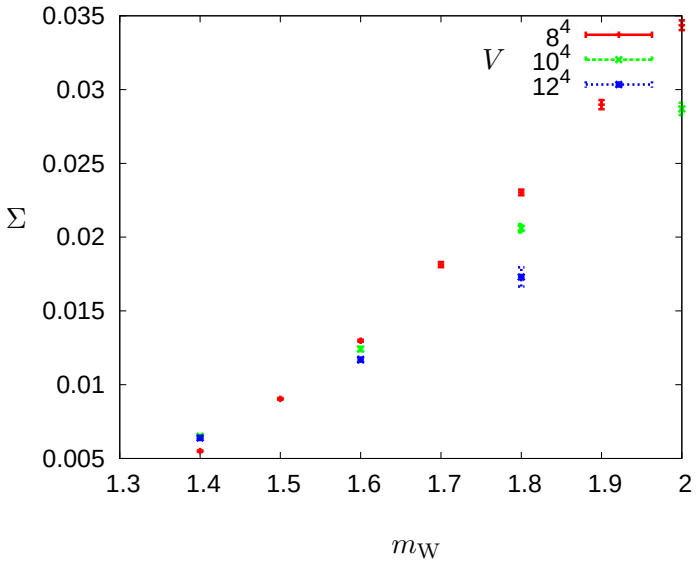

Figure 8.8: Chiral condensate Σ obtained from fits to the microscopic spectral density Eq. (7.12), cf. Table 8.3, as function of the Wilson mass m_{W} and lattice volume V.

.

V	β	No. config.	m_{W}	μ	$P(\nu=0)$	$P(\nu=1)$	$P(\nu=2)$	$P(\nu=3)$	$P(\nu\geq 4)$
8^4	2.2	8976	1.4	0.0	0.217	0.373	0.241	0.111	0.059
		60000	1.4	0.05	0.216	0.371	0.239	0.117	0.057
		30000	1.4	0.1	0.214	0.372	0.241	0.115	0.059
		20000	1.4	0.2	0.212	0.374	0.240	0.114	0.060
		20000	1.4	0.3	0.210	0.372	0.240	0.117	0.061

Table 8.4: Parameters for the simulations of quenched two-colour QCD at non-zero chemical potential μ: lattice volume V, coupling strength β, and number of gauge field configurations evaluated at fixed Wilson mass m_{W}. Here $P(\nu)$ is the relative frequency of the number ν of zero modes found in the spectra of the overlap operator. For comparison the results of the simulation at zero chemical potential are also shown.

8.2 Two-colour QCD at non-zero chemical potential

Introduction of a real-valued chemical potential renders the kernel Wilson-Dirac operator non-Hermitian and (after stereographic projection) results in a complex-valued eigenvalue spectrum of the overlap operator that is distorted non-trivially from the imaginary axis. The distortion, however, is dictated by the underlying symmetries of the operator and one expects the eigenvalues to appear in pairs on the real and imaginary axes and in quadruplets in the complex plane. The spectral densities associated with the eigenvalues near zero are considered to be described by universal functions derived from the non-Hermitian random matrix model introduced in Sect. 7.3. Due to the computational costs simulations of quenched two-colour QCD with chemical potential were only carried out on the 8^4 lattice with coupling strength $\beta = 2.2$ and Wilson mass $m_{\mathrm{W}} = 1.4$, and a few simulations were carried out on a 4^4 lattice. The gauge field configurations obtained from simulations of the 8^4 lattice at zero chemical were reused for evaluation of the spectra. Additional configurations had to be generated in order to obtain the statistics required for comparison of the densities associated with the complex eigenvalues of the overlap operator and RMT. The lowest-lying eigenvalues of $\mathcal{O}(10^4)$ spectra were evaluated applying the Krylov-Ritz method described in Appendix B.1 for several choices for the chemical potential. The simulation parameters are summarized in Table 8.4. In the following the spectrum of the overlap operator with chemical potential is investigated and the spectral densities obtained from simulations of the 8^4 lattice are compared to the predictions given by RMT.

8.2.1 Spectrum of the overlap operator

As introduced in Sect. 6.7, the overlap operator with chemical potential μ is given by

$$D_{\mathrm{ov}}(\mu) = 1 + \gamma_5 \varepsilon[\gamma_5 D_{\mathrm{W}}(\mu)] \,. \tag{8.8}$$

The real-valued chemical potential renders the Wilson-Dirac kernel $\gamma_5 D_{\mathrm{W}}(\mu)$ non-Hermitian, see Eq. (6.111). As a result of the breaking of Hermiticity the (unprojected) eigenvalues of the overlap operators get non-trivially distorted from the Ginsparg-Wilson circle. However, as proposed in Sect. 7.3, the underlying symmetries of the operator restrict its projected eigenvalues to come in purely imaginary pairs, real pairs, and complex quadruplets if the chemical potential is turned on. These restrictions on the spectra of the chiral lattice Dirac operator were indeed observed in the simulations carried out for this study and are briefly discussed in the following. Note that no further rotation of the eigenvalues is required after stereographic projection.

At zero chemical potential the unprojected eigenvalues $\lambda_{\mathrm{ov,u}}$ of the overlap operator are pinned

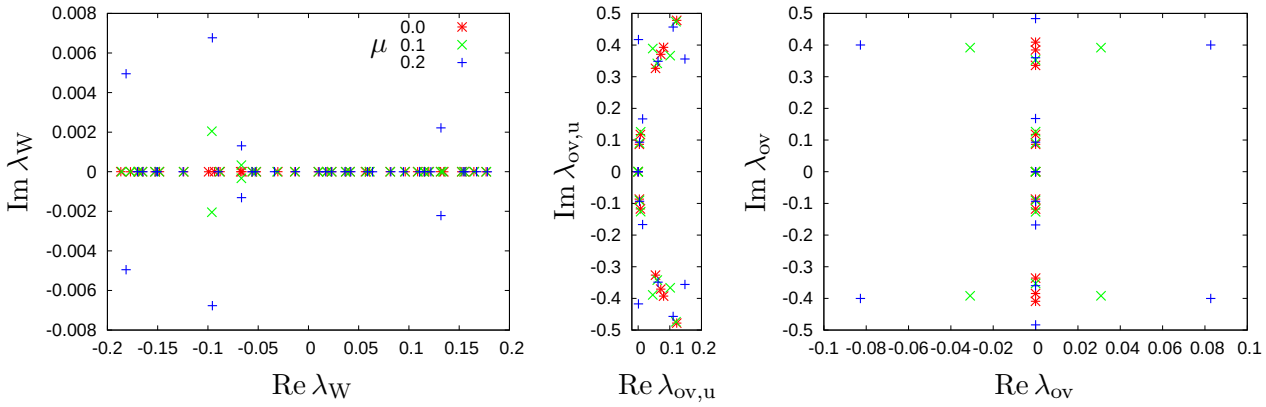

Figure 8.9: Spectrum of the kernel Wilson-Dirac operator, unprojected and projected spectrum of the overlap operator for chemical potential $\mu = 0.0, 0.1$, and 0.2 obtained from a sample configuration on the 8^4 lattice. The left plot shows the spectrum of the Wilson-Dirac operator $\gamma_5 D_{\mathrm{W}}(\mu)$. As the chemical potential is turned on Hermiticity is broken and complex conjugate pairs $(\lambda_{\mathrm{W}}, \lambda_{\mathrm{W}}^*)$ of eigenvalues occur. The middle plot shows the corresponding unprojected spectrum of the overlap operator with eigenvalues $\lambda_{\mathrm{ov,u}}$ that deviate from the Ginsparg-Wilson circle. The right plot shows the same spectrum but projected eigenvalues λ_{ov}. The spectrum exhibits one complex quadruplet. The colouring scheme is identical for all plots.

to the Ginsparg-Wilson circle Eq. (6.95). If the chemical potential μ is turned on, real eigenvalues and complex eigenvalues located outside the Ginsparg-Wilson circle do occur. The stereographic projection Eq. (6.106) maps

- eigenvalues on the circle to the imaginary axis,
- real eigenvalues to the real axis, and
- those eigenvalues lying neither on the circle nor being real into the complex plane.

The spectrum of the kernel Wilson-Dirac operator, the corresponding unprojected, and the projected spectrum of the overlap operator evaluated on a sample configuration on the 8^4 lattice are shown in Fig. 8.9. Deviations from the Ginsparg-Wilson circle do occur, and the projected spectrum exhibits one complex quadruplet as the chemical potential is turned on.

The spectral flow of the projected eigenvalues of the overlap operator with the chemical potential is shown in Fig. 8.10.[6] The spectra were evaluated on a sample gauge field configuration on a 4^4 lattice with coupling strength $\beta = 1.8$ and Wilson mass $m_{\mathrm{W}} = 2.3$. The simulation parameters were adopted from Ref. [68]. At zero chemical potential the eigenvalues are purely imaginary (and come in complex conjugate pairs). Some of the neighbouring eigenvalues on the imaginary axis attract each other as the chemical potential is increased. The first pair of degenerate imaginary eigenvalues forms around $\mu \approx 0.111$ (there is one such pair on each half of the imaginary axis due to chiral symmetry), and form the first quadruplet of eigenvalues which propagates into the complex plane as the chemical potential is increased. Even more quadruplets form at $\mu \approx 0.135, 0.154$, and 0.178. The depletion of the eigenvalues along the imaginary axis is clearly visible. The formation of real pairs of eigenvalues essentially follows the same pattern, however, the sample configuration does not exhibit such a pair. A real pair forms when a complex conjugate pair of imaginary eigenvalues propagates along the imaginary axis, meets at the origin at a certain strength of the chemical potential, and then starts propagation along the real axis. Alternatively a quadruplet hits the real

[6] See also Ref. [102] for the movie animation.

axis and forms two pairs of real eigenvalues. It was also observed on sample configurations that the formation of pairs and quadruplets is reversible, e.g., a pair of imaginary eigenvalues forms a quadruplet which again forms a pair of imaginary eigenvalues. In any case, the symmetries of the spectrum of the overlap operator at non-zero chemical are dictated by the breaking of Hermiticity as described in Sect. 7.3, and no violation was observed in the lattice studies.

Another observation made in the study of two-colour QCD with chemical potential are variations in the index of the overlap operator. The overlap operator still exhibits exact zero modes and the corresponding eigenvectors come with exact chirality. However, the relative frequencies $P(\nu)$ of the number ν of zero modes varies with the chemical potential on the 8^4 lattice. The frequencies are summarized in Table. 8.4. The variations in $P(\nu)$ are mild compared to the variations with the Wilson mass observed at zero chemical potential, cf. Table. 8.1, however they are clearly visible.

The connection between the index of the overlap operator and the eigenvalues of the Wilson-Dirac kernel is again

$$\mathrm{Tr}(\gamma_5 D_{\mathrm{ov}}) = \mathrm{Tr}\left[\gamma_5 + \varepsilon(\gamma_5 D_{\mathrm{W}})\right] = \mathrm{Tr}\left[\varepsilon(\gamma_5 D_{\mathrm{W}})\right] = n_+^{\mathrm{W}} - n_-^{\mathrm{W}} = 2\,\mathrm{index}\, D_{\mathrm{ov}} \;, \tag{8.9}$$

where n_+^{W} (n_-^{W}) is the number of eigenvalues of the kernel operator with positive (negative) sign of the real part. At zero chemical potential the kernel operator is Hermitian and its eigenvalues are real. The change of the sign of one of its eigenvalues results in a change of the index of the overlap operator by ± 1, cf. Sect. 8.1.1. This holds true also in the non-Hermitian case, with the matrix sign function defined by Eq. (6.101). If a real eigenvalue of the non-Hermitian Wilson-Dirac operator $\gamma_5 D_{\mathrm{W}}(\mu)$ changes the sign (as function of the chemical potential), the index of the overlap operator changes by ± 1. However, as shown in Fig. 8.9, the spectrum of the non-Hermitian Wilson-Dirac operator also exhibits pairs of complex eigenvalues. If such a pair of complex eigenvalues changes the sign, then $n_+^{\mathrm{W}} - n_-^{\mathrm{W}}$ changes by ± 4 and the index of the overlap operator changes by ± 2. This behaviour of the index was indeed observed on sample configurations by fine-tuning of the chemical potential. Variations of the index were also observed in a study of quenched three-colour QCD with chemical potential [74], and it is assumed that the variations are a lattice artefact that disappears in the continuum limit.

8.2.2 Analysis of the spectral density

Formally the normalized spectral density of the overlap operator is defined by

$$\rho_{\mathrm{ov}}(\lambda) = \langle \sum_k \delta(\lambda - \lambda_k) \rangle_{\mathrm{G}} \tag{8.10}$$

as ensemble average over gauge field configurations. It was shown in the previous section that the projected spectrum of the overlap operator at non-zero chemical potential exhibits purely imaginary, real, and also complex eigenvalues. Normalization of the spectral density therefore implies

$$\int_{\mathbb{C}} \mathrm{d}^2\lambda\, \rho_{\mathrm{ov}}(\lambda) = \int_{\mathrm{i}\mathbb{R}} \mathrm{d}\lambda\, \rho_{\mathrm{ov}}^{\mathrm{i}\mathbb{R}}(\lambda) + \int_{\mathbb{R}} \mathrm{d}\lambda\, \rho_{\mathrm{ov}}^{\mathbb{R}}(\lambda) + \int_{\mathbb{C}\setminus(\mathbb{R}\cup\mathrm{i}\mathbb{R})} \mathrm{d}^2\lambda\, \rho_{\mathrm{ov}}^{\mathbb{C}}(\lambda) = 4N_c V \;, \tag{8.11}$$

again with $N_c = 2$ in this study. The densities associated with the imaginary, real, and complex eigenvalues are predicted by non-Hermitian RMT, cf. Sect. 7.3. A two-parameter fit to non-Hermitian RMT, Eqs. (7.24) and (7.29), subject to (for a given ν)

$$\rho_{\mathrm{S}}^{(\mathrm{i})\mathbb{R}}(\xi) = \frac{1}{\Sigma V}\rho_{\mathrm{ov}}^{(\mathrm{i})\mathbb{R}}\left(\frac{\xi}{\Sigma V}\right) \;, \quad \rho_{\mathrm{S}}^{\mathbb{C}}(\xi) = \frac{1}{(\Sigma V)^2}\rho_{\mathrm{ov}}^{\mathbb{C}}\left(\frac{\xi}{\Sigma V}\right) \;, \tag{8.12}$$

allows to determine the chiral condensate Σ and also the Hermiticity breaking parameter $\hat{\mu} = 2\mu F\sqrt{V}$, from which the pion decay constant F can be determined. Analogous to the Hermitian case the eigenvalues of the overlap operator need to be rescaled to $\xi = \lambda_{\mathrm{ov}}\Sigma V$ to match RMT.

The study of quenched two-colour QCD was carried out for the chemical potentials $\mu = 0.05$, 0.1, 0.2, and 0.3. Densities associated with the purely imaginary and real eigenvalues were obtained by binning of the lattice data into a histogram according to Sect. 8.1.1. The low statistics of the complex eigenvalues prohibited a 2-dimensional binning of the lattice data. Therefore, the real part of the complex eigenvalues was integrated out and the resulting data binned along the imaginary axis. Fits of the lattice data to RMT were performed simultaneously to the densities associated with the three classes of eigenvalues using only two fit parameters. The results of the fits are summarized in Table 8.5 and plots for the simultaneous fits to $\nu = 0, 1$ are shown in Figs. 8.11 – 8.14 for each choice of the chemical potential. The fits were performed individually to those spectra of the overlap operator which exhibit 0 or 1 zero mode, and also simultaneously to the densities associated with $\nu = 0$ and 1. Discretization errors were compensated for by fitting of the densities integrated over the bin size. The upper limits of the fit, $\xi_{\mathrm{max}} = \lambda_{\mathrm{max}}\Sigma V$, were individually guided by the eye for each density and fit. In analogy to the Hermitian case the spectral support for the fit was estimated by integration over the microscopic spectral densities Eqs. (7.24) and (7.29),

$$N_{\mathrm{ev}} = \int_0^{\xi_{\mathrm{max}}^{\mathrm{i}\mathbb{R}}} \mathrm{d}\xi\ \rho_{\mathrm{S}}^{\mathrm{i}\mathbb{R}}(\mathrm{i}\xi) + \int_0^{\xi_{\mathrm{max}}^{\mathbb{R}}} \mathrm{d}\xi\ \rho_{\mathrm{S}}^{\mathbb{R}}(\xi) + 2 \int_0^{\xi_{\mathrm{max}}^{\mathbb{R}}+\mathrm{i}\xi_{\mathrm{max}}^{\mathbb{C}}} \mathrm{d}^2\xi\ \rho_{\mathrm{S}}^{\mathbb{C}}(\xi)\ , \tag{8.13}$$

taking into account only eigenvalues in the positive complex plane and on half the real axis. The factor of 2 for the complex eigenvalues mimics the symmetry of the density. The density associated with the complex eigenvalues was integrated up to $\xi_{\mathrm{max}}^{\mathbb{R}} + \mathrm{i}\xi_{\mathrm{max}}^{\mathbb{C}}$.

The fitted parameter Σ shows variations with ν and μ on the order of 5%. However, the pion decay constant F obtained from individual fits are almost constant in μ, but the results obtained from fits to different ν systematically differ by about 4%. The values of the pion decay constant obtained from simultaneous fits to $\nu = 0$ and 1 settle in between the results obtained from the individual fits. An interesting observation is the validity of the fit intervals. The interval $\xi_{\mathrm{max}}^{\mathrm{i}\mathbb{R}}$ for the purely imaginary eigenvalues scales with the chemical potential, whereas the interval $\xi_{\mathrm{max}}^{\mathbb{C}}$ for the complex eigenvalues along the imaginary axis is basically independent of μ. The real eigenvalues were fitted along the positive half-axis, taking into account all positive real eigenvalues obtained from the lattice simulations. Nevertheless, the spectral support in terms of the number of eigenvalues N_{ev} that contribute is $\mathcal{O}(1)$. This number is, however, slightly lower than the estimate obtained from simulations at zero chemical potential. Note that the fit intervals were independently guided by the eye. Other methods can lead to different estimates.

Universality in the spectra of the overlap operator turns out to be dependent on the parameter ν and the choice for μ. In the case $\nu = 1$ the lattice data is almost perfectly described by the RMT model for two-colour QCD with real chemical potential $\mu = 0.05$. However, with increasing strength of the chemical potential deviations from the predictions occur. This is reflected in the goodness-of-fit parameter χ^2/dof, which increases with the chemical potential. The RMT model describes the lattice simulations (with this particular choice for the simulation parameters) only in the regime where the finite-volume partition function of QCD is dominated by the zero momentum modes of the pion fields, i.e., in the ε-regime where $\mu L \ll 1$ (L is the lattice extension). The case $\nu = 0$ is somewhat different. The lattice data is adequately described by the univeral functions and RMT is valid even for simulations up to $\mu L = \mathcal{O}(1)$.

μ	ν	$\hat{\mu}$	Σ	F	χ^2/dof	$\xi_{\max}^{i\mathbb{R}}$	$\xi_{\max}^{\mathbb{R}}$	$\xi_{\max}^{\mathbb{C}}$	N_{ev}
0.0	0 – 3	0.0	0.00550(4)	—	1.2(4)	4.76	—	—	1.74
0.05	0	0.242(4)	0.00538(4)	0.03773(29)	0.8(4)	2.86	1.65	2.86	1.13
0.1	0	0.480(8)	0.00522(6)	0.03753(29)	1.0(5)	2.78	3.21	2.78	1.12
0.2	0	0.956(13)	0.00514(8)	0.03734(26)	1.4(5)	3.79	8.21	2.74	1.41
0.3	0	1.455(16)	0.00524(8)	0.03789(21)	1.4(4)	4.29	12.88	2.79	1.34
0.05	1	0.230(4)	0.00535(2)	0.0390(3)	1.1(5)	2.85	1.64	2.85	0.68
0.1	1	0.499(5)	0.00525(5)	0.03900(21)	1.7(5)	2.79	3.22	2.79	0.68
0.2	1	0.9830(10)	0.00534(6)	0.03838(19)	2.0(6)	3.93	8.52	2.84	1.01
0.3	1	1.493(11)	0.00544(5)	0.03887(15)	4.3(7)	4.46	13.38	2.90	$(*)$
0.05	$0+1$	0.2438(29)	0.00536(2)	0.0381(5)	1.0(3)	2.85	1.65	2.85	1.13
0.1	$0+1$	0.491(4)	0.00524(4)	0.03839(29)	1.4(4)	2.79	3.22	2.79	1.13
0.2	$0+1$	0.974(8)	0.00527(4)	0.0380(3)	1.7(4)	3.89	8.42	2.81	1.44
0.3	$0+1$	1.480(9)	0.00537(5)	0.03854(24)	2.9(4)	4.40	13.21	2.86	1.39

Table 8.5: Results for the chiral condensate Σ and the pion decay constant F from individual and simultaneous fits to the microscopic spectral densities Eqs. (7.24) and (7.29) for $\nu = 0$ and 1. Here μ is the chemical potential and $\hat{\mu}$ is the symmetry breaking parameter. The rescaled variables $\xi_{\max} = \lambda_{\max}\Sigma V$ indicate the fit interval as described in the text. The spectral support is provided by N_{ev} eigenvalues (for the simultaneous fit $\nu = 0$ was chosen). The results of the fit to the microscopic spectral density at zero chemical potential Eq. (7.12) are shown for comparison. The statistical errors were estimated by the bootstrap method (cf. Appendix B.2). For $\mu = 0.3$ and $\nu = 1$ the evaluation of the parameter N_{ev} was spoiled by instabilities in the numerical integration, indicated by the asterisk.

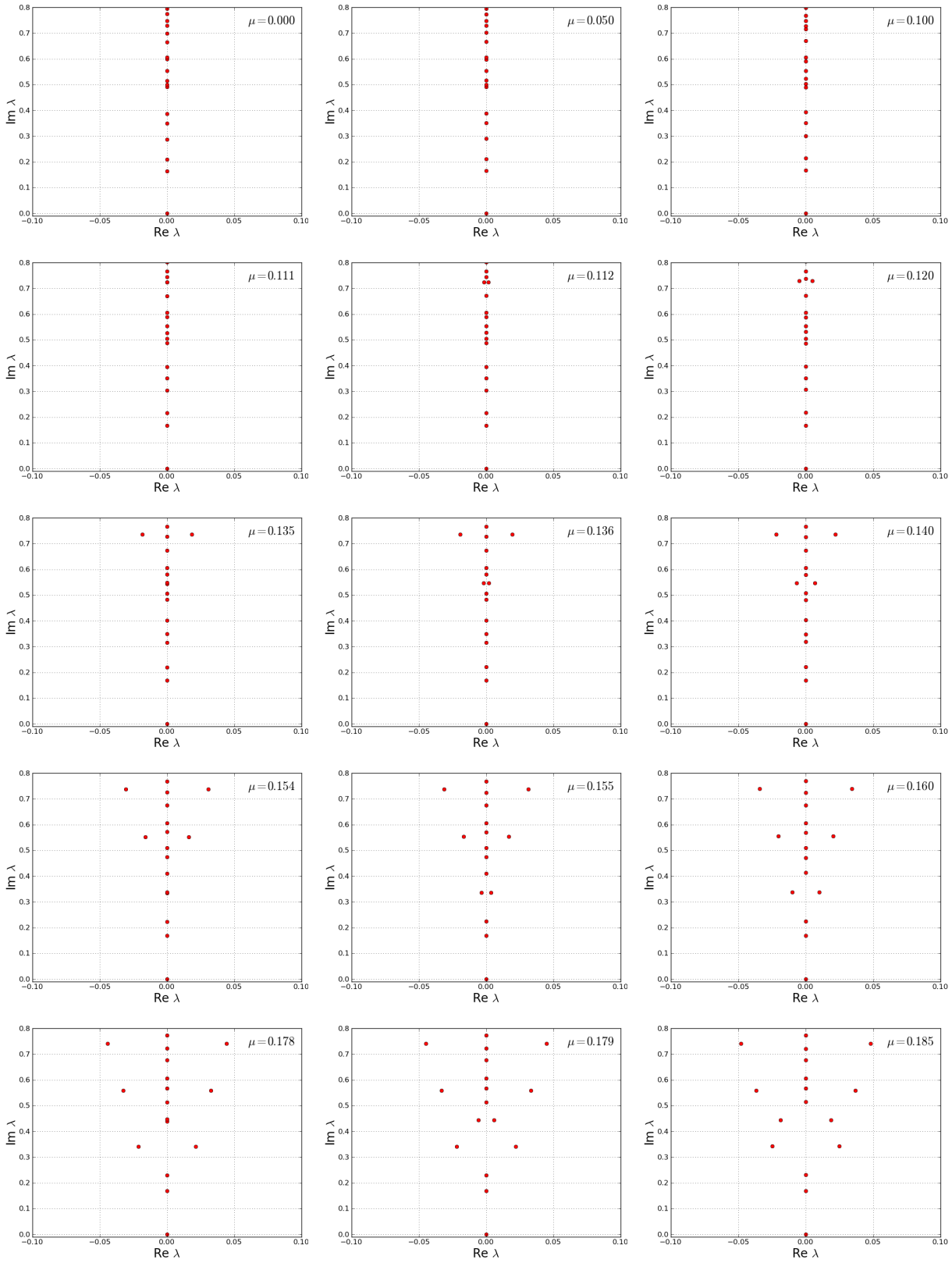

Figure 8.10: Snapshots of the spectral flow of the lowest-lying eigenvalues λ of the overlap operator with the chemical potential μ after stereographic projection. The spectrum was evaluated on a 4^4 lattice with coupling strength $\beta = 1.8$ and Wilson mass $m_W = 2.3$. The chemical potential is increased from 0.0 up to 0.185 in irregular steps. The spectrum is symmetric with respect to the real axis, exhibits one zero mode and shows the formation of complex quadruplets. See Ref. [102] for the movie animation.

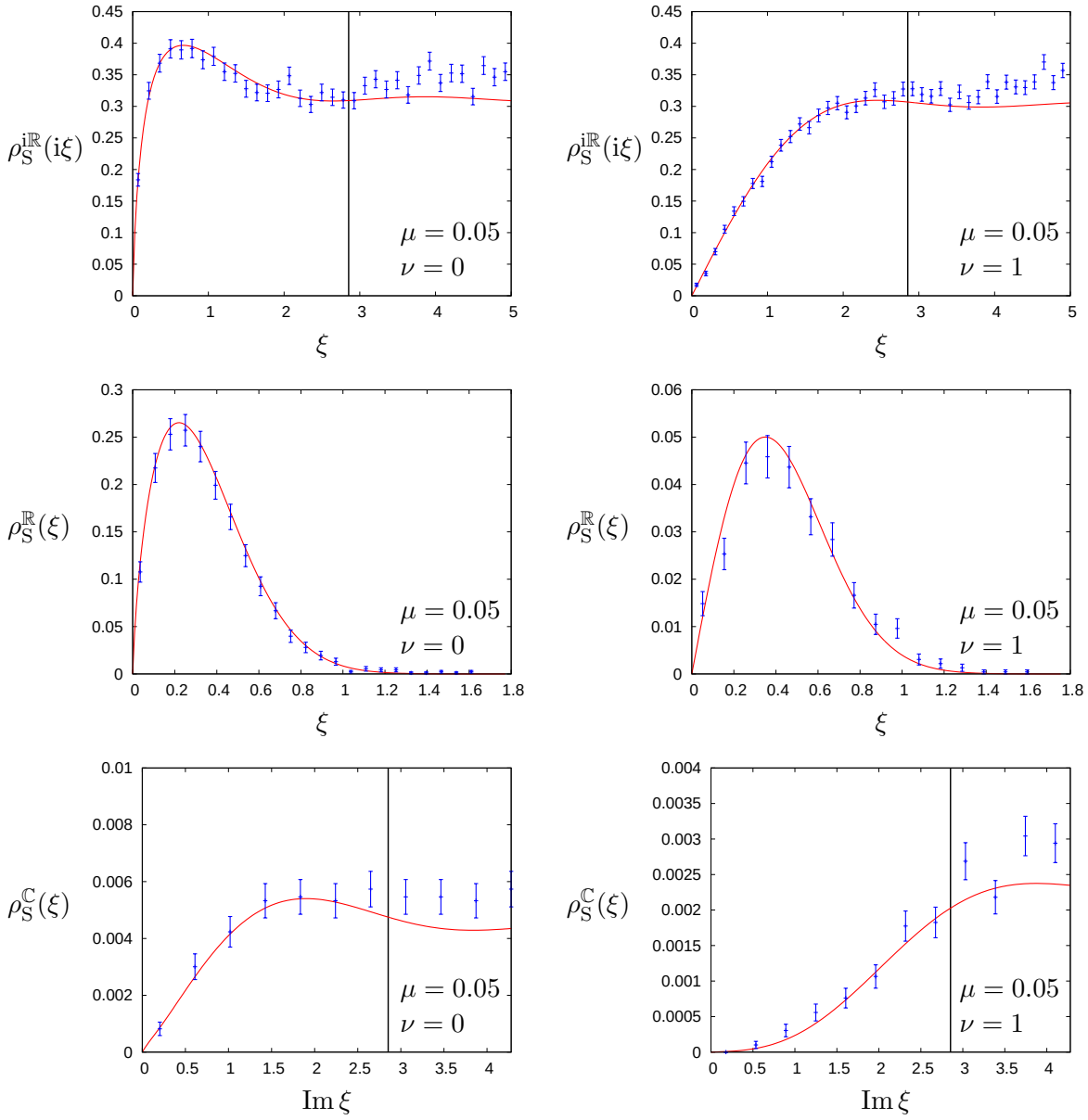

Figure 8.11: Density of the lowest-lying eigenvalues λ of the overlap operator obtained from simulations with chemical potential $\mu = 0.05$ on the 8^4 lattice. The eigenvalues of the overlap operator are rescaled to $\xi = \lambda\Sigma V$ to match the RMT prediction Eqs. (7.24) and (7.29), indicated by solid lines. The left column shows the densities for $\nu = 0$, the right column shows the densities for $\nu = 1$. The top row shows the density associated with the purely imaginary eigenvalues, the middle row shows the density associated with the real eigenvalues and the bottom row shows the density associated with the complex eigenvalues with the real part integrated out. Vertical lines indicate the limit $\xi_{\max}$.

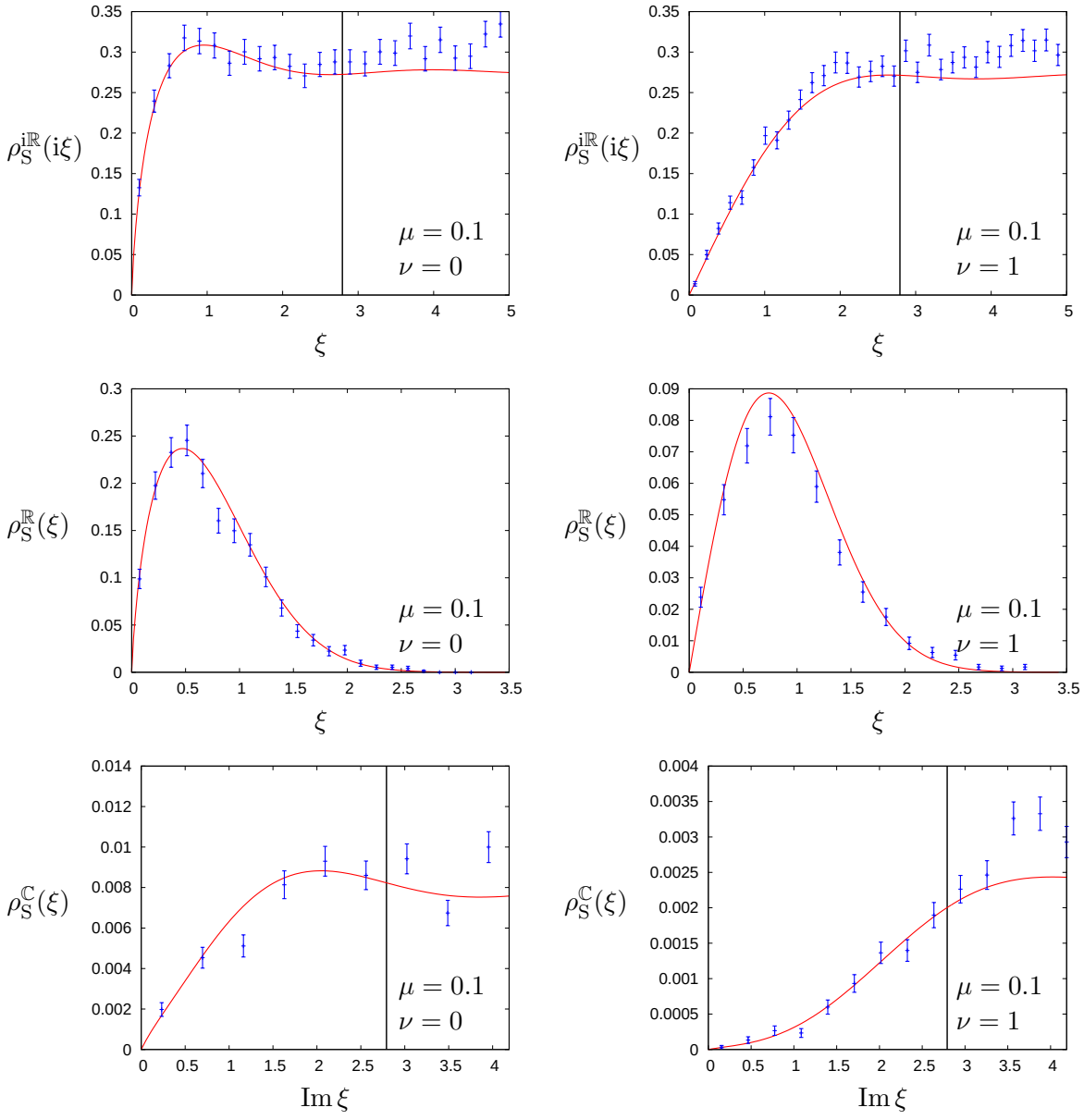

Figure 8.12: Density of the lowest-lying eigenvalues λ of the overlap operator obtained from simulations with chemical potential $\mu = 0.1$ on the 8^4 lattice. The eigenvalues of the overlap operator are rescaled to $\xi = \lambda\Sigma V$ to match the RMT prediction Eqs. (7.24) and (7.29), indicated by solid lines. The left column shows the densities for $\nu = 0$, the right column shows the densities for $\nu = 1$. The top row shows the density associated with the purely imaginary eigenvalues, the middle row shows the density associated with the real eigenvalues and the bottom row shows the density associated with the complex eigenvalues with the real part integrated out. Vertical lines indicate the limit $\xi_{\max}$.

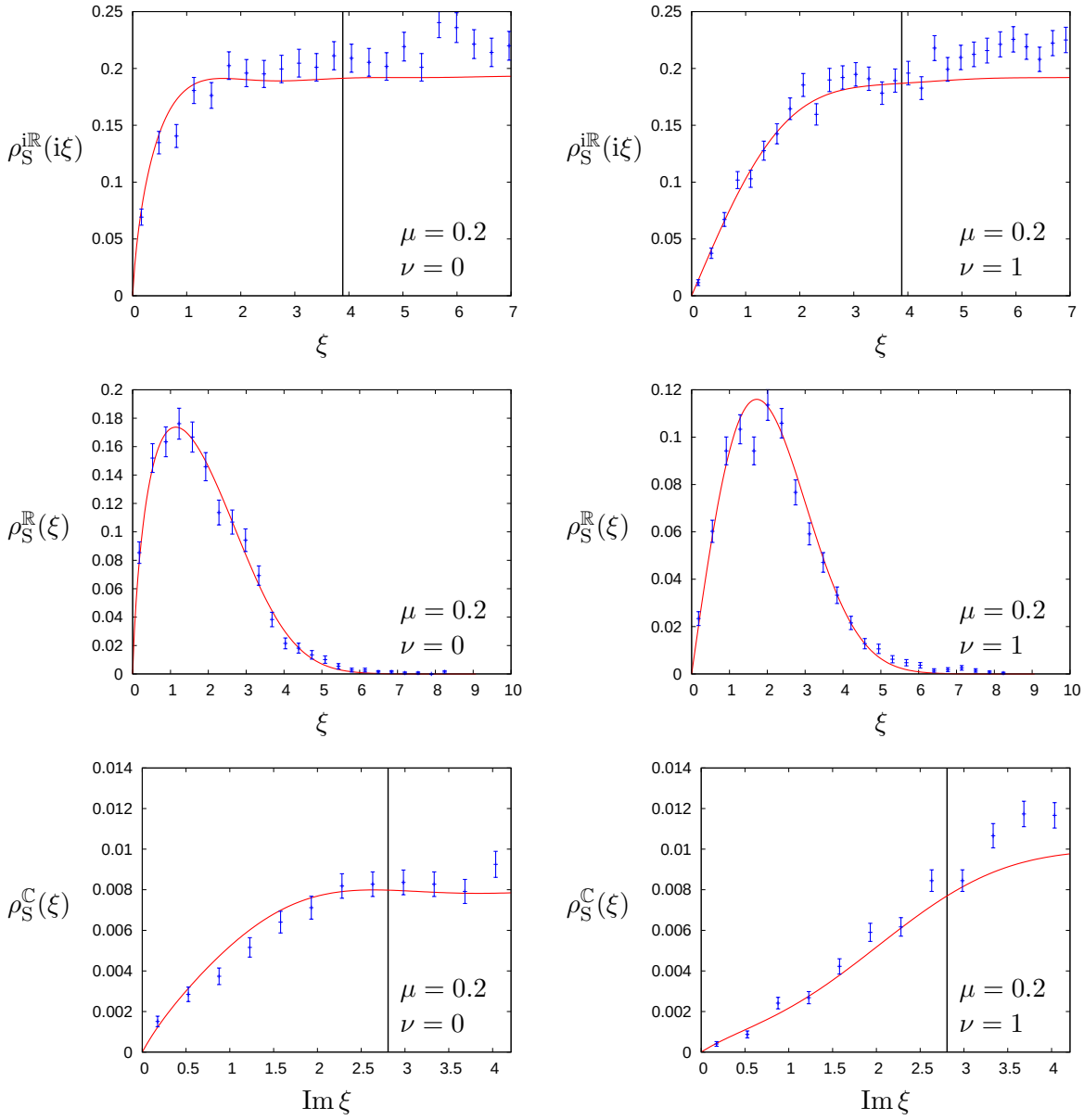

Figure 8.13: Density of the lowest-lying eigenvalues λ of the overlap operator obtained from simulations with chemical potential $\mu = 0.2$ on the 8^4 lattice. The eigenvalues of the overlap operator are rescaled to $\xi = \lambda\Sigma V$ to match the RMT prediction Eqs. (7.24) and (7.29), indicated by solid lines. The left column shows the densities for $\nu = 0$, the right column shows the densities for $\nu = 1$. The top row shows the density associated with the purely imaginary eigenvalues, the middle row shows the density associated with the real eigenvalues and the bottom row shows the density associated with the complex eigenvalues with the real part integrated out. Vertical lines indicate the limit $\xi_{\max}$.

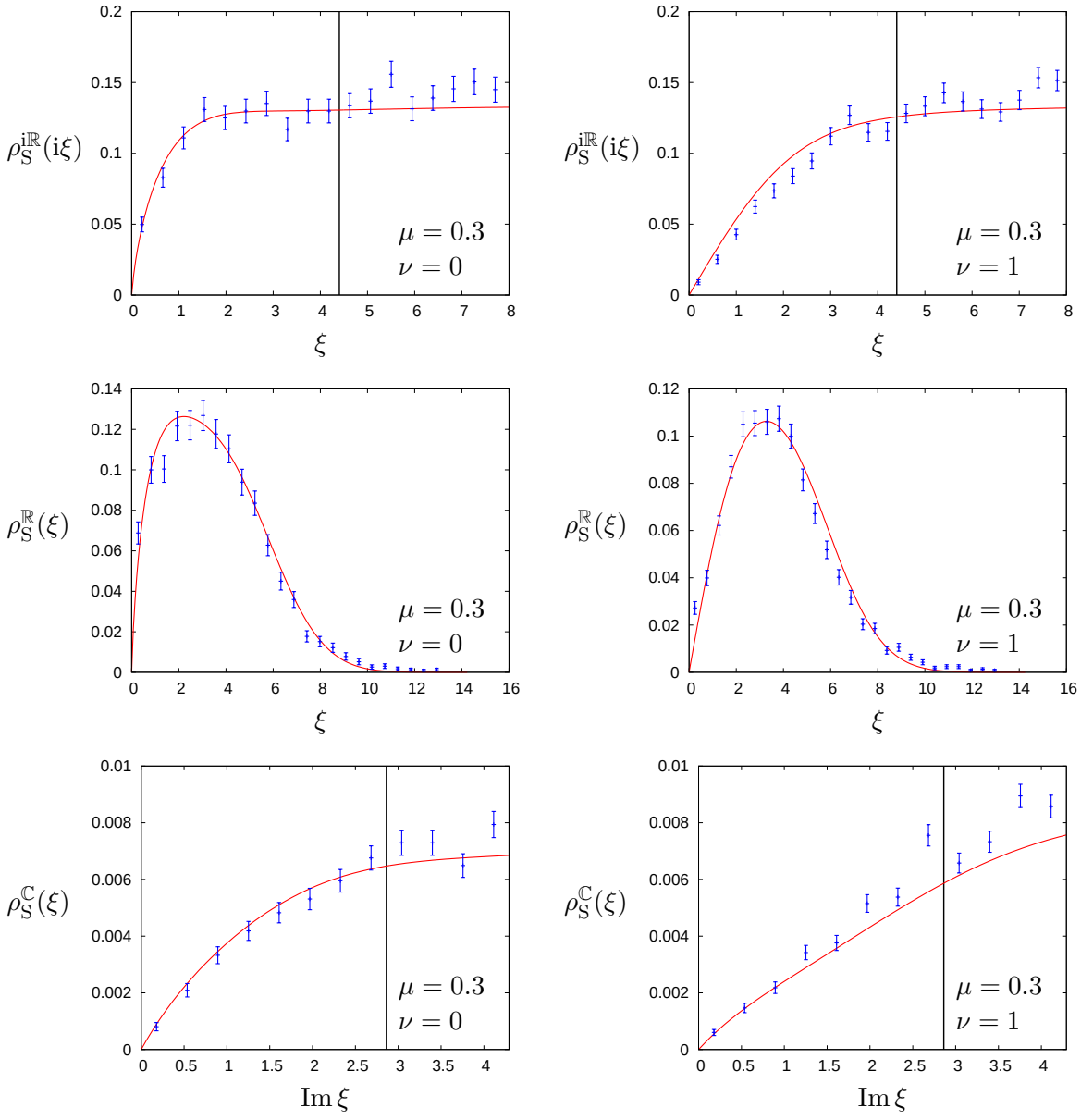

Figure 8.14: Density of the lowest-lying eigenvalues λ of the overlap operator obtained from simulations with chemical potential $\mu = 0.3$ on the 8^4 lattice. The eigenvalues of the overlap operator are rescaled to $\xi = \lambda\Sigma V$ to match the RMT prediction Eqs. (7.24) and (7.29), indicated by solid lines. The left column shows the densities for $\nu = 0$, the right column shows the densities for $\nu = 1$. The top row shows the density associated with the purely imaginary eigenvalues, the middle row shows the density associated with the real eigenvalues and the bottom row shows the density associated with the complex eigenvalues with the real part integrated out. Vertical lines indicate the limit $\xi_{\max}$.

Chapter 9

Summary

9.1 QPACE

In the first part of this thesis some details of the QPACE supercomputer were revealed. QPACE is a research project carried out by several academic institutions in collaboration with the IBM Research and Development Laboratory in Böblingen, Germany, and other industrial partners. The main goal was the design of an application-optimized scalable architecture especially designed for applications in Lattice QCD. QPACE relies on the IBM PowerXCell 8i microprocessor directly connected to a custom-designed network processor that is implemented on a Xilinx Virtex-5 FPGA.

The project officially started in 2008. Two installations were deployed in the summer of 2009 and the final design was completed in early 2010. The aggregate compute performance of the architecture is 200 TFlops in double precision. QPACE turned out to be a huge success. In November 2009 QPACE was the leading architecture on the Green 500 list of the most energy-efficient supercomputers in the world. The title was defended in June 2010, when the architecture achieved an energy signature of 773 MFLOPS per Watt in the Linpack benchmark. In the Top 500 list of most powerful supercomputers, QPACE ranked #110, #111, #112 in November 2009, and #131, #132, #133 in June 2010. The innovative water-cooling solution also influenced other supercomputer designs such as SuperMUC.

9.2 Lattice simulations of two-colour QCD

In the second part of this thesis the spectral properties of a Dirac operator that respects a lattice chiral symmetry, the overlap operator with Wilson-Dirac kernel, were studied on the lattice in the quenched approximation of QCD with gauge group SU(2). The dynamics of the Goldstone modes that arise in the phase of broken chiral symmetry is governed by the deep infrared spectrum of the Dirac operator. In the ε-regime the finite-volume partition function of pion effective theory associated with two-colour QCD is equivalent to the partition function of chiral random matrix theory of the orthogonal ensemble. Both Hermitian and non-Hermitian formulations of chiral random matrix theory exist. The latter can be applied to the formulation of QCD at non-zero baryon chemical potential. The microscopic spectral densities derived from random matrix models are universal functions, and one expects them to describe the correlations in the few lowest-lying eigenvalues of the overlap operator for simulations of the ε-regime on the lattice.

As a test for Hermitian chiral random matrix theory, lattice simulations were carried out on 8^4, 10^4, and 12^4 lattices and several choices for the Wilson mass parameter. The spectral density associated with the lowest-lying eigenvalue(s) of the overlap operator were compared to the predictions of random matrix theory. Due to the large statistics it was possible to investigate the spectra associated with non-trivial sectors of topological charge. It was found that, with the proper choice for

the Wilson mass parameter, the distribution of the lowest-lying eigenvalue(s) is indeed universal and described by random matrix theory. The chiral condensate was evaluated by a fit of the spectrum of the overlap operator to the predictions. The condensate shows a strong dependence on the Wilson mass. However, the variations in the fit values diminish towards the continuum limit.

Recently the non-Hermitian extension to chiral random matrix theory of the orthogonal ensemble was solved. This extension is equivalent to pion effective theory associated with two-colour QCD and non-zero baryon chemical potential in the ε-regime. Therefore, the spectral correlation functions derived from the non-Hermitian random matrix model should be applicable to simulations of Lattice QCD with baryon chemical potential. The real-valued chemical potential renders the Dirac operator non-Hermitian, and the non-trivial distortion of its spectrum into the complex plane is dictated by the underlying symmetries of the operator. As a test for the random matrix model a baryon chemical potential μ was introduced into the overlap operator along the lines of Ref. [74]. By simulation of quenched two-colour QCD on 4^4 and 8^4 lattices it was found that the spectrum of the non-Hermitian chiral lattice Dirac operator indeed exhibits pairs of purely imaginary and real eigenvalues, and also complex quadruplets, as predicted by the formal symmetry analysis. The spectral density of the overlap operator near zero also turned out to be universal. It was shown by simulations of the 8^4 lattice at fixed Wilson mass that the spectral density of the few lowest-lying eigenvalues obtained from the lattice simulations in the regime $\mu L \ll 1$ are indeed described by the microscopic spectral density derived from non-Hermitian random matrix theory. For the trivial sector of topological charge the predictions of random matrix theory are even valid in the regime $\mu L = \mathcal{O}(1)$.

In the end a few open questions remain to be answered:

- At zero chemical potential the fit value of the chiral condensate shows a strong dependence on the choice of the Wilson mass for the simulation parameters investigated. However, due to the computational costs of the simulations, the effect of the Wilson mass on the pion decay constant was not investigated in this study.

- The regime μL where random matrix theory applies to lattice simulations depends on the sector of topological charge.

- It was shown that Hermitian chiral random matrix theory applies to non-trivial sectors of topological charge. However, due to limited statistics only the sectors $\nu = 0$ and 1 were investigated in the non-Hermitian regime.

- The fit values of the pion decay depend on the sector of topological charge. In contrast, the fit values of the chiral condensate are rather insensitive to the number of zero modes in the spectra of the overlap operator.

- At non-zero baryon chemical potential the spectral support for the fit of the complex eigenvalues was chosen with rectangular shape in the complex plane. Other choices such as circular or elliptic shapes could be also used to resolve the applicability of non-Hermitian random matrix theory.

Appendix A

QPACE Addendum

A.1 Sources

The following table provides an overview on the most important administrative tools and test cases written in C. Dependencies on other QPACE-specific libraries are not shown. All programs have to be executed on the master server and perform remote access to the root-card and node-card using the *witchlib* library. An exception is `ncDcr`, which has to be executed directly on the node-card. The sources are located in the sub-directories of `trunk/hsit/` of the QPACE repository.

File	Description
witchlib.c	Root-card high-level functions library
witchlib.h	witchlib header
ncBoot.c	Node-card boot tool
xparse.c	Command line parser
asciitable.c	Ascii art tables
asciitable.h	asciitable header
ncVpd.c	Node-card VPD integrity test
ncDcr.c	Node-card DCR integrity test
dcr_slaves.h	DCR slaves description
ncScan.c	Node-card scan test
ncGetInfo.c	Node-card information tool
rcGetInfo.c	Root-card information tool
srcGetInfo.c	Superroot-card information tool
psuScan.c	Power supply information tool
rcSpiRd.c	Node-card SPI integrity test
rcSpiBackdoor.c	Node-card SPI backdoor integrity test
rcI2c.c	Root-card I^2C test
rcRs485.c	Node-card RS-485 integrity test

The following table provides an overview on the most important administrative tools and test cases written in Python, compliant to Python version 2.4. All scripts have to be executed on the master server. Wrapper scripts copy the test case executable onto the node-card. The test case executables are not listed. The sources are located in the sub-directories of `trunk/integration/` of the QPACE repository. The QPACE Front-end Client (QFC) is located in `trunk/front/qfc/`.

File	Description
iothd.py	Multithreading engine, required by several test cases
ethint.py	Ethernet stress test wrapper
switchmapping.py	Ethernet stress test switch mapping
rcSpiRd1M.py	SPI integrity check wrapper
iwcstress.py	IWC stress test wrapper
fft2d.py	FFT stress test wrapper
epc.py	EPC stress test wrapper
qc	QPACE Front-end Client
qfoil.py	QFC helper functions
qfengine.py	QFC multithreading, flash and fuqd engine
qfchelp.py	QFC help

The VHDL sources of the FPGA entities introduced in chapter 4 are listed below. The sources are located in the sub-directories of `trunk/node-card/FPGA/` of the QPACE repository.

Unit	Entity	File	Description
DCR master	`spi_backdoor_top`	spi_backdoor_top.vhd	Top level
	`spi_backdoor_dcr_master`	sync_backdoor.vhd	DCR master
	`dcr_arbiter`	dcr_master_switch.vhd	DCR arbiter
DCR slave	`dcr_slave_interface`	generic_dcr_slave.vhd	Sync. DCR slave interface
	`cdt_dcr_slave_interface`	cdt_dcr_slave.vhd	Async. DCR slave interface
		cdt_dcr_slave.ucf	Async. DCR slave UCF template
IWC	`iwc`	iwc-simple.vhd	IWC core logic
	`iwc_em`	iwcem.vhd	IWC Extension Module
UART	`uart_16550_dcr_top`	alternative_uart_16550_dcr_top.vhd	Top level w/ sync. DCR interface
	`uart_16550_top`	uart_16550_top.vhd	Top level
	`uart_buffered_tx`	uart_buffered_tx.vhd	Wrapper FIFO and transmitter
	`txunit`	uarttx.vhd	Transmitter logic
	`uart_buffered_rx`	uart_buffered_rx.vhd	Wrapper FIFO and receiver
	`rxunit`	uartrx.vhd	Receiver logic
		uart_16550_const.vhd	Constant declarations

A.2 DCR memory map

The DCR device tree is mapped into the Cell BE I/O real address space at base address

```
DCR_BASE_BE = 0x18000000000
```

with a size of

```
DCR_SIZE_BE = 0x00000037000
```

corresponding to 220 kByte of DCR address space. The base addresses `DCR_BASE_DEVICE` of the public DCR device register files are summarized with corresponding reference in Table A.1.

Device	`DCR_BASE_DEVICE`	Description	Reference
UART 0	`0x0000`	UART to Cell BE	Sect. 4.5.2
UART 1	`0x0010`	UART to Service Processor	Sect. 4.5.2
Ethernet EMAC	`0x0020`	Ethernet EMAC	Ref. [53]
Ethernet GBIF	`0x0030`	Ethernet GBIF	Ref. [53]
Ethernet GTX	`0x0040`	Ethernet GTX	Ref. [53]

Device	DCR_BASE_DEVICE	Description	Reference
CB	0x0080	Control Box	Ref. [53]
IWCEM	0x0210	Inbound Write Controller Extension Module	Ref. [53]
OWCEM	0x0220	Outbound Write Controller Extension Module	Ref. [52]
Torus RX 0	0x3000	Torus Receive link 0	Ref. [55]
Torus RX 1	0x3100	Torus Receive link 1	Ref. [55]
Torus RX 2	0x3200	Torus Receive link 2	Ref. [55]
Torus RX 3	0x3300	Torus Receive link 3	Ref. [55]
Torus RX 4	0x3400	Torus Receive link 4	Ref. [55]
Torus RX 5	0x3500	Torus Receive link 5	Ref. [55]
Torus TX	0x3600	Torus Transmit	Ref. [55]

Table A.1: Public DCR memory map.

Any register with I/O address `addr` in the Cell BE I/O address space and device-specific register address `reg` has to be addressed from the Cell BE relative to the base device address `DCR_BASE_DEVICE`. The target address has to be multiplied by 16, thus

```
addr = DCR_BASE_BE + (DCR_BASE_DEVICE + reg) << 4 .
```

The best practise to access the real address space from the Cell BE is to map the I/O DCR address space into the Cell BE's virtual address space, see Ref. [109] for technicalities. The C–code listed in Fig. A.1 returns the base pointer to `DCR_BASE_BE` in virtual memory.

All public DCR devices are also accessible via the Service Processor at base address `DCR_BASE_SP = 0x0`. No additional multiplication is required to address a device register,

```
addr = DCR_BASE_DEVICE + reg .
```

Note: The DCR device tree should not be modified without detailed knowledge about the device register files. Read or write access to any address outside the register files may cause DCR bus deadlock, node-card instability or checkstop.

```
1  /* map I/O DCR into virtual address space */
2  uint32_t* get_dcr_base(void)
3  {
4    /* open memory device */
5    int fd_dev_mem = open("/dev/mem", O_RDWR);
6    if (fd_dev_mem < 0) die("open /dev/mem failed!\n");
7
8    /* mmap */
9    uint64_t* addr_base = DCR_BASE_BE;
10   unsigned  pagesize  = getpagesize();
11   size_t    size      = (DCR_SIZE_BE/pagesize)*pagesize;
12   if (DCR_SIZE_BE % pagesize) size += pagesize;
13
14   void* mm = (unsigned char*)mmap64((void*)0, size,
15     PROT_READ | PROT_WRITE, MAP_SHARED, fd_dev_mem, addr_base);
16   if (errno) die("mmap failed!\n");
17
18   return (uint32_t*)mm;
19 }
```

Figure A.1: C–code for mapping the I/O DCR real address space into virtual address space.

A.3 witchlib API

The *witchlib* library provides high-level access to the root-card from the master server via TCP/IP. The library is built on top of the low-level functions provided by the *feenlib* library written by S. Solbrig. Each access requires a valid connection to the root-card of the type `feenRC_t`. If not quoted otherwise, a function returns 0 if the execution has been successful. In case of any error the code -1 is returned. A corresponding error message is also provided.

The *witchlib* supports for error statistics on SPI, I^2C, Service Processor, flash memory and Ethernet operations. The statistics are supported if the library is compiled with the flags `STATS` and `STATS_OUT`. Only the public functions are listed below.

---

void witchStatFeenRestart(feenRC_t)

Print statistics for feen restarts.

---

void witchStatInit(**int** retry, **int** mwait)

Initialize statistics counters. Restarts of the root-card connection are delayed by `mwait` milliseconds.

---

void witchStatOut(**void**)

Print statistics.

---

ssize_t witchPushFile(feenRC_t rc, **const char*** filename, **int** append, **const** uint8_t* data, size_t length, **char**** **const** message)

Copy file `filename` from the master server to the root-card. Returns the number of copied bytes.

---

ssize_t witchPullFile (feenRC_t rc, **const char*** filename, uint8_t* data, size_t length, **char**** **const** message)

Copy file from the root-card to `filename` on the master server. Returns the number of copied bytes.

---

int witchRCcpld0Reset(feenRC_t rc, uint8_t addr, **char**** **const** message)

Reset CPLD 0 register defined by `addr`.

---

int witchRCcpld0Read(feenRC_t rc, uint8_t* data, **char**** **const** message)

Read 1 byte from CPLD 0 into `data`.

---

int witchRCcpld0Addr(feenRC_t rc, uint8_t addr, **char**** **const** message)

Set CPLD 0 address `addr`.

---

int witchRCcpld0Data(feenRC_t rc, **int** data, **char**∗∗ **const** message)

Set CPLD 0 `data` (4 bytes).

---

int witchRCcpld0Comm(feenRC_t rc, uint8_t comm, **char**∗∗ **const** message)

Set CPLD 0 command `comm`.

---

int witchRCSpiMux(feenRC_t rc, uint8_t mux, **char**∗∗ **const** message)

Set CPLD 0 SPI multiplexer to `mux`.

---

int witchRCcpld1Reset(feenRC_t rc, uint8_t addr, **char**∗∗ **const** message)

Reset CPLD 1 logic block defined by `addr`.

---

int witchRCcpld1Read(feenRC_t rc, uint8_t∗ data, **char**∗∗ **const** message)

Read 1 byte from CPLD 1 into `data`.

---

int witchRCcpld1Addr(feenRC_t rc, uint8_t addr, **char**∗∗ **const** message)

Set CPLD 1 address `addr`.

---

int witchRCcpld1Data(feenRC_t rc, **int** data, **char**∗∗ **const** message)

Set CPLD 1 `data` (4 bytes).

---

int witchRCcpld1Comm(feenRC_t rc, uint8_t comm, **char**∗∗ **const** message)

Set CPLD 1 command `comm`.

---

int witchNCSpiBackdoorWrite(feenRC_t rc, uint8_t nc, **int** dcr_addr, **int** dcr_data, **char**∗∗ **const** message)

Write 4-byte data `dcr_data` to address `dcr_addr` via SPI backdoor on node-card `nc`.

---

int witchNCSpiBackdoorRead(feenRC_t rc, uint8_t nc, **int** dcr_addr, **int**∗ **const** dcr_data, **char**∗∗ **const** message)

Read 4-byte data `dcr_data` from address `dcr_addr` via SPI backdoor on node-card `nc`.

---

int witchNCvpdRead(feenRC_t rc, uint8_t nc, **int** vpd_addr, uint8_t* vpd_data, uint8_t length, **char**** **const** message)

Read `length` bytes from node-card `nc` VPD at address `vpd_addr` into `vpd_data`.

---

int witchNCvpdWrite(feenRC_t rc, uint8_t nc, **int** vpd_addr, uint8_t* vpd_data, uint8_t length, **char**** **const** message)

Write `length` bytes starting at `*vpd_data` to node-card `nc` VPD at address `vpd_addr`.

---

int witchNCCheckFlashRS485(feenRC_t rc, uint8_t nc, uint8_t* discovery, **char**** **const** message)

Check node-card `nc` presence via Service Processor call. If a node-card is discovered `discovery` returns 1 and 0 otherwise.

---

int witchNCCheckFlash(feenRC_t rc, uint8_t nc, uint8_t* discovery, **char**** **const** message)

Check node-card `nc` presence via SPI flash memory read. If a node-card is discovered `discovery` returns 1 and 0 otherwise.

---

int witchNCIdentifyBitStream(feenRC_t rc, uint8_t nc, **int** offset , **char*** **const** nwpRev, **char**** **const** message)

Get FPGA bitstream revision `nwpRev` via SPI flash memory at offset `offset` from node-card `nc`.

---

int witchNCIdentifySlof(feenRC_t rc, uint8_t nc, **char*** **const** slofRev, **char**** **const** message)

Get SLOF revision `slofRev` via SPI flash memory from node-card `nc`.

A.4 QFC full reference

The QFC provides unique access to the QPACE hardware. It is executed via the command line using

```
qc [options] [<mode>] <target> <action> [parameters]
```

In the following the full reference to all actions supported by the QFC is provided.

A.4.1 PSU specific actions

Only the modes `get` and `power` are supported for the PSUs as target. Each action is listed with a short description.

`qc get psu <list> status`

Retrieve status information including error status, input/output voltages, input/output currents, power consumption, temperatures and fan speeds.

`qc power psu <list> off`

Power off PSU.

`qc power psu <list> on`

Power on PSU.

A.4.2 Node-card specific actions

The modes `get`, `set`, `power`, `clear` and `flash` are supported for the node-cards as target. Each action is listed with a short description.

A.4.2.1 Get mode

The following items give an overview on the actions for the `get` mode on node-cards specified by the `<list>` parameter.

`qc get node <list> status`

Retrieve status information including boot status, power status and status changes.

`qc get node <list> config`

Retrieve node-card configuration information including boot configuration and clock settings.

`qc get node <list> version`

Retrieve version information for FPGA bitstream, SLOF and Service Processor firmware.

`qc get node <list> serial`

Retrieve the node-card's serial number.

`qc get node <list> temp`

Retrieve the temperatures in °C.

`qc get node <list> mac`

Retrieve the MAC address.

`qc get node <list> dump`

Retrieve the Cell BE FIRs from the VPD. Information includes the register dump, preliminary error decoding, timestamp and validity flag.

`qc get node <list> vpd <address> <length>`

Block-read `<length>` bytes of data starting at `<address>` from the VPD. The address must be specified by 4 hexadecimal digits without separator. Output is shown in rows of 16 bytes in hexadecimal representation.

A.4.2.2 Set mode

The following items give an overview on the actions for the `set` mode on node-cards specified by the `<list>` parameter.

`qc set node <list> clock <setting>`

Set the clock source to onboard clock (`<setting>` = 0) or global clock (`<setting>` = 1).

`qc set node <list> vpd <address> <hex0> [<hex1> ...]`

Block-write data to the VPD starting at `<address>`. Each data block `<hex`N`>` represents a data byte in hexadecimal representation. The start address is specified by 4 hexadecimal digits without separator.

`qc set node <list> mac [<hex5> ... <hex0>]`

Set MAC address. If no parameters are defined the MAC address associated with the node-card position is set. The MAC address can be set manually providing 6 data bytes in hexadecimal representation. In this case only a single node-card may be addressed.

A.4.2.3 Power mode

The following items give an overview on the actions for the `power` mode on node-cards specified by the `<list>` parameter.

`qc power node <list> off`

Hard-power off Cell BE and FPGA via Service Processor.

`qc power node <list> on`

Hard-power on Cell BE and FPGA via Service Processor. The node boots automatically into Linux if no error occures.

`qc power node <list> cycle`

Hard-power cycle Cell BE and FPGA via Service Processor. The node boots automatically into Linux if no error occures.

`qc power node <list> sp`

Hard-power cycle Service Processor. This does not affect the Cell and FPGA. The node boots automatically into Linux if no error occures.

`qc power node <list> noboot`

Hard-power on Cell BE and FPGA without booting into Linux.

`qc power node <list> reboot`

Soft-power cycle Cell BE logging into the Linux shell via secure shell and execute Linux command `reboot`. This requires the Cell BE to be bootet into Linux.

`qc power node <list> halt`

Soft-power down the Cell via the Linux command `halt`. This requires the Cell to be bootet into Linux.

`qc power node <list> reset`

Hard-reset the node-card by the external reset lines. This command hard-resets all devices on the node-card.

`qc power node <list> probe`

Perform a test login onto the node-card using secure shell. The `probe` command can be used to test successful boot into Linux.

`qc power node <list> magic`

Probe the node-card state using login via secure shell. If the probe fails a hard-power cycle via the Service Processor is initiated. Both steps are repeated up to five times with a delay time of ninety seconds between power cycling and probing.

`qc power node <list> woof`

Identical to `magic` but performs and evaluates a FlexIO link stress if probing Linux was successful.

A.4.2.4 Clear mode

The following items give an overview on the actions for the `clear` mode on node-cards specified by the `<list>` parameter.

`qc clear node <list> states`

Invalidate error status.

A.4.2.5 Flash mode

The following items give an overview on the actions for the `flash` mode on node-cards specified by the `<list>` parameter.

```
qc flash node <list> linux <file>
```

Copy FPGA bitstream and/or SLOF image `<file>` onto the node-card using secure copy. This flash mode requires the Cell BE to be bootet into Linux. **Subsequently the Cell BE has to be power-cycled using the Linux `reboot` command**. Update of the images fails for any other mode of power cycle.

```
qc flash node <list> fpga <file>
```

Copy FPGA bitstream image `<file>` onto the node-card via root-card. Before the flash operation the Cell BE is hard-powered off. The image is copied in chunks. This splitting allows other instances of `qc` to access the node-card during flash operation. Flashing via root-card is not atomic.

```
qc flash node <list> slof <file>
```

Copy SLOF image `<file>` onto the node-card via root-card. Before the flash operation the Cell BE is hard-powered off. The image is copied in chunks. The splitting allows other instances of `qc` to access the node-card during flash operation. Flashing via root-card is not atomic.

```
qc flash node <list> sp <file>
```

Copy Service Processor firmware image onto the node-card via root-card. Before the flash operation the Service Processor is power-cycled. Booting the updated image requires an additional power cycle of the Service Processor after successful flash operation.

Appendix B

Numerical Simulation

B.1 Krylov-Ritz method

Comparison of spectral properties of the overlap operator with random matrix theory requires the evaluation of the eigenvalues of the matrix

$$D_{\mathrm{ov}}(\mu) = 1 + \gamma_5 \,\mathrm{sgn}[\gamma_5 D_{\mathrm{W}}(\mu)] \,. \tag{B.1}$$

The Wilson-Dirac kernel D_{W} is a sparse matrix with dimensions $4N_cV \times 4N_cV$, defined by Eq. (6.85) for zero chemical potential μ and Eq. (6.108) for the non-zero case. At zero chemical potential the Wilson-Dirac kernel is γ_5-Hermitian and the eigenvalues of $\gamma_5 D_{\mathrm{W}}$ are real. Hermiticity is lost if a real-valued chemical potential is turned on and the spectrum of $\gamma_5 D_{\mathrm{W}}$ is complex. In contrast, the overlap operator is a dense matrix of the same dimension as the Wilson-Dirac operator, and does, besides the lattice chiral symmetry, not exhibit further symmetries. Its (paired) eigenvalues are restricted to lie on the Ginsparg-Wilson circle if the Wilson-Dirac kernel is Hermitian and are distorted from the circle if the chemical potential is turned on.

In this study QCD with gauge group SU(2) was analyzed. The volumes investigated comprise $V = 8^4, 10^4$, and 12^4 lattice points, thus the dimension of the overlap operator is up to $\mathcal{O}(10^5)$. Full matrix diagonalization scales cubic with the dimension of the matrix. Full diagonalization of the overlap operator is prohibitively expensive if the number of spectra desired is rather high, which is the case in this study. However, only a few of the lowest-lying eigenvalues are needed to compare the spectral properties of the overlap operator with random matrix theory.

An efficient algorithm that significantly reduces the compute time of the eigenvalue problem is given by the (restarted) Arnoldi method. This method allows for evaluation of only a portion of the full spectrum and is readily supported by the software package `ARPACK` [110]. The Arnoldi method is based on power iteration, in this particular case defining the Krylov subspace

$$\mathcal{K}_j(D_{\mathrm{ov}}, x) \equiv \mathrm{span}(x, D_{\mathrm{ov}}x, D_{\mathrm{ov}}^2 x, \ldots, D_{\mathrm{ov}}^{j-1}x) \tag{B.2}$$

of $\mathbb{C}^n$ with $j \ll \dim(D_{\mathrm{ov}})$ and some randomly chosen source vector x. `ARPACK` computes an orthogonal basis $Q = (q_1, \ldots, q_j)$ of $\mathcal{K}_j(D_{\mathrm{ov}}, x)$, which is then used for approximation of a portion of the spectrum, i.e., the approximation of the eigenvectors and eigenvalues of D_{ov}. However, the software only provides the source vector x. The matrix-vector product $D_{\mathrm{ov}}x$ has to be performed explicitely. Its evaluation is described in the following.

Computation of the matrix-vector product $D_{\mathrm{ov}}(\mu)x$ requires the evaluation of the matrix sign function $\mathrm{sgn}(\gamma_5 D_{\mathrm{W}}(\mu))$. In the case of vanishing chemical potential μ the kernel is Hermitian, and its spectrum is real. This is not the case for $\mu \neq 0$, where the spectrum is complex. However, for both cases a proper definition of the matrix sign function is required. In general, some function f

of a diagonalizable matrix $A = U\Lambda U^{-1}$, with diagonal eigenvalue matrix $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$, $U \in \mathrm{Gl}(n, \mathbb{C})$ and $n = \dim(A)$, can be expressed by the spectral form

$$f(A) = U f(\Lambda) U^{-1} \,, \tag{B.3}$$

with

$$f(\Lambda) = \mathrm{diag}\left(f(\lambda_1), \ldots, f(\lambda_n)\right) \,. \tag{B.4}$$

For some complex number $z \in \mathbb{C}$ the sign can be defined as

$$\mathrm{sgn}(z) = \frac{z}{\sqrt{z^2}} = \mathrm{sgn}(\mathrm{Re}(z)) \,, \tag{B.5}$$

with the cut of the square root chosen along the negative real axis. An equivalent definition of the matrix sign function for both Hermitian and non-Hermitian kernel A is given by

$$\mathrm{sgn}(A) = U\,\mathrm{sgn}(\Lambda)U^{-1} = U\,\mathrm{diag}\left(\mathrm{sgn}(\mathrm{Re}(\lambda_1)), \ldots, \mathrm{sgn}(\mathrm{Re}(\lambda_n))\right)U^{-1} \,. \tag{B.6}$$

The numerical approach to the matrix sign function is not to compute $f(A)$ directly, but to evaluate its action on the source vector x, i.e., to compute $y = f(A)x$. An efficient approximative method to compute the matrix sign function is the nested Krylov-Ritz method described in Ref. [88]. Roughly speaking, the approximation to the matrix-vector product $f(A)x \equiv \mathrm{sgn}(A)x$ consists of a dimensional reduction of the matrix A and the evaluation of the matrix sign function on the source vector x in the reduced space, which is then lifted back to the full space.

In the Hermitian case the (large) matrix A is projected onto the Krylov subspace $\mathcal{K}_k(A, x)$ of size k by the projection

$$A_k = PAP = V_k V_k^\dagger A V_k V_k^\dagger = V_k H_k V_k^\dagger \,. \tag{B.7}$$

Here the orthogonal projection matrix is given by $P = V_k V_k^\dagger$. The projected matrix A_k has the dimension of the original matrix A, $\dim(A_k) = \dim(A)$, but at most rank k. The k-dimensional image of A_k is the Ritz matrix $H_k \equiv V_k^\dagger A V_k$. The components of H_k are the projection coefficients of A_k in the orthonormal basis $V_k = (v_1, \ldots, v_k)$. For Hermitian A the one-sided Lanczos algorithm, see Ref. [88], provides a fast method to determine both the basis V_k of the Krylov subspace and the Ritz matrix H_k, the latter being tridiagonal and symmetric. Given V_k and H_k the Krylov-Ritz approximation to $f(A)$ consists of the evaluation of the matrix function only on the Ritz matrix, and the back-projection to the full space. One has

$$f(A) \approx V_k f(H_k) V_k^\dagger \,. \tag{B.8}$$

The full matrix-vector product $y = f(A)x$ is then approximated by

$$y \approx V_k f(H_k) V_k^\dagger x = |x| V_k f(H_k) e_1^{(k)} \,, \tag{B.9}$$

where the first basis vector v_1 of the Krylov subspace $\mathcal{K}_k(A, x)$ is chosen to be collinear with the vector x, i.e., $v_1 = V_k e_1^{(k)} \equiv x/|x|$. Here $e_1^{(k)}$ is the first unit vector of $\mathbb{C}^k$. If the matrix function $f(H_k)$ is known then the approximation to y is proportional to the linear combination of the basis vectors v_i with coefficients determined by the first column of $f(H_k)$,

$$y \approx |x| \sum_{i=1}^{k} f(H_k)_{i1} v_i \,. \tag{B.10}$$

In the non-Hermitian case the matrix function $f(A)$ can be approximated using the right Krylov

subspace $\mathcal{K}_k(A,x)$ with basis $V_k = (v_1,\ldots,v_k)$, and the left Krylov subspace $\mathcal{K}_k(A^\dagger,x)$ with basis $W_k = (w_1,\ldots,w_k)$. The bases V_k and W_k can be constructed using the two-sided Lanczos algorithm, see [88], which also provides the Ritz matrix H_k. Again the Ritz matrix is tridiagonal, but in this case not symmetric. The bases V_k and W_k are biorthogonal, i.e., $v_i^\dagger w_j = \delta_{ij}$. The projector of the full matrix A on the right Krylov subspace is given by $P = V_k W_k^\dagger$, and the projection of A on the Krylov subspace is

$$A_k = PAP = V_k W_k^\dagger A V_k W_k^\dagger = V_k H_k W_k^\dagger \tag{B.11}$$

with the Ritz matrix $H_k \equiv W_k^\dagger A V_k$. The full matrix-vector product $y = f(A)x$ is thus approximated by

$$y \approx V_k f(H_k) W_k^\dagger x = |x| V_k f(H_k) e_1^{(k)} \ , \tag{B.12}$$

again with $v_1 = V_k\, e_1^{(k)} \equiv x/|x|$ and $e_1^{(k)}$ being the first unit vector of $\mathbb{C}^k$. The approximation to y is given by Eq. (B.10) by the linear combination of the basis vectors v_i multiplied by $|x|$.

The size k of the Krylov-subspace has significant impact on the performance of the algorithm. The parameter k, and thus the compute time, can be kept small if an initial left/right deflation step is performed. In this step a small portion of the lowest-lying left/right eigenvalues and corresponding eigenvectors of A are evaluated. Then one has

$$y = f(A)x \approx \sum_{i=1}^{N_d} \mathrm{sgn}(\lambda_i)(\phi_{l,i}^\dagger \cdot x)\phi_{r,i} + |x'| V_k f(H_k) e_1^{(k)} \ , \tag{B.13}$$

where the $\phi_{r,i}$ ($\phi_{l,i}$) are the N_d right (left) eigenvectors of A with corresponding eigenvalues λ_i. In the Hermitian case only the right eigenvectors are needed and thus $\phi_{l,i} = \phi_{r,i}$. The vector x' represents the source vector x with all components along the eigenvectors removed. Deflation allows for efficient evaluation of $f(H_k) \equiv \mathrm{sgn}(H_k)$ by the matrix-iterative Roberts-Higham method. One iterates

$$H_k^{(n+1)} = \frac{1}{2}\left(H_k^{(n)} + (H_k^{(n)})^{-1}\right) \ , \tag{B.14}$$

and chooses $H_k^{(0)} \equiv H_k$, which is the original Ritz matrix. The Roberts-Higham method converges to the matrix sign function within a few iterations.

The compute time of the matrix-vector product can be reduced further by evaluation of $\mathrm{sgn}(H_k)e_1^{(k)}$ on a nested Krylov subspace of size $l \ll k$. Nesting reduces the computational effort from $\mathcal{O}(k^3)$ to $\mathcal{O}(l^3) + \mathcal{O}(kl)$. The approximation is

$$\mathrm{sgn}(H_k)e_1^{(k)} \approx V_l\, \mathrm{sgn}(H_l) e_1^{(l)} \ , \tag{B.15}$$

where V_l collects the basis vectors of the nested Krylov subspace. This basis can also be constructed by the one- or two-sided Lanczos algorithm. Here H_l is the resulting Ritz matrix and $e_1^{(l)}$ is the first unit vector of $\mathbb{C}^l$. Evaluation proceeds on the Krylov subspace $\mathcal{K}_l(H'_k, e_1^{(k)})$ with preconditioned Ritz matrix

$$H'_k = \frac{1}{2}\left(pH_k + (pH_k)^{-1}\right) \ . \tag{B.16}$$

Here $p \in \mathbb{R}^+$ is a preconditioning factor that leaves $\mathrm{sgn}(H_k)$ unchanged and is chosen such that the condition number of the (original) matrix A is minimized. For the (non-)Hermitian Wilson-Dirac kernel a good choice is $p = (5.3 \cdot |\lambda_{N_d}|)^{-1/2}$, with λ_{N_d} being the largest eigenvalue of $\gamma_5 D_\mathrm{W}$ (in absolute value) obtained from deflation.

In the following the basic steps of the algorithm are summarized. The first step is deflation and has to be evaluated only once. The steps 2–6 have to be evaluated for each source vector x provided

by ARPACK.

1. The N_d right eigenvectors ϕ_r and eigenvalues λ of the matrix A are evaluated using the Arnoldi method provided by ARPACK [110]. If A is non-Hermitian then additionally the N_d left eigenvectors ϕ_l are evaluated.

2. The components of the source vector x along the eigenvectors are removed,

$$x' = x - \sum_{i=1}^{N_d} (\phi_{l,i}^\dagger \cdot x)\phi_{r,i} \; . \tag{B.17}$$

   If A is Hermitian then $\phi_{l,i} = \phi_{r,i}$.

3. The one- or two-sided Lanczos algorithm is run with A and x'. One obtains the tridiagonal Ritz matrix H_k and the orthogonal basis V_k of $\mathcal{K}_k(A, x')$. Here $k \ll \dim(A)$.

4. The Lanczos algorithm is run with the preconditioned Ritz matrix H'_k defined in Eq. (B.16) and source vector $e_1^{(k)}$. One obtains the Ritz matrix H_l and the orthogonal basis V_l of the nested Krylov subspace $\mathcal{K}_l(H'_k, e_1^{(k)})$. Here $l \ll k$. The Roberts-Higham iteration on H'_k can be computed efficiently by application of a prior LU decomposition of pH_k and application of optimized linear algebra routines from the BLAS and LAPACK software libraries, see Ref. [88] for details.

5. The Roberts-Higham iteration Eq. (B.14) is run on H_l. One obtains an estimate for the matrix sign function $\mathrm{sgn}(H_l)$.

6. Computation of

$$y = \mathrm{sgn}(A)x \approx \sum_{i=1}^{N_d} \mathrm{sgn}(\lambda_i)(\phi_{l,i}^\dagger \cdot x)\phi_{r,i} + |x'|V_k V_l \,\mathrm{sgn}(H_l) e_1^{(l)} \; , \tag{B.18}$$

   where $\phi_{l,i} = \phi_{r,i}$ if A is Hermitian.

On the order of 10^6 spectra of the overlap operator, each with $N_{\mathrm{ev}} = 20 \ldots 40$ eigenvalues, were evaluated for this study. For deflation (step 1) the accuracy of the residuals was chosen $\varepsilon_d = \mathcal{O}(10^{-12})$. The relative accuracy of the matrix-vector product Eq. (B.18) is determined by the deflation gap and by the sizes k and l of the Krylov subspaces. An accuracy $\varepsilon_s = \mathcal{O}(10^{-8})$ was achieved with even $k = 350 \ldots 400$, $l \approx k/8$, and $N_d = 60 \ldots 80$ deflated eigenvalues. The accuracy of the residuals for the calculation of the eigenvalues of the overlap operator was $\varepsilon_{\mathrm{ev}} = \mathcal{O}(10^{-6})$.

The Wilson-Dirac operator was evaluated using CHROMA [107]. Linear algebra was performed using routines from BLAS and LAPACK optimized for local computer installations. All calculations were carried out on the Athene HPC cluster, the iDataCool HPC cluster, and the server cluster of the physics department at the University of Regensburg.

B.2 Statistical bootstrap

Dealing with some unknown distribution function X the bootstrap method provides a simple and efficient way to obtain non-parametric characteristics θ of the distribution X. Given only a sample distribution $\hat{X}_n = (x_1, \ldots, x_n)$ of X of size n and some sample statistic $\hat{\theta}_n$ derived from $\hat{X}_n$, one can give an estimate for the uncertainty in $\hat{\theta}_n$ using the bootstrap method, see Ref. [75]. This method is based on B-fold resampling of the items x_i of the sample distribution $\hat{X}_n$ of X. Each bootstrap

resample X_i is an independent and identically distributed random sequence from $\hat{X}_n$, i.e., a random sample of size n of the items x_i drawn with replacement. Let θ_i^* be the corresponding statistics of the bootstrap resample X_i, then one calculates

$$\sigma_B^2 = \frac{1}{B}\sum_{i=1}^{B}(\theta_i^* - \overline{\theta}^*)^2 \ , \tag{B.19}$$

with the mean of the bootstrap samples

$$\overline{\theta}^* = \frac{1}{B}\sum_{i=1}^{B}\theta_i^* \ . \tag{B.20}$$

The estimate for the unknown statistic θ of X is then given by

$$\theta = \hat{\theta}_n \pm \sigma_B \ . \tag{B.21}$$

The actual choice for the number B of bootstrap resamples differs widely in literature. In this study the bootstrap method was applied to estimate the error on the best-fit for the chiral condensate, the pion decay constant and the goodness-of-fit χ^2/dof. The spectral densities were constructed by resampling of sets of eigenvalues of the overlap operator (the sets correspond to the items x_i). The number of resamples was restricted to $B = 1000$ for all such estimates in this study. This choice for B was motivated by (i) the scaling of the error on σ_B by $\mathcal{O}(1/\sqrt{B})$, and (ii) the intense compute time required for fitting the densities. All compute-intensive calculations were carried out on the server cluster of the physics department at the University of Regensburg.

References

[1] Top 500 Supercomputer Sites (accessed 1.3.13), http://www.top500.org/.

[2] Cray XK7 "Titan" at Oak Ridge National Laboratory (accessed 1.3.13), http://www.olcf.ornl.gov/titan/.

[3] IBM BlueGene/Q "JUQUEEN" at Forschungszentrum Jülich (available 1.3.2013), http://www.fz-juelich.de/ias/jsc/EN/Expertise/Supercomputers/JUQUEEN/JUQUEEN_node.html.

[4] SuperMUC at Leibniz Rechenzentrum Garching (accessed 1.3.13), http://www.lrz.de/services/compute/supermuc/.

[5] Wikipedia article on K Computer (accessed 1.3.13), http://en.wikipedia.org/wiki/K_computer.

[6] Cray XMT (accessed 1.3.13), http://wwwjp.cray.com/downloads/XMT-Presentation.pdf,.

[7] Smith R., "nVidia launches Tesla K20 and K20X: GK110 arrives at last", Anandtech (accessed 1.3.13), http://www.anandtech.com/show/6446/nvidia-launches-tesla-k20-k20x-gk110-arrives-at-last.

[8] Intel Many Integrated Core architecture (accessed 1.3.13), http://www.intel.com/content/www/us/en/architecture-and-technology/many-integrated-core/intel-many-integrated-core-architecture.html.

[9] http://www.readwriteweb.com/archives/obama_budget_includes_126_million_for_exascale_com.php (accessed 1.3.13).

[10] http://www.zdnet.co.uk/news/emerging-tech/2012/02/16/eu-to-double-supercomputing-funding-to-12bn-40095059/ (accessed 1.3.13).

[11] European Exascale Project DEEP homepage (accessed 1.3.13), www.deep-project.eu.

[12] IDC White Paper #231528, "Server Transition Alternatives: A Business Value View Focusing on Operating Costs", (2012), (accessed 1.3.13), http://www-05.ibm.com/de/events/breakfast/pdf/IDC_Operating_Cost_2012_IBM_POWER7_231528.pdf.

[13] T. Scogland, B. Subramaniam, and W. Feng, "Emerging Trends on the Evolving Green500: Year Three", *IEEE International Symposium on Parallel and Distributed Processing Workshops and PhD Forum* (2011) 822.

[14] The Green500 List (accessed 1.3.13), http://www.green500.org/.

[15] SPE Runtime Management Library Version 2.2 (accessed 1.3.13), http://www.bsc.es/computer-sciences/programming-models/linux-cell/cell-be-sdks/sdk-31/cell-be-components/libspe-2.

[16] QPACE homepage at Forschungszentrum Jülich (accessed 1.3.13), http://www.fz-juelich.de/portal/EN/Research/InformationTechnology/Supercomputer/QPACE.html.

[17] Bergische Universität Wuppertal, Pressestelle (accessed 1.3.13), http://www.presse-archiv.uni-wuppertal.de/2009/1120_qpace.html, http://www.presse-archiv.uni-wuppertal.de/2010/0701_qpace.html.

[18] H. Schick, H. Penner, and O. Wohlmuth, "LAB: Slimline Open Firmware". Power Architecture Developer Conference, 2007.

[19] H. Baier, H. Boettinger, M. Drochner, N. Eicker, U. Fischer, Z. Fodor, A. Frommer, C. Gomez, G. Goldrian, S. Heybrock, D. Hierl, M. Hüsken, T. Huth, B. Krill, J. Lauritsen, T. Lippert, T. Maurer, B. Mendl, N. Meyer, A. Nobile, I. Ouda, M. Pivanti, D. Pleiter, M. Ries, A. Schäfer, H. Schick, F. Schifano, H. Simma, S. Solbrig, T. Streuer, K.-H. Sulanke, R. Tripiccione, J.-S. Vogt, T. Wettig, and F. Winter, "QPACE: power-efficient parallel architecture based on IBM PowerXCell 8i", *Comput. Sci. Res. Dev.* **25** (2010) 149.

[20] H. Baier, H. Boettinger, M. Drochner, N. Eicker, U. Fischer, Z. Fodor, A. Frommer, C. Gomez, G. Goldrian, S. Heybrock, D. Hierl, M. Hüsken, T. Huth, B. Krill, J. Lauritsen, T. Lippert, T. Maurer, B. Mendl, N. Meyer, A. Nobile, I. Ouda, M. Pivanti, D. Pleiter, M. Ries, A. Schäfer, H. Schick, F. Schifano, H. Simma, S. Solbrig, T. Streuer, K.-H. Sulanke, R. Tripiccione, J.-S. Vogt, T. Wettig, and F. Winter, "QPACE - a QCD parallel computer based on Cell processors", *PoS* (LAT2009) 001, 0911.2174.

[21] N. Meyer, S. Solbrig, T. Wettig, A. Auweter, and H. Huber, "Data centre infrastructure requirements", European Exascale Project DEEP (2012).

[22] N. Meyer, T. Wettig, S. Solbrig, A. Auweter, M. Ott, and D. Tafani, "Concepts for improving energy and cooling efficiency", European Exascale Project DEEP (2012).

[23] M. Pivanti, F. Schifano, and H. Simma, "An FPGA-based Torus Communication Network", *PoS* (LAT2010) 038, 1102.2346.

[24] H. Simma, "Architecture of the QPACE Torus Network". http://www2.fz-juelich.de/jsc/datapool/cell/eQPACE/simma-tnw-arch.pdf.

[25] H. Baier, S. Heybrock, B. Krill, F. Mantovani, T. Maurer, N. Meyer, I. Ouda, M. Pivanti, P. Pleiter, S. F. Schifano, and H. Simma, *High-Performance Computing Using FPGAs*, ch. High-speed torus interconnect using FPGAs. Springer (due may 31, 2013).

[26] A. Nobile and D. Pleiter, "QPACE system software: torus". http://hpc.desy.de/sites2009/site_hpc/content/e476/e75391/e63656/syssw-torus.pdf.

[27] S. Solbrig, "Synchronization in QPACE". STRONGnet Conference, Cyprus 2010 (accessed 1.3.13), http://www.physik.uni-regensburg.de/STRONGnet/documents/STRONGnet2010/solbrig.pdf.

[28] QPACE homepage at Deutsches Elektronen-Synchrotron (accessed 1.3.13), http://hpc.desy.de/qpace/.

[29] Embedded Linux/Microcontroller Project homepage (accessed 1.3.13), http://www.uclinux.org/.

[30] FMod-TCP DB (accessed 1.3.13), http://www.fiveco.ch/product-fmod-tcp-db.html.

[31] S. Williams, J. Shalf, L. Oliker, S. Kamil, P. Husbands, and K. Yelick, "The potential of the cell processor for scientific computing", in *Proceedings of the 3rd conference on Computing frontiers*, pp. 9–20. ACM, New York, NY, USA, 2006.

[32] Los Alamos National Laboratory HPC Roadrunner (accessed 1.3.13), http://www.lanl.gov/roadrunner/.

[33] J. A. Kahle, M. N. Day, H. P. Hofstee, C. R. Johns, T. R. Maeurer, and D. Shippy, "Introduction to the Cell multiprocessor", *IBM J. Res. Dev.* **49** (2005) 589.

[34] Cell Broadband Engine Architecture (accessed 1.3.13), https://www-01.ibm.com/chips/techlib/techlib.nsf/techdocs/1AEEE1270EA2776387257060006E61BA.

[35] Cell BE Programming Handbook (accessed 1.3.13), https://www-01.ibm.com/chips/techlib/techlib.nsf/techdocs/7A77CCDF14FE70D5852575CA0074E8ED.

[36] Intel comment on the IEEE 754 standard for binary floating-point arithmetic (accessed 1.3.13), http://www.intel.com/standards/floatingpoint.pdf.

[37] Wikipedia article on the Pentium FDIV bug (accessed 1.3.13), http://en.wikipedia.org/wiki/Pentium_FDIV_bug.

[38] Intel Xeon E7-8870 Specifications (accessed 1.3.13), http://ark.intel.com/products/53580/Intel-Xeon-Processor-E7-8870-30M-Cache-2_40-GHz-6_40-GTs-Intel-QPI.

[39] nVidia Tesla Data Center Solutions, Tesla M2090 (accessed 1.3.13), http://www.nvidia.com/object/preconfigured-clusters.html, http://www.nvidia.com/docs/IO/105880/DS-Tesla-M-Class-Aug11.pdf.

[40] G. Bilardi, A. Pietracaprina, G. Pucci, F. Schifano, and R. Tripiccione, "The Potential of On-Chip Multiprocessing for QCD Machines", *Lecture Notes in Comput. Sci.* **3769** (2005) 386.

[41] F. Belletti, G. Bilardi, M. Drochner, N. Eicker, Z. Fodor, D. Hierl, H. Kaldass, T. Lippert, T. Maurer, N. Meyer, A. Nobile, D. Pleiter, A. Schäfer, F. Schifano, H. Simma, S. Solbrig, T. Streuer, R. Tripiccione, and T. Wettig, "QCD on the Cell Broadband Engine", *PoS* (LAT2007) 039, 0710.2442.

[42] G. Goldrian, T. Huth, B. Krill, J. Lauritsen, H. Schicka, I. Ouda, S. Heybrock, D. Hierl, T. Maurer, N. Meyer, A. Schäfer, S. Solbrig, T. Streuer, T. Wettig, K.-H. S. D. Pleiter, F. Winter, H. Simma, S. Schifano, R. Tripiccione, A. Nobile, M. Drochner, T. Lippert, and Z. Fodor, "QPACE: Quantum Chromodynamics Parallel Computing on the Cell Broadband Engine", *Comput. Sci. Eng.* **10** (2008) 46.

[43] A. Nobile, "Solving the Dirac equation on QPACE", *PoS* (LAT2010) 034, 1109.4279.

[44] Y. Nakamura, A. Nobile, D. Pleiter, H. Simma, T. Streuer, T. Wettig, and F. Winter, "Lattice QCD Applications on QPACE", *Procedia Comput. Sci.* **4** (2011) 841, 1103.1363.

[45] Cell BE SDK Resources (accessed 1.3.13), http://www.bsc.es/computer-sciences/programming-models/linux-cell/cell-be-sdks/sdk-31.

[46] H. Simma, "Lattice QCD on the Cell Processor". Cell Cluster Meeting, Jülich 2007.

[47] P. Boyle, C. Jung, and T. Wettig, "The QCDOC supercomputer: hardware, software, and performance", *hep-lat/0306023* (2003) `hep-lat/0306023`.

[48] F. Belletti, F. Schifano, R. Tripiccione, F. Bodin, P. Boucaud, J. Micheli, O. Pene, N. Cabibbo, S. de Luca, A. Lonardo, D. Rossetti, P. Vicini, M. Lukyanov, L. Morin, N. Paschedag, H. Simma, V. Morenas, D. Pleiter, and F. Rapuano, "Computing for LQCD: apeNEXT", *Comput. Sci. Eng.* **8** (2006) 18.

[49] Xilinx Virtex-5 FPGA Family (accessed 1.3.13), `http://www.xilinx.com/support/documentation/virtex-5.htm`.

[50] Virtex-5 FPGA User Guide (accessed 1.3.13), `http://www.xilinx.com/support/documentation/user_guides/ug190.pdf`.

[51] I. Ouda and K. Schleupen, "Application Note: FPGA to IBM Power Processor Interface Setup", *IBM Research Report* (2008).

[52] S. Heybrock, "A high-performance network controller for the QPACE architecture", Diploma thesis, University of Regensburg, 2008.

[53] T. Maurer, "The QPACE Supercomputer, Renormalization of Dynamical CI Fermions and Axial Charges of Excited Nucleons", Phd thesis, University of Regensburg, 2011.

[54] Device Control Register Bus 3.5 Architecture Specifications (available 1.3.13), `https://www-01.ibm.com/chips/techlib/techlib.nsf/techdocs/2F9323ECBC8CFEE0872570F4005C5739/$file/DcrBus.pdf`.

[55] H. Simma. Notes on the torus link logic v0.15 (accessed 1.3.13), `http://moby.mib.infn.it/~simma/tnw/notes-link.pdf`.

[56] T. Flik, *Mikroprozessortechnik und Rechnerstrukturen.* Berlin: Springer, 7th ed., 2005.

[57] J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach.* Amsterdam: Morgan Kaufman, 4th ed., 2006.

[58] D. A. Patterson and J. L. Hennessy, *Computer Organization and Design: The Hardware/Software Interface.* Amsterdam: Morgan Kaufman, 3rd ed., 2007.

[59] P. P. Chu, *RTL Hardware Design using VHDL.* New Jersey: John Wiley and Sons, 1st ed., 2006.

[60] S. Kilts, *Advanced FPGA Design: Architecture, Implementation and Optimization.* New Jersey: John Wiley and Sons, 1st ed., 2007.

[61] National Semiconductor, PC16550D Universal Asynchronous Receiver/Transmitter with FIFO's (accessed 1.3.13), `http://www.national.com/profile/snip.cgi/openDS=PC16550D`.

[62] Nagios IT Infrastructure Monitoring (accessed 1.3.13), `http://www.nagios.org/`.

[63] Z. Fodor and C. Hoelbling, "Light Hadron Masses from Lattice QCD", *Rev. Mod. Phys.* **84** (2012) 449, `1203.4789`.

[64] J. Verbaarschot, "The spectrum of the QCD Dirac operator and chiral random matrix theory: the threefold way", *Phys. Rev. Lett.* **72** (1994) 2531, hep-th/9401059.

[65] J. Verbaarschot, "The spectrum of the Dirac operator near zero virtuality for $N_c = 2$", *Nucl. Phys.* **B426** (1994) 559, hep-th/9401092.

[66] J. Verbaarschot and I. Zahed, "Spectral Density of the QCD Dirac Operator near Zero Virtuality", *Phys. Rev. Lett.* **70** (1993) 3852, hep-th/9303012.

[67] T. Nagao and P. Forrester, "Asymptotic correlations at the spectrum edge of random matrices", *Nucl. Phys.* **B435** (1995) 401.

[68] R. Edwards and U. Heller, "Quark Spectra, Topology, and Random Matrix Theory", *Phys. Rev. Lett.* **82** (1999) 4188, hep-th/9902117.

[69] W. Bietenholz, S. Shcheredin, and K. Jansen, "Spectral properties of the overlap Dirac operator in QCD", *JHEP* **0307** (2003) 033, hep-lat/0306022.

[70] H. Fukaya, S. Aoki, T. W. Chiu, S. Hashimoto, T. Kaneko, H. Matsufuru, J. Noaki, K. Ogawa, M. Okamoto, T. Onogi, and N. Yamada, "Two-flavor lattice QCD in the epsilon-regime and chiral Random Matrix Theory", *Phys. Rev.* **D76** (2007) 054503, 0705.3322.

[71] G. Akemann, M. J. Phillips, and H.-J. Sommers, "The chiral Gaussian two-matrix ensemble of real asymmetric matrices", *J. Phys.* **A43** (2010) 085211, 0911.1276.

[72] G. Akemann, T. Kanazawa, M. J. Phillips, and T. Wettig, "Random matrix theory of unquenched two-colour QCD with nonzero chemical potential", *JHEP* **3** (2011) 66, 1012.4461.

[73] T. Kanazawa, T. Wettig, and N. Yamamoto, "Chiral random matrix theory for two-color QCD at high density", *Phys. Rev.* **D81** (2010) 081701, 0912.4999.

[74] J. Bloch and T. Wettig, "Overlap Dirac operator at nonzero chemical potential and random matrix theory", *Phys. Rev. Lett.* **97** (2006) 012003, hep-lat/0604020.

[75] C. Gattringer and C. B. Lang, *Quantum Chromodynamics on the Lattice: An Introductory Presentation*, vol. 788 of *Lecture Notes in Physics*. Berlin: Springer, 1st ed., 2010.

[76] S. Chandrasekharan and U.-J. Wiese, "An Introduction to Chiral Symmetry on the Lattice", *Prog. Part. Nucl. Phys.* **53** (2004) 373, hep-lat/0405024.

[77] J. B. Kogut, M. A. Stephanov, D. Toublan, J. J. M. Verbaarschot, and A. Zhitnitsky, "QCD-like Theories at Finite Baryon Density", *Nucl. Phys.* **B582** (2000) 477, hep-ph/0001171.

[78] A. Smilga, "Chiral symmetry and spectrum of Euclidean Dirac operator in QCD", hep-th/9503049.

[79] K. G. Wilson, "Confinement of quarks", *Phys. Rev.* **D10** (1974) 2445.

[80] M. Atiyah and I. Singer, "The index of elliptic operators V", *Ann. Math.* **93** (1971) 139.

[81] C. Gattringer and I. Hip, "On the spectrum of the Wilson-Dirac lattice operator in topologically non-trivial background configurations", *Nucl. Phys.* **B536** (1998) 363, hep-lat/9712015.

[82] H. B. Nielsen and M. Ninomiya, "A no-go theorem for regularizing chiral fermions", *Phys. Lett.* **B105** (1981) 219.

[83] P. Ginsparg and K. Wilson, "A remnant of chiral symmetry on the lattice", *Phys. Rev.* **D25** (1982) 2649.

[84] P. Hasenfratz, V. Laliena, and F. Niedermayer, "The index theorem in QCD with a finite cut-off", *Phys. Lett.* **B427** (1998) 125, `hep-lat/9801021`.

[85] T.-W. Chiu, "Topological Charge and The Spectrum of Exactly Massless Fermions on the Lattice", *Phys. Rev.* **D58** (1998) 074511, `hep-lat/9804016`.

[86] R. Narayanan and H. Neuberger, "A construction of lattice chiral gauge theories", *Nucl. Phys.* **B443** (1995) 305, `hep-th/9411108`.

[87] H. Neuberger, "Exactly massless quarks on the lattice", *Phys. Lett.* **B417** (1998) 141, `hep-lat/9707022`.

[88] J. Bloch and S. Heybrock, "A nested Krylov subspace method to compute the sign function of large complex matrices", *Comput. Phys. Comm.* **182** (2011) 878, `0912.4457`.

[89] F. Niedermayer, "Exact chiral symmetry, topological charge and related topics", *Nucl. Phys. Proc. Suppl.* **73** (1999) 105, `hep-lat/9810026`.

[90] P. Hernandez, K. Jansen, and M. Lüscher, "Locality properties of Neuberger's lattice Dirac operator", *Nucl. Phys.* **B552** (1999) 363, `hep-lat/9808010`.

[91] H. Neuberger, "Vector like gauge theories with almost massless fermions on the lattice", *Phys. Rev.* **D57** (1998) 5417, `hep-lat/9710089`.

[92] P. Hasenfratz and F. Karsch, "Chemical potential on the lattice", *Phys. Lett.* **B125** (1983) 308.

[93] J. Bloch and T. Wettig, "Domain-wall and overlap fermions at nonzero quark chemical potential", *Phys. Rev.* **D76** (2007) 114511, `0709.4630`.

[94] T. Guhr, A. Mueller-Groeling, and H. A. Weidenmueller, "Random Matrix Theories in Quantum Physics: Common Concepts", *Phys. Rept.* **299** (1998) 189, `cond-mat/9707301`.

[95] J. Verbaarschot and T. Wettig, "Random Matrix Theory and Chiral Symmetry in QCD", *Ann. Rev. Nucl. Part. Sci.* **50** (2000) 343, `hep-ph/0003017`.

[96] E. Shuryak and J. Verbaarschot, "Random Matrix Theory and spectral sum rules for the Dirac operator in QCD", *Nucl. Phys.* **A560** (1993) 306.

[97] P. Forrester, "The spectrum edge of random matrix ensembles", *Nucl. Phys.* **B402** (1993) 709.

[98] Scientific Python (accessed 1.3.13), `http://www.scipy.org/`.

[99] M. G. et al, "GNU Scientific Library Reference Manual - Third Edition", 2009.

[100] T. Hahn, "Cuba - a library for multidimensional numerical integration", *Comput. Phys. Comm.* **168** (2005) 78, `hep-ph/0404043`.

[101] K. Splittorff and A. D. Jackson, "The Ginsparg-Wilson relation and local chiral random matrix theory", *hep-lat/9805018* (1998) `hep-lat/9805018`.

[102] J. Bloch, F. Bruckmann, N. Meyer, and S. Schierenberg, "Level spacings for weakly asymmetric real random matrices and application to two-color QCD with chemical potential", *JHEP* **08** (2012) 066, 1204.6259.

[103] T. Banks and A. Casher, "Chiral symmetry breaking in confining theories", *Nucl. Phys.* **B169** (1980) 103.

[104] H. Leutwyler and A. Smilga, "Spectrum of Dirac operator and role of winding number in QCD", *Phys. Rev.* **D46** (1992) 5607.

[105] Slatec (accessed 1.3.13), http://people.sc.fsu.edu/~jburkardt/f_src/slatec/slatec.html.

[106] K. Langfeld, "Improved actions and asymptotic scaling in lattice Yang-Mills theory", *Phys. Rev.* **D76** (2007) 094502, 0704.2635.

[107] SciDAC Collaboration, R. G. Edwards, and B. Joo, "The Chroma software system for lattice QCD", *Nucl. Phys. Proc. Suppl.* **140** (2005) 832.

[108] R. Edwards, U. Heller, and R. Narayanan, "Spectral flow, condensate and topology in lattice QCD", *Nucl. Phys.* **B535** (1998) 403, hep-lat/9802016.

[109] R. Stevens and S. A. Rago, *Advanced Programming in the UNIX Environment*. Amsterdam: Addison-Wesley, 2nd ed., 2008.

[110] Arpack (accessed 1.3.13), http://www.caam.rice.edu/software/ARPACK/.

List of Publications

The following publications are directly related to this work:

F. Belletti, G. Bilardi, M. Drochner, N. Eicker, Z. Fodor, D. Hierl, H. Kaldass, T. Lippert, T. Maurer, N. Meyer, A. Nobile, D. Pleiter, A. Schäfer, F. Schifano, H. Simma, S. Solbrig, T. Streuer, R. Tripiccione, and T. Wettig, "QCD on the Cell Broadband Engine", *PoS* (LAT2007) 039

H. Baier, H. Boettinger, M. Drochner, N. Eicker, U. Fischer, Z. Fodor, G. Goldrian, S. Heybrock, D. Hierl, T. Huth, B. Krill, J. Lauritsen, T. Lippert, T. Maurer, J. McFadden, N. Meyer, A. Nobile, I. Ouda, M. Pivanti, D. Pleiter, A. Schäfer, H. Schick, F. Schifano, H. Simma, S. Solbrig, T. Streuer, K.-H. Sulanke, R. Tripiccione, T. Wettig, and F. Winter, "Status of the QPACE Project", *PoS* (LAT2008) 039

G. Goldrian, T. Huth, B. Krill, J. Lauritsen, H. Schick, I. Ouda, S. Heybrock, D. Hierl, T. Maurer, N. Meyer, A. Schäfer, S. Solbrig, T. Streuer, T. Wettig, D. Pleiter, K.-H. Sulanke, F. Winter, H. Simma, S. Schifano, R. Tripiccione, A. Nobile, M. Drochner, T. Lippert, and Z. Fodor, "QPACE: Quantum Chromodynamics Parallel Computing on the Cell Broadband Engine", *Comput. Sci. Eng.* **10** (2008), 46

H. Baier, H. Boettinger, M. Drochner, N. Eicker, U. Fischer, Z. Fodor, A. Frommer, C. Gomez, G. Goldrian, S. Heybrock, D. Hierl, M. Hüsken, T. Huth, B. Krill, J. Lauritsen, T. Lippert, T. Maurer, B. Mendl, N. Meyer, A. Nobile, I. Ouda, M. Pivanti, D. Pleiter, M. Ries, A. Schäfer, H. Schick, F. Schifano, H. Simma, S. Solbrig, T. Streuer, K.-H. Sulanke, R. Tripiccione, J.-S. Vogt, T. Wettig, and F. Winter, "QPACE – a QCD parallel computer based on Cell processors", *PoS* (LAT2009) 001

H. Baier, H. Boettinger, M. Drochner, N. Eicker, U. Fischer, Z. Fodor, A. Frommer, C. Gomez, G. Goldrian, S. Heybrock, D. Hierl, M. Hüsken, T. Huth, B. Krill, J. Lauritsen, T. Lippert, T. Maurer, B. Mendl, N. Meyer, A. Nobile, I. Ouda, M. Pivanti, D. Pleiter, M. Ries, A. Schäfer, H. Schick, F. Schifano, H. Simma, S. Solbrig, T. Streuer, K.-H. Sulanke, R. Tripiccione, J.-S. Vogt, T. Wettig, and F. Winter, "QPACE: power-efficient parallel architecture based on IBM PowerXCell 8i", *Comput. Sci. Res. Dev.* **25** (2010), 149

H. Baier, S. Heybrock, B. Krill, F. Mantovani, T. Maurer, N. Meyer, I. Ouda, M. Pivanti, D. Pleiter, S. F. Schifano, and H. Simma, "High-speed torus interconnect using FPGAs". In W. Vanderbauwhede and K. Benkrid (Eds), "High-Performance Computing using FPGAs", Springer, accepted for publication in July 2012 (due may 31, 2013)

Danksagung

Abschließend möchte ich mich bei einer Vielzahl von Personen bedanken ohne die diese Arbeit nicht zustande gekommen wäre. Die Konstruktion eines Supercomputers innerhalb der theoretischen Physik ist sicherlich ein eher exotisches Thema, nichtsdestotrotz nicht minder interessant wie Physik selbst.

Zu Dank verpflichtet fühle ich mich insbesondere Prof. Dr. Tilo Wettig für die Möglichkeit an QPACE mitzuwirken. Ich bedanke mich auch für die Teilnahme an weiteren interessanten Projekten wie iDataCool, DEEP und den QPACE Nachfolger.

Ich möchte mich auch bei allen Beteiligten von QPACE für die sehr gute Zusammenarbeit bedanken, insbesondere aber bei Prof. Dr. Andreas Schäfer, Prof. Dr. Dirk Pleiter, Dr. Stefan Solbrig, Dr. Thilo Maurer, Dr. Thomas Streuer, Dr. Dieter Hierl, Matthias Hüsken, Uwe Fischer, Benjamin Krill, Thomas Huth sowie auch Dr. Hubert Simma. Erwähnen möchte ich hier auch das Sekretariat der Teilchenphysik – Monika Maschek und Heidi Decock.

Besonderer Dank gilt Dr. Jacques Bloch für zahlreiche Diskussionen bezüglich des zweiten Teil dieser Arbeit.

Ganz besonders freue ich mich über den beständigen Kontakt zu ehemaligen Kollegen innerhalb IBM Deutschland. Vielen Dank an Uwe Fischer für die Organisation des jährlichen Ausflugs nach Sölden.

Besonderer Dank gilt auch Denise Trapp für ihre Geduld und ihren Beistand trotz meiner zeitweisen geistigen Abwesenheit.