Faculty of Physics and Astronomy

University of Heidelberg

Diploma thesis

in Physics

submitted by

Matthias Hock

born in Munich

August 2009

Test of Components for a Wafer-Scale Neuromorphic Hardware System

This diploma thesis has been carried out by Matthias Hock at the
Kirchhoff Institute for Physics
University of Heidelberg
under the supervision of
Prof. Dr. Karlheinz Meier

Test of Components for a Wafer-Scale Neuromorphic Hardware System

This thesis presents the results of testing two components of a new neuromorphic hardware chip, which is the basic unit of a wafer-scale system. The first topic is an asynchronous bus for transferring presynaptic neural events across the wafer using a packet-based protocol and low-voltage differential signaling. Maximum available data rates and power consumption are the main features, crosstalk and latencies are further aspects which have been investigated using a prototype chip. Crosstalk turns out to be a problem. The councilation circuitry does not work sufficiently. Despite crosstalk the results are mostly within expectations. Reliable transmission is possible for data rates up to 1.6GBit/s within a packet. The second topic is analog floating-gate memory cells. These will be used to store the analog parameters required for adapting and calibrating the neuron implementations. Another prototype chip is used to primarily test the HDL code for the controller and also the addressing circuitry for the memory array. A few measurements concerning performance of the cells, especially their accuracy, are also presented.

Test von Komponenten eines neuromorphem Hardware-Systems auf Wafer-Ebene

Die vorliegende Arbeit stellt die Ergebnisse von Tests an zwei Komponenten eines neuromorphen Hardware Chips vor. Dieser ist der Grundbaustein eines größeren, einen ganzen Wafer umfassenden, Systems. Der erste Teil der Arbeit befasst sich mit einem asynchronen Bus mit Paket-basiertem Protokoll. Dieser diente der Übertragung präsynaptischer neuronaler Ereignisse zwischen verschienden Neuronen auf dem Wafer. Ein prototypen Chip wurde entworfen und genutzt um die die maximal erzielbare Datenübertragungsrate sowie der Energieverbrauch zu bestimmt. Zusätzlich wurde das übersprechen zwischen benachbarten Leitungen und Lantenzen des Systems untersucht. Es stellt sich haerraus dass Überspechen die Übertragung beeinträchtigt da die Schaltungen zur Kompensation dieses Effekts nicht ausreichend sind. Abgesehen von dieser Problematik entsprechen die Ergebnisse den Erwartungen. Eine zuverlässige Übertagung mit einer Datenrate von bis zu 1,6 GBit/s innerhalb der Pakete ist möglich. Der zweite Abschnitt befasst sich mit analogen Floating-Gate Speicherzellen. Diese sollen zur Speicherung von Parametern verwendet werden, die zur Anpassung und Kalibrierung der Implementierung neuronaler Strukturen erforderlich sind. Ein weiterer Testchip wird eingesetzt um den HDL Code für den Controller sowie die Adressierungs-Logik für die Speichermatrix zu testen. Desweiteren werden einzelne Messungen zum Leistungsvermögens der Zellen, insbesondere bezüglich ihrer Genauigkeit, dargestellt.

Contents

Introduction **1**

1 Neuromorphic Hardware **3**
1.1 The Electronic Vision(s) Group as a Part of the FACETS Project 4
1.2 FACETS Stage 1 Hardware 4
1.3 FACETS Stage 2 Hardware 5
1.3.1 The HICANN Chip - Building Block of the Stage 2 Hardware 5
1.3.2 Wafer-scale Integration 5

2 The L1 bus prototype chip **9**
2.1 The L1 Communication System 9
2.1.1 Low Voltage Differential Signals 11
2.1.2 DLL 11
2.1.3 Crosstalk within the L1 System 12
2.1.4 The L1 Repeaters Circuitry 13
2.2 Developing the L1 Prototype Chip 16
2.2.1 Mimicking the Metal Densities of the HICANN Chip 18
2.2.2 Number of available IOs 18
2.2.3 Protection against Electrostatic Discharge 19
2.2.4 The Final Layout 21
2.3 Measurements with the L1 Prototype Chip 24
2.3.1 PCB 24
2.3.2 FPGA 24
2.3.3 Generating Serial Input 29
2.3.4 Final Measurement Setup 29
2.3.5 Employing a New Chip 31
2.4 Results 32
2.4.1 DLL 33
2.4.2 Basic Transmission of Data between r0 and r1 34
2.4.3 Delay 38
2.4.4 Horizontal Lines 40
2.4.5 Crosstalk 40
2.4.6 Power Consumption 44
2.4.7 Reasons for Limited Data Rate 47

3 Floating-Gate Memory Cells **51**
3.1 Floating-Gate Memory 52
3.1.1 Digital Floating-Gate Memory 52
3.1.2 The Analog Floating-Gate Cells in Stage 2 52
3.2 Floating-Gate Cells on the HICANN chip 54

3.2.1 Floating Gate Cell Array 54
3.2.2 Programming the Array 55
3.3 Testing the Floating-Gate Prototype Chip 57
3.3.1 Experimental Setup 58
3.3.2 ESD Protection Related Problems 60
3.3.3 Power-Up Sequence 61
3.4 Results 61
3.4.1 Testing of Memory Array and Controller 62
3.4.2 Performance of the Floating-Gate Cells 62

4 Conclusion **67**

Bibliography **69**

Acknowledgements **73**

Introduction

The human brain is one of the most complex systems nature has evolved. A network of approximately 10^{11} neurons enables homo sapiens to accomplish physical motion, sensory perception, memory and finally even abstract phenomena such as consciousness. We are all quite familiar with these functions. However, to this day, the underlying mechanisms are poorly understood. In the course of the past decades, the field of biology has made great progress in studying the behavior of single neurons. Using e.g. the squid giant axon as a model, scientists have gained an understanding of the fundamental physiology of a neuron, such as maintenance of a membrane potential and synaptic transmission. Sophisticated experimental methods have been developed. Implanted electrodes allow for monitoring the membrane potential of single, live cells. The invention of the patch-clamp technique [14] permits to investigate the processes at single or multiple ion channels in the cell's membrane. There are also techniques to depict the brain's activity with a certain spatial resolution, such as functional magnetic resonance tomography, see [13]. This allows for the investigation of the brain's structure on a large scale. However, observing an operating neural network remains a great experimental challenge.

Moreover, it is commonly assumed that especially the high-level functions are encoded rather in the topology of the network than within single neuron processes or within discrete areas. Billions of neurons interact via either facilitating or depressing synapses. The length of these connections varies by several orders of magnitude. While some areas display regular structures, often no relation between histological and functional structures is evident [7].

So far the possibilities of investigating the network structure with biological methods is limited. Creating abstract models is a feasible and therefore common approach for gaining further insight into the function of the human brain. The objectives of these models range from detailed descriptions of single neurons to approaches covering entire areas of the cortex. Based on these models the biological processes can be simulated with the help of computer systems. These systems operate sequentially. This implies the transition between discrete states using a limited number of processing units [35]. In contrast, information processing within the brain is carried out by all its neurons in parallel. This discrepancy leads to poor performance of computer-based simulation systems, [23].

An alternative approach to this issue is the development of neuromorphic hardware which can overcome the bottle neck of sequential processing. Neuromorphic hardware consists of eletronic circuits emulating biological neurons. As does the natural antetype, these circuits operate in parallel. As a result, these systems provide improved scalability compared to software-based simulators and are capable of operating in real-time or even faster.

Progress in the field of neuroscience will depend on combining all the methods mentioned. This can only be achieved by an interdisciplinary team with expertise in biology, physiology, mathematics, computer science, electrical engineering and physics.

This diploma thesis is a small contribution to the most technical side of neuroscience, the development of neuromorphic hardware. The new hardware system currently being developed in the Electronic Vision(s) group will provide equipment for experiments at a scale and with a biological relevance superior to today's neuromorphic hardware.

Outline

This thesis begins with a general description of neuromorphic hardware, its advantages and its limitations compared to simulators. I the following an insight into the work of the Electronics Vision(s) group as a part of the FACETS research project is presented, including a survey of the two hardware systems which are developed within FACETS.

There are two main parts, as two very different components of a new neuromorphic hardware system have been tested, the L1 communication system and the floating-gate memory cells.

The L1 communication system

Chapter 2 discusses the L1 bus system which is used to transmit presynaptic neural events between different neurons in the new Stage 2 hardware system. After a general discussion of the L1 communication system including several technological aspects of it, the focus is shifted to a prototype chip developed during this thesis. The chip contains some components of the L1 system and is used for tests of basic features of the L1 bus. The results are presented in section 2.4.

The Floating Gate Memory Cells

Chapter 3 discusses the floating gate memory cells used to store analog parameters of hardware neuron models. Again a prototype chip, designed by Sebastian Millner, is tested. The focus is on the controller code and addressing circuitry of the memory array which is used in the Stage 2 hardware. Also few measurements concerning analog properties have been performed. The results are presented in section 3.4.

1 Neuromorphic Hardware

The available computing power increased dramatically within the last years, but we are still far from simulations in biological real-time for networks with neuron numbers even close to the number of neuron in the brains of mammals. Even subsections, like e.g. V1, the first processing layer of the human visual system, are far beyond the scale accessible by simulators. The restrictions for simulators using a limited number of processing units derive from the conceptional discrepancy to the massive parallelism of the neural processes in the brain [23].

Therefore another solution is needed to perform experiments with large networks in reasonable time. Developing analog neuromorphic hardware seems to be a promising attempt to make progress. This means to implement analog circuits e.g. in VLSI[1] CMOS[2] technology which physically mimic the behavior of neurons and synapses instead of numerically solving differential equations which describe the cells. The observables typically investigated in biological neurons are mapped to equivalent ones in electronic circuits. The artificial neurons all operate in parallel and in continuous time. Therefore neuromorphic systems are able to overcome the limitations of established software based simulators, especially concerning scalability and operating speed. Some implementations of artificial neurons are able to run orders of magnitude faster than their biological antetypes. Furthermore, the power consumption of a chip emulating a neural network is usually much lower than the power consumption for a computer system simulating the same network. The possibility of acceleration, compared to biology, recommends neuromorphic hardware especially for experiments where large parameter spaces have to be swept or for the investigation of long-term learning processes. Drawbacks are a lack of flexibility concerning the implemented neuron model, restrictions for the available ranges of parameters as well as limited bandwidths for monitoring the activity within the network. The idea to develop neuromorphic circuits was first proposed more than 30 years ago by Mead and Mahowald [22, 21]. Since then several groups have developed a variety of different neuromorphic hardware systems. An overview of current projects is given in [28].

Nevertheless, up to now neuromorphic hardware has not become an established tool in neuroscience. This may change in the next years. On the one hand neuroscientists depend more and more on experiments with large networks in order to gather deeper insight into the brain's structures and functions. On the other hand, remarkable progress has been made in the development of neuromorphic hardware. Sophisticated, configurable neuron models are implemented [31]. Various network architectures can be realized due to flexible routing capabilities. Both aspects are important to allow for experiments with biological relevance.

The general progress made in microelectronics, leading to smaller structures and higher operating frequencies, also helps to improve the potential of neuromorphic hardware. This allows for a larger number of neurons on a chip and higher bandwidth for communication interfaces. Finally neuromorphic hardware might not be just a fast and power-saving replacement or addition for established simulators. Due to its time-continuous operation it provides

[1] Very Large Scale Integration

[2] Complementary Metal-Oxide-Semiconductor

the possibility of interactive experimental setups. Several parameters can be tuned during an experiment and the consequences for the network become visible immediately.

Besides its relevance for neuroscientific research, neuromorphic hardware is also an interesting approach for several technical applications. For instance sensor systems and motor controllers based on such systems have been developed [25, 18]. The parallel and fault-tolerant operation of neural networks is also assumed to be an inspiration for architectures required in future computing technologies beyond CMOS, which may be based on large numbers of individually unreliable devices, [1].

1.1 The Electronic Vision(s) Group as a Part of the FACETS Project

The Electronic Vision(s) Group at the Kirchhoff-Institute for Physics is one of 15 members of the FACETS project, which is funded by the European Union as a part of the FET[3] framework. The aim of FACETS is to investigate the fundamental computing principles of biological nervous systems and also to evaluate the possibilities of using these paradigms for future computation technology. This involves biologists, performing measurements in living cells, as well as modelers, trying to derive mathematical principles from the biological data and run simulations. The third topic on which FACETS participants work is the development of neuromorphic hardware. Two different approaches are pursued here.

A group at ENSEIRB[4] builds neurons that precisely implement a Hodgkin-Huxley Model[12] with a high precision, working in biological real time. This aims at enabling their chips to directly interact with real biological systems. To realize the time constants of real neurons in chips large capacitances and very precise small conductances are necessary. This leads to large circuits, therefore currently not more than five neurons fit into a single chip. Information on this system can be found e.g. in [5].

On the other hand a collaboration of the Eletronic Vision(s) group and a group from the TU Dresden[5] is building large scale networks with neurons working up to 10^5 times faster than biological neurons. Two different stages are under development, described in the following sections. If the progress made within FACETS are taken into account, it seem reasonable that neuromorphic hardware has the potential to become a valuable tool in neuroscience in the near future. More information on the FACETS project and the participating groups can be found in [8].

1.2 FACETS Stage 1 Hardware

The first neuromorphic system developed by the Vision(s) Group for FACETS is called the Stage 1 hardware. It is based on an ASIC[6] called ”Spikey”, which contains 384 neurons, implementing a leaky integrate-and-fire model, see [4], on a $5 \times 5mm^2$ die. Typically it runs with a speed-up factor of 10^5 compared to biological real time. The behavior of the neurons can be adjusted by means of many parameters, for instance their fireing threshold or leakage conductance, but these parameters are global for all neurons or at least common

[3] Future Emergent Technology Initiative

[4] Ecole Nationale Supérieure d’Electronique, ’Informatique et Radiocommunications de Bordeaux, France

[5] Hochparallele VLSI-Systeme und Neuromikroelektronik, Technische Universität Dresden, Germany

[6] Application Specific Integrated Circuit

for blocks of one fourth of the total amount of neurons. For inter-neuron communication 50k programmable conductance based synapses with 4 bit weight resolution are available. To build larger networks it is possible to connect up to 16 boards, each carrying one Spikey chip, with a common backplane system, see [33, 27]. Currently the fourth version of the Spikey chip is under development. A detailed description of the Stage 1 hardware can be found in [10], the possibilities of operating it as well as results of experiments performed on Spikey chips are shown in [2, 24, 15] and [4].

1.3 FACETS Stage 2 Hardware

The numbers of neurons involved in every biological system of interest, within the scope of FACETS, exceed the numbers of neurons available on currently existing neuromorphic hardware by several orders of magnitude. Reducing this gap between biological and electronically realized neuron count was the aim when initializing the development of the Stage 2 system. The main component of this hardware system is a new chip called "HICANN" [7], developed by the Electronic Vision(s) group in collaboration with the TU Dresden. A new approach for building large networks by utilizing of multiple, interconnected HICANN chips is called *wafer-scale integration.*

1.3.1 The HICANN Chip - Building Block of the Stage 2 Hardware

The HICANN chip is the main building block of the Stage 2 system. It contains 512 analog neurons as well as a communication system enabling to directly transfer neural events to other HICANN chips. In total there are about 115k synapses available. If the number of neurons is decreased, it is possible to build neurons with up to 15k synapses each. Compared to the Stage 1 system the neuron model was improved to a so called "adaptive exponential integrate-and-fire" model with conductance based synapses [3]. The number of adjustable neuron parameters is increased, due to the more sophisticated model, and most of them are now individually programmable for every neuron. This large number of analog parameters is enabled by the use of floating gate memory cells, capable of storing non volatile voltages in range of 0 to 1.8V with 8 bit resolution and likewise currents in range of 0 to $2.5 \mu A$. These cells are further discussed in chapter 3. Another important advancement is the possibility to directly transmit neuronal events ("spikes") between different HICANN chips with the Layer 1[8] bus. The L1 bus is in focus of this work and will be discussed in chapter 2. The size of a single HICANN chip is $5 \times 10 mm^2$, this allows to use MPW[9] runs for prototyping. At the moment the first prototype of a HICANN chip is being produced, initial tests will be done by the end of summer 2009.

1.3.2 Wafer-scale Integration

Instead of connecting boards containing single chips to obtain larger networks as done in case of the Stage 1 system, for Stage 2 another way is chosen. As mentioned before every HICANN got the possibility to directly interchange neural events with other HICANNs via the L1 bus. This capability is an essential prediction for wafer-scale integration. Multiple

[7] High Input Count Analog Neural Network

[8] The name results from historic reasons

[9] Multi Project Wafer

identical HICANNs are produced on the same die, connected by their L1 buses, building a larger net. Whole wafers, 20cm of diameter, containing about 350 HICANN chips, can be produced. Such a wafer will be capable of emulating up to 180k Neurons and more than 40M synapses.

The wafer-scale integration involves raises various technical challenges. In chip production usually the same pattern is implemented repeatedly over the wafer in a two dimensional stepping process. In case of the process used for Stage 2, 48 so called "reticles", each with a size of $20 \times 20mm^2$ and implementing exactly the same circuitry, are placed on the wafer. It is not possible to route any wires across the border of a reticle. Eight HICANN chips fit into a reticle and their L1 links are directly connected. To enable integration on the whole wafer it is necessary to create L1 links crossing the borders of the reticles. This is done in a post-processing step at the Fraunhofer Institute for Reliability and Microintegration, appending an additional metal layer which is not limited to any substructure but spreads across the whole wafer. This layer is used for connecting the L1 links of adjacent HICANN chips located in different reticles.

Another problem is the connection of the wafer to power supply and external communication links. The post processing layer provides pads, which are used to connect the wafer with help of elastomeric connectors[10] to a PCB[11]. The assembly of the whole system can be seen in Figure 1.1. The elastomeric connectors have already been successfully tested in a setup consisting of a PCB and a wafer with post-processing structures, but no active electronics. For more information concerning the assembly tests see [36].

Power consumption is a critical issue for this device. While for single chips a large heat spreader can be attached, in case of wafer-scale integration every square centimeter of silicon is not allowed to produce much more heat than can be cooled with one square centimeter of heat spreader. Actually the system is designed for a maximum power consumption of 1kW for the wafer. The expected average power consumption is about 500W, leading to a power density of about $1.6W/cm^2$, which is feasible with standard air cooling. It must be taken into account that 1kW at 1.8V leads to a total current of more than 500 amperes which need to be transfered to the wafer through the elastomeric connectors. Communication interfaces and control circuits located at daughter boards attached to the mainboard are expected to consume additionally about 1kW of power.

The transmission of configuration data and neural events between the HICANN chip and the host computer system of the Stage 2 hardware is utilized in two steps. On every HICANN a block called "DNC interface" is implemented, capable of transmitting data to the DNC[12] chips. These chips are located on the communication daughter boards mentioned above. The DNC chips are connected to an FPGA which establishes an ethernet connection to transmit the data to the host computer. The DNC chips as well as the underlying protocol, named Layer 2, have been developed by the TU Dresden. Via the host computer system it will be possible to further increase the system's capabilities by using several Stage 2 wafers combined.

For more information concerning the Stage 2 system see [32].

[10] "Zebra elastomeric connectors", www.fujipoly.com

[11] Printed Circuit Board

[12] Digital Network Interface

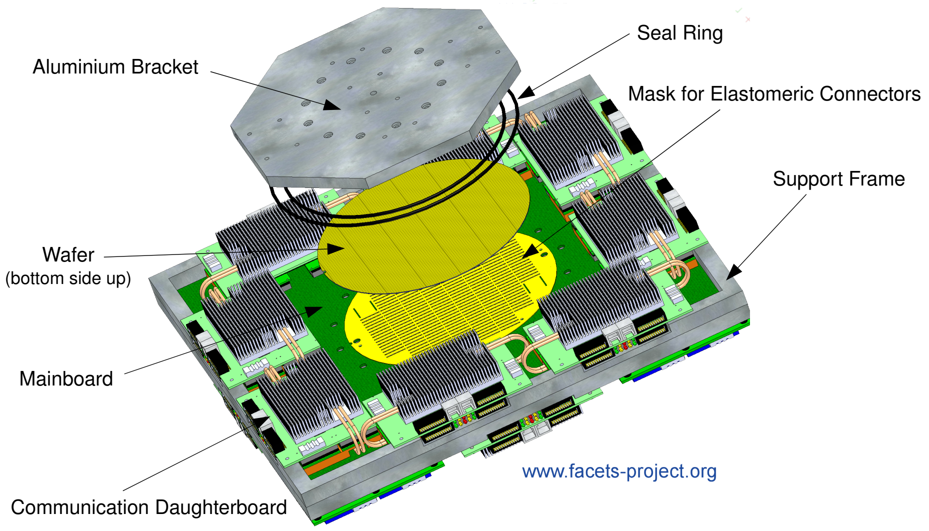


Figure 1.1: The Stage 2 system assembly. The wafer, utilizing about 350 HICANN chips, is connected to the mainboard through elastomeric connectors. The mainboard provides only routing capabilities. Active electronics for communication interfaces and controlling of the power supply are located at daughter boards. Figure by D. Husmann de Oliviera

2 The L1 bus prototype chip

The neural events in the Stage 2 system are transmitted between neurons by the L1 bus. It allows for time-continuous neuron-to-neuron communication across the wafer and was developed and implemented by Dr. Johannes Schemmel. During this diploma thesis a prototype chip containing components of the L1 communication system was designed and tested. The simulations concerning the L1 system mentioned in this chapter have been carried out by Dr. Johannes Schemmel. Most of the information concerning the repeaters presented in the following was also provided by him in personal conversations.

Besides the general possibility of sending data between repeaters, some more issues need to be addressed. A very important aspect of the L1 system is for example its power consumption, as it accounts for a considerable amount of the total power consumption of the system. Low latency is also an urgent issue, as, in the worst case, a signal crosses a maximum of about 20 repeaters, and the delay of these adds up. Many L1 lanes close to one another are running in parallel across the HICANN chip, so the problem of crosstalk between adjacent L1 lanes arises.

The routing of the L1 connections is very closely related to how the neurons which are to be modeled are mapped to the hardware neurons. All hardware constraints, especially the L1 capacities, must be taken into account. In [9] the algorithm realizing the mapping and the L1 routing is described. Simulations show that the implementation of biologically relevant networks is possible with an efficient usage of the available hardware.

2.1 The L1 Communication System

The L1 bus can be characterized as an asynchronous serial low voltage transmission using differential signals. The protocol is packet-based. One packet contains 8 bits, start and stop bit enclosing 6 payload data bits. Since the static power consumption of the repeaters is minimal for a positive input, this is chosen to be the inactive signal. Start and stop bit are therefore 0.

The system needs to facilitate a very flexible and programmable network topology. Since packets may have to travel distances of up to 20cm across the wafer it is necessary to amplify the signals and restore the timing regularly. This is done by repeaters, the basic circuitry of the L1 system. The general arrangement of the L1 components on the HICANN chip is shown in Figure 2.1. The repeaters are located at the edges of the HICANN chip in such a way that the L1 signals transit a repeater at least every 15mm. A more detailed illustration of the arrangement of the repeaters at the edges of the HICANN chips is given in Figure 2.7.

The route of a neural event from a neuron's output to a synapse located at another neuron on the same wafer is shown in Figure 2.2. Every neuron has a configurable six bit number. The outputs of 64 neurons are connected to an asynchronous priority encoder, since they share one L1 lane. The priority encoder decides which neuron's spike is sent first to the L1 sender in case several neurons fire simultaneously. The neuron with the highest number gets

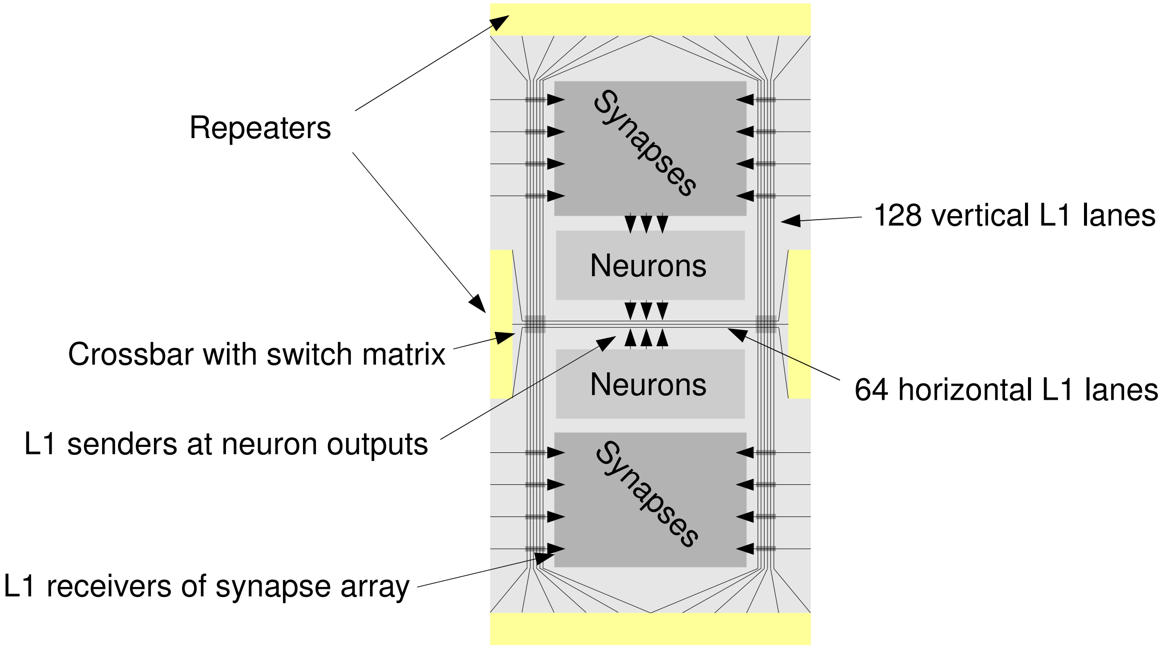

Repeaters
Synapses
128 vertical L1 lanes
Neurons
Crossbar with switch matrix
Neurons
64 horizontal L1 lanes
L1 senders at neuron outputs
Synapses
L1 receivers of synapse array

Figure 2.1: Block diagram of the HICANN chip, showing the arrangement of the L1 components.

the highest priority to send its spikes. Every neuron buffers one spike at its output. If it fires for a second time before the priority encoder has enabled it to send the previous one, the first spike is dropped. The parallel CMOS output of the priority encoder, representing the number of the neuron that fired, is serialized and transformed to a low voltage differential signal by an L1 sender. The L1 packet is routed crossing an arbitrary number of HICANN chips, repeatedly amplified by repeaters, until it reaches a HICANN chips on which a target neuron is located. Here the L1 packet is received and transformed to a low voltage parallel signal. Reducing the signal level to 0.9V helps to decrease the power consumption of this path to about 25% compared to a corresponding 1.8V system. Programmable address decoders connected to synapses are located along this parallel signal path. These synapses are finally transmitting the spike to the target neuron.

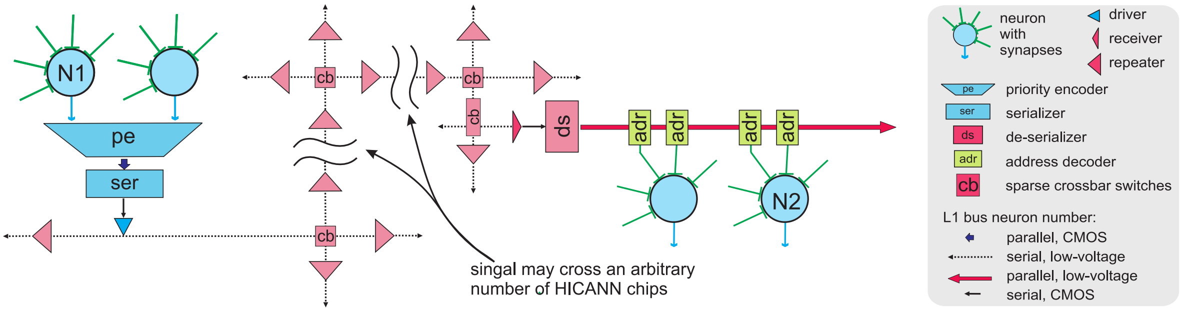


Figure 2.2: Schematic diagram of the route a neural event takes from the neuron labeled N1 to its target neuron N2. Taken from [32].

The repeaters are the main building block of the L1 system, senders and receivers use the same components implemented in the repeaters. A repeater receives the serial L1 data and uses some latches to convert it into parallel data. Afterwards the data is serialized again by a multiplexer and sent to the next section of L1 wire. The time-stamps for sampling in the receiver and sending are derived with help of a DLL[1].

In the following some general aspects of the L1 transmission are discussed, preparing the more detailed discussion of the repeater circuitry given in 2.1.4.

2.1.1 Low Voltage Differential Signals

The L1 bus utilizes low voltage differential signaling. This technology uses two separate wires carrying complementary signals with small amplitudes, typically below 200mV. For high-speed and low-power applications this is a better choice than the usual single-ended signals.

In case of single-ended transmission normal CMOS logic levels are typically used. This is necessary as the ground potential is often affected by noise or slight offsets. At high frequencies strong drivers are necessary to completely charge or discharge the capacitance of the buses wires to the logic levels in a sufficient period of time, which is much shorter than the period of a bit. This leads to a high power consumption, drivers requiring a lot of chip area and strong disturbances on the power supply net of the chips as well as strong crosstalk between the buses wires due to very high slopes.

In contrast small amplitudes allow for high frequencies and low power consumption. However, noise immunity is decreased. This is solved by using differential signals. The receiver functions as a differential amplifier, sensitive only to differences between both wires. As most external disturbances occur symmetrically on both wires, they are not visible for the receiver. Of course both wires have to be routed close together and with the same impedance to ensure that external disturbances have a symmetric impact. Also a common mode, typically in the range of half the supply voltage, is applied to the signals. Therefore slight shifts of ground or supply potential between sender and receiver do not affect the transmission. The drawbacks of this technology are the more complex driver and receiver circuitries as well as two wires being required for every signal. For an increasing number of applications the advantages justify the additional effort, due to increasing operating frequencies. Examples for the use of low voltage differential signals in current commercial computing technology are USB[2] or PCIe[3].

2.1.2 DLL

The L1 system does not have a global clock. The power consumption of a complete clock tree as well as the difficulties to synchronize a clock across an entire wafer were the reasons for use asynchronous transmission for the L1 bus. The time-stamps for sampling and sending of the single bits are generated locally in every repeater or receiver using a DLL which derives the timing information from received data packets. Only the senders obtain input from a clock which is generated in the digital part of the chip. These clocks are not synchronized for the different HICANN chips across the wafer.

[1] Delay Locked Loop

[2] Universal Serial Bus

[3] Peripheral Component Interconnect Express

In general a DLL is used to add a controlled phase shift to a digital signal. Between input and output of the DLL an adjustable delay element is placed. A phase detector controls the delay element to gather a fixed phase correlation between the input and the output. DLLs are mostly used to generate defined phase shifts between clock signals.

In the L1 system DLLs are used to generate time-stamps for communication. The delay consists of 24 equal adjustable delay elements forming a line, named 0 to 23, all controlled by the same voltage named V_{CTRL}. The phase detector adjusts V_{CTRL} in order to align the falling edge of the start bit delayed by 16 elements with the incoming rising edge of the stop bit. Thereby the time enclosed by this frame is divided into 16 equidistant bins. Every second time-stamp is in the middle of the expectation window for a single bit, providing a time-stamp to sample the bit. Also the timing for sending data is derived from the DLL. The controlling inputs of the multiplexer which generates the serial data are connected to the delay-line with a distance of two delay elements between them. As a result, the output of the multiplexer is switched to the next input every two time bins, which equals to one bit period of the incoming packet. This is illustrated in Figure 2.5, where a schematic overview of a L1 repeater, discussed in more detail in 2.1.4, is shown.

The DLL needs an initial training to set V_{CTRL} and lock on the signal correctly. The first step is to send only packets containing neuron number 0, to prevent the DLL from detecting other bit transitions than the one from the frame. It is also possible to give a starting value for V_{CTRL} to prevent the DLL from locking on multiples of the correct frequency. Once locked on the signal, a mask applied to the input signal is covering the arbitrary bit transitions between start and stop bit which occur during the regular network operation. Only an expectation window for the rising edge of the stop bit, plus/minus a half time bin width, is visible for phase detection to dynamically correct V_{CTRL}. This is on the one hand necessary because V_{CTRL} is stored on a capacitor. Due to leakage it is necessary to refresh it regularly. On the other hand slight drifting, for example caused by temperature variations, must be compensated. If the value of V_{CTRL} drifts too far from the correct value the rising edge of the stop bit does no longer occur within the expectation window and the DLL is not able to lock on the signal. This leads to an upper limit for a maximum distance between the L1 packets, considered to be larger than 1ms in the technical time domain, which equals to 10s in biological time (at a speedup of 10^4). If this maximum time is exceeded in regular network operation it is necessary to insert additional events. This is no restriction as in most experiments a certain Poisson background is used to stimulate the network, sufficient to keep the DLLs locked even with out further neural activity.

2.1.3 Crosstalk within the L1 System

As mentioned before, differential signals have the great benefit that every disturbance occurring on both wires is automatically canceled by the receiver. However, even small disturbances appearing on only one wire of a pair are likely to cause errors, due the low amplitudes. This is a decisive problem on the HICANN chip: On every side of the chip 128 pairs of L1 bus lanes are running next to one another with a minimum distance between them. Capacitive and magnetic coupling between adjacent wires of different pairs cause crosstalk. In case of VLSI capacitive effects are dominating because of the small distances and the high dielectric constant of silicon dioxide, see [6]. In case of the L1 system crosstalk is a source of disturbances which is much closer to one wire of a pair than to the other. The impact is not exactly the same on both wires and therefore not canceled in the differential receiver. This was identified as a serious problem in simulations using coupled transmission line models performed by Dr. Johannes Schemmel.

The simplest measure to suppress this effect is to twist the wires of the differential pair twice for every second bus lane. On the HICANN chip such crossings are located after 1/4 and 3/4 of the entire distance. Hence one wire of a bus lane is running the same length in parallel to the positive and to the negative wire of the adjacent pairs, which cancels most of the crosstalk. This is not completely sufficient because the amplitude, and therefore the crosstalk applied to its neighbors, decreases while the signal is propagating the wire. The direction in which the lanes are used may change from experiment to experiment, and therefore it is not possible to permanently adapt the position of the crossings to this effect. Another feature for crosstalk councilation is implemented to further improve signal quality. The asymmetric impact on the wires of a pair can be compensated by increasing it artificially on the wire that is less affected.

This is done by capacitors which connect the L1 wires at the receiver to the next but one wire of the adjacent pairs. An illustration of the crosstalk councilation using capacitors is given in Figure 2.3. In every repeater there are two capacitors of 52fF capacitance each which can be connected to the next but one wire on the right side and likewise two capacitors for the left side. They are individually configurable. It has to be tested whether enabling only a single capacitor or both capacitors simultaneously provides best crosstalk councilation.

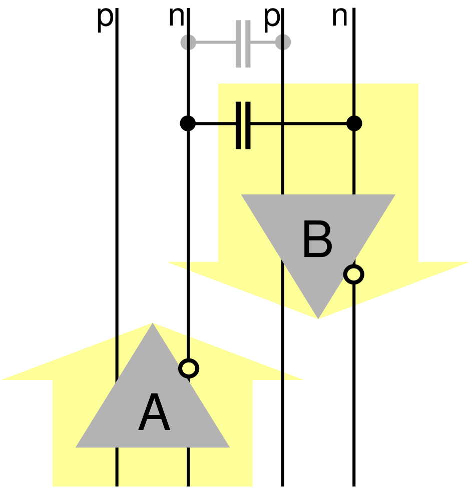


Figure 2.3: Illustration of crosstalk compensation using capacitors, exemplified for the receiver of repeater B. The capacitor in light gray symbolizes the capacitance between the negative wire of A and the positive wire of B. The dark capacitor is located in the receiver of B in order to balance the impact of the crosstalk caused by the negative wire of A on both wires ending at B.

2.1.4 The L1 Repeaters Circuitry

The repeaters represent the main circuit of the L1 communication system. They can be divided into several stages. The first stage seen by an incoming signal is the receiver, which consists of a differential amplifier to generate CMOS levels from the low voltage differential input signal. It is followed by six latches working as a deserializer. The next step is the parallel interface. This allows for accessing the latches in order to read and write parallel data. In the last step the parallel bits are serialized again by a multiplexer and converted to a differential signal in the driving stage. The timing information for both, the deserializing by triggering the latches at different times as well as the multiplexer is provided by the DLL. It is possible to configure the repeaters data flow direction according to the routing requirements. A simplified schematic of a repeater is shown in Figure 2.5.

Receiver

The differential amplifier is a critical component because it is consuming a significant static bias current in the range of $100\mu A$, independent of the L1 activity. Since there are 260k receivers this bias is an important aspect for the power consumption of the whole wafer. This static power dissipation is caused by the bias current for the differential pair which is the first stage of the receiver. The bias is generated in each receiver by one half of a distributed current mirror. This reproduces the current applied to the unique other half which is connected to an external pin to set the bias for all receivers. The CMOS level data stream from the differential amplifier is stored in six latches, enabled one after another by the timing signals of the DLL. The first latch is enabled by the third time-stamp from the DLL, the second by the fifth time-stamp and so on. Therefore this the signal is always sampled in the middle of the expected period of the incoming bit.

Parallel Interface

The latches mentioned above are connected to a parallel interface, which permits directly read or write parallel data from or into the latches. The ports for the parallel data are named TDI (Test Data Input) and TDO (Test Data Output). It is also possible to enable an input for an external clock named TCLKI. In this case the DLL is locking on the edges enclosing the high period of a test clock applied to TCLKI, instead of the frame of received packets. To simulate packets with a 2.0GBit/s data rate e.g. the clock needs a frequency of 125MHz. There is also a test clock output, named TCLKO, providing a high signal for the duration of a received packet.

Serializing Multiplexer

For sending, the data stored in parallel in the latches must be serialized. This is achieved with help of an 8-to-1 multiplexer. The inputs 0 and 7 are connected to ground to generate start and stop bit. The latches are connected to the inputs 1 to 6. The control inputs of the multiplexer are enabled by signals from the DLL. The start bit is connected to the output when the first edge of the incoming packet has passed the fifth delay element. The first data bit is then connected to the output when the seventh delay element is passed. Thus, the repeater starts sending after receiving the second data bit of the incoming packet. This adds an amount of three bit periods to the total delay caused by the repeater.

Driver

The driver generates differential signals from the CMOS level data stream generated by the serializer. The large RC time constant of the long L1 wires requires a strong preemphasis.

This is realized by connecting the output directly to VDD or ground to generate a sufficient slope at the edges of each bit. Only if the bit stream is constant for more than a single bit the output is connected to V_{OL}[4] or V_{OH}[5] instead. V_{OL} and V_{OH} are the limits of the nominal differential voltage used for the L1 system. Typically the amplitude is around 200mV and the common mode approximately 700mV. At the driver, the signal's peak-to-peak voltage is close to 1.8V caused by the preemphasis. Nevertheless, due to the capacitance of the wire only a low voltage signal arrives at the receiver. Prior to every change in the bit stream the differential lines are shorted to equalize their potential. This helps to reduce power consumption, as charging with the opposite polarity is required.

In Figure 2.4 L1 signals, measured directly at the driver and after 1cm of on-chip wire, are compared. The preemphasis is clearly visible at the sender.

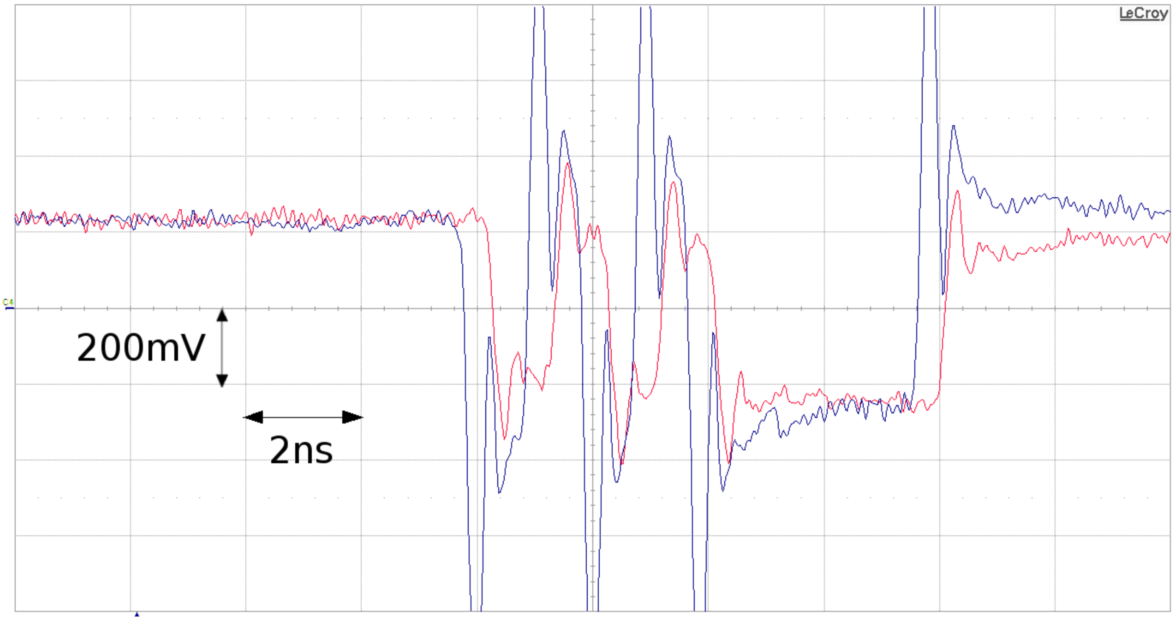


Figure 2.4: Comparison of the L1 signals directly at the sender (blue) and after propagating 1cm of on-chip wire (red). The preemphasis is clearly visible at the sender. The signals have been measured at the outputs of r0 and r2, which are sending the same bit pattern simultaneously, `101000` enclosed by the start and the stop bit. The phase shift is caused by the propagation delay of the signal on the 1cm wire connected to r2.

The possibility to send data in both directions makes two drivers in every repeater necessary. Depending on the configured data flow direction only one of them is enabled.

Configuration Memory

Each repeater has a memory to save its configuration, consisting of 8 SRAM cells. Two bits are used to enable or disable the repeaters and to determine the repeaters direction. Two additional bits are used to enable or disable the parallel data in and out ports, including the

[4] Voltage Out Low
[5] Voltage Out High

test clocks. The four remaining bits are used to enable the crosstalk compensation capacitors for each repeater individually.

Separate Drivers and Receivers

As shown in Figure 2.2 there are not only repeaters within the L1 system, but also separate drivers and receivers.

The drivers are simply repeaters with a disabled receiver, but both drivers are activated simultaneously, sending data in either direction at the same time. The data about to be sent is connected to the parallel test data input. This setup was chosen because it requires minimum additional design effort. Locking the DLL, which is necessary to generate the time-stamps for the serializing multiplexer, is carried out with help of the test clock input of the parallel interface. The clock applied to the drivers is generated in the digital part of the HICANN chip and determines the data rate with which the entire L1 system operates, as all following repeaters and receivers lock their DLLs on the signals sent by the initial drivers.
The separate receivers are also basically repeaters. In this case the drivers are not only disabled but also omitted in the layout. The neuron number received is assigned to additional buffers via the parallel test data output. The buffers convert the signals into low-voltage parallel signals.

The L1 system is designed to operate with a data rate of up to 2.0GBit/s within a packet. The netto data rate is lower, since start and stop bit must be taken into account and since the absolute data rate also depends on the distance between the packets. For a repeater the minimum interval between two packets is three bit periods due to the DLL timing scheme, shown in Figure 2.5. The repeater starts sending after the second data bit of the incoming packet has been received. In the final system the minimum pause on the L1 bus will be even longer. A interval of one packet length is caused by the priority encoder. In the following all data rates mentioned represent the data rate within a packet, not the absolute data rate.

2.2 Developing the L1 Prototype Chip

The task was to design a prototype chip that allows for testing a maximum number of aspects of the L1 bus system, which still fits onto a "Miniasic" of an IMEC[6] MPW[7] run. These chips have a size of $1.5 \times 1.5 mm^2$ and were produced in the same 180nm single-poly-6-metal process as the Stage 2 hardware.

First the most simplest possible setup, two repeaters connected to each other, was implemented. These two repeaters are named r0 and r1 in the following. They are located next to one another in the prototype chip, but the wire connecting them is 1cm long. The metal densities surrounding these wires are mimicked to match the conditions present on the HICANN chip. As mentioned before crosstalk seems to be an important issue within the L1 bus system. Therefore two additional repeaters were implemented, named r2 and r3. These repeaters are driving wires which run in parallel to the connection between r1 and r0 to mimic the adjacent L1 lanes. The connection between r0 and r1 will be abbreviated by $r0 \rightleftharpoons r1$ in the following. During preliminary tests of possible layouts it turned out rather the number of available bond

[6] Interuniversity Microelectronics Centre, Belgium
[7] Multi Project Wafer

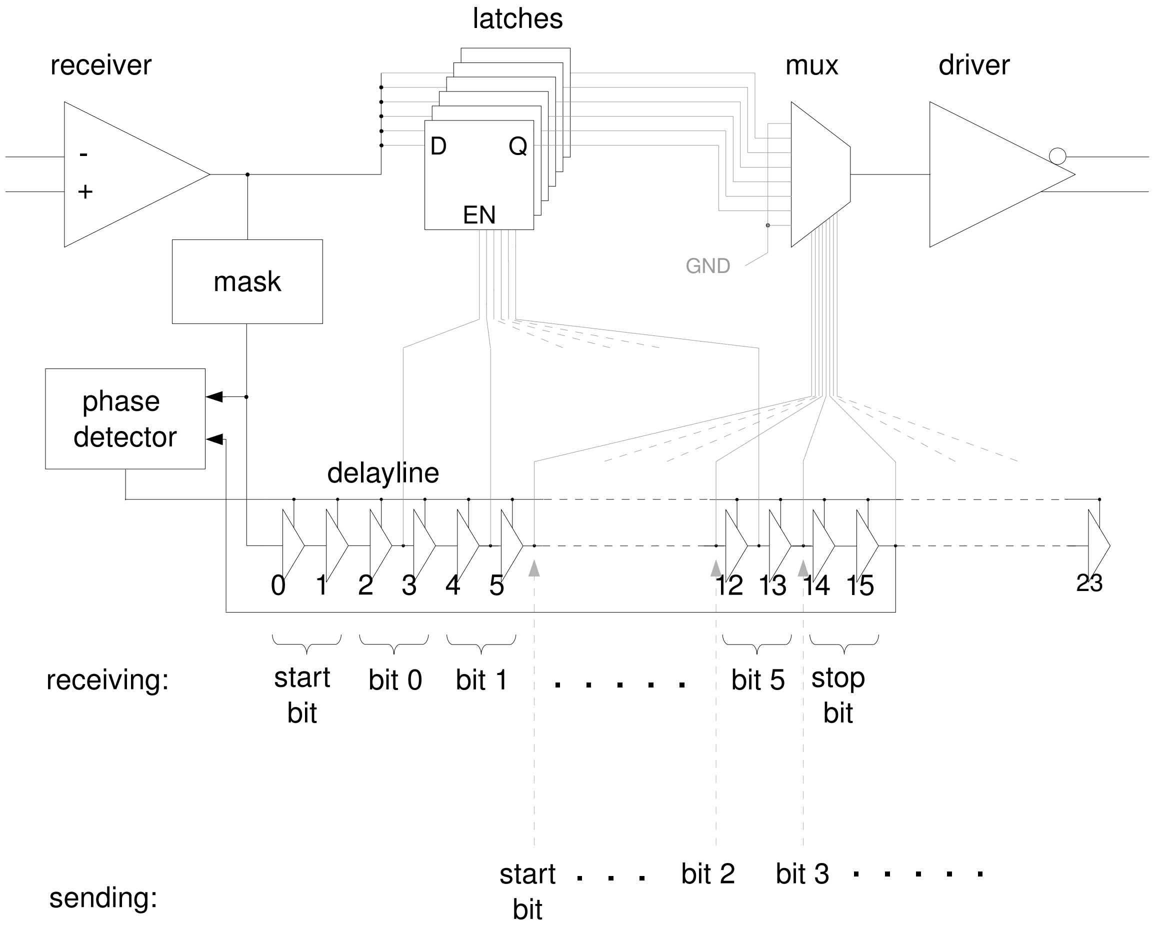


Figure 2.5: Simplified block diagram of a repeater. The serial data received by the differential amplifier is turned into parallel data by dynamic data capture latches. The time-stamps enabling the single latches one after another are derived with the help of a 24-tap DLL. The adjustable delay-line is controlled by a phase-detector, which aims at aligning the rising edge of the stop bit with the falling edge of the start bit delayed by 16 elements. Thereby the frame defined by the start and the stop bit is divided into 16 equidistant time bins. A mask covers the arbitrary bit transitions of the payload data. For sending, the parallel data is serialized again by a multiplexer. The timing for enabling the multiplexer's inputs is again derived from the delay-line. The parallel interface permitting access to the parallel data is omitted. Based on a schematic from [32]

pads that can be placed at its outline than the chip area available will be the limiting factor for the functionality of the chip.

2.2.1 Mimicking the Metal Densities of the HICANN Chip

In order to test the L1 system, a setup of the transmission lines as similar as possible to the one in the HICANN chip is necessary to model the parasitic capacitances of the wires correctly. The vertical L1 bus wires are mainly located on metal 6, as this layer is significantly thicker compared to the others. Therefore it provides a resistance which is decreased to one third of the resistance provided by the other layers. The wires are of minimum width, which is $1.2\mu m$ for metal 6. There is distance of $1.4\mu m$ between them. The horizontal L1 wires are also mostly located in the metal 6 layer, only in regions where the horizontal lines cross the vertical ones, they are routed on metal 5. Due to the higher resistance of metal 5 the width of the wires is increased to $4.2\mu m$.

Despite the few horizontal lines there is hardly any metal 5 below the L1 buses to reduce the capacitance of the wires. On the HICANN chip a very dense net of power supply wires on metal 4 is intended to shield the bus lanes from noise caused by the digital circuitry below the bus lanes. The prototype chip has a dense net of metal 4 wires. It has no other function aside from modeling parasitic capacitances. There are strict design rules for the total density of metal in every layer. Furthermore it is recommended to distribute this metal as evenly as possible. To order to provide an adequate and evenly distributed metal density so-called "filler cells" are added to every design by the producer. To ensure the metal densities affecting the L1 system are not modified it is possible and necessary to add a special layer to block the filler cells.

The prototype was intended to be compliant with the design rules referring to the entire chip. However, in the area below the differential wires the rules are violated and the resulting risk was taken into account. In the latest version of the HICANN chip the metal densities are even more exceptional metal densities than in the prototype. This might be a risk, especially for the prototype chips produced in MPW runs. When whole wafers are produced the metal densities are no longer as critical as in MPW runs, because process parameters can be controlled better and adapted within a certain range to these special requirements.

2.2.2 Number of available IOs

As mentioned above, the number of IOs available on the chip limits its functionality. First the number of IOs realizable in the layout was maximized. Afterwards as much functionality as possible with the given number of IOs was implemented. Some of the pins are shared by two repeaters. Therefore a correct configuration becomes important to prevent two drivers connected to the same line from being active simultaneously.

Bond pads of minimum size require a pitch of about $100\mu m$. In the IO cells metal is integrated in such a way that it connects to form ring around the chip, distributing the supply voltage to all the IO cells. At the corners of the chip a filler is needed, which bends the power wires by 90 degrees. Normally a triangular shape is used to do this. However, this blocks a considerable length at the chip's edge, about $500\mu m$ per corner, reducing the number of available IOs. Taking this into consideration, only about 40 pads can be placed at the chip's circumference.

Since the Faraday IO cells themselves are only $62\mu m$ wide, it is possible to place them directly

next to each other and stagger the connected bond pads, obtaining an effective pitch of $62\mu m$. To save additional space the corners were designed manually in an unusual way, integrating the ground IO cells into the power-ring. As a result 62 bond pads could be placed on the chip, as shown in 2.9.
These staggered pads had not been used in the institute before. so there was no experience whether bonding will work with the equipment available. It was known before that these pads are close to the limits of normal ultrasonic bonding with aluminum wires. The chip is placed directly onto a PCB with a resolution of $100\mu m$. This leads to a minimum pitch of $200\mu m$ for the bond pads on the PCB. Two sides of the chip feature 21 bond pads on a length of $1.4mm$. The corresponding pads on the PCB are distributed over a length of $4.2mm$ due to the limited resolution of the PCB. Given a maximum distance between the pads on the chip and the PCB of about 3mm for reasonable length of the bond wires this leads to angles of about 30 degrees at the corners, relative to an orthographic line. These angles were difficult to realize for pads at the inner position of the staggering without touching a wire of the outer pads. Bonding was made accomplished by Ralf Achenbach after several hours of experimenting. An automatic bonder was used in manual mode, running the wires in S-shapes. This permits angles at the chip's corners of less than the 30 degrees required for a straight connection between the pads on the chip and the PCB. Never the less the gap between the bond wires, which have a diameter of $25\mu m$, is in a range of only $10\mu m$ at the corners of the chip, see Figure 2.10.

In order to reduce the numbers of IOs required the repeaters share several pins. The parallel data inputs as well as the outputs of r0 and r1 are directly connected. It is important to ensure that the TDO of both repeaters are never enabled simultaneously, were as it is not useful to configure both to use the same input. On the HICANN chip all repeaters use the same 8 bit wide bus for configuration data. The address is 7 bit wide. In case of only four repeaters, as in the prototype chip, 2 bits are enough to address them. The remaining bits where statically connected to ground. For r2 and r3 their parallel TDI is used as data input. On the chip they are combined to the 12 bit wide port named TDI23. Since data for sending is applied only in operation, the IOs are also used during configuration. In case a signal named SELECT is high some transmission gates are enabled, additionally connecting the TDI23 to the configuration data lines. After configuration SELECT is released and the configuration data lines are pulled to ground. The analog parameters of the repeaters, V_{CCAS}, I_{BIAS}, V_{OL} and V_{OH} are global for all repeaters on the chip, as they will be global on the HICANN chip.

2.2.3 Protection against Electrostatic Discharge

An important issue in integrated microelectronics is the protection against electrostatically generated charges, discharging into a chip. This is referred to as ElectroStatic Discharge, abbreviated by ESD. Since both chips, the L1 prototype as well as the floating-gate chip suffered from different problems related to their ESD protection, a brief survey of the basic concepts of ESD protection is given in the following. For an extensive discussion of state-of-the-art ESD technology, see [30].

According to Hossein [30] up to 70% of IC failure may be caused by ESD problems. Static charges easily reach voltages beyond several kilovolt. This can cause damage either by the high electrical fields or by the high impulse currents induced if connected to a pin of a microelectronic device. Therefore every IO of a chip must be protected against ESD events. The simplest way to provide basic protection is to connect the pin via diodes to ground and

supply voltage, each directed in reverse-bias direction as long as the voltage at the pin does not exceed the range between ground and supply voltage, see Figure 2.6. Every over-voltage at the pin in either direction is connected to supply or ground by a forward-biased diode. This is applicable only if the diodes are fast enough and capable of withstanding high impulse currents.
Another possibility to achieve ESD protection are GGNMOS[8] devices. In this case a transistor is used. Its drain is connected to the IO pad, source, gate and bulk are connected to ground. During normal operation this transistor is turned off. What happens in case of an ESD event can be explained by a parasitic bipolar transistor built by the n-doped drain and source with the p-doped substrate. When the voltage at the pin exceeds about 6V the bipolar transistor turns on, clamping the ESD event to ground. A detailed explanation of this process can be found in [30]. A complementary structure built with a PMOS protects against negative ESD events.

Designing sufficient ESD protection devices is challenging. In case of an ESD event they will be operating close to their physical limits. Therefore it is hardly possible to simulate this situation. Furthermore, chip area is valuable, so ESD protection structures should be as small as possible. Even if enough area is available, every ESD protection measure adds further parasitic capacitance. This decreases the performance of the IO, especially when it comes to high frequency applications. In professional ESD design, aside from simulations, real ESD structures are still tested and damages occurring are investigated under an electron microscope to reinforce the devices exactly where they failed.

If an ESD event is clamped to the supply voltage another problem arises. It must be ensured that the difference between supply and ground does not increase to a level capable of destroying the chip. This is done properly by using a large transistor with triggering electronics, which shorts the supply voltage to ground in case of over-voltage. There are different possibilities for triggering. Some are sensitive to the absolute voltage, others are sensitive to extreme slopes in the supply net. In case of prototype chips, used only in laboratories with ESD protected workstations, this is usually not necessary. Since only weak ESD events occur, the small amounts of charge induced into the chip normally do not lead to a critical increase of the supply voltage if the chip's power net provides a certain capacitance.

On the L1 prototype chip, professionally designed ESD cells for analog and digital 3.3V IOs from a library available from Faraday[9] have been used. These cells provide reliable protection for all 3.3V IOs, but what happens to the 1.8V analog IOs was not considerated appropriately. These cells are working with the GGNMOS and complementary PMOS structures mentioned above. Their trigger point is above 5V. In case of 1.8V transistors, the gate oxide typically breaks down at voltages well below the trigger point of the ESD structures which are designed for 3.3V devices. So ESD protection for these inputs does not work, they can be destroyed before ESD protection becomes active. Furthermore the capacitance of the power net of a minimum-size prototype ASIC containing little active circuitry is very low. Therefore even ESD events which are clamped to VDD can cause problems. This leads to insufficient ESD protection, which is the most plausible explanation for damaged input pins occurring during work with the L1 prototype chip, see 2.3.5.

[8] Grounded Gate NMOS

[9] Faraday Technology Corporation, www.faraday.tech.com

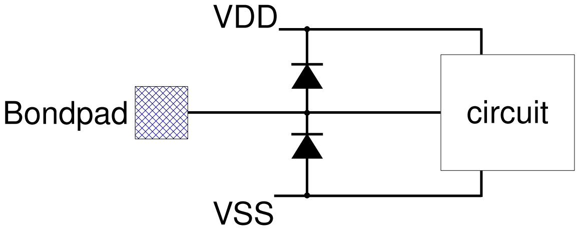


Figure 2.6: Basic ESD protection utilizing diodes. The diodes provide a discharge path for ESD events as soon as the voltage at the pin exceeds the range between power supply and ground

2.2.4 The Final Layout

The final chip supports four repeaters, two connected by a wire simulating a L1 lane that is vertically crossing a HICANN chip. Two additional repeaters are driving wires, running along this connection to cause crosstalk, as described in 2.2. These connections are an arrangement of three differential pairs, running in parallel with a distance of $1.4\mu m$ between adjacent wires. Figure 2.7 illustrates which repeaters of the repeater arrangement on the HICANN chip are mimicked by the repeaters on the prototype chip.

To fit onto the chip the wires are running in serpentines, covering most of the chip. The remaining differential IOs of the repeaters 0 and 1 are connected to bond pads. Also the crosstalk wires end in bond pads. This allows for measuring L1 signals after they have propagated 1cm on L1 wires with an oscilloscope. A general overview of which IOs are accessible via bond pads and probe pads for all four repeaters can be found in Figure 2.8.

In order to simulate active horizontal[10] L1 lanes at two points pass transistors are located at $r0 \rightleftharpoons r1$, one transistor connects it to 5mm, the second one to 10mm of additional wire. The 5mm wire is named h1, the 10mm wire is named h2.

Because of limited IO number it was not possible to place a receiver at their ends or connect them directly to bond pads. A bond pad would also apply a huge amount of capacitance, which is not a realistic emulation of the original HICANN setup. By enabling the pass transistors it is only possible to investigate the impact of the capacitance added by the extra wires on the performance of the transmission via $r0 \rightleftharpoons r1$. To have at least a chance of measuring the signal quality at the end of the horizontal lines, probe pads are applied. However, measuring the L1 signals with sufficient bandwidth on a wafer prober requires considerable effort.
The possibility to see the delay control voltage, named V_{CTRL}, of the DLL to check whether it has locked correctly is important for debugging. For r0 and r1, V_{CTRL} is buffered using operational amplifiers. V_{CTRL} of r0 and r1 are individually accessible via IO pads. Another IO pad was used for debugging. One bit of the configuration memory of r1 was connected to the outside, to see whether configuration works.
Since some area was left, Sebastian Millner suggested to test a metal finger capacitor that was placed in one corner of the chip. It is completely independent of the rest of the circuitry on the chip.

The layout of the final chip is presented in Figure 2.9. In Figure 2.10 a photograph of a

[10] This refers to the direction of their antetypes in the HICANN chip, not to their alignment in the prototype chip

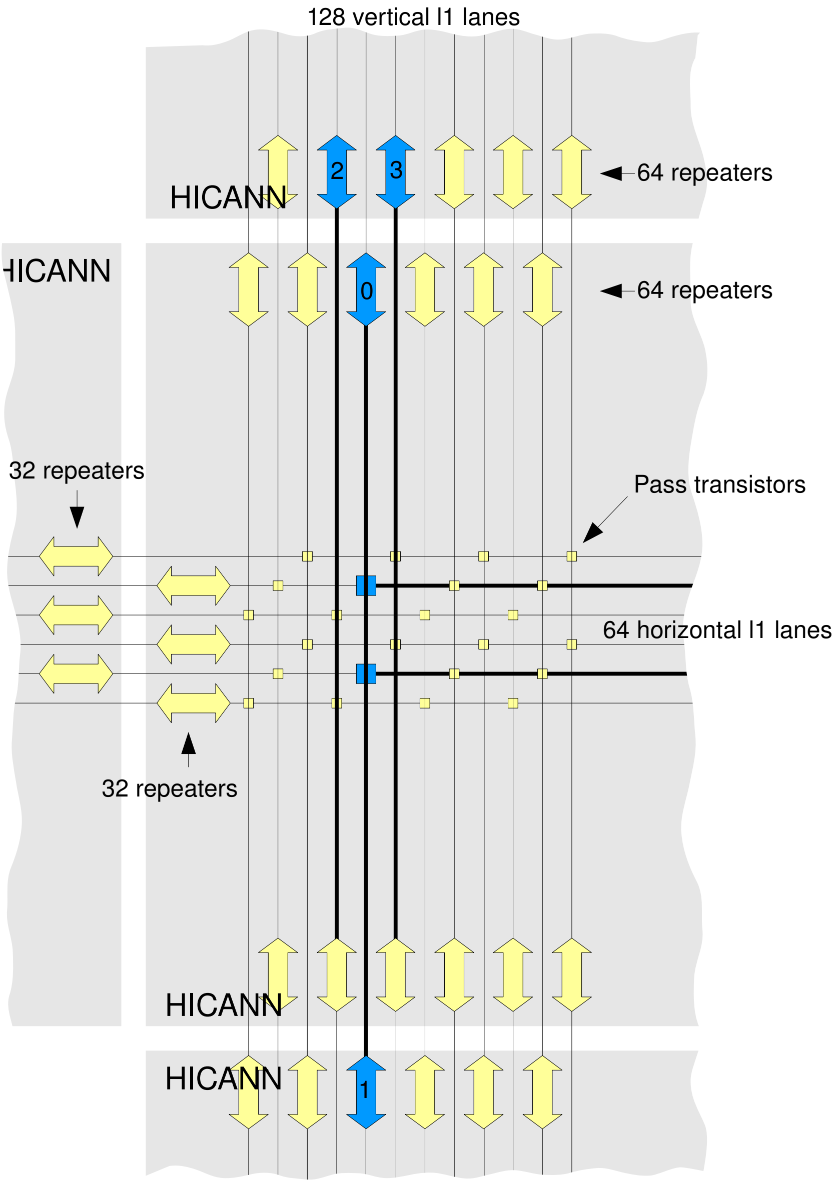


Figure 2.7: Block diagram of the repeater arrangement on the HICANN chip. The pair of both differential wires of one L1 lane is depicted by a single line. At the edge of a HICANN chip only the signals of every second L1 lane are amplified by a repeater. On the adjacent HICANN chip the position of the repeaters is shifted by one L1 lane, so the remaining signals are amplified. Therefore every L1 signal propagating on vertical L1 lines is amplified repeatedly, once on every HICANN chip. The repeaters of the horizontal bus lanes are arranged analogously. The parts of the L1 structure mimicked by the prototype chip are highlighted (blue). The numbers refer to the names assigned to the respective repeaters in the prototype chip.

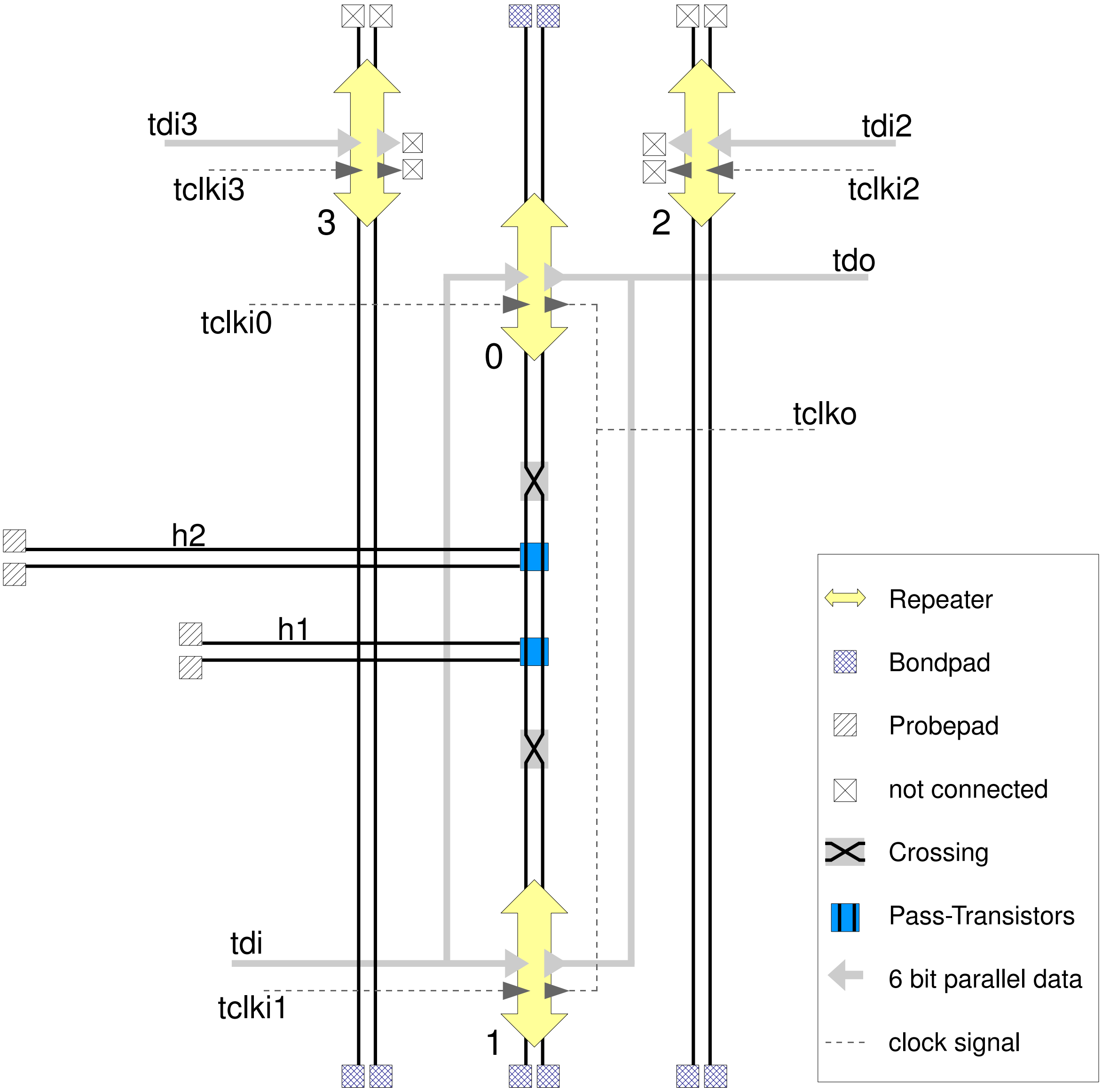


Figure 2.8: Overview of the components implemented in the L1 prototype. The connection between repeater 0 and repeater 1 is subject to most experiments. For r0 and r1 either the serial differential IOs or the parallel IOs, requiring a test clock, can be used. The repeaters 2 and 3 are intended to test the impact of crosstalk. For r2 and r3 only the parallel inputs are available, since they are only used as senders. Their differential outputs are connected to bond pads which permits measuring L1 signals after propagating 1cm of on-chip wire. For the digital signals, such as the test clocks or the parallel data IOs, the corresponding bond pads are omitted in the figure.

chip bonded to a PCB is shown. A section of the wires connecting the repeaters is shown in Figure 2.11.

To validate the function of the chip simulations were necessary. They have been carried out by Dr. Johannes Schemmel by implementing the L1 prototype chip in his testbench for the repeater circuitry. This lead to the important result that the transistors of some transmission gates were designed too small to program the following SRAM cells at first. In this case it would have been impossible to configure the repeaters and to use the chip in any way.

2.3 Measurements with the L1 Prototype Chip

For testing, a setup consisting of a PCB carrying the chip and connecting it to a multipurpose FPGA[11] development board was built. A high speed AWG[12] and a high speed oscilloscope was used to generate and view serial L1 data.

2.3.1 PCB

The PCB was designed to provide power supply and generate all adjustable voltages. A series of sma[13] connectors allows connection of the AWG to all differential inputs. The chip itself is attached in COB technique[14], this provides best possible signal quality for the serial differential output of the chip.

The PCB also provides two connectors which allows for attachment to an FPGA development board. The lines connecting the chip and the sma connectors have to be designed with correct impedance, 50Ω to ground, and matching termination resistors to provide proper signal quality. The sma connectors are only used to provide input from the AWG. Measuring the L1 packets on the differential IOs which are configured as outputs requires some effort, since the power of the signal is very small. To reduce the load it is possible to disconnect the wires to the sma connectors and the termination resistors using solder-jumpers located very close to the chip. Only small pads for connecting active differential probes remain connected.

2.3.2 FPGA

The experimental setup is based on an FPGA development board from Avnet[15], supporting a Xilinx virtex5 110T. It provides also a number of interfaces such as RS232, Ethernet, USB, PCIexpress and several more. The PCB carrying the chip is attached by two exp connectors[16] to the FPGA board.

The FPGA is used for configuring the repeaters as well as sending and receiving data using the parallel test data IOs. Furthermore it is capable of sending data packets within r2 and r3 to cause crosstalk on $r0 \rightleftharpoons r1$. If serial data inputs are used, the FPGA also controls the AWG. This is described in 2.3 in more detail. The FPGA was programmed in VHDL[17] with support from the Xilinx EDK software[18]. This tool provides examples and wizards that help

[11] Field Programmable Gate Array

[12] Arbitrary Waveform Generator

[13] Sub-Miniature-A, coaxial cable connector specified for frequencies up to 18GHz

[14] Chip On Board, the chip is directly bonded to the PCB

[15] Avnet Technology Solutions, www.avnet.com

[16] Expansion connector, a standard from Avnet to connect daughter boards to their development boards

[17] Very High Speed Integrated Circuit Hardware Description Language

[18] Xilinx Embedded Development Kit

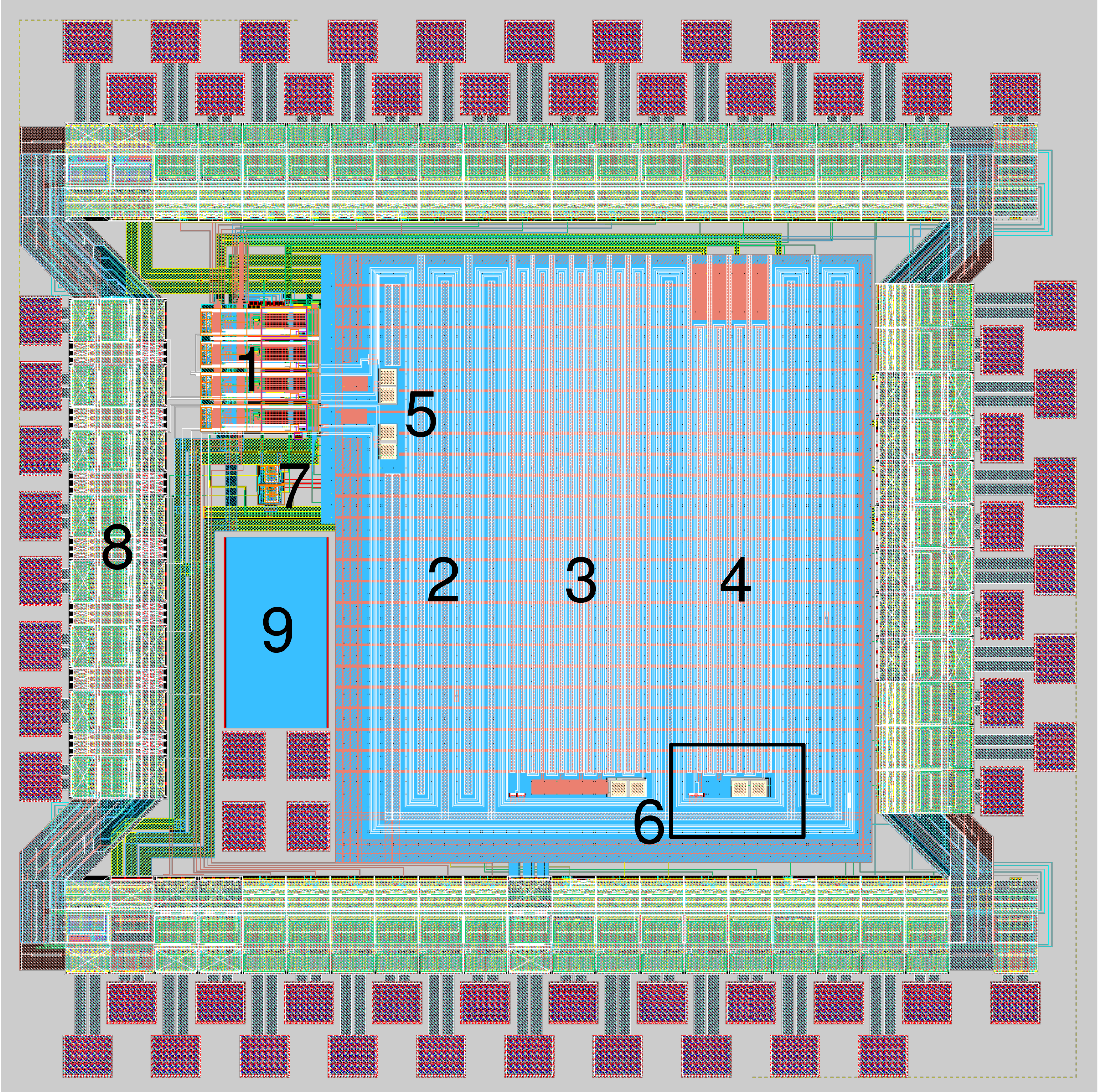


Figure 2.9: Layout of the L1 prototype chip. (1) The four repeaters; (2) Bundle of three parallel L1 lanes, $r0 \rightleftharpoons r1$ is in the middle, between the lanes connected to r2 and r3; (3) h1, the 10mm horizontal line (4) h2, the 5mm horizontal line; (5) probe pads at the beginning and at the end of $r0 \rightleftharpoons r1$; (6) Pass transistors to enable h1 and h2, probe pads at the ends of h1 and h2. A detailed view of the framed section is shown in Figure 2.11; (7) Operational amplifiers buffering V_{CTRL} of r0 and r1; (8) Differential IOs; (9) Metal finger capacitance;

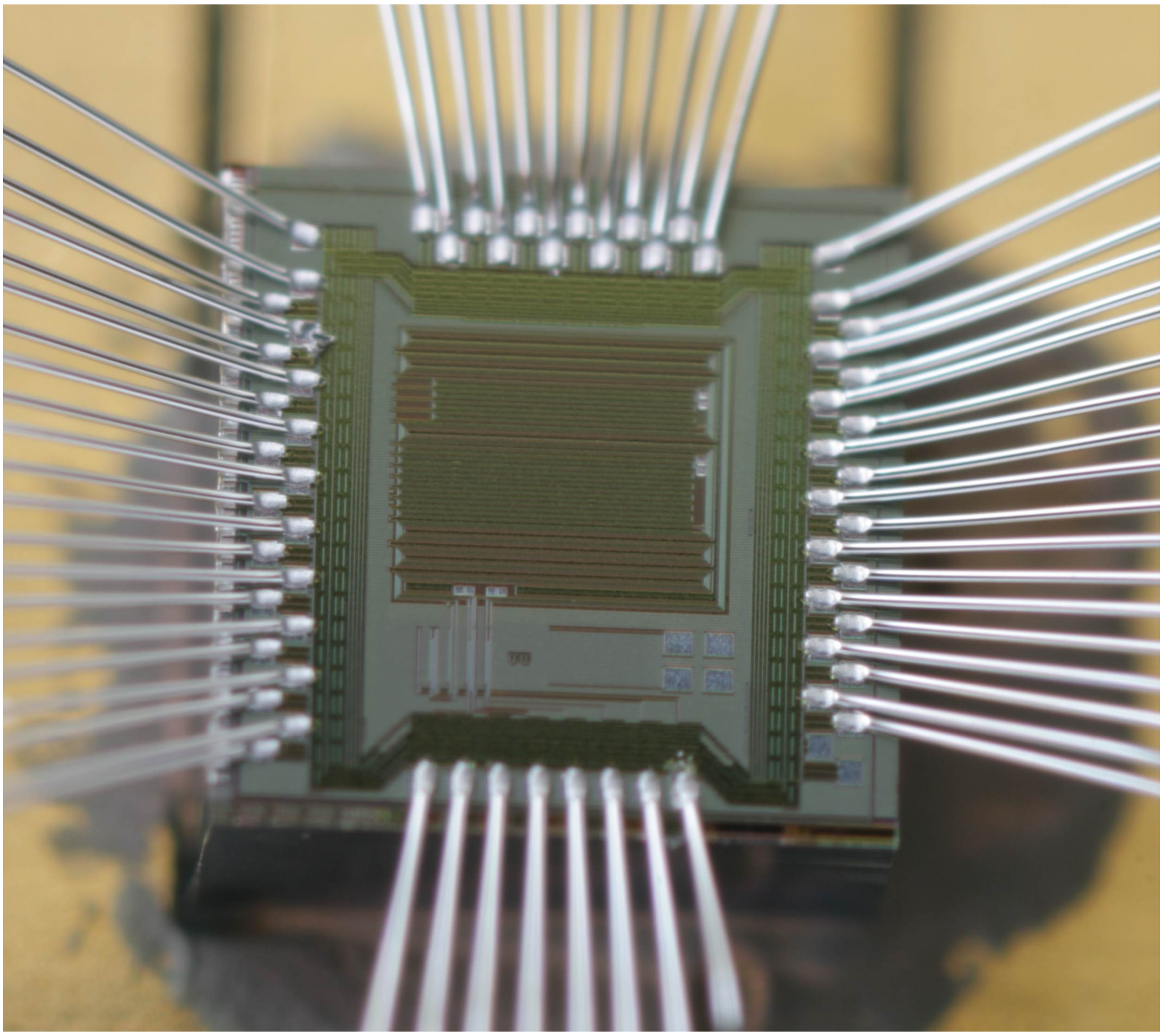


Figure 2.10: Photograph of a bonded L1 prototype chip. In this photograph the four repeaters are located at the lower left corner of the chip. The parallel structures covering most of the chip's surface are the L1 wires running in serpentines. Note the close proximity of the bondwires to one another.

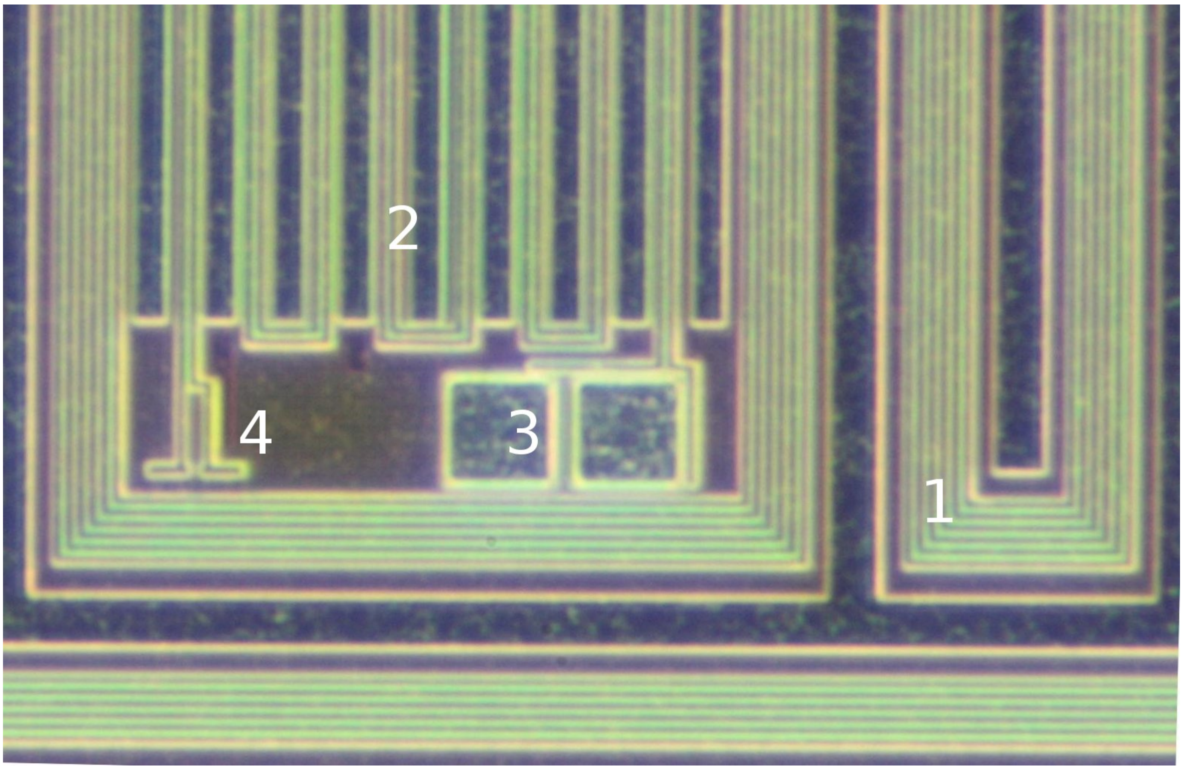


Figure 2.11: Detail of the L1 wires on the L1 prototype chip. (1) Bundle of the three parallel L1 lanes; (2) h1: the 5mm horizontal line; (3) Probe pads at the end of h1; (4) Locatio of the pass transistors connecting h1 to $r0 \rightleftharpoons r1$.

to implement a microblaze[19] softcore processor and use the IO interfaces available on the board. In the microblaze processor C programs can be executed, but not all C instructions are supported. The clock frequency for the whole design, including the microblaze processor, is 125MHz.
EDK also provides a template to connect modules written in VHDL, called "peripherals" in terms of the EDK software, to the microblaze processor via a simplified interface for the PLB[20]. A number of 32 bit registers defined in the vhdl code is used for communication. These registers are also accessible by the C code with help of the PLB. A memory address of the microblaze is mapped on every register. Reading data from one of this addresses means that they are read from the register and the data is send via the PLB to the processor. Writing is performed analogously. The data written to a dedicated address is transfered via the PLB and sets the target register to the respective value. This communication scheme is rather easy to use. The management of the bus is done in the background, almost invisible for the user, by the code automatically generated by the EDK software.

Aside from the 125MHz clock, also 187.5MHz and 250MHz are generated with help of DCMs[21]. These are necessary to generate signals for the test clock inputs of the chip to run it with data rates of 1.5 and 2.0GBit/s. Another useful feature of EDK is its support of simulations. It provides an automatically generated script for every design that can be executed directly in modelsim[22] and enables simulation of the whole system, including the C code running in the microblaze processor. Also, a template for a vhdl testbench is generated, so one can assign signals to the input ports of the design about to be simulated.

The setup finally consists of a vhdl module doing the low level controlling of the chip. Basic tasks performed by the vhdl within few clock cycles are for example reset, configuration, sending and receiving data via $r0 \rightleftharpoons r1$, sending crosstalk and activating crossbar transistors or crosstalk councilation. The sequence of these basic tasks is determined by a C program running in a microblaze processor which is connected to the vhdl module via the PLB.
To give an example, the sequence of a configuration proceeds as follows: The program running in the microblaze processor writes the data for the chips configuration memory as well as the address of the repeater to be configured via the PLB to the assigned registers. There is a register for instructions, a bit is assigned to every basic task. The vhdl code gets the instruction to configure by enabling the dedicated bit in the instruction vector. This triggers a sequence with a duration of three clock cycles. In the first one SELECT is enabled, the configuration data is applied to TDI23 and the repeater address is set. In the next clock cycle the EN signal is activated to write the data to the memory of the addressed repeater. Finally, an acknowledge bit in the instruction vector is set, signaling to the C program to be ready for the next instruction. The design fit into the FPGA without difficulties, since the virtex5 provides plenty resources. In the end its capacities were use to less than 15%. For programming the FPGA is connected to a PC using a USB to JTAG converter.

[19] A 32 bit processor with RISC architecture, designed to be implemented into Xilinx FPGAs
[20] Processor Local Bus, a bus used for processors with IBM powerPC architecture
[21] Digital Clock Manager
[22] Digital simulation environment from Mentor Graphics

2.3.3 Generating Serial Input

To send serial L1 data to the chip a high speed AWG[23] is available. It is capable of generating single waveforms with more than 1 million samples as well as sequences of shorter waveforms.

To use custom waveforms the AWG has the possibility to import text files containing analog or 8 bit values for every sample. Files with analog values for all possible L1 packets from 0 to 63 where generated with help of a python[24] script. A number of five samples per bit have been chosen, so the output data rate is 2.0GBit/s if the AWG is running at maximum speed, 10GS/s for dual channel operation. For experiments with data rates above 2.0GBit/s it is possible to use the "interleaved mode" combining both channels and reaching 20GS/s.

There are two different possibilities of sequencing waveforms, the "hardware sequencing" and the "software sequencing" mode. The hardware mode allows chaining of completely arbitrary waveforms, up to a total number of about 1 million samples. It works simply by combining the individual waveforms to a single large one and writing it into the playback memory at once. Therefore the possibilities of controlling the sequence during runtime by external signals are very limited. Much more controlling features are feasible in the software sequence mode, with the drawback that every waveform used in the sequence must have exactly 960 samples.

The software mode allows for loading blocks of 960 samples to the playback memory during operation, permitting manipulation of the sequence at runtime. Basically there are two different external signals to control the AWG. One the one hand by using a trigger input it is possible to wait for this signal before the next waveform is generated. On the other hand an "event" input allows for looping a waveform until an event occurs or for jumping to an arbitrary waveform in the sequence in case of an event.

The output of the AWG is not directly connected to the chip. A so-called "bias tee" is placed in between. It has two inputs, one for AC, one for DC signals. The AC input is coupled to the output by a capacitor. The DC input is connected to the output by an inductor. This permits to add the L1 DC offset of about 700mV to the AC signals of the AWG. Furthermore this device has a protective function. The outputs of the AWG are sensitive to any accidentally applied DC voltage, more than 1V may destroy it. The bias tee provides reliable protection against any errors on the PCB or mistakes of the experimenter which result in a DC voltage at the sma connectors of the board.

2.3.4 Final Measurement Setup

An overview of the complete measurement setup for the L1 prototype chip is given in Figure 2.12. As mentioned before the chip is configured and controlled with an FPGA development board which is also capable of sending and receiving L1 data with help of the parallel test data inputs/outputs. The code running in the microblaze processor is controlled via RS232 with a terminal on the PC. The FPGA is able of controlling the sequence in the AWG using its trigger and event inputs, as described in 2.3.3. A multimeter is used to view V_{CTRL} of r0 and r1, a logic analyzer helps to debug the software and directly watch received parallel data. The L1 signals sent by the chips can be measured with a high speed oscilloscope[25], using suitable active high bandwidth differential probes[26]. For measurements with best signal

[23] Tektronix AWG 7102, 2x10GS/s

[24] An interpreter based programming language with scripting support

[25] LeCroy SDA Zi318, 4x 40GS/s

[26] LeCroy WL600, 7.5GHz

quality and minimum load applied to the output certain special resistors, provided with the probes, are soldered directly to the PCB, very close to the chip, the probe is attached to these resistors.

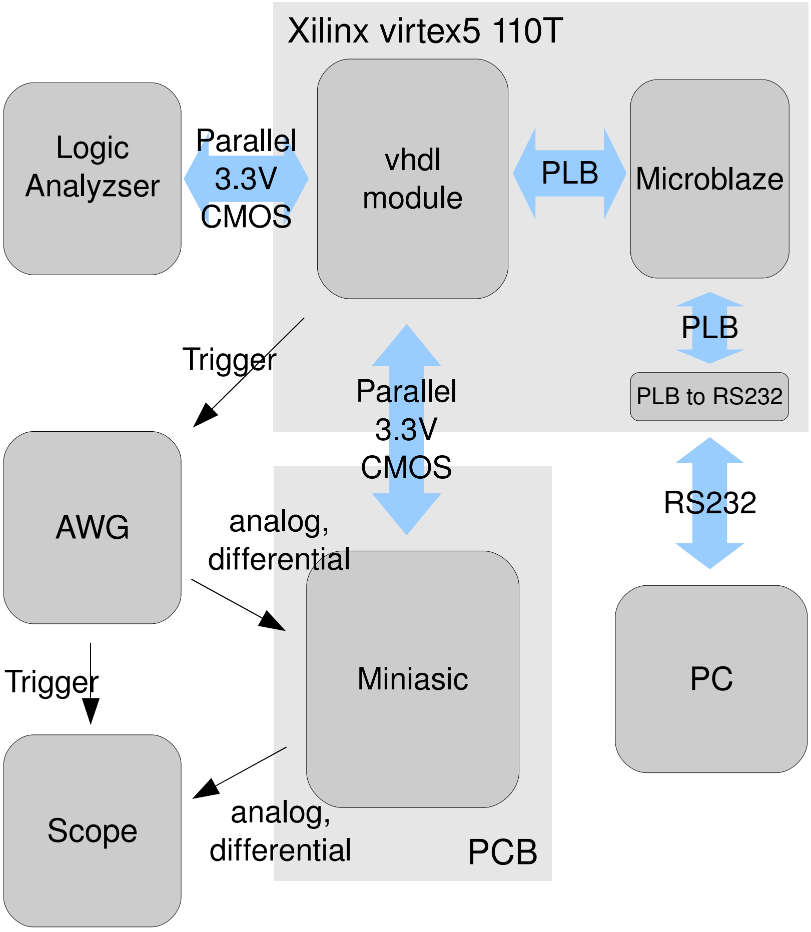


Figure 2.12: Block diagram of the experimental setup for measurements with the L1 prototype chip. The entire setup is controlled by a C code running in the microblaze processor implemented in the FPGA. The FPGA is used to configure the repeaters and can utilize the parallel data ports of the chip. Serial differential input to the chip is provided by the AWG. Serial output of the chip can be measured with an oscilloscope.

The simplest way to measure error rates on $r0 \rightleftharpoons r1$ with good statistics is to send and receive data using the parallel data IOs. It does not require much effort to generate random data in the FPGA, send it over $r0 \rightleftharpoons r1$ in any direction and compare the sent data with the one received to detect errors. Unfortunately the only chip that did not suffer from serious ESD problems or other fatal damages had a problem with some bond wires of the parallel TDI. More information about the individual technical problems that occurred while working with the chips are described in section 2.3.5.

Therefore another, more complex, setup to measure error rates was developed and used for most experiments. Instead of the parallel input from the FPGA, serial input generated by the

AWG was used. The sequence running in the AWG starts by looping packets containing only zeros to lock the DLL. A measurement is started by the FPGA applying an event signal. This leads to generation of a series of packets containing the numbers 0 to 63. Before each one the AWG is waiting for a trigger signal. The FPGA is always sampling the latest received packet at the parallel TDO, just before sending the the next trigger impulse. Afterwards the FPGA checks whether the last packet was received correctly before sampling and then triggering again. When the packet with number 63 has been sent the sequence jumps back to the 0 loop to keep the DLL locked until a new event is sent.

Because every waveform containing one packet of 40 samples must to have 960 samples in total for using the software sequencer, this setup is not suitable for measurements with high rates of packets. Since bursts of packets are prohibited by the AWG, the code in the FPGA was not optimized for high packet rates. This results in a total packet to break ratio not larger than about 1:100. Usually random data is necessary to test data transmissions. In this case the order of the packets is always the same. Due to the large temporal distance between the packets affection of a packet by the previous one seems impossible. Hence counting up the packets number should not affect the measurements.

The number of errors detected in a run from 0 to 63 is sent to the PC via RS232. Additionally the errors are summed up for a larger number of packets, in most cases 1 million. The typical runtime for the evaluation of 1M packets is about 10 seconds. The most time consuming process is sending the results for every run of 64 packets to the PC.

2.3.5 Employing a New Chip

A new chip is first glued to a PCB and bonded. Every IO of the chip is tested with a multimeter for abnormal resistances to either ground or VDD right after bonding, to see whether any ESD damages have already occurred. If no or no fatal damage has been found the PCB is assembled and again the IOs are checked. All of this was done at ESD protected workstations and with an anti-static wrist strap. Nevertheless several damages, most likely related to ESD events, were found, most of them directly after bonding. In the end only two chips did not suffer from fatal damages to 1.8V IOs at this point.

The next step is to connect the PCB to the setup and test whether configuration works, using the bit of the r1 configuration memory accessible by a bond pad. The following test for basic functionality of the chip consists of locking of the DLLs of r0 and r1 to TCLK signals. This can be checked by measuring their V_{CTRL}. Sending packets with any repeater to their differential outputs and observing them with the oscilloscope is another initial test for the repeaters. In Figure 2.13 a measurement of an L1 packet sent by r2 and measured after propagating the 1cm wire on the chip.

In case this initial tests have been successful, transmissions between r0 and r1 can be tested. This failed for one of the two chips mentioned. The repeaters r0 and r1 worked correctly on their own, they were able to lock their DLL and send data to their differential outputs, but it was not possible to transmit any packet between them. The chip behaved as though if there was no connection between the repeaters. Maybe at some point the long wire is broken.

The only chip able to send and receive data between the repeaters was damage at the parallel test input, caused by mechanical problems with the bond wires. Two of the input bits are shorted to ground, which prevented testing of all possible data packets. Hence, the AWG solution described above was developed and used for most measurements.

After a series of measurements with the AWG setup another damage on the chip was found.

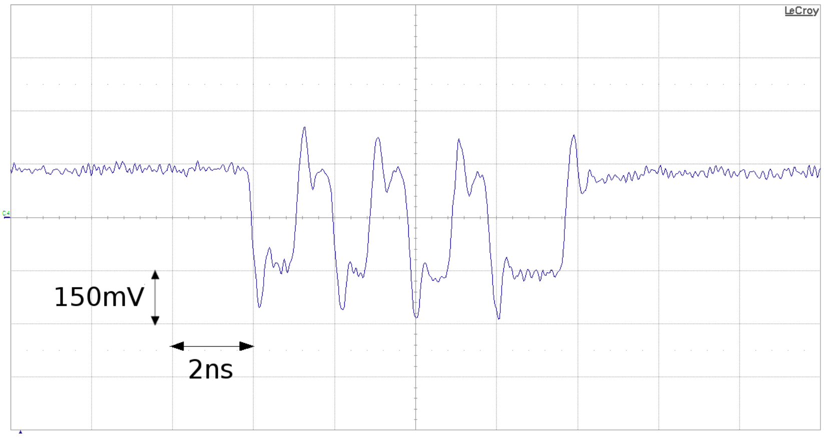


Figure 2.13: A typical L1 packet, containing `101010` as payload data framed by the start and the stop bit. Measured at the end of the 1cm wire connected to r2. The data rate is 1.0GBit/s.

In the experiments done carried out previously it was necessary to chose a setting of V_{CCAS}, a parameter affecting the receiver, see 2.4.2, that differs much from the expected value to enable any data transmission. Still, the performance was very poor. Investigations concerning this issue lead to the result that the transmission between AWG and r1 did not function properly. When evaluating the signals from the AWG at the input of the chip with an oscilloscope, they were distorted. The reason for this was finally identified. The resistance between the positive input and ground was too low. The cause this low resistance remains uncertain, but this problem was bypassed by increasing the DC offset individually for the positive signal until the signal visible at the input was symmetric. A setting of 600mV for the negative and 800mV for the positive input lead to the best results.
A potential cause for the low resistance is a damage to the second driver of the repeater r1. The differential pins of the repeaters can work as inputs or outputs. At r1 the ones accessible by pads are used as inputs. Nevertheless a second driver, disabled in this configuration, is also connected to them. An ESD damage to the NMOS transistor of the driver's output stage can cause a low resistance to ground.

2.4 Results

Several measurements have been performed with the L1 prototype chip. The results are presented in this section. First the DLL is tested and characterized. Afterwards the transmission between the repeaters r0 and r1 is tested under variations in different parameters. Besides the error rate, delay and power consumption of the repeaters is measured. Finally the impact of connecting the additional horizontal L1 lines as well as crosstalk from the adjacent wires is investigated. It turns out that transmission is reliable up to a data rate of about 1.6GBit/s in most cases.

2.4.1 DLL

Locking of the DLL to differential or test clock signals is very robust. In many cases it is not even necessary to toggle the DLL reset, which sets the DLLs control voltage to the adjustable starting voltage. Nevertheless, using the reset leads to reliable locking if the reset voltage is in the range of $\pm 200mV$ from the correct value. With a fixed value of 700mV for the DLL reset voltage, it is possible to lock the DLL to packets between 8 and 4ns length. This equals to data rates of 1.0 to 2.0GBit/s. V_{CTRL} was measured for various data rates. The result is shown in 2.14. For every data point three measurements have been performed. The standard deviation within this three measurements is below 1mV, therefore no error bars are visible in the diagram.

As mentioned in 2.1.2 the DLL needs a minimum data rate to prevent loss of locking due to leakage of the capacitor that stores V_{CTRL}. This was measured with a fixed data pattern of `010101`, which is assumed to be the worst case, transmitted from r1 to r0 at 1.2GBit/s. The data was generated by the AWG and applied to the differential input of r1. The time distance between the single packets is increased until the first error in transmission occurs. This is repeated three times and leads to an minimum packet frequency of $(960 \pm 80)Hz$ in the hardware time domain, this is typically equal to approximately 0.01Hz in biology. The error derives from the width of steps in which the packet frequency was changed. A rate of 0.01kHz is easily exceeded in every experiment of biological relevance.

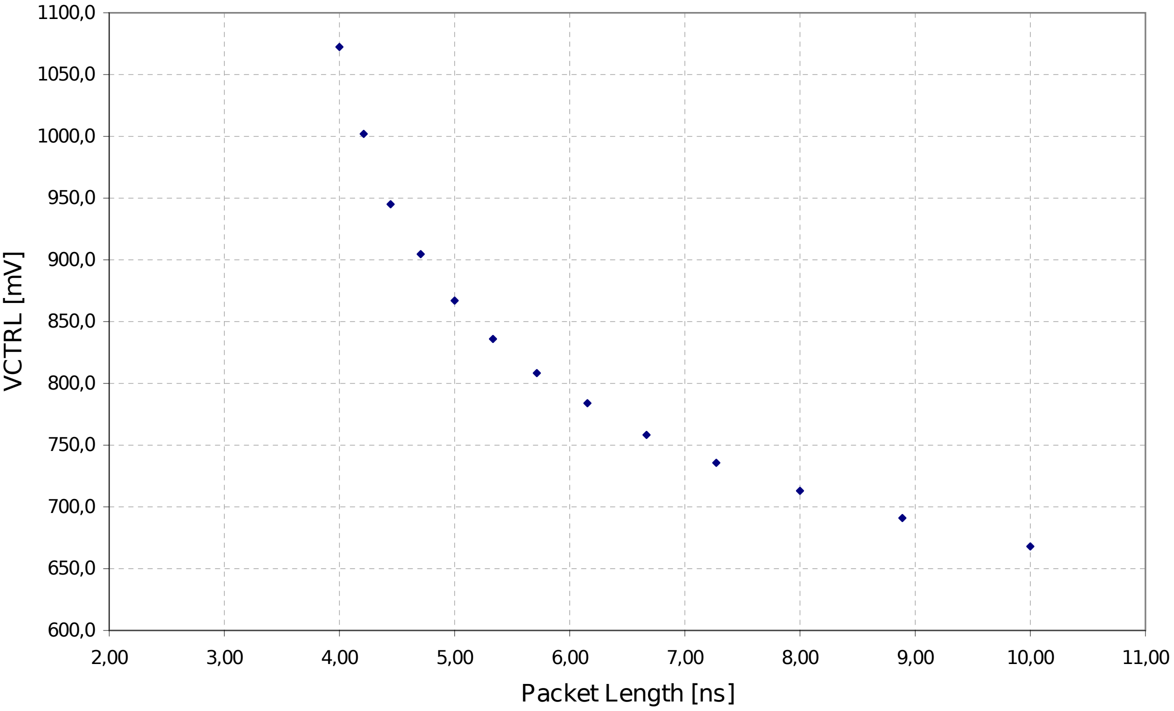


Figure 2.14: V_{CTRL} of the DLL plotted against the length of arriving L1 packets which were generated with the AWG. The range measured corresponds to data rates from 0.8 to 2.0 GBit/s. The errors of the data points are too small to be visible in the diagram.

Due to parasitic capacities and device variations the time bins generated by the DLL are not exactly equidistant. This can be seen in Figure 2.15. These variations are in a range of $\pm 100ps$ compared to the average width of the time bins. It would be interesting to investigate this effect for a large number of different repeaters as systematical variations can be partially corrected in the DLL layout.

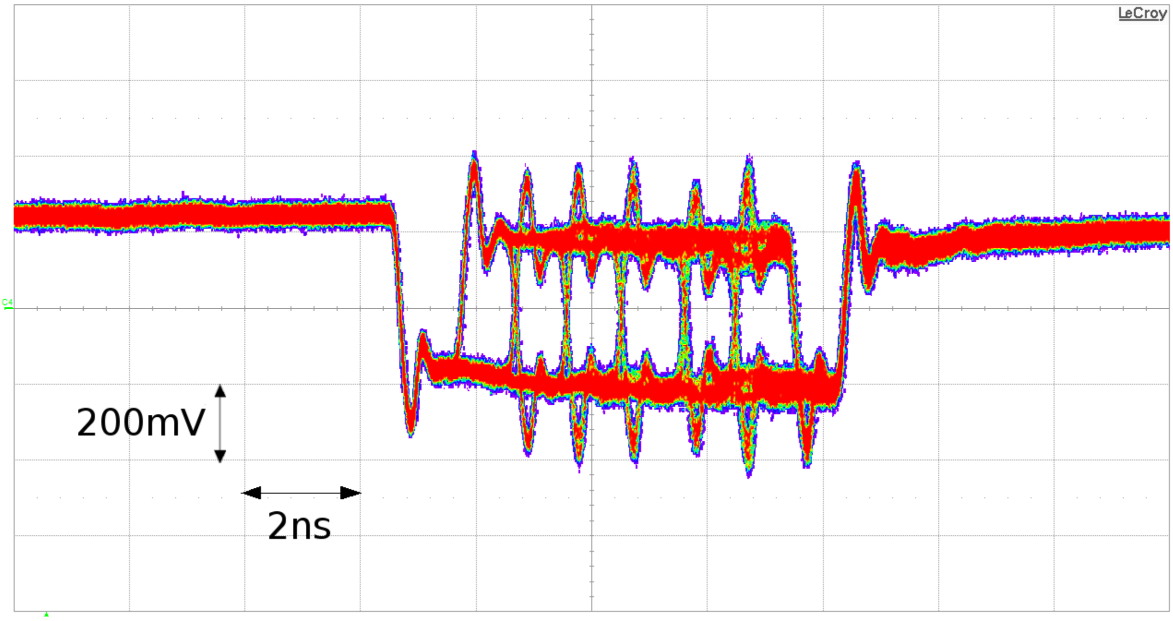


Figure 2.15: Persistence measurement of about 10M packets at a data rate of 1.0GBit/s. In the packets sent, the numbers from 0 to 63 are encoded repeatedly. The signal was measured at the end of the 1cm on-chip wire connected to repeater 2. The time bins generated by the DLL are not exactly equal in length, especially bit 4 is shorter than the others.

In summary the DLL works in the way it was intended to. The results comply to the values expected based on simulations.

2.4.2 Basic Transmission of Data between r0 and r1

The most important aspect, investigated with help of the prototype chip, is the question if, and under which conditions, the transmission of data packets between the repeaters r0 and r1 works. The experimental setup was already described. Data is sent to r1 by the AWG and received by the FPGA. To ensure the error rates measured are really the ones for the transmission between r1 and r0, in a first step the transmission between AWG and r1 is tested.

For data transmission via $r0 \rightleftharpoons r1$ the adjustable parameters of the repeaters are swept over certain ranges to see the impact on the error rates and find optimum settings. The analog voltages have to be set manually. Therefore the single parameters are assumed to be independent from one another to reduce the effort for the measurements to a reasonable amount. This was also suggested by the results of some preliminary measurements, which did not show any correlations.

The bias current for the differential amplifier in the receiver was chosen to be fixed at $I_{BIAS} = 110\mu A$. This setting leads to an acceptable power consumption and sufficient performance of the receivers. A higher bias current may improve the available data rate for the receiver. However, this is not required, as the measurements presented in the following show that the receiver is not the limiting aspect when it comes to transmissions between repeaters.

The cases of activated pass transistors and crosstalk are discussed later in the dedicated sections 2.4.4 and 2.4.5.

Receiver

A correct measurement of error rates for the transmission from r1 to r0 is only possible if no errors occur when r1 receives the data sent from the AWG. This means the AWG has to provide a significantly superior signal quality at the receiver of r1 compared to the signals arriving at the receiver of r0. Measurements of the AWG signals with a differential probe show that the signal quality at the chip is good. To further increase reliability for this transmission the amplitude of the AWG is set to 700mV.

The error rate for the transmission from the AWG to r1 is measured with exactly the same setup used for testing $r0 \rightleftharpoons r1$, despite the fact that the parallel output of r1, instead of the one from r0, is sampled and checked for errors by the FPGA. The result is shown in Figure 2.16. For every data point three times 1M packets have been measured, the error is given by the standard deviation within the three measurements. Up to 1.8GBit/s the data sent by the AWG is reliably received. As mentioned before, the limit for transmissions between r1 and r0 is in the approximately 1.6GBit/s, well below the limit of the r1 receiver.

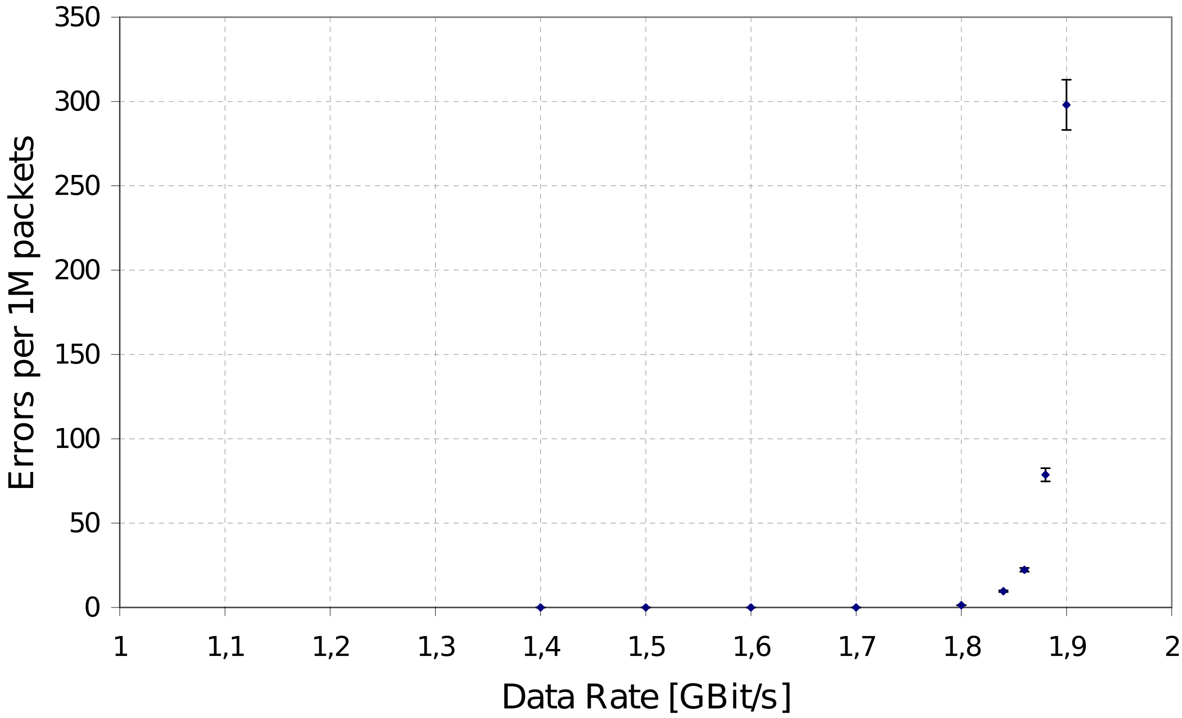


Figure 2.16: Error rate for the transmission from the AWG to r1 plotted against the data rate. Despite the good signal quality provided by the AWG, the receiver is not capable of receiving data reliably at a rate higher than about 1.8GBit/s. The error bars denote the standard deviation over three repetitions.

Impact of V_{CCAS}

Tuning of the voltage V_{CCAS} helps to compensate process variations that lead to an asymmetry between NMOS and PMOS devices in the receiver. The chip has an internal circuitry to generate the optimum value, derived from the actual conductances of NMOS and PMOS devices located on the same die. For the chip used the internally generated V_{CCAS} is $(1580 \pm 10)mV$, the error is estimated from variations of V_{CCAS} on several days of experimenting. To prove that the internal value is derived correctly, it is possible to force V_{CCAS} to an externally applied voltage. This is done for a range from 1500mV to 1750mV. The resulting error rates are measured. The data rate is set to 1,7GBit/s. This operating speed leads to a significant error rate and therefore reasonable statistics when measuring only 1M packets. The result can be seen in 2.17. For every data sample 5 times 1M packets have been measured, the error is given by the standard deviation within these five runs.
The internal V_{CCAS} seems not to have the optimum value. The error rate decreases for higher values of V_{CCAS}. As this effect is currently not understood and the difference in the error rates between the internal value and the best value in the measurements is not that large, in the following experiments the internal V_{CCAS} is used.

In the actual HICANN chip the internal V_{CCAS} is derived in the same manor as in the prototype chip, but it is also possible to force it to a value stored in a floating gate cell. The results of the measurements presented suggest to test the final HICANN chip with a V_{CCAS} even set to 1.8V. If this works it simplifies the system. In future revisions the floating gate cells storing V_{CCAS} can be used for other purposes. Why the generating circuit does not derives the optimum value, as it is expected to do, is not understood at the moment and needs to be investigated by simulations and measurements. Some more chips would be helpful to rule out an unusual behavior of the single chip used for the presented measurements.

The Differential Voltage

For V_{OL} and V_{OH} the common mode as well as the difference between them are parameters which need to be adjusted. The common mode needs to meet the optimum operating point of the differential amplifier at the receiver. So the common mode was swept with an amplitude of 200mV from 500 to 1400mV and the error rate for 1M packets was measured three times for every data point at a data rate of 1.7GBit/s. The error is the standard deviation within the three runs. The result can be seen in 2.18. It turns out that a value around 700mV seems to be the optimum, but as long as the common mode stays above 600mV the receiver is robust against shifts concerning this parameter. From 600mV downwards the error rate increases dramatically.

Next the amplitude was swept with a common mode of 700mV. It is expected that if the amplitude is too high the drivers are no longer capable of switching polarity reliably at a change in the bit stream. On the other hand, in case of a small amplitude the receivers are more likely to detect wrong polarity. Underlying could be either a systematic error, an offset of the differential amplifier or statistic errors due to noise.
The result of this measurement is plotted in Figure 2.19. Again 1M packets are measured three times for every data point. At amplitudes greater than 250mV the error rate increases, most likely because the drivers are no longer able to switch the polarity of the wire within a single bit period. The correct detection works down to an amplitude of 5mV. Below this voltage it is difficult to adjust V_{OL} and V_{OH} reliably with sufficient accuracy. Attempts to

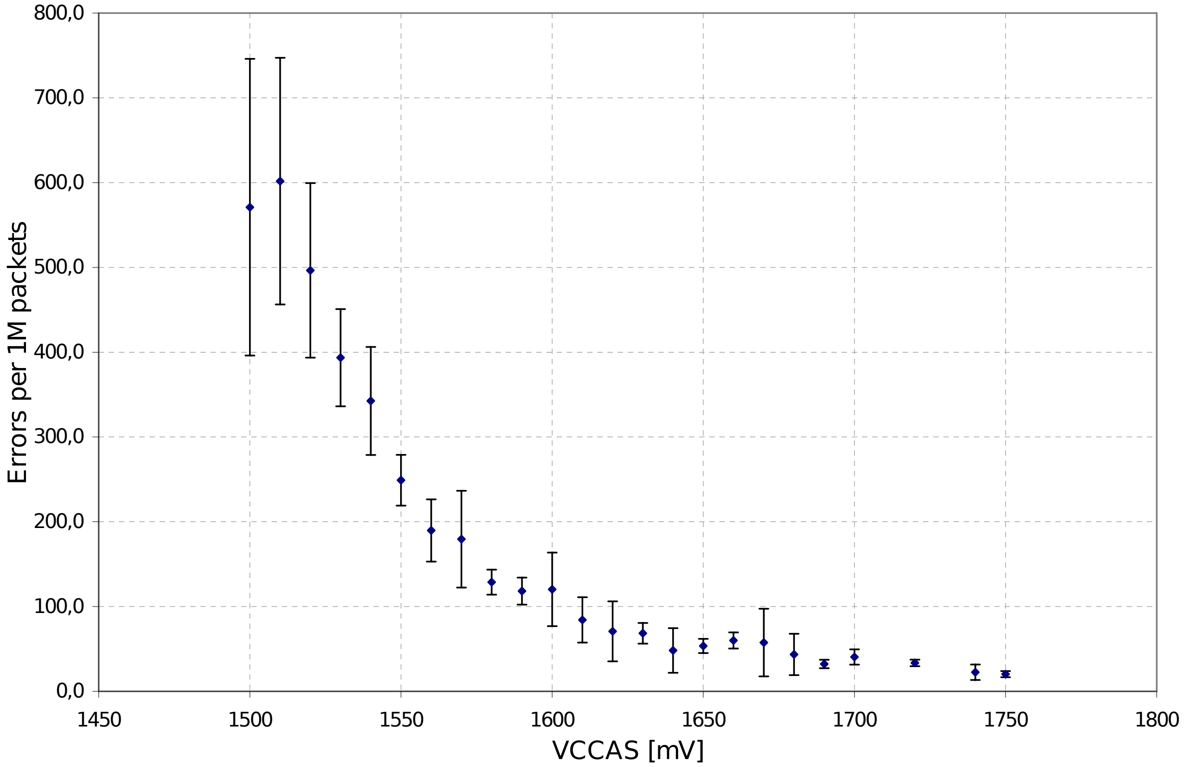


Figure 2.17: Impact of V_{CCAS} variations on the error rate for data sent over $r0 \rightleftharpoons r1$ at 1.7GBit/s. The error bars denote the standard deviation over five repetitions. The value of V_{CCAS} derived internally in the chip is 1580mV. It seems larger values for V_{CCAS} can improve performance.

measure with an amplitude of only 2mV amplitude lead to error rates far exceeding the range shown in the diagram.

It is shown that variations of $\pm 50mV$, starting from 200mV, do not affect the error rate significantly. The offset of the differential amplifier seems to be below 5mV.

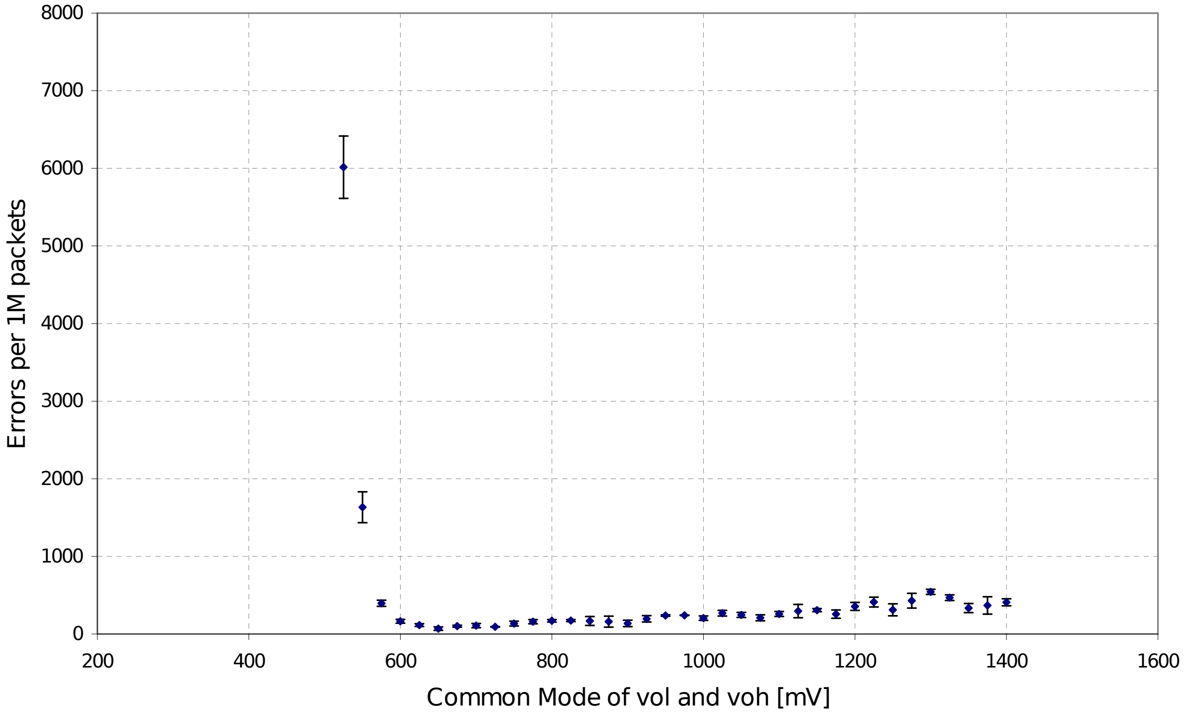


Figure 2.18: Impact of variations in the common mode of V_{OL} and V_{OH} on the error rate of data transmission via $r0 \rightleftharpoons r1$ at 1.7GBit/s. The error bars denote the standard deviation over three repetitions.

In conclusion, the measurements show that the system is sufficiently robust against variations of the analog parameters.

2.4.3 Delay

The delay for a single repeater cannot be measured directly with the L1 prototype chip. It is derived from a combination of two different measurements. First of all the propagation delay caused by the 1cm L1 wire on the chip is determined. Therefore r0 and r2 are sending packets containing zeros, using their parallel data inputs. Their TCLK inputs are shortened to ensure both are sending the packets simultaneously. Probes are placed at the differential outputs of r0 and r2. While the output of r0 is directly connected to the bond pads, the signal of r2 propagates on the 1cm on-chip wire at first. The time difference displayed on the oscilloscope was measured with a result of $\Delta t_{propagation} = (240 \pm 50)ps$. The large relative error is the estimated uncertainty when defining the edges of the signals on the oscilloscope.

Next the total delay for a transmission via $r0 \rightleftharpoons r1$ is measured. The AWG sends data to the serial input of r1. A differential probe is located there. Another probe is placed at the serial output of r0. The delay for the first edge and the last edge of a packet was measured

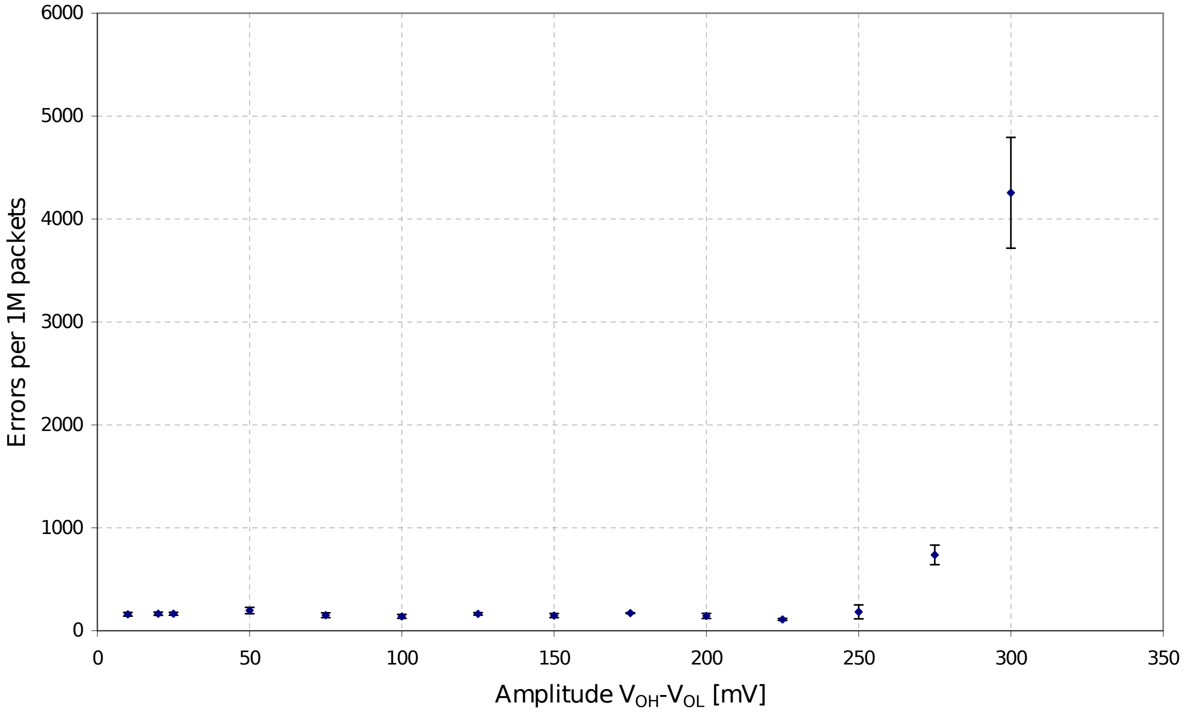


Figure 2.19: Impact of variations in the amplitude of the differential voltage, V_{OH} minus V_{OL}, on the error rate of data transmission via $r0 \rightleftharpoons r1$ at 1.7GBit/s. The error bars denote the standard deviation over three repetitions.

with an oscilloscope for different data rates. The average of both measurements is taken as Δt_{total} for the respective data rate. The errors on this values are estimated by the uncertainty when defining the edges of the packets on the oscilloscope.
Now the delay per single repeater is calculated by $\Delta t_{repeater} = \frac{1}{2}(\Delta t_{total} - \Delta t_{propagation})$. The results for the determined delay per repeater are shown in Figure 2.20. The errors is the result of a calculation for the propagation of uncertainty, based on the errors of Δt_{total} and $\Delta t_{propagation}$.
A delay of three bit periods is caused inevitably by the timing scheme of the repeater. This contribution to the total delay is of course dependent on the data rate used. In Figure 2.20 the delay per repeater minus this inevitable delay is also shown. The remaining delay caused by receiver and driver seems to be independent of the data rate.

In simulations the delay caused by a repeater and 1cm of L1 wire is 2.3ns at 2.0GBit/s. The value measured for this setup is $\Delta t_{repeater} + \Delta t_{propagation} = (3.0 \pm 0.16)ns$.

For experiments performed on the hardware a delay of less than 1ms in biological time for the transmission of a neural event is desirable, which equals to 100ns in the hardware time domain at a typical speed-up of 10^4. In the worst case a signal is routed across a wafer crossing about 20 repeaters. This allows a maximum delay of 5ns per repeater. As long as the data rate of the L1 system is set to more than 1.0GBit/s the repeaters fulfill this requirement, even when taking the additional delay caused by priority encoder and synapse address decoders into account.

2.4.4 Horizontal Lines

As described before, L1 routing requires horizontal L1 lanes which can be connected to the vertical lanes by programmable pass transistors. With respect to these the maximum wire length a signal passes between two repeaters totals to 15mm. It is not possible to test such a vertical-horizontal-vertical connection with the prototype chip. It is only possible to enable horizontal wires attached to $r0 \rightleftharpoons r1$. Since this causes an additional capacitive load on $r0 \rightleftharpoons r1$, reduced performance is expected.
In Figure 2.21 the error rate at various frequencies is plotted for different settings of the pass transistors which enable the horizontal wires. For every data point three times 1M packets were sent, the error is the standard deviation of the three runs. No significant impact of additional wires of 5 or 10mm of length could be observed, compared to the error rate measured without any horizontal wire active.
It seems the capacitive load is not the limiting factor for the transmission in these cases. Only when h1 and h2 are active simultaneously a limited reduction in performance becomes evident. The transmission is reliable up to 1.6GBit/s, independent of the activated horizontal lines, as long as no crosstalk occurs. The question of crosstalk is discussed in the next section, 2.4.5.

2.4.5 Crosstalk

Crosstalk has been identified as an important aspect within the L1 system. The worst case is a repeater that receives data while the adjacent repeaters are sending, as the amplitude of the senders, due to the strong preemphasis, is very high directly at the sender. For this situation hardly any transmission is possible in the simulations if no councilation based on capacitors

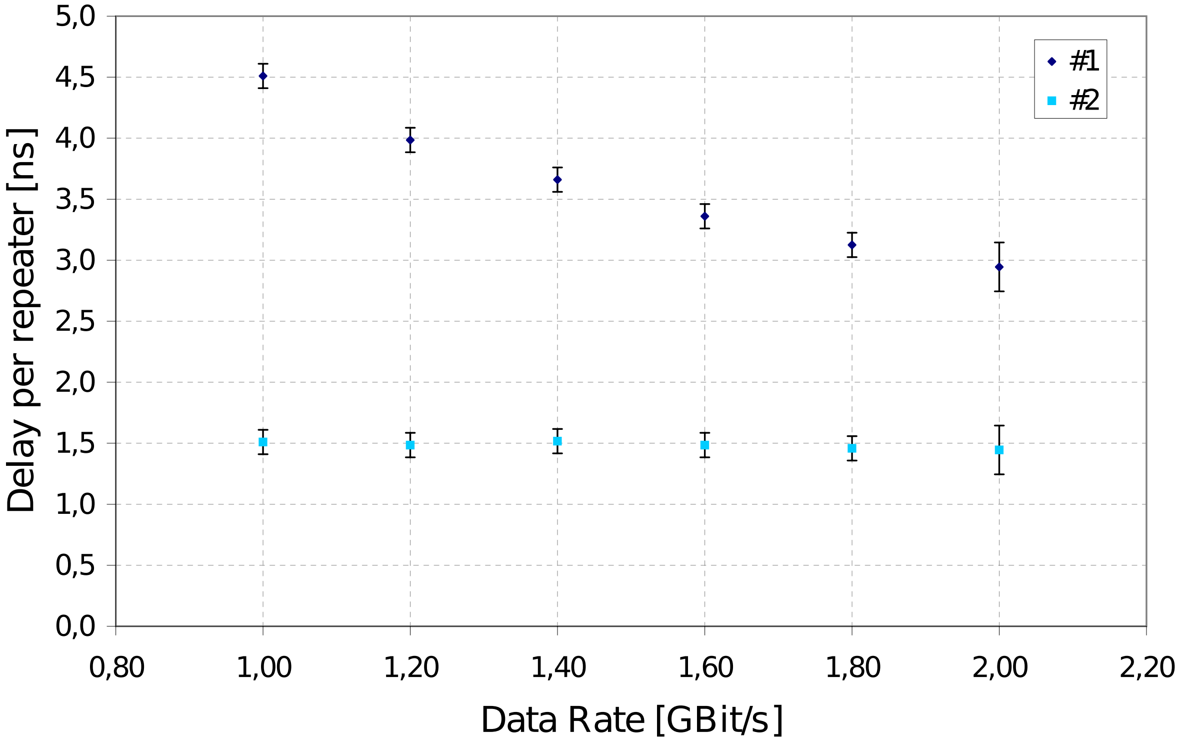


Figure 2.20: The delay caused by a single repeater plotted against the data rate. #1 shows the total delay. The inevitable delay of three bit periods in the repeater is subtracted to obtain #2. The remaining delay seems to be independent of the data rate. The errors result from the estimated uncertainty when determining the edges of the signals on the oscilloscope.

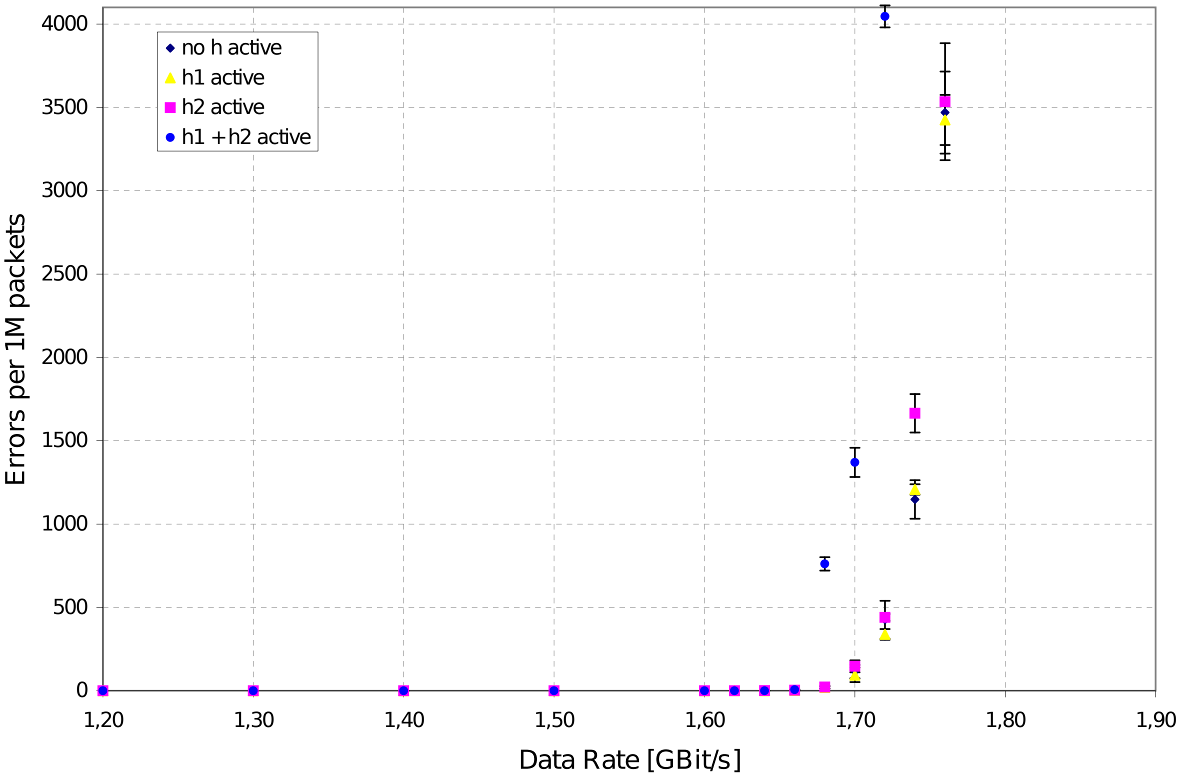


Figure 2.21: Error rate for data transmission via $r0 \rightleftharpoons r1$ plotted against the data rate for different combinations of activating the the pass transistors. Neither the activation of h1 (5mm) nor the activation of h2 (10mm) has any significant impact, compared to the case that no pass transistors is enabled. Up to a data rate of 1.65GBit/s the error rates are below 2 per 1M packets. Only in case both lines available are activated, performance is reduced. The error bars denote the standard deviation over three repetitions.

capacitance $[fF]$	Errors per 1M packets	Standard deviation
0	240	54
52	399	249
104	551	353

Table 2.1: Error rate for transmission via $r0 \rightleftharpoons r1$ in case of severe crosstalk. The results for three different settings of the crosstalk councilation capacitors are presented. For every setting 20 times 1M packets have been evaluated.

is used. The strong decrease of the amplitude of the signals while propagating along the wires is the reason why the crossings are not completely sufficient for suppressing the crosstalk.

For the measurements the worst case was modeled. In this case r1 sends data to r0, while r2 and r3, located next to r0, are sending in the opposite direction. The data rate of the transmission via $r0 \rightleftharpoons r1$ is 1.6GBit/s, the usual setup with the AWG sending to r1 and the FPGA sampling the parallel output of r0 is used. The repeaters r2 and r3 are sending at a data rate of 2.0GBit/s to increase their impact. The packets contain various neuron numbers, r2 is counting up from 0 to 63, r0 is counting down. The packet rate on $r0 \rightleftharpoons r1$ is below an average L1 activity. To ensure the probability of two packets being sent at the same time by two different repeaters hence affecting each other by crosstalk must be in a realistic range. Therefore the packet to pause ratio for r2 and r3 is chosen to be 1:3, which is rather high.

The resulting error rate is measured for three different amounts of councilation capacitances, the result is shown in Table 2.1. For every value 20 times 1M packets have been measured.

The result was not as expected. First of all it seems that a larger capacitance in the councilation circuit increases the error rate, instead of decreasing it. The remarkably high standard deviation needs to be discussed also. This could indicate an insufficient experimental setup. At least within a number of 1M packets some correlation between the time points the different repeaters are sending seems to occur. However, with a number of 20 individually triggered measurements this should not effect the average error rate significantly. Especially since any correlations in time are not related to the chosen capacitance, the results are comparable.
The most important question is why the councilation does not function properly. One must be very careful when deriving the correct configuration of the crosstalk councilation for the given situation. This was checked several times, nevertheless some measurements with other configurations were performed to double-check that the configuration used is correct. None of them lead to better results than the most likely correct configuration, which are shown above.

The entire issue of crosstalk strongly depends on the exact properties of the wires, especially their resistance and capacitance. If the models used for the simulations on which the crosstalk councilation is based are not precise enough, it is possible that the capacitances have completely wrong dimensions. Watching the packets sent by r2 or r3 at the end of the 1cm wire for example show a better signal quality than the simulations for this situation.
Another possible reason for the crosstalk related problems might be a coupling through the power supply. When r2 or r3 send a packet, the power supply is possibly perturbed to a degree which disturbs r0 while receiving or distorts a packet sent simultaneously. An attempt to rule this out was measuring with a setup in which r2 and r3 are sending simultaneously exactly before and never during the transmission of a packet on $r0 \rightleftharpoons r1$. With the setup

including the AWG to send the packets it was not possible to ensure that the packets of r2 and r3 are sent precisely before the ones from r0, as the setup is insufficient for appliances that require such a precise timing. Adaptation would require an extensive redesign of the software. The time gap between the packets sent by the crosstalk receivers and r1 must be very small in order to see any effect. Again, the ability of using the parallel test data input of r1 and the test clocks would solve the problem, as it is easy to send impulses with a defined phase shift to the test clock inputs.

Finally the parallel ports were used at a speed of 1.5GBit/s with just two different data patterns, omitting the damaged input bits. This does not allow for measuring any comparable error rates, but the packets of the crosstalk receivers end in the exact moment r1 starts sending. The logic analyzer is used to view the received data. Triggering on errors in the transmission did not lead to any result during several millions of packets. Therefore it seems probable that there is no, or at least no massive, problem with crosstalk through the power supply.

The relevance of these results for working with the HICANN chips is not clear yet. Finally only tests with the original setup, and hence realistic activity on the L1 buses, can show whether the crosstalk occurs to the degree it does in the tests performed with the prototype chip. These experiments aimed at setting up a worst case scenario. Perhaps under typical conditions the error rate is below 100 errors per 1M packets. This might be tolerable in typical experiments. Based on tests with the first prototype of a HICANN chip one must decide whether it is necessary and possible to further address the crosstalk problem.

2.4.6 Power Consumption

As mentioned in the beginning, power consumption is an important issue for the wafer-scale system. Taking estimations on the power consumption of the different systems on the wafer into account, L1 communication consumes almost half the total power during operation. The reason is the bias of the receivers as well as the DLL, the only circuits on the wafer consuming a relevant amount of static power. There are about 260k receivers on a wafer. Each one comes along with a DLL.

It is not feasible to directly determine the power consumption of a single repeater. The only possibility is to measure the change of the total current drain on the 1.8V supply of the chip, depending on the number of enabled and active repeaters.

First the current I_{vdd}, consumed by the chip, is determined for the case that all repeaters are disabled. The result is $I_{disabled} = 483 \pm 2\mu A$. The error is estimated by variations for this value during the measurement. A lower value is expected since a disabled repeater should consume hardly any current, and there is not much more circuitry on the chip. This can be caused by the damage at the input of r1, mentioned in 2.3.5. Another possible explanation would be logic-gates floating to undefined states when the repeaters are disabled. Undefined logic states cause a static power consumption as neither the NMOS nor the PMOS transistors of the affected gate are in cut off mode. This allows for a static current from supply to ground. As long as no undamaged chip is available, it is not possible to further investigate the reasons for the high power consumption in case all repeaters are disabled.

Next, single repeaters and combinations of several repeaters are enabled. To make sure the repeaters are completely inactive, the parallel data inputs and the test clock input are activated and statically set to zero. In Table 2.2 the results measured for I_{vdd} are shown in the second column, the repeaters listed in the first column are the enabled ones. The third

repeaters enabled	I_{vdd} $[\mu A]$	I_{static} $[\mu A]$
r0	822	339
r1	829	321
r2	802	319
r3	1175	692
r0 + r1	1140	657
r1 + r2	1123	640
r0 + r2	1142	659

Table 2.2: Static power consumption of the repeaters. I_{vdd} represents the total current consumption of the chip for the setup specified in column 1. To obtain I_{static}, the current consumption in case no repeater is enabled, $I_{disabled}$, is subtracted.

column shows the difference between I_{vdd} and the previously determined $I_{disabled}$. The error for I_{vdd} in these measurements is estimated to be below $5\mu A$, based on readout accuracy.
As a disabled repeater is supposed to consume hardly any power, this difference, termed I_{static} in the following, should give the total power consumption of an enabled repeater without any activity. The error for I_{static} results from the errors of $I_{disabled}$ and I_{vdd} and equals to approximately $6\mu A$. It is eye-catching that r3 consumes much more than the other repeaters, most probably it is damaged in some way and will not be taken into account for the measurements related to power consumption. Nevertheless r3 is able to send packets. This has been confirmed with an oscilloscope.

The average of I_{static} per single repeater resulting from the values given in Table 2.2 is $I_{static} = (326 \pm 8)\mu A$. The error is given by the standard deviation between the different repeaters.

This value is larger than expected. The bias current is $110\mu A$. The DLL consumes about $50\mu A$ according to simulations. Damages causing the additional current are rather unlikely since the repeaters 0, 1 and 2 are consuming almost exactly the same amount of current, even if enabled in different combinations of two repeaters, see the last 3 lines of Table 2.2. It is more likely that parts of the logic in the inactive repeater are floating to undefined levels. In this case the power consumption of a repeater with very low activity should be even beyond the values measured in the static case.
The activity-dependent current consumption of r1 is shown in Figure 2.22. It receives data packets from the AWG and sends them to r0, which is disabled to measure only the current consumption of r1. The additional wire h1 is enabled to increase the capacitive load. Two different fixed patterns are sent in independent experiments. As the worst case, with a maximum number of bits changing, the packet `010101` is sent repeatedly. The lowest amount of power is necessary to send packets containing just zeros, `000000`. The repetition frequency of the packets is varied between 8kHz and 100MHz for both cases. The errors estimated by readout accuracy are below 5mV and therefore not visible in the diagram.

For very low repetition rates the current tends to about $I_{dynamic} = 165\mu A$. This is significantly lower than the value determined for $I_{static} = 326\mu A$ and is close to the expected value of $160\mu A$ as a result of the static power consumption of the receiver and the DLL. Therefore it seems that in case of no L1 activity some logic gates float into undefined states.

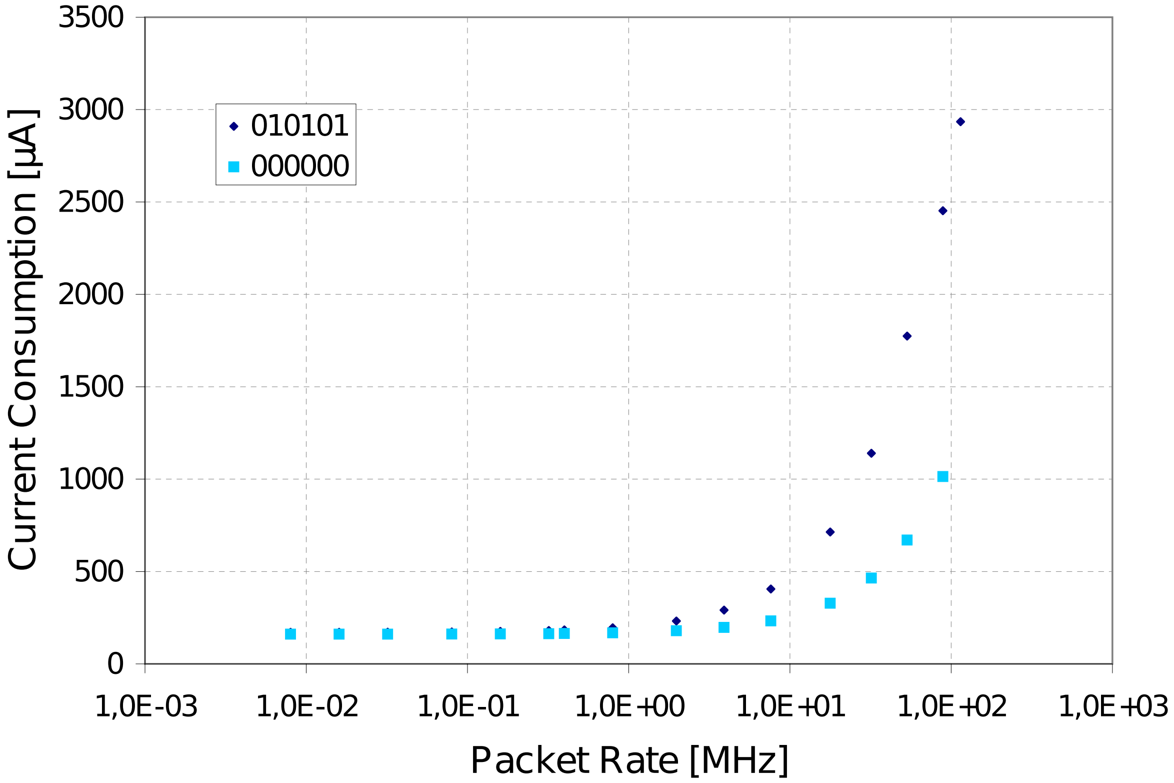


Figure 2.22: Current consumption of a repeater plotted against the packet rate for two different data patterns. `010101` is the pattern leading to the highest power consumption, whereas `000000` is the pattern requiring least power. The data rate is 1.6GBit/s. The errors are to small to be visible in the diagram.

2.4.7 Reasons for Limited Data Rate

So far only the resulting error rates of a transmission in dependency of other parameters have been discussed. Now the question of the type of errors is addressed and possible reasons are discussed briefly.
For further evaluations of the errors occurring the software was modified to not only count errors but also to generate statistics on how often which individual bit failed. These statistics are evaluated for different situation in which a significant error rate was observed for the transmission of data.

First of all the situation of the AWG sending data to r1 is tested at a data rate of 1.89GBit/s. In Figure 2.25 a) the number of errors in every single bit, relative to the total number of incorrect bits, resulting from a measurement evaluating 100k packets is shown. In Figure 2.25 b) the for a transmission via $r0 \rightleftharpoons r1$ for a data rate of 1.74GBit/s. All other parameters are at their respective optimum values. As shown in the diagram most of the errors occur at the end of the packet. In Figure 2.25 c) Another means of provoking errors was chosen. At a speed of only 1.60GBit/s V_{CCAS} is reduced until errors occur, in this case 1500mV. Again most of the errors occur among the last bits. Figure 2.25 d) shows the distribution of the errors in case of a transmission with 1.56GBit/s but heavy activity at the adjacent wires. The disturbances caused by crosstalk are supposed to affect every bit to the same degree. The distribution of the incorrect bits is more even, but still the last bits are more often inaccurate.

Obviously the last three bits are generally more likely to be incorrect, as soon as a process of sending a packet by a repeater is involved. This might be related to a degradation of the supply voltage during the sending of the packets. Furthermore systematic uncertainties of the DLL sum up during a packet and are therefore maximum for the last bits.

At a data rate of 2.0GBit/s a fatal error concerning the last bits can be observed at the sender. This is shown in Figure 2.23. The preemphasis to generate the rising edge of the stop bit does not work in some cases. Figure 2.24 shows a persistence measurement at 2.0GBit/s. The last bits are obviously often distorted. Further experiments show that this arises only if the last data bit is 1. The preemphasis in this situation definitely works at lower speed. Therefore it is not possible that a general mistake in the driver's logic is responsible for this. It was also never observed during simulations. This suggests a problem with the power supply. While sending the packet the supply voltage may decrease. As a consequence the logic gates responsible for enabling the preemphasis at the end of the stop bit may be affected. However, in order to avoid such problems the chip has different 1.8V supply voltage rails. The one with the highest load, which is used by the drivers, is independent from the one for the logic circuitry. So there is no completely convincing explanation for this problem.
The repeaters used in the HICANN chip has additional blocking capacitances for power, so maybe this problem does not occur there. On the other hand the L1 prototype chip has only four repeaters in a row, in the HICANN chip up to 20 are located next to one another, connected to the same power lines.

The sender is not the only element not capable of working at 2.0GBit/s. The receiver is also limited, as shown by measurements using the AWG which provides good signal quality. This was already discussed in 2.4.2. Errors during receiving occur at data rates above 1.8GBit/s.

Finally, simulations of the repeaters show that operating at 2.0GBit/s is only possible if the chip is in a typical or fast corner. Unfortunately it was not possible to acquire any information on the process parameters for the chip. On every wafer there are test structures that allow the producer to measure many parameters concerning the quality of the chips in the so called ”wafer acceptance test”. The files with the results of this test for several wafers have been provided to the author. However, due to confusions with the labeling of the chips, the IMEC was not able to specify on which of these wafer the L1 prototype chips were located.

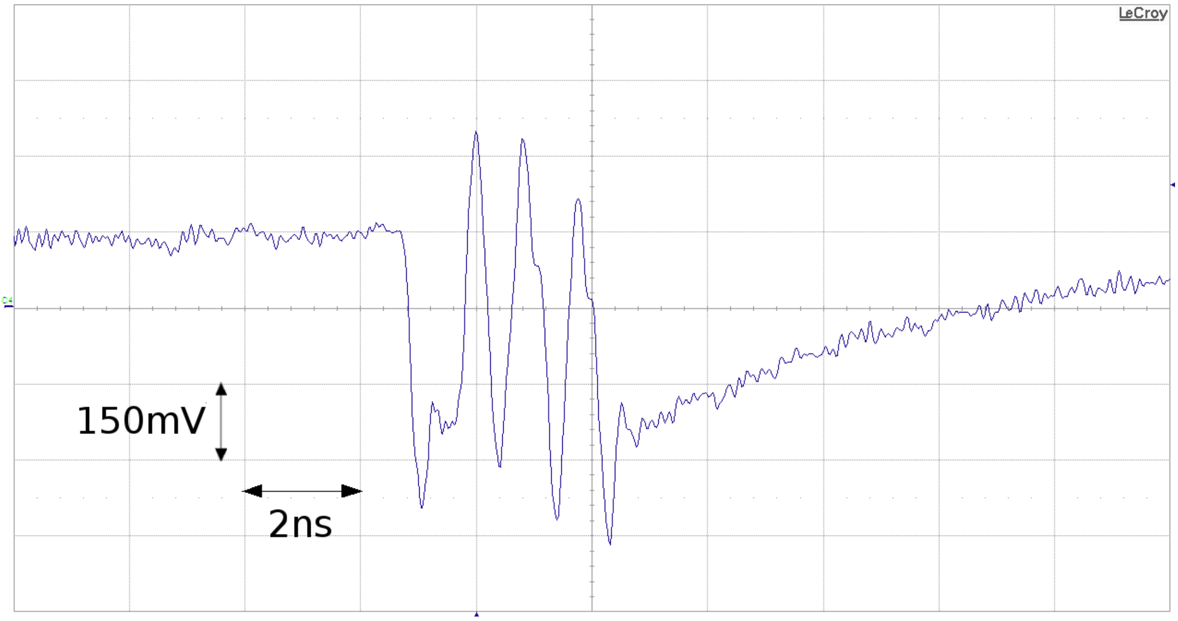


Figure 2.23: Packet containing the data `010101` at a data rate of 2.0GBit/s, measured after propagating 1cm of on-chip wire at the pads connected to r2. The preemphasis at the end of the packet is not generated correctly.

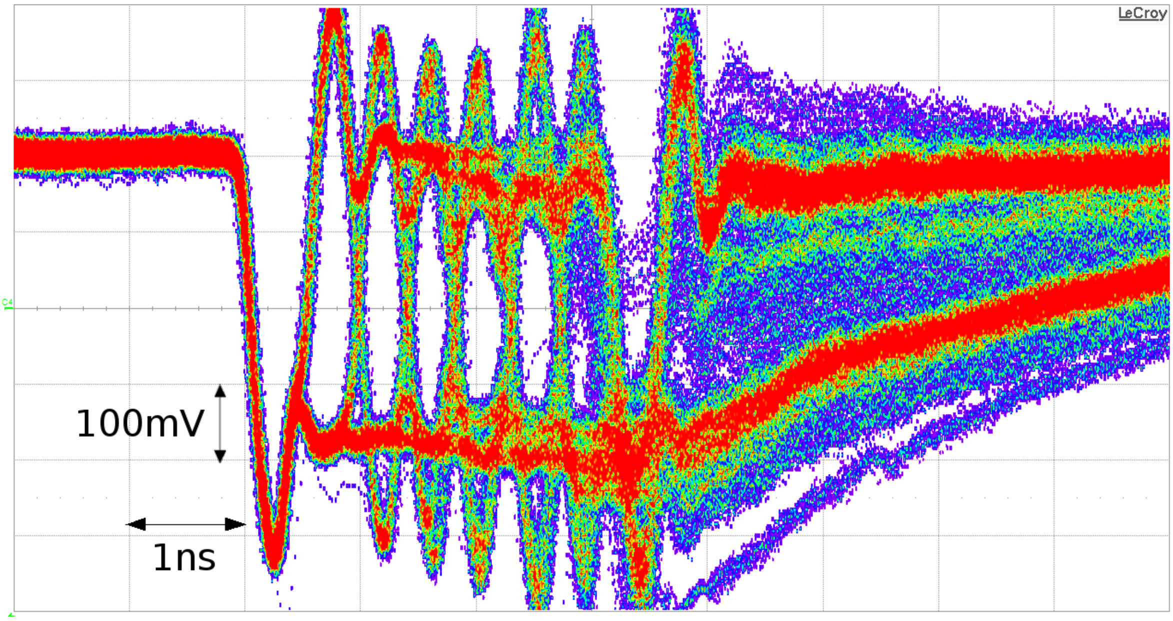
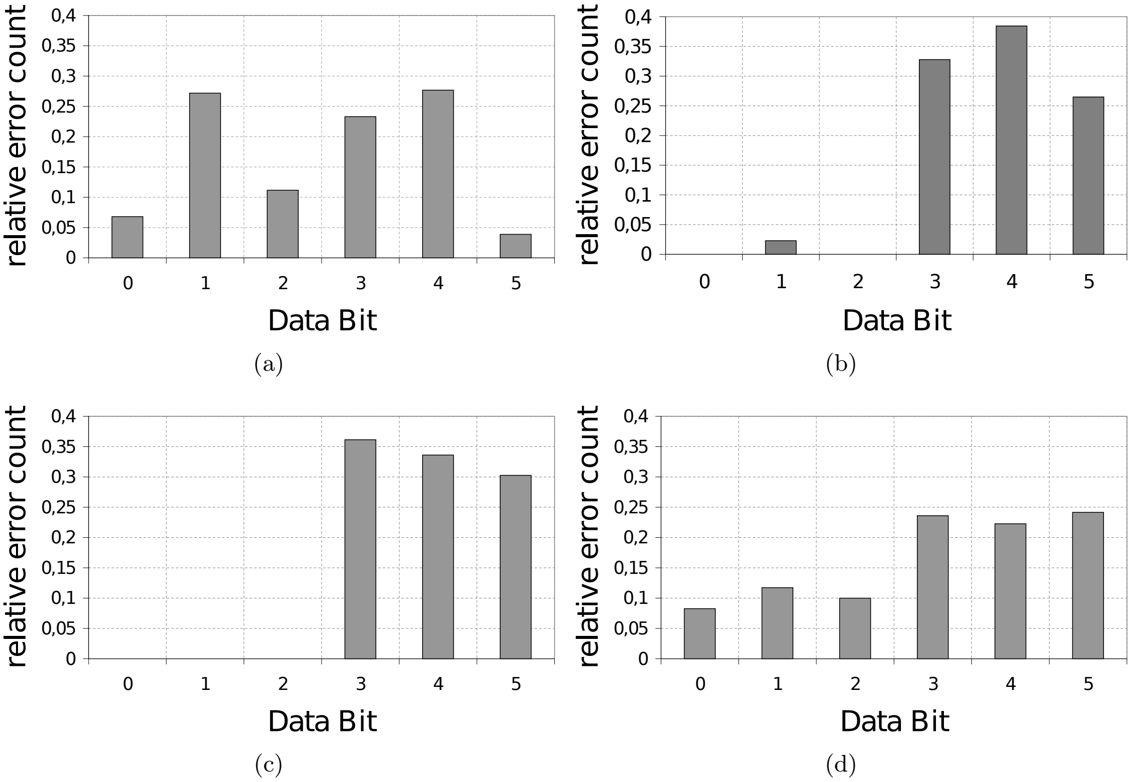


Figure 2.24: Persistence measurement of about 10M packets at a data rate of 2.0GBit/s. In the packets sent, the numbers from 0 to 63 are encoded repeatedly. The signal was measured at the end of the 1cm on-chip wire connected to repeater 2. Especially at the end of the packet the sender is not capable of generating the bits correctly. The failure of the preemphasis at the stop bit is also clearly visible.



(a)

(b)

(c)

(d)

Figure 2.25: The number of errors for a single bit in relation to the total number of incorrect transmitted bits measured during 100k packets is displayed for different setups resulting in high error rates. a) Distribution of the incorrect bits in case the AWG sends data to r1 at a rate of 1.89GBit/s. b) Distribution of the incorrect bits for a transmission via $r0 \rightleftharpoons r1$ in case of a data rate of 1.74GBit/s. c) Distribution of the incorrect bits for a transmission via $r0 \rightleftharpoons r1$ at only 1.6GBit/s, while V_{CCAS} is reduced to 1500mV. d) Distribution of the incorrect bits in case of high activity on the adjacent lines, so crosstalk causes errors in the transmission. The data rate is 1.56GBit/s. The most striking result is that the last three bits are most likely to be incorrect as soon as an L1 sender is involved in the transmission.

3 Floating-Gate Memory Cells

The implementation of the neurons provides a number of parameters to adjust their behavior. On the one hand this possibility is important to adapt the characteristics of the artificial neurons to the characteristics of the natural cells they are intended to mimic, on the other hand to eliminate mismatch, a significant issue within analog VLSI circuits, by calibration. The neurons implemented on the HICANN chip have 24 adjustable parameters, 12 voltages and 12 currents, e.g. the resting potential or the firing threshold.

To provide programmable analog voltages the straightforward solution is to store a digital value and use a DAC. Since DACs, especially if a certain resolution and accuracy is necessary, consume a lot of chip area, this is not a good choice if a lot of independent values are needed. To increase the density of analog value storage, it is possible to use a capacitor for every parameter. The capacitors are charged one after another by one DAC, similar to common digital dynamic memory. Due to leakage it is required to refresh the capacitors repeatedly during operation. It has to be ensured that recharging does not interfere with normal operation of the chip, which requires a complex controller for the DAC and the addressing circuitry.
On the Spikey chip of the Stage 1 hardware such a combination of capacitors and a DAC is used. Refresh cycles are required every few microseconds. Due to the drawbacks of this solution most of the neuron parameters are the same for all neurons located in the same quarter of the chip. So calibration of single neurons is not possible. Furthermore the flexibility of the network architecture is strictly limited if different types of neurons are used in one experiment.

Aiming at much larger and more flexible networks, a different solution, based on analog floating-gate memory cells, is implemented in the Stage 2 hardware. The cells used in the HICANN chip have been developed by Dr. André Srowig[1] and will be discussed in the following. The advantages of these cells - they are nonvolatile, power- and space-saving - made it possible to implement neurons with 24 parameters individually adjustable for every neuron. Several test with a prototype chip, containing an array of 3096 cells, were performed. The chip was designed by Sebastian Millner, based on circuitry designed by Dr. André Srowig. The tests focus on the controller for the cells, written by Sebastian Millner, and the addressing circuitry. The performance of the cells is a side issue. Measurements concerning the analog performance, such as the available output range and accuracy have been performed previously, see [34, 20].

[1] Dr. André Srowig is a former member of the Electronic Vision(s) group

3.1 Floating-Gate Memory

3.1.1 Digital Floating-Gate Memory

Today, floating-gate cells are common nonvolatile binary storage devices, used in so-called ”flash memory”. They consist of a MOS transistor whose gate is completely isolated by surrounding silicon dioxide, also referred to as ”floating”, and an additional gate above the first one, termed the control gate. A schematic cross section of a standard floating-gate cell can bee seen in Figure 3.1. The first experiments concerning storing charge on insulated gates of transistors were performed in 1967, see [16].

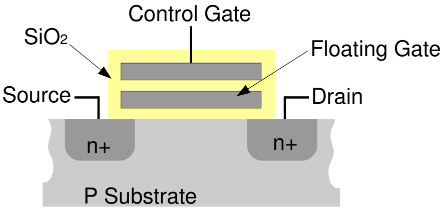


Figure 3.1: Cross section of a common floating-gate cell.

Electrons can be transferred onto or from the floating gate by Fowler-Nordheim-Tunneling, if the potential difference between source and the control gate is large enough (see e.g. [17]). If source is connected to ground and the control gate is connected to a positive voltage, typically more than 5V are required to enable a significant number of electrons to tunnel onto the gate. If the potentials are interchanged, the electrons tunnel in the opposite direction.

In commercial digital floating-gate devices an additional mechanism to transfer electrons to the floating gate is used, it is named ”hot electron injection” [29]. A voltage is applied between source and drain of the floating-gate transistor, accelerating the electrons in its channel. A fraction of these electrons is scattered upward and, in combination with a positive potential at the control gate, capable of overcoming the insulation barrier. This mechanism is not used in the cells which will be discussed in the following.

Without a rather high voltage at the control gate, allowing for tunneling, the electrons are trapped on the floating gate. The amount of charge on the floating gate determines the conductance of the underlying transistor. Therefore it can be used as nonvolatile memory. Typically one cell is used to store one bit, represented by two different conductance states of the transistor. More information on floating-gate devices can be found e.g. in [19], chapter 4.

3.1.2 The Analog Floating-Gate Cells in Stage 2

There are two main differences between the common cells described above and the cells used on the HICANN chip. On the one hand they have to store analog values. Therefore the writing process needs to be controlled and interrupted when the set point is reached. On the other hand, there is no possibility in a standard CMOS process of placing a second gate over a floating one. An architecture different from the common one described above is required.
In total three transistors are necessary to form one cell. Their interconnected gates form the floating gate. One of the transistors is used as the output of the cell, the remaining two are necessary for writing. Since source and drain of these two transistors are shorted they work only as capacitors coupled to the floating gate. A schematic of the cells can be seen

in 3.2. This architecture, which allows for floating-gate devices in a single-poly-process, was first demonstrated in 1994, see [26].

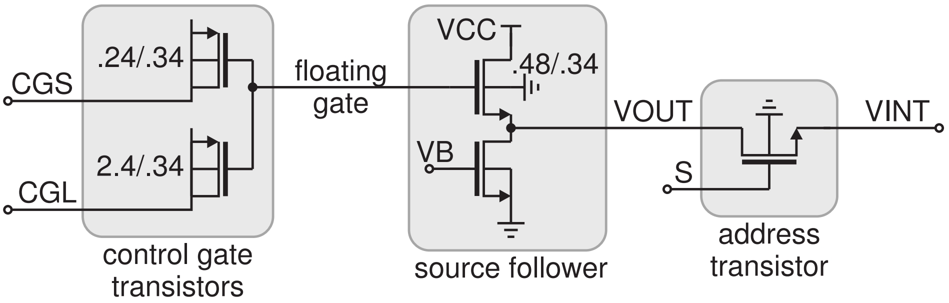


Figure 3.2: Schematic diagram of a floating-gate cell used ion the HICANN chip. The interconnection of the gates of three transistors form the floating gate. The control gate transistors are required to program the cell. The source follower converts the current of the output transistor into a voltage. The output voltage of the cell can be connected to the analog VINT bus by an addressing transistor. Taken from [34]

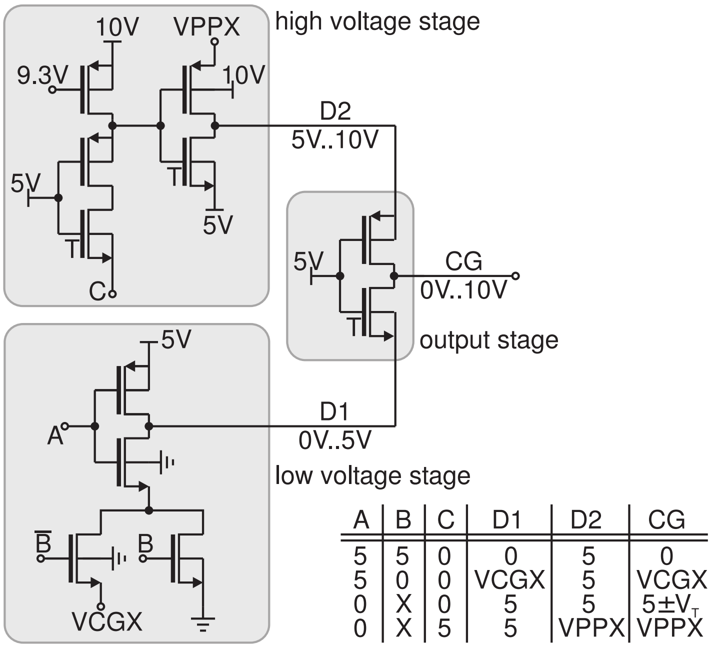
One of the programming transistors, named CGL (Control Gate Large) is ten times larger than the second one, called CGS (Control Gate Small). The difference in area causes CGL to couple much stronger to the floating gate. This is used to pull the floating gate to the necessary potentials for writing. CGS is where the tunneling currents, either charging or discharging the gate, occur, since the maximum potential differences arise there. The output transistor is an NMOS transistor. Therefore the output voltage is higher if the number of electrons trapped on the gate is less than in an uncharged state. In order to increase the output value of a cell, CGS is connected to ground and CGL is connected to a positive voltage called VPP2. This causes electrons to tunnel from the gate to CGS. Decreasing the stored value works analogously with interchanged potentials at CGL and CGS. The electrons then tunnel from CGS to the floating gate.

The cells are designed to store output voltages covering a range from 0 to 1.8V with a nominal resolution of 8 bit. To allow for reasonable writing currents over the entire range a relatively high positive potential VPP2 is required. Typically it is set to 11V is used. A voltage of 11V is rather high compared to the usual levels in microelectronics. The standard process used for production of the HICANN chip only supports 1.8 or 3.3V transistors. Therefore special high-voltage driver circuits are required to switch VPP2. A diagram of these driving circuits is shown in Figure 3.3. These circuits use triple-well-transistors[2] to allow for switching of voltages up to two twice the gate breakdown voltage of the transistors used. More detailed information concerning this issue can be found in [34].

The charging process is performed in short pulses of switching CGL to ground and CGS to VPP2 or vice versa, depending on the direction the actual output of the cell is intended to be changed. After each programming pulse the output of the cell is connected to a comparator to check whether it has reached its desired output value or whether another pulse has to be applied. The set point is provided by a 10 bit DAC[3] operating in a range between 0 and 1.8V.

[2] An additional well of n-doted silicon is enclosing the transistor, this allows bulk insulation for NMOS devices.

[3] Digital-to-Analog Converter



A	B	C	D1	D2	CG
5	5	0	0	5	0
5	0	0	VCGX	5	VCGX
0	X	0	5	5	$5 \pm V_T$
0	X	5	5	VPPX	VPPX

Figure 3.3: High-voltage driver for switching the control gates to the potentials required for programming the cells. Taken from [34].

Some parameters of the neuron are determined not by voltages but also by currents. Therefore two different types of memory cells, either with voltage or current output, are available. Figure 3.2 shows a voltage cell. The current through the output transistor is converted to a voltage by the source follower. When used on the HICANN chip the cell's output VOUT is connected to the dedicated neuron's circuit. It can also be switched on the analog bus VINT which is connected to the comparator during the writing process by the addressing transistor. For current cells the current through the output transistor is applied to a current mirror that provides two outputs. One of these outputs is connected to the neuron. The second output multiplies the current by four and can be switched by an addressing transistor to an analog bus which is terminated by a resistor. The voltage drop over this resistor is evaluated by the comparator to decide whether the cell reached its set point.

3.2 Floating-Gate Cells on the HICANN chip

3.2.1 Floating Gate Cell Array

To save area on the chip the cells are arranged in arrays of 24 columns and 129 rows, sharing resources such as DAC and comparator, address circuitry and high voltage drivers. Every second column, beginning with 0, contains voltage cells, the remaining ones are current cells. One line with its 24 entries represents the parameters for one neuron, which are 12 voltages and 12 currents. Only line 0 does not provide neuron parameters, but bias currents and adjustable voltages for other components of the system, e.g. VCCAS for the receivers of the L1 system mentioned in 2.1.4.
In every column the CGLs are connected to a common high-voltage driver and the CGSs of one line are sharing a high-voltage driver as well. The DAC providing the set point of the

cells to the comparator, which controls the charging process, is a R-2R DAC with a resolution of 10 bit. Therefore the range from 0 to 1.8V for voltage cells and 0 to $2.5\mu A$ for current cells is divided into 1024 steps. The array of floating gate cells as it is described here, containing 3096 cells, requires in total $14400\mu m^2$ of chip area. So, taking address circuitry, DAC and high voltage drivers into account, the average area required on the HICANN chip to store one parameter is about $200\mu m^2$.

3.2.2 Programming the Array

Addressing and programming of the cells in the array is performed by a controller that was written in system verilog[4] by Sebastian Millner. This controller is implemented in standard cell logic on the HICANN chip. Since the floating-gate cells are absolutely essential to use the HICANN chip and the controller is used for the first time in the prototype chip, extensive testing was necessary.

The controller assigns signals to the 26 inputs of the array. In the other direction, from the chip to the controller, information is transmitted by only one signal, the output of the comparator. In general, the array can be in "read mode" or "write mode". In read mode all column and line drivers are set to ground. In write mode the drivers set the control gate lines to VPP1, a voltage approximately in the middle of VPP2 and ground. Typically 5V are used.

There are two separate address decoders for columns and lines, both working basically as shift registers with an inverted first latch. A cell can be addressed by applying the corresponding numbers of shift pulses for its column and line number. In read mode this activates its address transistor and connects the cell's output to the comparator.

In write mode the drivers of the selected column and line are set to VPP1, as all the others. But only the selected column and the selected line are changing their output to either ground or VPP2 in case a strobe signal, triggering a programming pulse, is applied. The other cells within the same column or line as the selected cell are not affected by this programming as either their CGL or their CGS is set to 5V. So they are exposed to a maximum potential difference of 6V at their control gates, which is not sufficient for enabling a significant amount of electrons to tunnel. Only the selected cell is exposed to the full potential difference of 11V, causing it to change its value.

After a programming pulse the array is set to the read mode. Now the address transistor for the respective cell is activated, connecting the cells output to the comparator. If the set point was not reached, the write mode is activated again and another programming pulse triggered. In order to save programming time, the array has an additional feature in its addressing scheme. In write mode it is also possible to address whole columns at once, applying the programming pulse to the drivers of all lines simultaneously. After the pulse every cell is checked individually in the read mode by the comparator. A cell that reached its set point is omitted from the next programming pulse by setting the respective line driver to 5V. Figure 3.4 illustrates a writing process, omitting one cell in a column where the output value of other cells is increased.

This scheme of programming is inefficient for writing random data into the array, e.g. it is not possible to increase the value of one cell and decrease the value of another one in the same column at the same time. However, for the use in the HICANN chip this is reasonable,

[4] A hardware description language

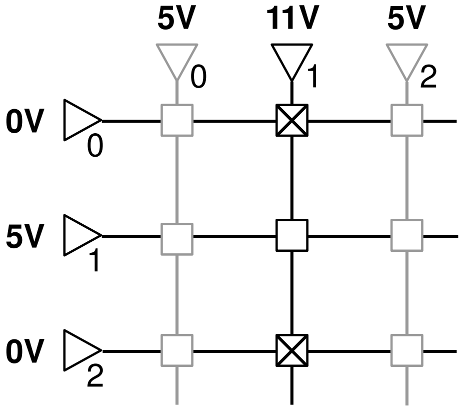


Figure 3.4: Illustration of the programming scheme used within the array. The setting of the potentials corresponds to a programming process increasing the output voltage of the cells within column 1. The cell located in line 1 has already reached its set point and is therefore omitted from the programming process by switching its CGS to 5V.

as a column contains the same parameters for 128 different neurons. In most cases basically the same value is written into all cells within a column. Only the first line, providing parameters for other circuits than neurons, needs to be programmed separately. To further save programming time there are in fact two banks of RAM. While the contents of the first one is programmed to a column, the other on can be filled with the data for the column to be written next. In fact, the final controller implemented in the HICANN chip does not support the programming of single cells described above, as in most cases a whole column needs to be written.
There are several parameters for the writing and comparison process, which have to be set correctly to allow for a sufficient compromise of writing speed and accuracy. All time differences, including the ones adjustable by the parameters mentioned below, are measured in units of clock cycles. Originally the controller was intended to run at 50MHz.

Pulse Length:

First of all, the basic length of the writing pulses for voltage and current cells are adjustable independently within a 5 bit range each. Short pulses result in longer writing times since many pulses have to be applied. This results in a larger number of comparisons to be done, which consume a significant amount of time, see *comparator read time*. Long pulses lead to short writing times, but accuracy may decrease if the amount of charge transfered during one pulse has an impact larger than 1 LSB [5] on the cell's output. This occurs especially at low values. The *accelerator* parameter discussed below represents a solution to this problem. Typical pulse duration is about 16 clock cycles.

[5] Least Significant Bit

Comparator Read-Time:

A crucial parameter for improvement of accuracy, but decreasing writing speed is the read time for the comparator. The output of the cell needs to pull the wire to the comparator, which is up to about $1800\mu m$ long, to its current value, before a reasonable comparison can be done. The *comparator read-time* parameter determines the delay between the activation of the address transistor of the cell and the sampling of the comparator. This is especially important for the current cells at low values. In this case currents of a few μA or less have to load the capacitance of the wire before a correct voltage drop at the resistor can be measured. Since comparison is necessary after every single charging pulse and a typical value for the comparator read-time is 64 clock cycles, this has a significant impact on the total writing time.

Accelerator:

The higher the actual value of the cell, the less electrons are tunneling per pulse during programming. To accelerate the writing of high values the controller uses a special speed-up function. If the cell is not ready after an adjustable number of pulses, the controller doubles the duration of a single pulse. This *accelerator* parameter needs to be set carefully. If pulse length is increased too rapidly, accuracy is affected. So a balance between *pulse width* and *accelerator* must be found to achieve maximum performance. This parameter was typically set to 32.

Max Cycle:

There is a maximum number of write pulses the controller applies to a single cell in case it does not reach its set point. If this number, adjustable by the parameter *max cycle* within a 7 bit range, is exceeded, the address of the cell is written into an error register and the controller proceeds with the next cell to be written next. This results in a timeout for the writing process if a cell is damaged or the set point cannot be reached due to insufficient settings of other parameters.

3.3 Testing the Floating-Gate Prototype Chip

A prototype chip containing an array of 24 × 129 cells to test the floating gate memory was designed and submitted to an IMEC MPW run earlier by Sebastian Millner and tested for this diploma thesis. At the prototype chip the VINT bus is accessible by a bond pad so the output of every cell can be measured. Furthermore the outputs of three voltage cells and three current cells are directly accessible by bond pads. To permit direct measurements of a larger number of cells, the chip has 440 probe pads. Unfortunately they are located so close to each other that it is hardly possible to connect the needle of a wafer prober to a dedicated one. There is a high risk to contact more than one pad.

The array implemented in the prototype chip is the same used on the HICANN chip, despite one discrepancy. As mentioned in 3.1.2 the voltage drop caused by the output of a current cell crossing a resistor is sampled by the comparator to check the current value of the cell. In a first version of the array, the output of the cells was multiplied by two before connected to a $400k\Omega$ resistor. During programming much time is consumed by waiting for the small output current to load the capacitance of the wire leading to resistor and comparator. To increase the programming speed, the multiplier of the current mirror was changed to four. Unfortunately, for the array implemented in the prototype chip the resistor was not adapted to this change. In the HICANN chip a $180k\Omega$ resistor is used instead. The result of this is a reduced maximum output current of the cells in the prototype chip. The maximum output current of the cells is limited to approximately $1.1\mu A$. A range from 0 to $2.5\mu A$ is desired for the final system and will most likely be available with the corrected resistor.

3.3.1 Experimental Setup

The experimental setup for testing of the FG chip is very similar to the one for the repeaters, see 3.5. The chip was directly bonded at a PCB which is connected to the FPGA development board mentioned before.

FPGA

The structure of the software is analog to the one used in the experimental setup for the L1 prototype, see 2.3. Again a system consisting of microblaze processor and a vhdl module, connected to each other by the PLB, is used. The vhdl module instantiates the controller for the floating gate array, written in verilog[6], which is implemented in the HICANN chip. The Xilinx EDK software supports verilog as well as vhdl. Nevertheless it involves some effort to build a project with mixed vhdl and verilog source codes. The controller has an OCP[7] interface to obtain the data that is designated to be written into memory cells. The vhdl code basically emulates an OCP interface to send data from the C program, running in the microblaze processor, to the controller.

PCB

The PCB provides the necessary supply voltages for the chip. It is designed with 5V logic level IO cells. However, the exp-connector of the FPGA supports only 2.5V or 3.3V logic. So a possibility to shift logic levels has to be implemented on the PCB.

In case of the single output of the chip, it is directly connected to the FPGA by a serial resistor of $15k\Omega$. In case of a high signal from the chip the voltage at the FPGA pin increases until its ESD protection turns on. The serial resistor is limiting the resulting current to prevent damage.

In case of the 26 inputs of the prototype chip active electronics is required to increase the voltage level of the signals. A long search for suitable level shifting buffers did not lead to a satisfying result. Therefore it was decided to make use of discrete devices for this purpose. A pull-up resistor and a NMOS transistor to pull down are connected to the pins of the chip. The FPGA controls the NMOS transistor. With this solution all output signals of the FPGA have to be inverted. Despite the range of the signal frequencies controlling the floating gates of only a few MHz, the problem of signal quality was underestimated at first. For the pull-up resistor $2k\Omega$ were chosen originally, but it turned out that this results in a too small slope of rising edges. When performing experiments with high programming frequencies short pulses did not even cross the threshold of the chips' input buffers.

[6] A hardware description language

[7] Open Core Protocol

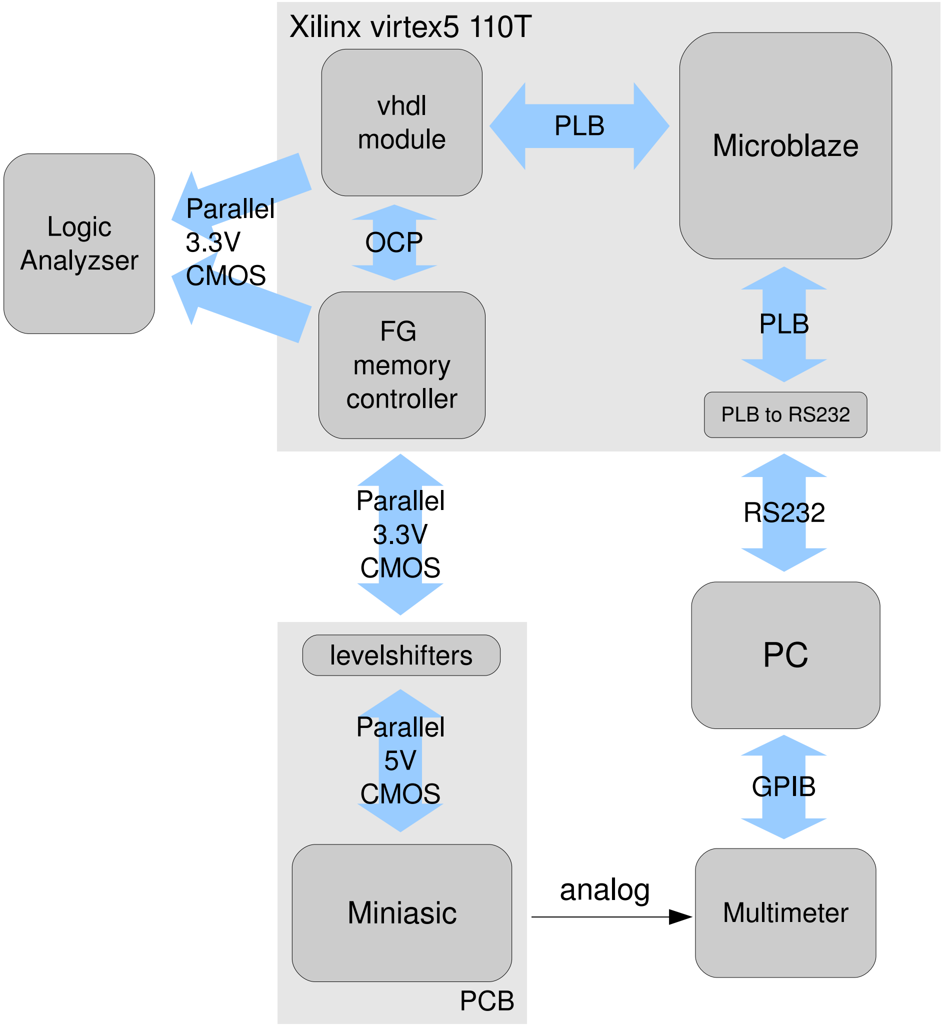


Figure 3.5: Experimental setup for measurements with the floating-gate prototype chip. The controller for the cells is implemented in the FPGA and programs the floating-gate cells on the chip. The values which are to be written are provided by a program running in the microblaze processor. The logic analyzer is used only for debugging. For automated measurements the output values of the cells can be measured with the multimeter which is connected to a PC running LabView software. The measurements are triggered via RS232 by the C program after a new value has been programmed.

Once aware of this problem a value of 470Ω for the pull-up was evaluated to provide sufficient performance of the level shifting circuit. Another unexpected problem with the digital inputs was discovered. Due to the layout of the PCB as well as the FPGA board a lot of crosstalk occurred, strong enough to overcome the threshold of the NMOS transistor used in level shifters. So additional pulses were triggered and operation of the chip was disturbed. To eliminate this, $2k\Omega$ resistors to ground were added directly at the gates of the level shifting transistors for termination. A second measure against crosstalk was to choose another setting for the output drivers of the FPGA, it provides several possibilities to set their behavior. The strength of the drivers is adjustable in steps of 2mA in a range from 2 to 24mA. Furthermore the slope of the outputs can be set to a "slow" mode. A setting of just 2mA with "slow" slope in combination with the resistors leads to a proper signal quality.

Automatic Measurements

First measurements were done simply with a multimeter or an oscilloscope. In the following a setup capable of performing measurements automatically to obtain statistics was required. This was realized with help of the LabView software. A computer running LabView was connected to the FPGA development board via RS232 and a multimeter was connected via GPIB to the PC. The sequence of the measurement is controlled by the C code running in the microblaze processor. Every time a new value has been programmed a trigger signal is sent via RS232 to the PC where LabView starts a measurement with the multimeter. After the multimeter has finished sampling it sends a signal via the same pathway back to the FPGA, so the next cell or value can be programmed and a new measurement triggered. The values measured are written into a file, the only analysis done directly in LabView is determining the standard deviation of all values in case the same value is written repeatedly. This allows a quick evaluation of the programming accuracy, helping to improve parameter settings.

3.3.2 ESD Protection Related Problems

Also in case of the floating gate chip ESD was an important issue. The ESD protection structures were designed by Dr. André Srowig as normal diodes to supply and ground. Some problems occurred because of the multiple number of supply voltages, especially since one of them is much higher than usual. As it is pointed out in [30] chips with multiple supply voltages need ESD structures from every pin to every supply voltage separately to provide reliable protection. In case of the floating gate chip this would require an enormous design effort and a considerable amount of area on the chip due to the four independent supply voltages. The digital 5V inputs and outputs are protected using reversed diodes to VPP1 and ground. The same is done for the analog outputs of floating gate cells. The output transistors of the voltage cells are 3.3V NMOS and therefore protected from ESD events. In case of the current mirrors for the current cells' outputs 1.8V PMOS transistors were used. Their gate breakdown limit is below 5V. In this case the ESD protection cannot be expected to work. Nevertheless no problems with these pins have been observed.

Another aspect are the high voltages at VPP2, which is about 11V, and at VBP, which is typically VPP2 minus 0.7V. Because of limited breakdown voltages of the transistors it is not possible to directly install reversed diodes between VBP or VPP2 to ground. This is avoided by installing the diode to VPP1 instead of ground. Due to these particular ESD structures in the chip a correct power-up sequence is necessary. This topic is discussed below in 3.3.3.

Furthermore there was a mechanical problem with the IO cells of the FG chip. The distance between the bond pad and the active electronics of the ESD structures is just $4\mu m$. To prevent the bonder from destroying the diodes, the bond wires where placed close to the opposite border of the pad. Therefore accidental contacts to the scripeline became more likely. This happened several times. In most cases rebonding was successful, whereas one chip got permanently damaged.

3.3.3 Power-Up Sequence

Initially a lot of problems occurred, when the power supply of the PCB was switched on without special care. The high voltage and the ESD diodes between VPP2 and VBP respectively and VPP1 were the cause of damage during power-up. Once aware of this problem the necessary constraints to avoid damage were derived from the schematic diagram, and appropriate measures were implemented. VDD and VP are not causing any difficulties since they are well below any breakdown voltages of the devices used, but there are some restrictions concerning VPP2 and VPP1. VPP2 must not be switched on before VPP1. Otherwise the breakdown voltage of the diode between VBP and VPP2 is exceeded. Furthermore many transistors of the high voltage drivers would be destroyed by gate breakdown.

On the other hand VPP1 is not allowed to be switched on before VPP2 is larger than 5V. In this case the blocking capacitors of VPP2 and VBP are rapidly charged through their ESD diodes to 5V. The diodes may withstand an ESD event, but the blocking capacitors have 6.8nF capacitance and therefore the total amount of energy transiting the diode is much larger, capable of destroying it. In the first chip that was connected to the power supply the diode between VBP and VPP1 was, most likely, destroyed in exactly this manor. Obviously VPP1 and VPP2 need to be increased in parallel up to about 5V, afterwards VPP2 has to be further increased. Switching off needs to happen in the exact opposite order. Therefore it is hardly possible to guarantee a reliable power-up and -down sequence with simple RC-filters. For the prototype chip the problem was solved by slowly increasing or respectively decreasing the main supply voltage. VPP1 and VPP2 are derived from this main supply voltage, with linear voltage regulators on the PCB. To protect the chip against power failure or accidental switching off some passive elements were added. A zener diode prevents VPP1 from being larger than VPP2 and additional capacitor prevent VPP1 from decreasing faster than VPP2.

To use the correct power up sequence and protection against failure of single supply voltages is especially important for the final system. This becomes more complicated as every reticle can be switched on and off separately. Every single reticle must follow the correct procedure.

3.4 Results

As mentioned before, the experiments performed with the floating-gate chip focused primarily on testing the addressing circuitry of the array and the controller code.

The functionality of the voltage cells itself was already tested extensively by Dr. André Srowig and Jan-Peter Loock, see [34, 20]. Therefore it was not urgent to test the performance, for instance accuracy or programming speed, systematically and comprehensively.

The experiments were carried out only weeks before the tape out of the first prototype of the HICANN chip. So digital logic, such as the controller code, was the only aspect which could be adapted to the results of the tests at this stage of the development. In the following the findings from testing the digital part, addressing circuitry and controller, are discussed. Later several measurements of basic analog characteristics are presented.

An extensive search for a parameter setup that provides best programming time to accuracy ratio, with respect to the requirements for the HICANN chip, is desirable. Due to limited time and technical problems described in 3.4.2, it was not possible to realize this during this diploma thesis. The setup for fully automated measurements described in 3.3.1 was therefore used only in few cases.

3.4.1 Testing of Memory Array and Controller

At first, some simulations were required to debug the C code and the vhdl code of the experimental setup. It needed to be confirmed that the parameter settings as well as the values to be programmed to the cells are transmitted correctly via the OCP interface to the controller. After the input of the controller was confirmed to be correct the next step was to investigate the output of the controller. This was addressed initially in simulations and later with the complete setup, using a logic analyzer to watch the signals. This was done in collaboration with Sebastian Millner, who developed the controller code and is therefore more familiar with its operation.

First attempts to write a cell failed due to an error in the addressing logic, preventing the controller to address the column with number 0. Having fixed this problem, the output of the controller seems to be correct. Nevertheless it was not possible to change the output of any accessible cell. No reason for this, neither in the software nor on the PCB was found. In the end a new PCB with a new chip was taken into operation. After replacing the chip writing of the cells worked. However low values were not written correctly. It turned out that the level shifters of the PCB were responsible for this. As described in 3.3.1 the pull-up resistors were too large at first, cutting the short pulses which are especially important at low values. After exchanging the pull-up resistors basic programming worked and the ranges of the parameters described in 3.2.2 needed to be checked.

When writing with a set point of 1023 to a cell the corresponding value of 1.8V was not reached at the output, despite a *pulse width* setting of 32 and *max cycle* set to its maximum. A second writing process to the same cell with the same setting was able to increase the output level significantly. This indicates that 128 pulses are not sufficient in this case. As a result the range for max cycle was enlarged from 7 to 8 bit. In general the operation of the cells worked better with a clock frequency below 10MHz, typically 7.8MHz have been used. Hence the possibility of scaling down the clock frequency for the writing processes in relation to the OCP clock was added by Sebastian Millner.

3.4.2 Performance of the Floating-Gate Cells

Only few measurements on the cells have been performed, most of them primarily to test the controller rather than to gather precise information concerning the cells' performance. More detailed measurements and evaluations concerning the cells' characteristics can be found in [20].

The performance of the cells was far below expectations during the measurements presented in the following. However, a systematic optimization of the parameters has never been performed. This is related to a mechanical problem. It was not possible to bond the pin connected to the VINT bus, on the only chip that functioned properly, aside from this aspect.

Despite many attempts with different settings of the bonder, for an unknown reason the bond wire did not stick on this pad of the chip. Lack of time prevented testing more chips. The VINT pad is the only way to access the output of all the cells on the chip. Without this pad no variations from cell to cell can be taken into account, a very important aspect for an evaluation of the cells' performance.
Likewise, an extensive tuning of the parameters, based on data from a maximum of three cells, does not lead to a generally usable result. In the following some results of quantitative experiments performed with the directly accessible cells are presented. All measurement have been performed with a clock frequency of 31.25MHz for the OCP interface and an operating frequency of 7.8MHz for the floating-gate array.

Voltage Cells

The output of a voltage cell during a writing process can be seen in Figure 3.6. The cell is programmed with a set point of 0 at first, this results in an output voltage of about 70mV. In the process shown its value is increased, the given set point was 1023, which leads to an output voltage of about 1.48V. This writing process was aborted since the number of pulses specified by the *max cycles* parameter, in this case 128, was applied, before the designated output value of 1.8V was reached. The maximum output voltage observed in experiments with the prototype chip is approximately 1.6V. In Figure 3.7 an enlarged section of the writing process displayed in Figure 3.6 is shown. The effect of the *accelerator* parameter, an increase of the pulse width during the programming process, can be seen.

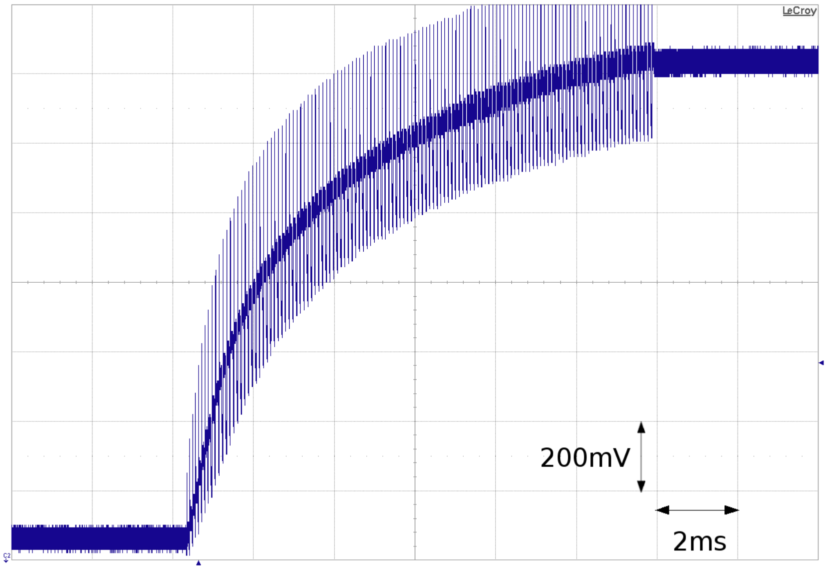


Figure 3.6: Programming process of a voltage cell. The output of a voltage cell is measured with an oscilloscope during the process. The output value of the cell changes from 70mV to about 1.48V. The set point of 1023 which corresponds to an output value of 1.8V is not reached within the 128 programming pulses specified by the *max cycle* parameter. Therefore the precess is interrupted.

Several experiments concerning the repeat accuracy have been done by writing a cell repeatedly to a certain value. In between, the cell is written to 0. A number of 200 measurements

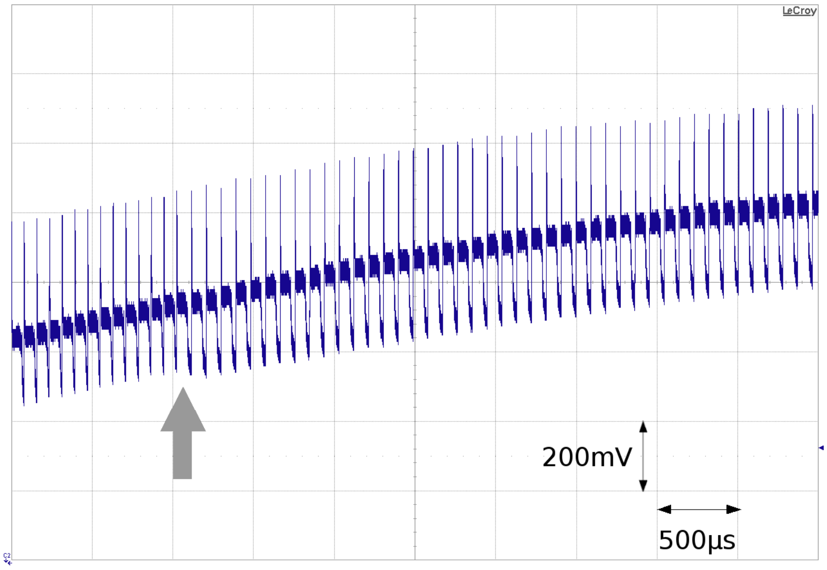


Figure 3.7: Section of the programming process displayed in Figure 3.6. The effect of the *accelerator* parameter is visible: The width of the programming pulses is doubled at the point marked by the arrow.

with a set point of 512 and a *pulse width* of 32 leads to mean value of 889.5mV, the standard deviation of a single value is 8mV. For a set point of 512 an output of 900mV is expected, but the absolute value depends on the value of VB in the output source follower, see Figure 3.2. VB was not tuned to reproduce exact absolute values. The standard deviation of the single values is more interesting. In order to achieve a nominal 8 bit resolution in the range from 0 to 1.8V a standard deviation significantly below 7mV is required. This is not reached for the measurement described above, but a *pulse width* of 32 is rather large and might be the limiting factor.

Current Cells

The current cells generate a current at their output in a range from 0 to $1.1\mu A$, this is too small to be measured reliably with a normal multimeter. To simplify measurements the prototype chip has current mirrors, multiplying the output of the current cells connected to IO pads by 40. Still the accuracy of the multimeter needs to be taken into account. A resistor of $22k\Omega$ is used in series to the multimeter when measuring the currents to simulate a load for the current output. The currents mentioned in the following refer to the multiplied output of the chip.

In Table 3.1 the results for repeated writing of the same value with two different parameter setups are compared. In the first case the cell was directly programmed with a normal parameter setting, where all parameters are in the middle of their respective adjustment ranges. Especially the width of the charging pulses, which is critical for accuracy, is set to 32. In order to achieve a reduced deviation of the output values a modified programming scheme is tested. In the second setup a sequence of two writing processes is used. First the cell is written to a set point of 490 with a reduced pulse width of 16. In a second step the pulse width is further reduced to 4 and the cell is written to the target set point of 512. In both

	Set Point	Pulse Width [clock cycles]	Mean Output Current [μA]	Error on the Mean Value [μA]	Standard Deviation [μA]
Setup 1	512	32	23.4	0.34	1.5
Setup 2	512	16 + 4	21.9	0.14	0.6

Table 3.1: Results from measuring the accuracy of the cell's output. The data shown is based on 20 independent programming processes with a set point of 512. In setup 1 the cell is directly written to the target set point of 512. The width of the programming pulses is 32 clock cycles. In order to obtain a smaller deviation of the resulting output voltages a sequence of two writing processes is used in setup 2. First the cell is written to a set point of 490 with a pulse width of 16. In a second step it is programmed to the target value of 512 by programming pulses of only four clock cycles in width. Note the difference in the resulting mean values. The same cell was used in both setups.

cases the mean value is based on the result of 20 independent programming processes.
In the second case in fact a lower standard deviation is reached. Here the resolution is most possibly limited by the accuracy of the multimeter. According to the manual of the multimeter ([11]) it is limited to $1\mu A$ for absolute measurements. The relative accuracy for measurements carried out shortly after one another is most likely better, but still not sufficient to test whether the cells reach an 8 bit resolution, which requires to distinguish steps of $0.18\mu A$ reliably. It is eye-catching, that the mean value of all measurements is different for both setups. Even if resolution of the multimeter is limited, the shift seems to be significant, when the errors on the mean value are taken into account.

This phenomenon, an increase of the mean value when the programming process consists of two steps was also observed in measurements involving voltage cells. This cannot be explained with a change in VB, as both measurements were done within minutes. If this happened at high set points, it would indicate that the first writing process did not reach the set point, but was interrupted by the timeout. When a second writing process is applied the additional number of pulses would be able to further increase the programmed value. However, in case of the programming processes with setup 1 there was no timeout, which would have lead to an entry in the list of failed cells. Furthermore, this cannot happen while writing a set point of 512, as long as writing of larger values is possible with the same parameters. Currently there is no explanation for this behavior and further investigations are required.

In Figure 3.8 measurements of a current cells' output over the whole accessible range is shown. For every data point three measurements have been done, the error bars represent the standard deviation. The output covers in mostly linear progression the whole range up to $45\mu A$. At low values an offset is visible. The average standard deviation of the single data points is $1.3\mu A$ which complies with the standard deviation observed in the measurement of the repetitive writing of the set point 512 shown above. The *pulse width* is also set to 32, as in setup 1 of the previous measurement.

To evaluate the cells' performance appropriately, a large number of additional measurements, especially with many different cells is required. The results need to be evaluated in terms of differential and integral linearity, not only by the standard deviations of measure-

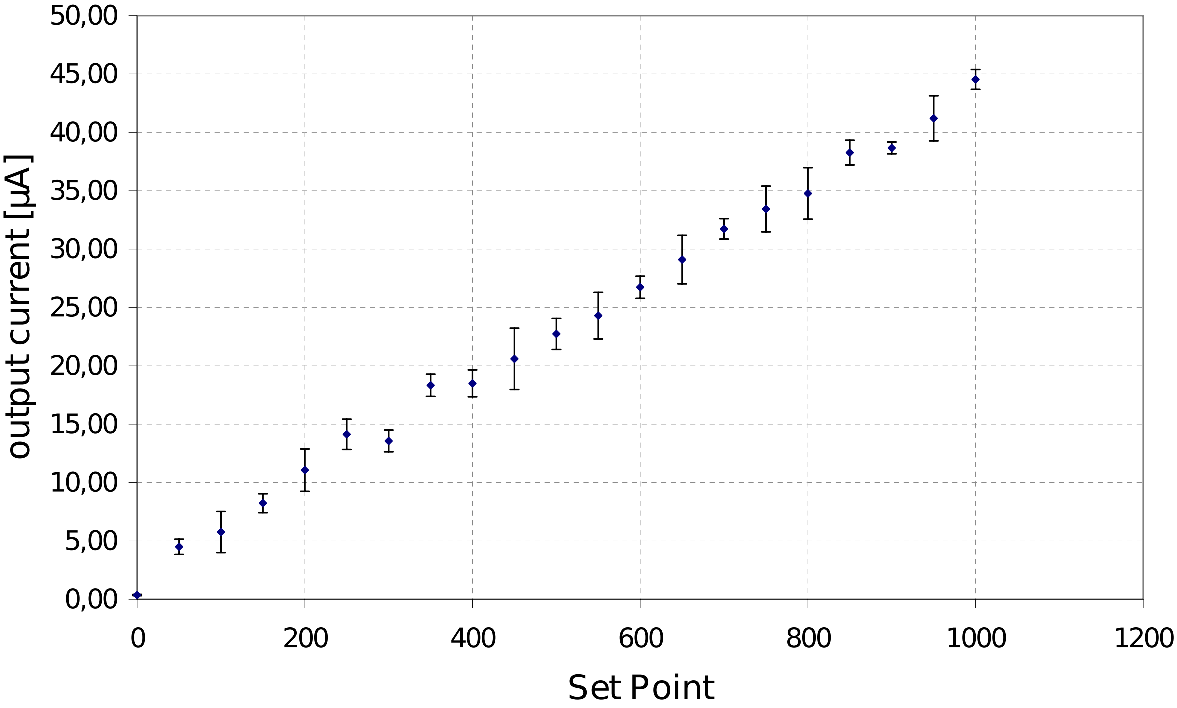


Figure 3.8: The output of a current cell was measured over the whole range of set points from 0 to 1000. The error bars denote the standard deviation over three repetitions.

ments done for a certain set point.

4 Conclusion

This thesis included the designing and testing of the L1 prototype chip and secondly performing measurements with the floating-gate prototype. In the following some conclusions drawn from the results are stated.

The L1 Repeaters

Performing experiments with the L1 prototype chip has been severely affected by various technical problems. Repeatedly employing new chips and developing workarounds for malfunctions of the chips. Aside from evident damage, especially ESD problems may have lead to unrecognized consequences. Therefore the significance of most results is limited. It is extremely unlikely that any damage improves the performance of the repeaters. Thus, have to be most of the results have to be seen rather as a lower bound than an absolute value. Nevertheless, the principle ability of sending packets between repeaters at a reasonable data rate was confirmed. The transmission is quite robust against variations in the analog parameters. This simplifies the usage of the L1 system.
Only concerning crosstalk some uncertainties remain. Further investigation is required. Especially a means to simulate the L1 wires more precisely would be important to improve the crosstalk councilation circuitry. With realistic activities on the wires, the error rates might be small enough to allow for experimenting with the Stage 2 hardware without a modification in crosstalk councilation.

The question why the L1 system does not reach the data rate originally planned cannot be answered conclusively. It is certain that more than one aspect in the repeaters is limiting the bandwidth to less than 2.0GBit/s. Obvious factors are that the sender has problems generating the preemphasis at the stop bit at high data rates. The receiver is also limited as was shown in measurements with the AWG. However, this does not explain all errors which occurred. As a consequence of the results presented here, capacities for the supply voltage in the repeaters for the HICANN chip were added. This may improve performance of the L1 system significantly.

An additional receiver at the end of a horizontal wire would have been interesting to see up to determine the maximum data rate at which transmission via a 15mm wire, discontinued by a pass transistor, works. Even with the L1 prototype chip as it is many more interesting measurements are possible. However, at some point it necessary to proceed with tests on the original setup. One can only transfer the results from a prototype chip with four repeaters to a large and complex chip such as HICANN to a certain degree. According to the results presented here, it seems likely that the transmission of the neural event with the L1 system will be sufficient in the Stage 2 system. It cannot be anticipated which exact data rate will finally be achieved.

The Floating-Gate Memory Cells

The quantitative measurements carried out with the floating-gate cells were not satisfying. To properly characterize the behavior of the cells a number of additional experiments are of need. This was not possible due to limited time and technical problems. Especially access to all cells on the floating-gate prototype chip with a functioning pin to access the VINT bus would have been useful. For instance, the variation of the output voltage for different cells programmed to the same value is a decisive aspect which could not be investigated so far. Therefore whether the performance of the cells is sufficient for the correct operation of the HICANN chip could not be proven. The flexibility of the controller providing many parameters which can be adapted to the characteristics of the cells and to the requirements in accuracy and writing speed should allow for reasonable use of the cells.

Nevertheless, the experiments performed with the prototype chip lead to some important results. The necessity of a correct power-up sequence became evident. Now this issue can be taken into account for building the experimental setup for a single HICANN prototype as well as for the complete Stage 2 system. The tests with the controller also lead to some relevant results. Several bugs in the code were found and the addressing of the cells in the array was proven to function correctly. Again working with the chip was often affected by technical problems. Bonding the chip directly to the PCB was not the best approach in case of this chip. For every new chip required a new PCB and soldering a number devices onto the PCB was necessary. In total about 100 smd[1] devices, most of them used for shifting the logic levels, including the 120 pin connectors mating the ones on the FPGA board, were required. The use of 3.3V IOs at the floating-gate prototype chip would have simplified working with the chip significantly.

Undoubtedly, the development of the Stage 2 hardware is an ambitious project. A number of technical problems arise when designing and establishing a neuromorphic system of this dimension. Simulations prove that solutions to these problems have been found. Having accomplished this, the next step is to verify the functionality of the circuitry in silicon. The prototype chips discussed in this thesis were the first components of Stage 2 to be tested. Although the prototypes did not meet all expectations, their basic functionality could be confirmed. Therefore the technical evaluation of the first HICANN chips can begin in the near future. Correct operation of the HICANN chip would represent a decisive element in the progress of Stage 2 development.

[1] Surface Mounted Device

Bibliography

[1] The international technology roadmap for semiconductors. www.itrs.net, 2007.

[2] Johannes Bill. Self-stabilizing network architectures on a neuromorphic hardware system. Diploma thesis (English), University of Heidelberg, HD-KIP-08-44, 2008.

[3] R. Brette and W. Gerstner. Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *J. Neurophysiol.*, 94:3637 – 3642, 2005.

[4] Daniel Brüderle. *Neuroscientific Modeling with a Mixed-Signal VLSI Hardware System.* PhD thesis, Ruprecht-Karls University, Heidelberg, 2009.

[5] Laure Buhry, Sylvain Saïghi, Wajdi Ben Salem, and Sylvie Renaud. Adjusting neuron models in neuromimentic ics using the differential evolution algorithm. In *Proceedings of the 4th International IEEE EMBS Conference on Neural Engineering*, 2009.

[6] Anantha Chandrakasan, William J. Bowhill, and Frank Fox. *Design of High-perfomance Microprocessor Circuits.* IEEE Press, 2001.

[7] Rodney J. Douglas and Keavan A.C. Martin. Mapping the matrix. the ways of neocortex. *Neuron*, 56(2):226–238, October 2007.

[8] FACETS. Fast Analog Computing with Emergent Transient States – project website. `http://www.facets-project.org`, 2009.

[9] J. Fieres, J. Schemmel, and K. Meier. Realizing biological spiking network models in a configurable wafer-scale hardware system. In *Proceedings of the 2008 International Joint Conference on Neural Networks (IJCNN)*, 2008.

[10] Andreas Grübl. *VLSI Implementation of a Spiking Neural Network.* PhD thesis, Ruprecht-Karls-University, Heidelberg, 2007. Document No. HD-KIP 07-10.

[11] Hewlett Packard. *Multimeter HP 34401A*, October 1992.

[12] Alan Lloyd Hodgkin and Andrew F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *J Physiol*, 117(4):500–544, August 1952.

[13] Scott A. Huettel, Allen W. Song, and Gregory McCarthy. *Functional Magnetic Resonance Imaging.* Sinauer Associates, 2004.

[14] E. R. Kandel, J. H. Schwartz, and T. M. Jessell. *Principles of Neural Science.* McGraw-Hill, New York, 4 edition, 2000.

[15] Bernhard Kaplan. Self-organization experiments for a neuromorphic hardware device. Diploma thesis (English), University of Heidelberg, HD-KIP-08-42, 2008.

[16] D. Khang and S. M. Sze. A floating -gate and its applications to memory devices. *The Bell System Technical Journal*, 1967.

[17] M. Lenzlinger and E. H. Snow. Fowler-norheim tunneling into thermally grown $si0_2$. *Journal of Applied Physics*, 1969.

[18] M. Anthony Lewis, Ralph Etienne-Cummings, Avis H. Cohen, and Mitra Hartmann. Toward biomorphic control using custom aVLSI chips. In *Proceedings of the International conference on robotics and automation.* IEEE Press, 2000.

[19] Shih-Chii Liu, Jörg Kramer, Giacomo Indiveri, Tobias Delbrück, and Rodney Douglas. *Analog VLSI: Circuits and Principles.* The MIT Press, 2002.

[20] Jan-Peter Loock. Evaluation of a floating gate memory cell in single-poly umc 180 nm cmos-process for implementations in neural networks. Diploma thesis (German), University of Heidelberg, 2006.

[21] C. A. Mead. *Analog VLSI and Neural Systems.* Addison Wesley, Reading, MA, 1989.

[22] Carver A. Mead and M. A. Mahowald. A silicon model of early visual processing. *Neural Networks*, 1(1):91–97, 1988.

[23] Abigail Morrison, Carsten Mehring, Theo Geisel, Ad Aertsen, and Markus Diesmann. Advancing the boundaries of high connectivity network simulation with distributed computing. *Neural Comput.*, 17(8):1776–1801, 2005.

[24] Eric Müller. Operation of an imperfect neuromorphic hardware device. Diploma thesis (English), University of Heidelberg, HD-KIP-08-43, 2008.

[25] Thomas Netter and Nicolas Franceschini. A robotic aircraft that follows terrain using a neuromorphic eye. In *Conf. Intelligent Robots and System*, pages 129–134, 2002.

[26] Katsuhiko Ohsaki. A single poly eeprom cell structures for use in standard cmos processes. *IEEE Journal of Solid-State Circuits*, 29(3), March 1994.

[27] Stefan Philipp. *Design and Implementation of a Multi-Class Network Architecture for Hardware Neural Networks.* PhD thesis, Ruprecht-Karls Universität Heidelberg, 2008.

[28] Sylvie Renaud, Jean Tomas, Yannick Bornat, Adel Daouzli, and Sylvain Saïghi. Neuromimetic ICs with analog cores: an alternative for simulating spiking neural networks. In *Proceedings of the 2007 IEEE Symposium on Circuits and Systems (ISCAS2007)*, 2007.

[29] J.J. Sanchez and T.A. DeMassa. Review of carrier injection in the silicon/ silicon-dioxide system. In *IEEE Proceedings G: Circuits, Devices and Systems*, volume 138, pages 377–389. IEEE Press, 1991.

[30] Hossein Sarbishaei. *Electrostatic Discharge Protection Circuit for High-Speed Mixed-Signal Circuits.* PhD thesis, University of Waterloo, 2007.

[31] J. Schemmel, D. Brüderle, K. Meier, and B. Ostendorf. Modeling synaptic plasticity within networks of highly accelerated I&F neurons. In *Proceedings of the 2007 IEEE International Symposium on Circuits and Systems (ISCAS'07).* IEEE Press, 2007.

[32] J. Schemmel, J. Fieres, and K. Meier. Wafer-scale integration of analog neural networks. In *Proceedings of the 2008 International Joint Conference on Neural Networks (IJCNN)*, 2008.

[33] J. Schemmel, A. Grübl, K. Meier, and E. Muller. Implementing synaptic plasticity in a VLSI spiking neural network model. In *Proceedings of the 2006 International Joint Conference on Neural Networks (IJCNN'06)*. IEEE Press, 2006.

[34] André Srowig. Analog floating gate memory in a 0.18μm single-poly cmos process. Internal FACETS documentation, 2007.

[35] A. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42:230–265, 1937.

[36] Holger Zoglauer. Development of a wafer scale integration prototype. Diploma thesis (German), University of Heidelberg, HD-KIP-09-28, 2009.

Acknowledgements

Finally I want to express my gratitude to everyone who supported this work, especially:

Prof. Dr. Karlheinz Meier for supervision and the friendly admission in the Electronic Vision(s) group.

Prof. Dr. Peter Fischer for the second opinion.

Dr. Johannes Schemmel for supervision, and a lot of helpful and interesting discussions on technical issues.

Sebastian Millner for help with the cadence Software and collaboration on the floating-gate cells.

Dr. Andreas Grübel for help during chip design.

Dr. Stefan Philipp for support in any questions related to FPGAs and VHDL.

Ralf Achenbach for help in the clean room, especially for the bonding of the L1 prototype chip.

Simon Friedmann for support during learning C and Python.

Dr. Daniel Brüderle for help concerning LaTeX and general aspects of thesis writing.

Markus Dorn for communication to the IMEC and assistance during tape out of the chip.

Dan Husmann de Oliviera for support with the Allegro software during PCB design.

All who helped with proof-reading.

Christine Engeland for general support, endless proof-reading and massive cake supply.

All members of the Vision(s) group, for great teamwork and a better not to mention number of thrilling tabletop soccer matches.

Erklärung:

Ich versichere, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Heidelberg, den 03.08.2009

....
(Unterschrift)