

RUPRECHT-KARLS-UNIVERSITÄT HEIDELBERG



Ranjeet Kuruvilla
KIRCHHOFF-INSTITUT FÜR PHYSIK

Design and Implementation of an User-Friendly
Hardware-Setup for a Chip-Based Neuromorphic
System

Diplomarbeit

HD-KIP-11-28

Fakultät für Physik und Astronomie
Ruprecht-Karls-Universität Heidelberg

Diplomarbeit
im Studiengang Physik
vorgelegt von
Ranjeet Kuruvilla
aus Kurichy, Kerala, Indien

Januar 2011

Entwicklung und Aufbau einer anwenderfreundlichen Hardwareumgebung für ein neuromorphes Chipsystem

Die Diplomarbeit wurde von **Ranjeet Kuruvilla**
ausgeführt am

KIRCHHOFF-INSTITUT FÜR PHYSIK
RUPRECHT-KARLS-UNIVERSITÄT HEIDELBERG

unter der **Betreuung von**
Prof. Dr. Karlheinz Meier

Design and Implementation of an User-Friendly Hardware-Setup for a Chip-Based Neuromorphic System

The presented thesis describes the development of the system *DajaSittah*. It has the purpose to implement the VLSI ASIC *Spikey*, a neuromorphic ASIC that emulates a neuronal network. The Printed Circuit Board allows its user to set up experiments with up to four *Spikey*-ASICs while the user can monitor the membrane-potentials of each ASIC via a high-frequency-connector. *DajaSittah* contains of two PCBs named *Daja* respectively *Sittah*: The ASICs are placed on *Daja* while the other electronic devices are placed on *Sittah*. The System connects those four ASICs with a Virtex5-FPGA via 40 differential transmission-lines, which transmit data with a data rate of up to 800MBit/s. In order to control the system VHDL-Code has been implemented for serial communication between FPGA and the specific electronic hardware on *Sittah*. This implementation allows the user to control the system via a PC. *DajaSittah* can be used to set up experiments for demonstration and teaching purposes. It therefore provides an economic and handy simplification of the current hardware.

Entwicklung und Aufbau einer anwenderfreundlichen Hardwareumgebung für ein neuromorphes Chipsystem

Die vorliegende Arbeit beschreibt die Entwicklung des Systems *DajaSittah*. Seine Aufgabe besteht darin den VLSI ASIC *Spikey* zu betreiben, ein neuromorpher ASIC zur Emulation eines neuronalen Netzes. Die Platine ermöglicht es Experimente mit bis zu vier *Spikey*-ASICs zu betreiben, während der Nutzer die neuronalen Membranspannungen über eine Hochfrequenz-Buchse beobachten kann. *DajaSittah* besteht aus zwei Platinen namens *Daja* bzw. *Sittah*: Die ASICs befinden sich *Daja*, während die elektronischen Bauteile auf *Sittah* platziert wurden. Das System verbindet diese vier ASICs mit einem Virtex5 mittels 40 differentiellen Leitungen, welche Daten mit einer Datenrate von bis zu 800MBit/s übertragen. Zur Ansteuerung des Systems musste VHDL-Code zur seriellen Kommunikation zwischen FPGA und den einzelnen elektronischen Bauteilen entwickelt werden. Dieser Code erlaubt es dem Nutzer das System über einen PC. *DajaSittah* soll später dazu eingesetzt werden Experimente zu Lehrzwecken und Demonstrationen durchzuführen. Damit stellt es eine kostengünstige und handliche Vereinfachung der bisherigen Hardware da.

Inhaltsverzeichnis

1. Einleitung	1
2. FACETS Hardware	3
2.1. FACETS Stage1 Hardware chip	3
2.1.1. Spikey Daten und mehrere Spikeys in einer Kette	3
2.2. Recha	5
2.2.1. Hochfrequenz-Buchsen und Analog-Digital-Converter	5
2.2.2. Digital-Analog-Converter	6
2.3. Backplane und Nathan	7
3. Grundlagen der Signalübertragung	10
3.1. Datenleitungen	10
3.1.1. Masse-bezogene Datenleitung	10
3.1.2. Differentielle Datenleitungen	10
3.2. Störungen in der Datenübertragung	11
3.3. Leitungsimpedanz, Leitungsterminierung und Stützkapazität	14
3.4. Seriell-Kommunikationsinterface	18
4. Komponenten des Boards	20
4.1. Virtex5	20
4.2. Spannungsregler	22
4.3. P-Channel MOSFET	23
4.4. Operationsverstärker	24
4.5. Differential Line Driver	24
4.6. Differential Line Receiver	25
4.7. Digital-Analog-Converter	25
4.8. Analog-Digital Converter	26
4.9. Analog-Multiplexer	26
4.10. Temperatursensor	28
4.11. CPLD	28
5. Board Layout	30
5.1. Board Dimensionen	31
5.1.1. Aufteilung in IC-Board und <i>Spikey</i> -Board	31
5.1.2. Nutzen	32
5.2. Versorgungsspannung und Lagenaufbau	32
5.3. Membranspannungen	35

5.4. Neuronenspannungen	36
5.5. Complex Programmable Logic Device	37
6. Kommunikation zwischen FPGA und CPLD	38
7. Zusammenfassung	44
A. Spikey Chip	46
A.0.1. Footprints	46
A.0.2. Bondingpläne	49
B. Schaltungen	52
B.1. Lagenaufbau und wichtige Stromwerte	60
C. Materialliste	61
D. Fehler auf der Platine	64
Literaturverzeichnis	66
Danksagungen	67

1. Einleitung

Wohl kein anderes Organ, egal ob im Menschen, in Reptilien oder in einem Insekt, strahlt die gleiche Faszination auf Forscher aller Richtungen aus wie das Gehirn. Das mag daran liegen, dass selbst das Gehirn des einfachsten Organismus den Naturwissenschaftlern Rätsel aufgibt, während es Ingenieuren und Medizinern Lösungen für neue Technologien und zur Heilung von Krankheiten liefert.

Eine Möglichkeit für die Forschung an neuronalen Netzwerken bietet spezielle neuromorphe Hardware, welche diese Netzwerke emulieren. Die Neuronen werden in neuromorpher Hardware durch elektronische Bauteile aufgebaut, welche in ihrem Zusammenspiel das Geschehen im Gehirn emulieren. Dabei wird versucht die Anzahl Neuronen stetig zu steigern, um grössere Netze auf gleichbleibendem Raum herzustellen. Neuromorphe Hardware unterscheidet sich von normalen Computersystemen dadurch, dass Berechnungen nicht mehr sequentiell auf wenigen komplexen Bauteilen, sondern parallel auf vielen wenig komplexen Elementen stattfinden.

FACETS und Electronic Vision(s) Electronic Vision(s) bildete in Kooperation mit Partnern die interdisziplinäre FACETS-Projekt. Fast-Analog-Computing-with-Emergent-Transient-States, kurz FACETS, ist ein von der Europäischen Union finanziertes Forschungsprojekt im Bereich der Neurowissenschaft. 15 Arbeitsgruppen aus sieben Ländern arbeiten an biologischen Experimenten und an der Modellierung neuromorpher Hardware.

Die Nachfolge von FACETS wurde durch das Projekt BrainScales angetreten. Mitglieder von FACETS und der Electronic Vision(s)-Gruppe entwickeln neuromorphe Hardware und führen Experimente an neuronalen Netzen durch.

DajaSittah Das neu entwickelte System *DajaSittah* soll eine Vereinfachung der Handhabung des in der Electronic Vision(s)-Gruppe verwendeten Spikey-Systems ermöglichen: Es soll handlicher und mit einem modernen FPGA leistungsfähiger als die bisherige Hardware sein. Es verbindet den Computer über einen Virtex5-FPGA¹ mit dem neuronalen Chip *Spikey*. Es besteht aus den beiden Platinen *Daja* und *Sittah*, welche aufeinander gesteckt werden. Die gegenüber der bisherigen Hardware herbeigeführte Vereinfachung besteht darin, dass nun Experimente auf einem handlichen und kostengünstigen System mit bis zu vier Chips und dadurch mit einem grösseren neuronalen Netz ausgeführt werden können.

Daja1 besitzt einen *Spikey*-Footprint. Nachdem *DajaSittah* zur Fertigung gesendet wurde, wurde eine verbesserte Version *DajaSittahV2* entwickelt. Auf dem neuen System

¹Field Programmable Gate Array

1. Einleitung

wurden Fehler beseitigt und einige weitere Bauteile eingefügt; *Daja1V2* wurde neu arrangiert und eine neue Version von *Daja2V2* und *Daja4V2* enthält Footprints für zwei bzw. vier *Spikey*-Chips. Der Code für den CPLD² ist derart, dass er nach kleinen Modifikationen auf beiden Plattformen einsetzbar ist. Die Unterschiede von *DajaSittahV2* zu *DajaSittah* werden in jedem Kapitel entsprechend genannt. Im Anhang finden sich technische Details zu dem System. Eine Bemerkung bzgl. der Namensgebung: *DajaSittah* beschreibt Eigenschaften für beide Platinen *Daja* und *Sittah*, *DajaSittahV2* beschreibt Eigenschaften für beide Platinen *DajaV2* und *Sittah*.

Gliederung der Arbeit Die vorliegende Arbeit besteht aus sechs Kapiteln. Die FACETS-Hardware wird in Kapitel 2 beschrieben um die wichtigen Aspekte der Funktionsweise dieser Hardware zu verdeutlichen. Kapitel 3 beschreibt Grundlagen der Signalübertragung, welche für das Design des Boards notwendig sind. Die für das Board verwendete Elektronik findet sich in Kapitel 4; dabei wurden nur Spezifikationen und Gründe für die Wahl des jeweiligen Bauteils aufgeführt. Kapitel 5 beschreibt das eigentliche Boardlayout von *DajaSittah*. Kapitel 6 beschreibt die Kommunikation zwischen dem FPGA und dem CPLD. In der gesamten Arbeit werden Vergleiche von *DajaSittah* zu anderen Leiterplatten wie *Recha* gezogen.

²Complex-Programmable-Logic-Device

2. FACETS Hardware

Die FACETS Hardware bildet unter anderem die neuromorphe Hardware für Experimente an neuronalen Netzwerken.

2.1. FACETS Stage1 Hardware chip

Der FACETS Stage1 Mikrochip *Spikey* emuliert ein künstliches neuronales Netzwerk in Hardware.

Er ist in einen digitalen Teil, welcher Kontrollinstanzen implementiert und einen analogen Teil, welcher das Netzwerk enthält, aufgeteilt. Das Netzwerk selbst ist in zwei Blöcke aufgeteilt, jeder implementiert 192 Neuronen, die jeweils 256 synaptische Eingänge besitzen. Die Synapsen sind blockweise in einem Rechteck, vergleichbar einer Verbindungsmatrix, angeordnet. Jede der 256 Synapsenzeilen pro Block wird durch einen Synapsentreiber getrieben. Wird nun im Digitalteil ein charakteristisches Aktionspotential, *Spike*, abgearbeitet, so wird der entsprechende Synapsentreiber ausgelöst und fährt eine Dreiecksrampe mit konfigurierbarer Anstiegs- und Fallzeit¹ ab. Die Synapse liegt nun am Kreuzungspunkt zwischen Synapsentreiberzeile und Neuronspalte mit einem konfigurierbaren, digitalen Gewicht. Das Aktionspotential führt nun am Neuron zu einer Änderung der Leitfähigkeit hin zum exzitatorischen Umkehrpotential im Falle anregender Eingänge bzw. zum inhibitorischen Umkehrpotential im gegenteiligen Falle. Die Membranspannung am Neuron entwickelt sich in Folge dieser Änderungen der Leitfähigkeiten und kann evtl. die Feuerschwelle überschreiten, wodurch ein digitaler Ausgangspuls ausgelöst, sowie die Membranspannung resettet wird. Die dadurch erzeugten digitalen *Events*, wobei die Quellnummer des Neurons sowie der Feuerzeitpunkt kodiert darin sind, werden vom Chip am Datenausgang ausgegeben.

2.1.1. Spikey Daten und mehrere Spikeys in einer Kette

Spikey Eingabedaten bestehen aus zwei zehn Bit synchronen differentiellen Bussen, wobei eine Leitung den Takt, eine weitere Leitung ein Steuersignal und die restlichen acht Bit die Daten fuer das Neuronale Netzwerk überträgt. Ein über diese beiden Acht-Bit-Leitungen übertragenes Datenpaket enthält 64 Bit verteilt über vier Takte. *Spikey* erkennt ein neues Datenpaket daran, dass das Steuersignal beim Eingang eines neuen Datenpakets für einen Takt auf HIGH gezogen wird.

Spikey selbst sendet seine Daten wieder als 64 Bit Datenpakete über zwei zehn Bit synchrone differentielle Datenbusse, die ebenfalls pro Bus einen Takt und ein Steuersignal enthalten.

¹diese entsprechen den Zeitkonstanten für das Aktionspotential

2. FACETS Hardware

Spikey kann eintreffende Daten auf zwei Arten behandeln:

1. *Spikey* kann die Eingabe-Daten annehmen. Er leitet die Daten in sein Neuronales Netzwerk und sendet seine Ergebnisse weiter. Ein READ-Bit im Datenpaket wird gesetzt, um dem nachfolgenden Bauelement zu signalisieren, dass dieser Datenstrom die Daten eines neuronalen Netzwerks enthalten. *Spikey* nimmt die Eingabe-Daten nur dann an, wenn der Adress-Vektor des Datenpakets seiner eigenen Adresse entspricht und das READ-Bit nicht gesetzt ist.
2. *Spikey* kann die Daten ignorieren und über seinen Output weiterleiten; der Chip arbeitet dann im Prinzip als SHIFT-Register. *Spikey* handelt so, wenn der Adress-Vektor des Datenpakets seiner eigenen Adresse nicht entspricht oder das READ-Bit gesetzt ist.

Diese Konstruktion ermöglicht es, dass mehrere *Spikeys* in einer Kette, eine sogenannte Daisy-Chain, hintereinander geschaltet werden können. Ein Datenstrom wird nur von jenem *Spikey* mit der richtigen Adresse bearbeitet und von jedem anderen *Spikey* in der Kette weitergeleitet.

Diese Konstruktion verhindert auf der anderen Seite, dass Chip direkt miteinander kommunizieren können; es ist technisch nicht möglich, dass ein beliebiger *Spikey* die Daten direkt an seinen Nachfolger sendet. In einer Daisy-Chain laufen die Daten von einem Chip zurück zum FPGA, welcher die Daten für den nächsten Chip neu adressiert, bevor der FPGA neue Daten an den nachfolgenden Chip in der Kette schickt.

Die für die Daisy-Chain zur Verfügung stehende Bandbreite wird unter den *Spikey*-Chips geteilt. Ein weiteres Problem besteht darin, dass durch die im *Spikey* entstehenden Latenzen, egal ob die Daten verarbeitet oder weitergeleitet werden, die Anzahl der verschaltbaren Chips in einer Daisy-Chain begrenzt ist.

Um die vorher erwähnten Synapsentreiber gezielt anregen zu können existieren zehn digitale FireIn-Leitungen (siehe [Grübl, 2007, Seite 127]). Auf diesen Leitungen wird ein Ein-Takt langer Impuls gegeben, welcher direkt einzelne Synapsentreiber anregt.

Die Arbeitsweise des *Spikey* wird durch sechs Pins und einen differentiellen Takt beeinflusst. Liste 2.1.1 zeigt eine knappe Beschreibung dieser Pins (Details unter [Grübl, 2007, Seite 32]).

- RESET ermöglicht das Zurücksetzen des gesamten Chips
- PLL_RESET, PLL_BYPASS, PLL_LOCKED setzen Signale für den Zustand des Phase-Locked-Loop-Systems im Chip
- BS_MODE setzt den Boundary-Scan-Modus im Chip
- CI_MODE setzt einen *Bypass-Mode* im Chip.
- C_DELAY setzt die Funktionsweise einer Delayline, welche dem Dateneingangssinterface nachgeschaltet und dem Datenausgangssinterface vorgeschaltet ist.
- Differentieller Takt EXT_CLK liefert das Taktsignal für den Chip

Um die Eigenschaften der Neuronen, Synapsen und Synapsentreiber zu setzen (weitere Details in [Grübl, 2007, Seite 176]), benötigt das neuronale Netzwerk zur Arbeit die folgenden analogen Strom- und Spannungsparameter:

- V_{REST} setzt die Offset-Spannung eines Neurons.
- V_{START} setzt das Startpotential der Synapsentreiber.
- V_{M} setzt den Korrelationsparameter der Synapsen.
- V_{CASDAC} setzt den Referenzspannung für einen in *Spikey* eingebauten Digital-Analog-Converter.
- I_{START} setzt den Referenzstrom für den genannten Digital-Analog-Converter.

Am *Spikey* lassen sich an acht Pins die Membranpotentiale ausgewählter Neuronen auslesen, an einem weiteren Pin können Neuronenströme gemessen werden².

2.2. Recha

Recha³ ist das Trägerboard des *Spikey*-Chips. Es stellt die notwendige Elektronik und die Spannungsversorgung zur Verfügung um einen Chip zu betreiben. Diese Platine wird auf *Nathan* mittels Aufsteckverbindern aufgesetzt.

2.2.1. Hochfrequenz-Buchsen und Analog-Digital-Converter

Um einzelne Neuronen im *Spikey*-Chip überwachen zu können stellt *Spikey* OUT-AMP und einen IBTEST zur Verfügung, dargestellt in Abbildung 2.1.

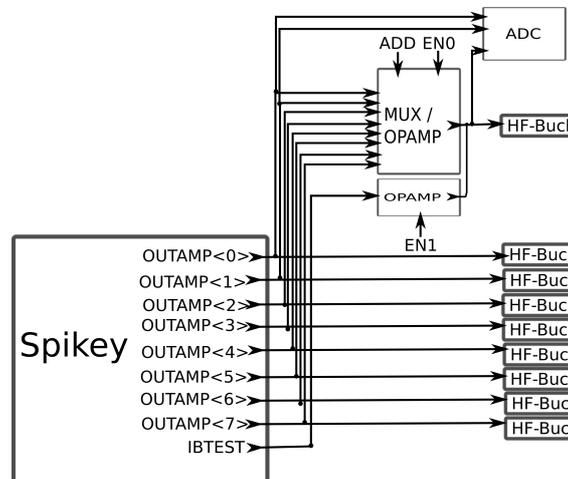


Abbildung 2.1.: Schema der Leitungen zum Auslesen der OUT-AMP Kanäle und des IBTEST-Pins.

²Welches Neuron an welchem Kanal anliegt, wird von dem Benutzer konfiguriert.

³Recha wurde entwickelt von A. Grübl und überarbeitet durch B. Ostendorf.

2. FACETS Hardware

Diese Kanäle geben die im neuronalen Netzwerk auftretende *Spikes* wieder. Die OUT-AMP-Pins liefern Spannungen, der IBTEST-Pin liefert einen Strom, welcher über einen Pulldown-Widerstand in eine Spannungsleitung verwandelt wird.

Jeder OUT-AMP Kanal ist an eine Hochfrequenz-Buchse von der Firma LEMO angeschlossen. Sie laufen des weiteren an einen Hoch-Frequenz Video-Multiplexer mit integriertem Operations-Verstärker (das verwendete Modell ist der MAX4312EEE von MAXIM [*Maxim Integrated Products*, a]). Der Ausgang dieses Multiplexers läuft an die neunte HF-Buchse.

IBTEST ist an seinen eigenen Operations-Verstärker (AD8063 von Analog Devices [*Analog Devices*, c]) angeschlossen, dessen Ausgang ebenfalls an die neunte HF-Buchse läuft: Jene Leiterbahn, welche vom OUT-AMP Multiplexer stammt, bildet zusammen mit der Leitung des IBTEST-Operations-Verstärkers eine TRISTATE-Leitung.

Ermöglicht wird das, da sowohl OUT-AMP Multiplexer als auch der IBTEST-Operationsverstärker je einen ENABLE-Pin besitzen. Ein gemeinsames ENABLE-Signal läuft an beide Pins, wird aber vor dem Mutliplexer invertiert, um zu verhindern, dass die TRISTATE-Leitung gleichzeitig von zwei Bauteilen getrieben wird. Zwei der OUT-AMP Kanäle und die TRISTATE-Leiterbahn führen ebenfalls an den Analog-Digital-Converter (AD7924BRUZ [*Analog Devices*, b]). Dieser ADC wandelt diese analogen Daten in digitale um und sendet ihn an den *Nathan* FPGA. Diese Daten sind notwendig zur Kalibration der Neuronenspannungen eines Chips⁴

2.2.2. Digital-Analog-Converter

Zwei Digital-Analog-Converter mit integriertem Operationsverstärker (beides MAX5253ACAP [*Maxim Integrated Products*, b]) liefern die analogen Spannungen des *Spikeys*.

Das Zusammenspiel der DACs ist in Abbildung 2.2 dargestellt. Der DAC auf *NATHAN* liefert das analoge Potential V_{REFDAC} und ein zweiter DAC auf *Recha* liefert V_{M} , V_{Rest} , V_{Start} und V_{CASDAC} .

⁴Um Daten während des Betriebs zu digitalisieren wäre ein schnellerer Video-ADC von Nöten, welcher nicht auf *Recha* verbaut wurde.

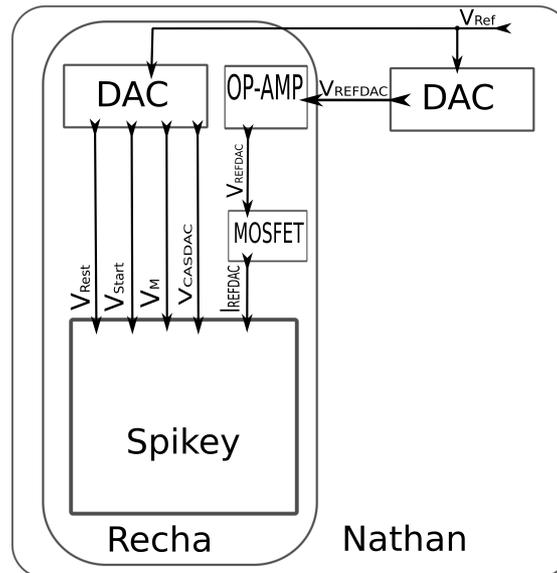


Abbildung 2.2.: System zur Lieferung der DAC-Spannungen.

Das Referenzpotential für beide DACs ist an das gleiche $V_{\text{Ref}} = 1.25\text{V}$ angeschlossen⁵. Der im DAC eingebaute Operationsverstärker allerdings ermöglicht es dem DAC maximal 1.8V analoger Ausgangsspannung zu liefern⁶. Der maximal lieferbare Referenzstrom⁷ ist $I_{\text{refdac, max}} = 25.0\mu\text{A}$.

2.3. Backplane und Nathan

Nathan und *Backplane* stellen die Infrastruktur her, so dass ein *Spikey* mit einem FPGA und mehrere FPGA untereinander und mit dem PC kommunizieren können.

Recha ist ein Aufsteckboard, das auf *Nathan* durch zwei 100-Pin-Stecker aufgesteckt nach dem Schema von Abbildung 2.3) wird.

⁵Dieser Wert entspricht der Referenzspannung des ADC.

⁶Die Berechnung von maximal möglichen I_{OUT} wurde entnommen von [Grübl, 2007, Seite 117].

⁷Die Berechnung von der maximal möglichen I_{refdac} wurde entnommen von [Grübl, 2007, Seite 117].

2. FACETS Hardware

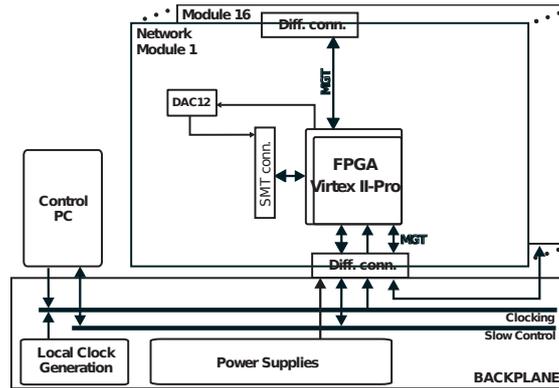


Abbildung 2.3.: Nathan Aufsteckboard auf Backplane (entnommen von [Ostendorf, 2007, Seite 33]).

Ein Virtex2-FPGA stellt die Kommunikation via differentiellen Leitungen zwischen dem Chip auf *Recha* und der Backplane her.

Die Hauptaufgabe der Backplane besteht darin die FPGA untereinander und mit dem PC zu verbinden, so wie in Abbildung 2.4 dargestellt. Der FPGA bedient sich dazu seiner eingebauten differentiellen Multi-Gigabit-Transceiver, die maximal eine Datenrate von 3.125Gb/s erreichen.

Die *Nathans* sind vom Gesichtspunkt der Kommunikation untereinander in einem 2D-Torus angeordnet: Jeder beliebige *Nathan* kann mit vier Nachbarn direkt kommunizieren und im schlechtesten Fall läuft die Kommunikation über drei zwischengeschaltete *Nathan*-FPGAs.

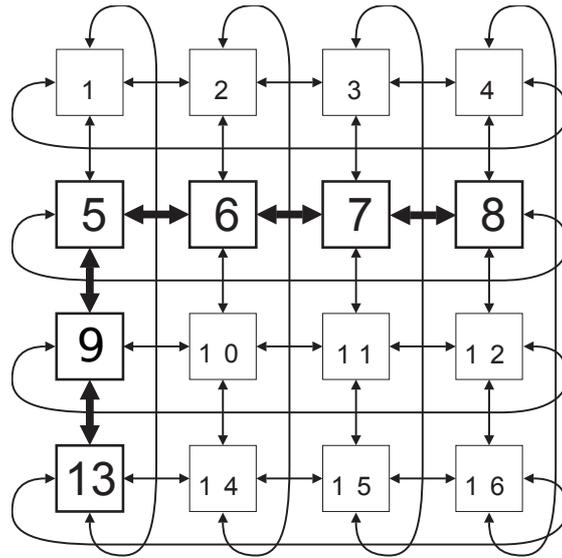


Abbildung 2.4.: 2D-Torus des Spikekeynetzes auf der Backplane (Bild entnommen von [Ostendorf, 2007, Seite 32])

In 2.4 beispielsweise können die Daten falls nötig von Knoten 13 zu Knoten 8 zu kommen, indem sie über die Knoten 9, 5, 6 und 7 laufen.

3. Grundlagen der Signalübertragung

Beim Design einer Leiterplatte ist die Signalübertragung von essentieller Bedeutung. Auf diesem Board werden Signalleitungen verlegt, von denen einige Leitungen Daten mit einigen 100MHz Signalgeschwindigkeiten übertragen.

3.1. Datenleitungen

Datenleitungen übertragen Informationen zwischen einem sendenden und einem empfangenden Bauteil über elektrisch leitfähige Leitungen.

3.1.1. Masse-bezogene Datenleitung

Die Masse-bezogenen Datenleitungen auf *DajaSittah* bestehen jeweils aus einer Verbindung zwischen einem Sender und einem Empfänger. Der Empfänger interpretiert die Spannungsdifferenz zwischen dem anliegenden Spannungspegel gegenüber MASSE. Der Stromverbrauch ist abhängig von der Datengeschwindigkeit.

Bei den auf *DajaSittah* verwendeten digitalen Masse-bezogenen Datenleitungen interpretiert der Empfänger dabei ein HIGH oder eine logische 1, wenn an der Leitung ein Spannungspegel angelegt wird, LOW oder eine logische 0 wird interpretiert, wenn MASSE anliegt. Die verwendeten analogen Datenleitungen übertragen Spannungswerte als Parameter, die vom *Spikey*-Chip verwendet werden.

3.1.2. Differentielle Datenleitungen

Eine digitale differentielle Datenleitung besteht aus zwei Verbindungen, der sogenannten positiven und der negativen Leitung, zwischen Sender und Empfänger¹. Die Leitungen übertragen nicht jeweils den vollen Spannungspegel bzgl. MASSE wie bei Masse-bezogenen Datenleitungen, sondern auf jeder Leitung eine Offsetspannung zzgl. der Hälfte des Pegels mit entgegengesetztem Vorzeichen. Der Empfänger interpretiert die Spannungsdifferenz zwischen den Spannungen beider Leiterbahnen und ignoriert die absoluten Spannungen der beiden Einzelleitungen: Liegt nur die Offsetspannung an, verschwindet die Spannungsdifferenz.

¹Details für einen Low Voltage Differential Signaling-Standart finden sich [*National Semiconductor*, 2004].

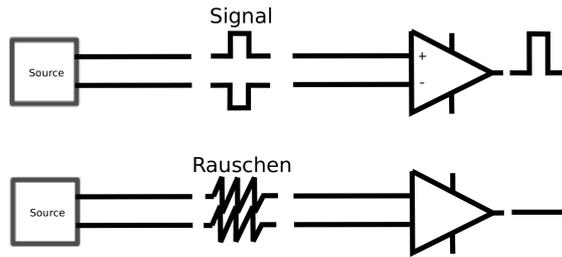


Abbildung 3.1.: Differentieller Puls und differentielles Rauschen

In Abbildung 3.1 werden die Vorteile ersichtlich: Da beide Einzelleitungen nahe beieinander verlaufen, beeinflusst Rauschen unter anderem durch Spannungsfuktuationen oder Schaltvorgängen in den angeschlossenen Bauteilen beide Leitungen in etwa gleich stark. Dieser Einfluss schlägt sich in einer Verschiebung des Spannungspegels der Einzelleitungen nieder. Der Empfänger aber ignoriert solche absoluten Verschiebungen, da sich die Differenz der Spannungen nicht ändert.

Differentielle Datenübertragung wird in dieser Arbeit verwendet, da sie Möglichkeit bietet Daten mit mehreren 100MHz zu übertragen.

3.2. Störungen in der Datenübertragung

Datenleitungen unterliegen Störeinflüssen und Interferenzen, die die Signalqualität besonders bei hoher Datengeschwindigkeit beeinträchtigen (nähere Informationen finden sich in [Grübl, 2003, Seite 12]).

Reflexion Die Gefahr von Reflexionen besteht darin, dass sie das eigentliche Signal überlagern und verfälschen können. Reflexionen entstehen, wenn Wellenpakete im Leiter auf Abschnitte treffen, in denen sich die Impedanz ändert. Impedanzsprünge kommen in den mittleren Abschnitten des Leiters vor, in der Regel allerdings in der Nähe des Senders, des Empfängers, von Durchkontaktierungen², Kondensatoren, Widerständen, etc.³. Das Verhältnis r zwischen Signalamplitude U_{Signal} und Reflexionsamplitude $U_{\text{Reflexion}}$ (entnommen aus [Grübl, 2003, Gleichung 2.8]) ist definiert durch

$$r = \frac{U_{\text{Reflexion}}}{U_{\text{Signal}}} = \frac{Z_{\text{LAST}} - Z_0}{Z_{\text{LAST}} + Z_0} \quad (3.1)$$

, wobei Z_0 die Leitungsimpedanz und Z_{LAST} der Eingangswiderstand des Bauteils ist. r hängt ab von der Impedanz des Leiters und der Bauteile, die angeschlossen sind; idealerweise sollte r verschwinden, was eintritt, wenn $Z_{\text{LAST}} \approx Z_0$ ist. Sie ist 1, wenn die Impedanz Z_0 verschwindet.

²Im allgemeinen Sprachgebrauch als Via bezeichnet

³Eine nähere Beschreibung der Impedanz findet sich in Paragraph 3.3

3. Grundlagen der Signalübertragung

Transmissions-Verstärkung Bei der Transmissions-Verstärkung wird eine Welle bei einem Impedanzsprung in seiner Amplitude verstärkt (entnommen aus [Grübl, 2003, Gleichung 2.8]). Dieser Effekt tritt auf, wenn die Impedanz nach dem Sprung höher ist als die Impedanz vor dem Sprung. Dabei wird der Transmissionskoeffizient t berechnet nach

$$t = 2 * \frac{Z_{Last}}{Z_0 + Z_{Last}} = 1 + r \quad (3.2)$$

mit Z_0 als Impedanz vor dem Sprung und Z_0 nach dem Sprung. Dabei kann $t \approx 2$ werden, wenn $Z_{Last} \gg Z_0$, also genau dann, wenn eine starke Reflexion stattfindet.

Verstärkte Wellen können dem angeschlossenen Bauteil schaden: Ein Bauteil würde durch diesen Effekt eine höhere Spannungsspitze als eventuell toleriert erhalten, welche ohne Gegenmaßnahmen das Bauteil in Mitleidenschaft ziehen kann.

Überschwingen Überschwingen ist ein Effekt, bei dem die Amplitude eines Signals verstärkt wird. Es entsteht unter anderem dadurch, dass beim Umschalten des Zustands einer Datenleitung das Signal dem gewünschten Pegel einpendelt, bis es dem Pegel entspricht.

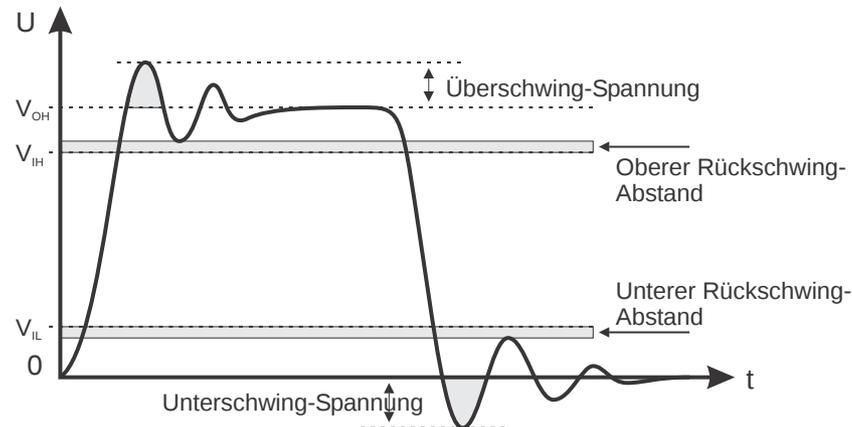


Abbildung 3.2.: Beispiel eine überschwingenden Signals (Bild entnommen von [Grübl, 2003, Seite 11]).

Überschwingen wird bei der Transmissionsverstärkung zu einem Problem, wenn die Überschwingamplitude ungefähr der Differenz zwischen einem Logikpegel und der jeweiligen Schaltschwelle entspricht. Viele Bauteile verfügen über sogenannte *Clamping-Dioden*, die das Überschwingen absorbieren. Allerdings zerstören grosse Ströme diese Dioden (siehe [Grübl, 2003, Seite 12]). Leiterbahnen sollten daher terminiert und in der Impedanz konstant gehalten werden.

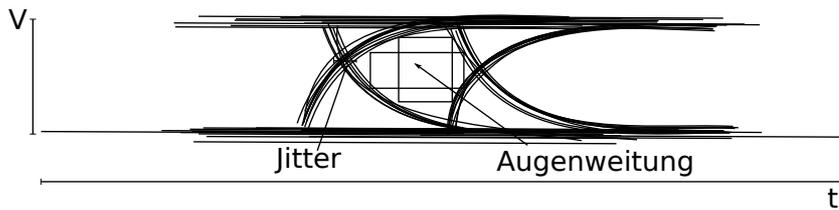
Übersprechen zwischen Leitungen Elektrische Leitungen erzeugen ein elektromagnetisches Feld, welches wiederum mit anderen Leitungen interferiert. Das von einer Leitung

verursachte Feld

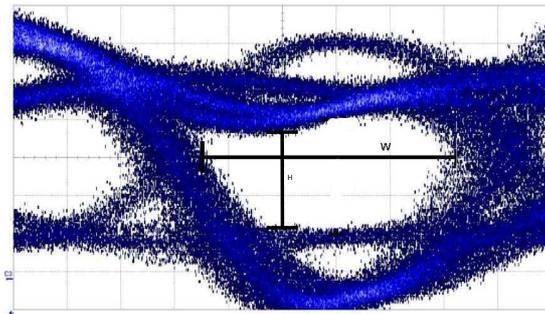
$$\vec{E} = \frac{Q}{4\pi\epsilon_0\epsilon_r} \frac{\vec{r}}{r^3} \quad (3.3)$$

betrifft Leitungen antiproportional zu $\frac{1}{r^3}$ mit der Entfernung. Dieser Effekt tritt unter anderem bei hochfrequenten Signalleitungen auf, wenn sich das elektromagnetische Feld permanent ändert. In Konsequenz müssen Leitungen mit einem gewissen Abstand untereinander verlegt werden, damit das Feld einer Leitung den Pegel der anderen Leitung nicht zu stark beeinflusst.

Augendiagramm Als Utensil zur Beurteilung, ob eine Datenübertragung möglich ist, gilt das Augendiagramm, welches in Abbildung 3.3 dargestellt ist⁴.



(a) Skizze eines Augendiagramms. Die Öffnung wird durch die beiden Rechtecke ausgedrückt.



(b) Foto eines Augendiagramms

Abbildung 3.3.: Beispiel eines Augendiagramms (Bild entnommen von [Grübl, 2003, Seite 14]), sowohl als Skizze als auch als Foto, welches an einem Oszilloskop mit Nachleuchten aufgenommen wurde.

Ein Augendiagramm wird für eine digitale Leitung erzeugt, indem permanent ein Bitpaket fester Grösse über die zu messende Datenleitung gesendet wird. In diesem Bitpaket werden alle bis auf ein Bit auf HIGH gesetzt; die Position des LOW-Bit wird zufällig ausgewählt.

⁴Das Augendiagramm erhält seinen Namen, weil in der Mitte des Diagramms eine Öffnung besteht, welches an ein Auge erinnert.

3. Grundlagen der Signalübertragung

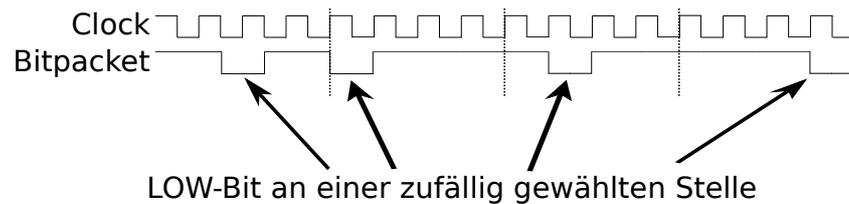


Abbildung 3.4.: Beispiel eines Bitpakets aus vier Bits mit zufällig gewältem LOW-Bit.

Die Höhe der Öffnung wird bestimmt durch die Spannungspegel, die Breite durch die Dauer, in der der Zustand definiert ist.

Am Augendiagramm erkennt man an der Öffnung des Auges, an der Relation zwischen Höhe und Weite des Auges, welche Signalqualität vorliegt; ein grosses Verhältnis oder ein weit geöffnetes Auge bedeutet ein gut definiertes Signal: Die Anstiegszeit ist klein, die Leitung erreicht schnell genug den gewollten Spannungspegel, das Signal ist in Phase. Umgekehrt bedeutet ein kleines Verhältnis zwischen Höhe und Weite der Öffnung oder ein geschlossenes Auge ein verschwommenes unter Umständen undefiniertes Signal: Die Anstiegszeit ist zu lang oder die Phasenzusammenhänge sind nicht eindeutig.

3.3. Leitungsimpedanz, Leitungsterminierung und Stützkapazität

Um die im Abschnitt 3.2 genannten Probleme zu minimieren, ist es notwendig, Leitungen zu terminieren und bei ihnen die richtige Impedanz sicherzustellen. Eine Terminierung schützt Bauteile vor Störungen, die auf der Leitung, aber auch Leitungen vor Störungen, die durch das Bauteil verursacht werden.

Leitungsimpedanz Die Impedanz eines Leitungsabschnitts gibt an, welcher Strom bei welcher Spannung durch diesen Abschnitt fließt.

Um eine saubere Datenübertragung zu gewährleisten, müssen alle Leitungsstrukturen Impedanz-kontrolliert verlegt werden (siehe Abschnitt 3.2), sodass jeder Abschnitt die Impedanz des vorhergehenden Abschnitts aufweist. Um das zu bewerkstelligen, sind neben den Eigenschaften der Leitung auch die Eigenschaften des Boards zu berücksichtigen.

Am Ausgang des Senders oder am Eingang des Empfängers treten Impedanzsprünge auf bedingt durch den Widerstand des Eingangs bzw. des Ausgangs, weswegen die Terminierung am Ende einer Leitung bei dem Empfänger oder dem Sender durchgeführt werden muss.

Eine Lage, auf der eine Impedanz-kontrollierte Leitung verlegt wird, benötigt eine Referenzlage mit einem möglichst konstantem Spannungspotential.

Stützkapazität Elektronische Bauteile benötigen eine stabile statische Stromversorgung. Stützkapazitäten erfüllen die Aufgabe die Stromversorgung für ein Bauteil zu stabilisieren. Durch eine Spannungsversorgung wird ein Kondensator aufgeladen. Schwankt

3.3. Leitungsimpedanz, Leitungsterminierung und Stützkapazität

die durch das Bauteil aufgenommene Energie oder die Energieversorgung selbst, wird die Differenz zwischen aufgenommener und gelieferter Energie von dem Kondensator entweder aufgenommen oder abgegeben. Die Kapazität muss ausreichen, sodass genügend Energie aufgenommen werden kann, darf aber nicht zu hoch ausfallen, da er für die Energieversorgung eine weitere Last darstellt.

Serielle Terminierung Bei der Serienterminierung in Abbildung 3.5 wird direkt am Ausgang des Senders ein Widerstand in Serie eingebaut⁵

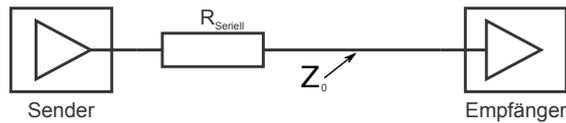


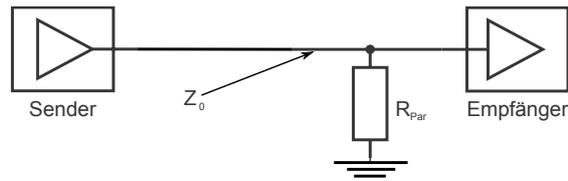
Abbildung 3.5.: Serienterminierung, bei der Widerstand nahe am Sender platziert ist.

Das Prinzip der Serienterminierung arbeitet mit Reflexionen, anstatt sie zu beseitigen, um eine saubere Datenübertragung zu gewährleisten. Der Widerstand am Sender, welcher der Impedanz der Leitung entsprechen muss, fungiert als Spannungsteiler, das Signal läuft mit halber Amplitude weiter zum Empfänger. Wie vorher angesprochen, wird am Sender durch den Impedanzsprung das Signal reflektiert. Das reflektierte Signal überlagert das einlaufende Signal und die Amplitude wird verdoppelt. Die rücklaufende Welle wird durch den Quelleninnenwiderstand absorbiert, es entstehen keine weiteren Reflexionen. Diese Form der Terminierung funktioniert bei einer Konfiguration mit nur einem Sender und einem Empfänger; bei mehreren Sendern bzw. Empfängern stören sich die verschiedenen Reflexionen gegenseitig.

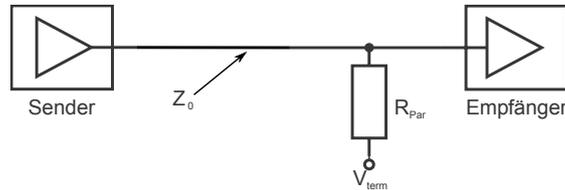
Parallele Terminierung Bei paralleler Terminierung (abgebildet in Abbildung 3.6) wird die Leitung am Empfänger über einen Widerstand mit MASSE verbunden. Der Wert des Widerstands entspricht dabei der Impedanz der Leitung.

⁵Diese Art der Terminierung ist die einzige, die am Sender und nicht am Empfänger terminiert wird.

3. Grundlagen der Signalübertragung



(a) Terminierung zu MASSE.



(b) Terminierung zu einer Terminierungsspannung.

Abbildung 3.6.: Parallelerterminierung (entnommen von [Grübl, 2003, Seite 18]).

Der Terminierungswiderstand verhindert, dass auf Empfängerseite ein Impedanzsprung auftritt, Reflexionen werden so in der Entstehung unterbunden. Da Strom nach MASSE abfließt, erhöht sich der Stromverbrauch bei einem HIGH-Pegel. Als Strom-sparende Alternative lässt sich der Widerstand an eine Terminierungsspannung verbinden. Dadurch fließt, während das Signal auf LOW gesetzt ist, Strom von der Terminierungsspannungsquelle richtung Leitung, Strom fließt von der Leitung richtung Terminierungsspannungsquelle, falls das Signal auf HIGH gesetzt ist: Der Stromverbrauch wird dadurch über im Mittel halbiert.

Thevenin Terminierung Anstatt einem Widerstand als Verbindung zwischen Leitung und Terminierungsspannung wie bei der Parallelerterminierung benötigt die Theveninterminierung einen Widerstand zu MASSE und einen zur Logikspannung. Der Strom fließt bei einem HIGH-Pegel von der Leitung über den Widerstand nach MASSE und bei einem LOW-Pegel von der Logikspannung über einen Widerstand in den Leiter.

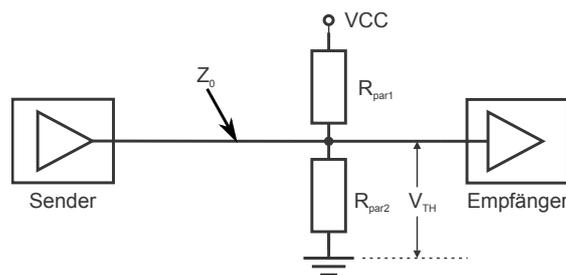


Abbildung 3.7.: Thevenin-Terminierung (entnommen von [Grübl, 2003, Seite 18]).

Die Thevenin-Terminierung basiert auf dem Thevenin-Theorem: Jede Kombination

3.3. Leitungsimpedanz, Leitungsterminierung und Stützkapazität

aus Widerständen und Spannungsquellen ist äquivalent einem System mit nur einem Widerstand und nur einer Spannungsquelle (nähere Informationen dazu finden sich in [Paul Horowitz, 2006, Seite 11]).

Der Vorteil der Theveninterminierung besteht darin, dass auf Erzeugung einer Terminierungsspannung durch eine separate Spannungsquelle verzichtet wird.

Differentielle Terminierung Bei paralleler Terminierung an differentiellen Leitungen werden beide Leitungen via Widerstand verbunden, wie in Abbildung 3.8 dargestellt.

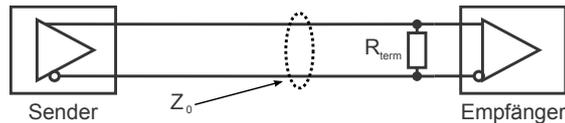


Abbildung 3.8.: Parallele Terminierung an differentiellen Leitungen (entnommen von [Grübl, 2003, Seite 21])

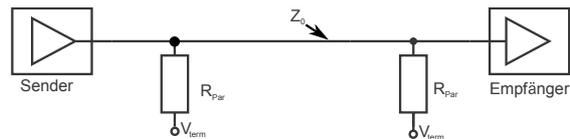
Der Wert dieses Widerstands entspricht dabei der Impedanz der gesamten differentiellen Leitung. Der Widerstand wirkt dabei wie bei der parallelen Terminierung und absorbiert Reflexionen. Strom fließt von der positiven Leitung in die negative, wenn ein HIGH-Bit übertragen wird; bei einem LOW-Bit wird kein Strom übertragen.

Stub Series Terminated Logic Class 2 Der Stub Series Terminated Logic Standard für unidirektionale Kommunikation (siehe [Association, 2002]) beschreibt eine Reihe an Terminierungstechniken für Datenübertragung bei hoher Frequenz⁶.

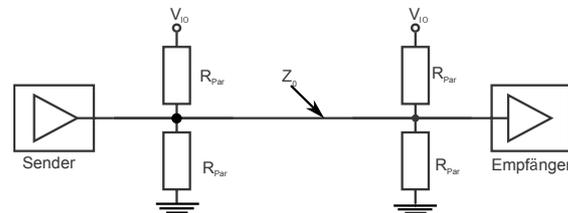
Abbildung 3.9 zeigt den gleichen Standard sowohl als Parallelterminierung als auch als Theveninterminierung, die auf Grund des Thevenintheorems 3.3 äquivalent sind.

⁶Es werden hier nur die zwei für das Design des Boards relevanten Standards beschrieben, weitere finden sich in [Xilinx, 2009, Seite 234].

3. Grundlagen der Signalübertragung



(a) SSDL2-Terminierung zu einer Terminierungsspannung



(b) SSDL2-Terminierung zur Versorgungsspannung

Abbildung 3.9.: Beispiel einer SSDL2 Leitung mittels Theveninterminierung und Parallelterminierung

In diesem Zusammenhang wird die Leitung am Sender und am Empfänger jeweils seriell und parallel terminiert. Der Widerstandswert der Serierterminierung entspricht dabei je der Hälfte der Leitungsimpedanz, der Wert der Parallelterminierung entspricht der vollen Impedanz.

3.4. Serielles Kommunikationsinterface

Ein Problem entsteht, wenn man weniger Datenleitungen zur Verfügung stehen hat, als Signale zu übertragen sind. Diese Signale können seriell über eine Datenleitung statt parallel über mehrere Datenleitungen übertragen werden. Das serielle Interface implementiert die Kommunikation zwischen zwei oder mehr Bauteilen mit vier Datenleitungen. Das in dieser Arbeit verwendete Verfahren enthält Elemente des SPI-Standarts. In dieser Konfiguration für eine serielle Kommunikation sind genau ein MASTER-Device und ein SLAVE-DEVICE vorhanden: Sind mehrere SLAVE-Devices in einer Chain vorhanden, wird jeweils SDO eines SLAVE mit SDI des nachfolgenden SLAVE verbunden. Alle Bauteile besitzen vier Ein- bzw. Ausgänge:

1. Serial-Data-Input SDI dient als Dateneingang
2. Serial-Data-Output SDO dient als Datenausgang
3. Seriel-Chip-Select SCS dient als Signal, dass Daten ausgetauscht werden. Dieses Signal wird gesetzt durch MASTER. SCS ist in dieser Konfiguration LOW-active.
4. Serial-CLOCK SCK dient als Takt für die serielle Kommunikation als auch als Takt für die SLAVE-Devices selbst.

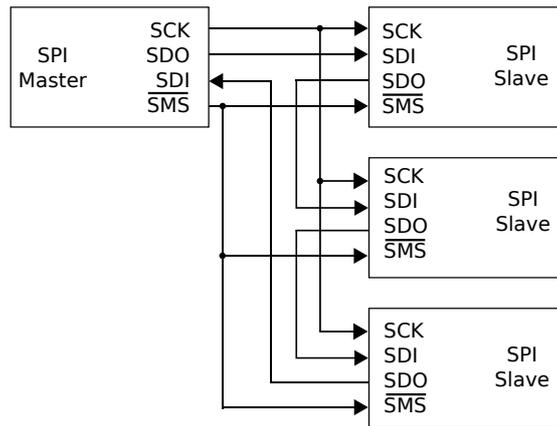


Abbildung 3.10.: Beispiel einer SPI-Kette als serielle Konfiguration mit einem Master und drei Slaves.

MASTER enthält ein Datenregister mit einer Anzahl an Bits; jeder SLAVE enthält ein Datenregister und die Anzahl an Bits in den Datenregistern aller SLAVES entspricht der Anzahl an Bits im MASTER-Datenregister. Sobald SCS auf LOW gezogen wird, tauscht MASTER sein Datenregister mit den SLAVE-Devices. Die Anzahl der Takte, in der SCS auf LOW gezogen bleibt, entspricht der Anzahl der Bits im Datenregister von MASTER, damit bei jedem Datenaustausch das MASTER-Device sein ganzes Datenregister an die SLAVE-Devices senden kann et vice versa. Realisiert wird der Datenregister als Shift-Register, sodass Daten bei jedem Datenaustausch gleichzeitig aufgenommen und gesendet werden; nach dem Datenaustausch befindet sich im Datenregister daher die Daten aller SLAVE-Devices und in den Datenregistern der SLAVE-Devices die Daten von MASTER. Die Datenleitungen SDI, SDO, SCS und SCK müssen parallel und mit gleicher Länge verlegt werden, damit alle vier Signale synchron von dem Sender zum Empfänger laufen.

SPI ist ein eingetragener Standard, allerdings muss beim Zusammenschalten einzelner SPI-Devices darauf geachtet werden, dass jedes Device mit der gleichen Takt-Frequenz arbeitet⁷.

⁷Leider ist es nicht sichergestellt, dass der SPI-Standard auf allen zusammengeschalteten SLAVE-Devices auf die gleiche Art implementiert ist oder ob die Devices in einer Kommunikationskette funktionieren.

4. Komponenten des Boards

Auf *DajaSittah* können aus diversen Gründen nicht alle elektronischen Bauteile, welche auf *Recha* verwendet werden, eingebaut werden. Ein Grund dafür besteht darin, dass anstatt einem Chip auf *DajaSittah* vier Chips verwendet werden.

Eine weitere wichtige Rolle spielt die Frequenzbandbreite der einzelnen Bauteile; eine Anzahl an Elektronik, die auf *DajaSittah* zum Einsatz kommen, wurden nach ihrer Fähigkeit ausgewählt Signale im Frequenzbereich bis zu 100MHz zu treiben. Bauteile dieser Frequenz konnten teilweise nicht gefunden werden.

DajaSittah kann zur Versorgung der gesamten Elektronik nur zwei Spannungswerte bereitstellen: 1,8V und 3,3V. Generell liegt das Problem darin, dass viele Bauteile eine höhere Versorgungsspannung benötigen, vorallem wenn die Bauteile für hohe Frequenzen ausgelegt sind.

Ein weiterer Faktor sind die Kosten der Bauteile und die Möglichkeit diese Elektronik zu erwerben. *DajaSittah* wurde von Anfang an darauf ausgelegt günstig hergestellt werden zu können. Kostspielig werden Bauteile unter anderem dann, wenn sie für hohe Frequenzen ausgelegt wurden oder eine hohe Anzahl an Kanälen aufweisen, um alle vier *Spikeys* bedienen zu können.

Bei einigen Bauteilen ist daher auf ein Viereck aus Frequenzbandbreite, vorhandenen Kanälen, Spannungsversorgung und Preis zu achten.

4.1. Virtex5

Das Virtex5 XC5VLX110T stellt die Grundlage für das Design von *DajaSittah* da und ist gleichzeitig die Verbindung zwischen *DajaSittah* und dem PC. *DajaSittah* wurden speziell dazu entwickelt mechanisch auf das Virtex5-Board ML505 aufgebracht zu werden und mit Virtex5 zu kommunizieren.

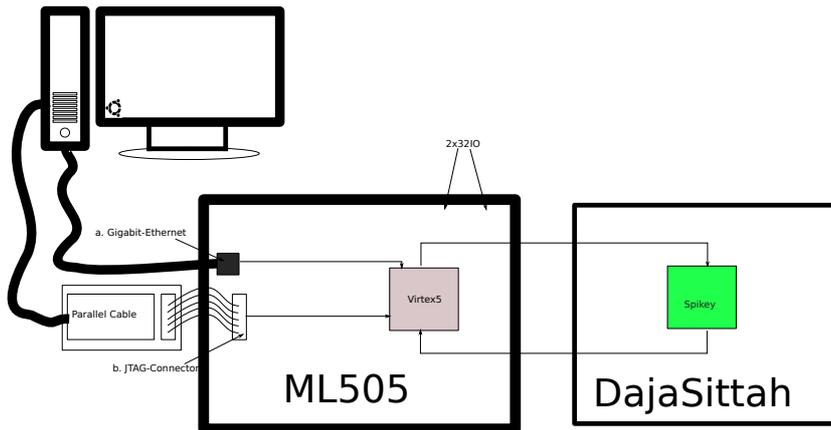


Abbildung 4.1.: Der ML505 als Schnittstelle zwischen Computer und dem Board *DajaSittah*.

Abbildung 4.1 stellt dar, wie der Computer den ML505 via einer 10/100/1000-Ethernet-Schnittstelle ansprechen kann oder via einem JTAG-Kabel programmiert wird. Der Virtex5 ist ein Xilinx-FPGA der Virtex-Serie¹. Er wurde 2006 eingeführt und liefert mehr Leistungsfähigkeit als der der Virtex2-Pro, der auf *Nathan* verbaut wurde. 560 IOs in 17 IO-Banken erlauben die Kommunikation des FPGA mit der Aussenwelt. Dieses Virtex5-Modell verfügt über 7200 Slices und 480kB an RAM.

Der FPGA ist in der Lage den Takt zu erzeugen um *DajaSittah* zu betreiben (weitere Informationen zum Erzeugen des Takts finden sich in Abschnitt 3.4). Er ist mit sechs Clock-Management-Tiles ausgestattet, die insgesamt zwölf Digital-Clock-Manager beinhalten. Der Takt mit einem Oszillator wird erzeugt (Punkt 12 in Abbildung 4.2). Er erzeugt eine Frequenz von 100MHz, die durch das Board in niedrigere Frequenzen konvertiert wird. Der FPGA verfügt über GigaBit-Transceiver, welche Daten mit bis zu 3,75Gb/s Baud-Rate übertragen.

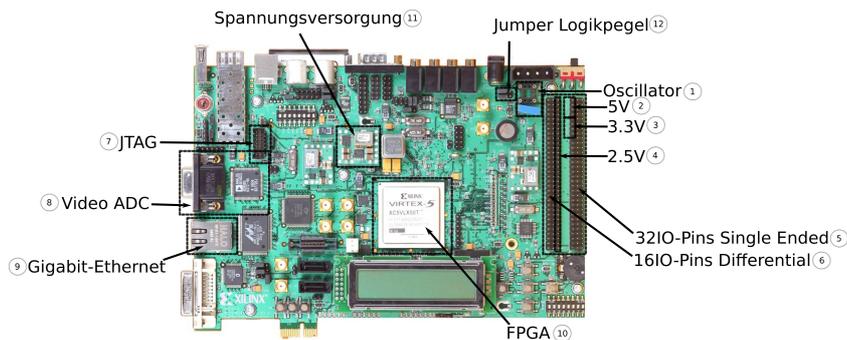


Abbildung 4.2.: Der ML505 mit einem Virtex5 XC5VLX50T; das Board welches unterscheidet sich von dem XC5VLX110T durch seinen FPGA (markiert sind wichtige Bauteile auf dem Board)

¹ mittlerweile ist die Version 7 erhältlich), welche für Audio-/Videokommunikation ausgelegt ist.

4. Komponenten des Boards

Der ML505 verfügt über 32 Masse-bezogene IOs und 16 differentielle IO-Paare, die als 32 Masse-bezogene IO-Pins verwendet werden können, wodurch insgesamt 64 Pins zur Verfügung stehen. Der FPGA erlaubt es die IO-Pin intern nach SSTL2 (siehe Paragraph 3.3) mittels DCI² zu terminieren, sodass die entsprechende Leitung nicht extern terminiert werden muss. Die IO-Pins werden in unserem Fall mit dem Logikpegel 3,3V betrieben. Derart terminierte IO-Pins erreichen eine Übertragungsgeschwindigkeit von 800MB/s.

ML505 ist in der Lage Strom und Spannung über acht Pins der 64 IOs zu liefern. Die Spannungswerte können 2,5V 3,3V und 5V annehmen. Diese Spannungsversorgung ist nicht genügend Rauschfrei, weswegen sie in unserem Fall nicht verwendet wird, Bauteile auf *DajaSittah* direkt mit Spannung zu versorgen³. Insgesamt sind diese Spannungspins in der Lage 10A an Strom zu liefern. Des weiteren bietet der ML505 einen MASSE-Anschluss. Diese MASSE ist nicht Rauschfrei, weswegen sie nur bedingt dazu verwendet wird Bauteile mit einem MASSE-Anschluss zu versorgen. Dieser MASSE-Anschluss dient dazu, Bauteile mit einem MASSE-Anschluss zu versorgen, welche mit einer digitalen Spannungsversorgung versorgt wird⁴

Der ML505 ist kommerziell zu erwerben und beherbergt die notwendige Infrastruktur, welche für *Backplane*, *Nathan* und *Recha* designed werden mussten.

4.2. Spannungsregler

Zur Herstellung der Energieversorgung mit den richtigen Spannungen werden fixe und justierbare lineare Low-Dropout-Spannungsregler⁵ verwendet.

Die von *DajaSittah* benötigten Spannungen betragen 1,8V und 3,3V um im Allgemeinen Bauteile zu versorgen und Logikpegel zu definieren, sowie 1,65V um die Low-Voltage-Digital-Signaling-Leitungen mit Terminierungsspannung zu versorgen. Zudem sind die analoge und digitale Stromversorgung strikt zu trennen. Der ML505 liefert an *Sittah* drei Spannungsquellen: 2,5V, 3,3V und 5V, die insgesamt 10A an Strom liefern können. All diese Versorgungen sind Rausch-belastet, weshalb sie in der Regel nicht direkt verwendet werden können.

Bei der Auswahl eines Spannungsreglers ist auf das Gehäuse zu achten. Bei Stromstärken von 1A bis 2A muss die Wärmeabfuhr beachtet werden: Dem kann begegnet werden, indem man das Bauteil in einem grossen Gehäuse erwirbt und auf eine grosse Kontaktfläche zu MASSE achtet. Geeignet für solche Bauteile ist das Package DPAK⁶. Es bietet eine breite Fläche, die Abwärme bzw. Strom abgeben kann, an die Umgebung bzw. an MASSE.

- Die digitalen 3,3V werden direkt vom ML505 übernommen. Rauschen auf dieser

²Digitally controlled impedance

³stattdessen werden sie via Spannungsreglern in niedere Spannungen umgewandelt.

⁴Der MASSE-Anschluss wird über den Spannungsregler erzeugt, siehe dazu 5.2.

⁵In diesem Zusammenhang wird nur noch das Wort Spannungsregler verwendet

⁶Genaue Spezifikationen finden sich im Datenblatt [*ON Semiconductor*, a, Seite 1] oder [*ST Microelectronics*, 2008, Seite 1]

Leitung ist ausreichend gering um eine digitale Versorgung herzustellen.

- Die analogen 3,3V werden über einen REG1117-A von Texas Instruments (siehe [Texas Instruments, a]) erzeugt. Die angehängten Bauteile benötigen einige Milliampere an Strom, ein Leistungs-starker Spannungsregler ist nicht notwendig.
- Die digitale und analoge 1,8V-Versorgung wird von je von einem LD293xx von ST (siehe [ST Microelectronics, 2008]) erzeugt. Beide liefern 3A Strom. Sie sind beide verbunden mit der 2,5V-Versorgung des ML505.
- Die Terminierungsspannung wird durch einen justierbaren NCP565 (siehe [ST Microelectronics, 2008]) und einem angehängten Spannungsteiler erzeugt. Der NCP565 wird versorgt durch die 5V-Versorgung des ML505.
- Die Referenzspannung für Digital-Analog-Converter und Analog-Digital-Converter werden durch die 1,8V Referenzspannungsquelle REF3318 erzeugt (siehe [ON Semiconductor, b]).

Der verwendete justierbare Spannungsregler benötigt zum Betrieb einen ADJUST-Pin. Dieser Pin ist angeschlossen an einen Spannungsteiler, von dem ein Widerstand an MASSE und ein weiterer an die Ausgangsspannung des Reglers angeschlossen ist. Dabei bestimmen die Widerstandswerte des Spannungsteilers, welche Spannung an den ADJUST-Pin geliefert wird und damit welche Spannung vom Spannungsregler geliefert wird. Ersetzt man einen der Widerstände durch einen Potentiometer, so lässt sich die Ausgangsspannung fein einstellen.

4.3. P-Channel MOSFET

Für das zu bauende Board wird derselbe Transistor in der gleichen Konfiguration verwendet, wie er auf *Recha* (es wurde der BSS84 verwendet [NXP Semiconductors N.V., 2008]) zum Einsatz kam. In Kombination mit dem Operationsverstärker liefert er den Referenzstrom für den *Spikey*-Chip, der Source-Anschluss ist angeschlossen an die 1,8V analoge Spannungsversorgung und Gate an eine analoge Digital-Analog-Converter-Spannung angeschlossen.

Der Transistor muss für *Spikey* einen Referenzstrom I_{REFDAC} von bis zu $2.5\mu A$, das bedeutet für vier *Spikey* bis zu $10\mu A$ leisten (dieser Wert wurde errechnet in [Grübl, 2007, Seite 117]).

Es ist nicht sichergestellt, dass verschiedene *Spikey*-Chips die gleiche Menge an Strom verbrauchen; Wären zwei Chips an dem gleichen Referenzstrom angeschlossen, ist es möglich, dass ein Chip mehr und der andere Chip weniger Strom verbraucht. Um alle *Spikey*-Chips separat mit ihrem eigenen Referenzstrom zu versorgen, kommen vier Transistoren zum Einsatz, die jeweils an einem Ausgang des Operationsverstärkers angeschlossen sind.

4.4. Operationsverstärker

Das *Sittah*-Board benötigt zwei Operationsverstärker um Spannungen zu verstärken.

Verstärkung der neuronalen Spannungen Wie in Abschnitt 2.1 beschrieben, kann die neuronalen Spannungen an Hand von neun Kanälen gemessen werden.

Um diese Membranspannung bearbeiten zu können, ist es notwendig einen Operationsverstärker mit genügend hoher Frequenzbandbreite zu finden.

Laut Boris Ostendorfs Berechnungen ergibt sich für die Signalfrequenz (siehe [Ostendorf, 2007, Seite 36]) ein Wert von 35MHz. Wie in Abschnitt 4.7 erwähnt, ist dieser Operationsverstärker

Verstärkung zur Erzeugung von Referenzstrom für Spikey Ein Verstärker verstärkt die analoge Spannung, bevor sie vom Transistor in Strom umgewandelt wird (siehe Abschnitt 4.3).

4.5. Differential Line Driver

Differential Line Driver dienen dazu Masse-bezogene Datenleitungen in Low-Voltage-Differential-Signaling-Leitungen umzuwandeln (der Low-Voltage-Differential-Signaling-Standard wurde beschrieben in Unterabschnitt 3.1.2).

Das Bauteil empfängt ein Signal U_{In} und sendet es über zwei Leitungen weiter, die wie in den Formeln 4.1 erzeugt werden.

$$U_+ = U_{Common-Mode} + \frac{U_{In}}{2} \quad (4.1)$$

$$U_- = U_{Common-Mode} - \frac{U_{In}}{2} \quad (4.2)$$

In unserem Design ist der Line Driver zur Übertragung hoher Frequenzen bis zu 400MHz bzw. 800MBit vorgesehen.

Desweiteren wurde nach Bauteilen mit nur einem Kanal gesucht: Bauteile mit mehreren Kanälen wären nur in einem Gehäuse mit sehr dünnen Pins zu erwerben. (Das Bauteil, dass die Erwartungen am ehesten erfüllt, ist der SN65LVDS1 von Texas Instruments, siehe [Texas Instruments, b]), er leistet maximal 315MHz bzw. 630MBit. Wir verwenden für unser Design das Package SOT-23, da es einfach zu verlöten ist. Der Common-Mode liegt in der Mitte zwischen MASSE und Logikpegel.

Die Spannungsversorgung ist 3,3V digital, welches dem Logikpegel der zu übertragenen Daten entspricht. Er verbraucht pro Bauteil etwa 25mW an Leistung. Der zum Betrieb der Line Receiver benötigte Leistungswert ist wichtig, da 18 Bauteile verwendet werden.

4.6. Differential Line Receiver

Differential Line Receiver dienen dazu Low-Voltage-Differential-Signaling-Leitungen in Masse-bezogene Datenleitungen umzuwandeln (der Low-Voltage-Differential-Signaling-Standard wurde beschrieben in Unterabschnitt 3.1.2).

Das Bauteil empfängt Signale mit Low-Voltage-Differential-Signaling auf zwei Zweigen U_+ und U_- und sendet ein Signal über eine einfache Leitung weiter, die wie Formel 4.3 erzeugt wird.

$$U_{\text{Einfach}} = U_+ - U_- \quad (4.3)$$

Für den Line Receiver gelten die gleichen Bedingungen wie für Abschnitt 4.5.

Das Erzeugen eines Masse-bezogenen Datensignals aus zwei Leitungen ist aufwendiger als das Erzeugen von zwei Datenleitungen aus einer einfachen Datenleitung. Der Grund dafür liegt darin, dass bei den zwei Leitungen je die Hälfte eines Logikpegels erreicht werden muss und die beiden Leitungen sich in der Amplitude negieren. Eine Masse-bezogene Datenleitung dagegen muss für ein Bit den vollen Signalhub erreichen. Daher sind Line Receiver für hohe Frequenzen schwerer zu finden als entsprechend schnelle Line Driver. (das schnellste gefundene Bauteil ist der SN65LVDT2 von Texas Instruments, siehe [Texas Instruments, b]). Er erreicht nach Spezifikation 200MHz bzw. 400MBit. Damit bleibt das Bauteil hinter dem ursprünglich gewünschten Wert von 400MHz bzw. 800MBit/s zurück.

Die Spannungsversorgung ist 3,3V digital, welches dem Logikpegel der zu übertragenen Daten entspricht. Er verbraucht pro Bauteil etwa 60mW an Leistung; das ist mehr, als der entsprechende Line Driver verbraucht, da der Receiver wie in Abschnitt 4.5 erwähnt zum Erreichen des vollen Hubs mehr Energie benötigt. Der zum Betrieb der Line Receiver benötigte Leistungswert ist wichtig, da 18 Bauteile verwendet werden.

4.7. Digital-Analog-Converter

Ein einzelner Digital-Analog-Converter⁷ dient der Aufgabe die fünf analogen Parameter zu erzeugen (siehe [Paul Horowitz, 2006, Seite 614]). Der DAC des *Recha* wurde nicht verwendet, da jener DAC nur vier Kanäle zur Erzeugung von analogen Spannungen zur Verfügung stellt. Der verwendete DAC (AD5668, siehe [Analog Devices, a]) besitzt acht Kanäle: Vier Kanäle zum Erzeugen der analogen Parameter für alle Chips zusammen, vier weitere zum Erzeugen der Referenzströme für jeden Chip einzeln. Der DAC beansprucht eine Referenzspannung von 1,8V, welche es dem DAC erlaubt eine maximale analoge Spannungen von 3,3V zu liefern. Wie der ADC wird der DAC über einen Drei-Kanal-SPI angesprochen (siehe Paragraph 6). Des weiteren stehen dem DAC ein *clr*- und *ldac*-Pin zur Verfügung, welche die analogen Parameter für alle Kanäle gleichzeitig löscht bzw. zum gleichzeitig auf den programmierten Wert setzt. Die Energieversorgung läuft über eine 3,3V-Spannungsversorgung.

⁷Abkürzung DAC

4.8. Analog-Digital Converter

Wie in Unterabschnitt 2.2.1 beschrieben ist der Analog-Digital Converter⁸ auf *Recha* eingesetzt, um die analogen Membranspannungen des *Spikey* zu digitalisieren, bevor sie an den FPGA auf *Nathan* gesendet wurden (Funktionsweise beschrieben in [Paul Horowitz, 2006, Seite 621]). Die Daten, welche der ADC an den CPLD sendet, sind notwendig, um die *Spikey*-Chips zu kalibrieren. Dabei wurde der ADC von *Recha* für *DajaSittah* übernommen (das Modell ist der AD7924BRUZ, siehe [Analog Devices, b]).

Der ADC kommuniziert via SPI mit vier Kanälen; die Logikpegel der SPI-Pins werden gesetzt durch eine externe Referenzspannung V_{DRIVE} , die in diesem Fall an die 3,3V-Spannungsversorgung von *DajaSittah* angeschlossen ist. Er erreicht eine Auflösung des Gerätes von 10MBit, welches die analoge Spannung in 1024 Einheiten unterteilen kann.

4.9. Analoger Multiplexer

Dieses Board benötigt analoge Multiplexer zum Multiplexen der OUT-AMP-Kanäle des *Spikey*-Chips.

Der Multiplexer, welcher auf *Recha* aufgesetzt wurde (siehe Unterabschnitt 2.2.1), kann nicht verwendet werden, da das FPGA-Board die 4V-Versorgung nicht liefert, die dieser Multiplexer als minimale Spannungsversorgung liefert.

Der ursprüngliche Wunsch bestand darin das Board mit zwei HF-Buchsen auszustatten um zwei verschiedene Kanäle vom gleichen oder von verschiedenen *Spikeys* gleichzeitig zu messen.

Dafür wären zwei 32-Kanal-Multiplexer für alle 32 OUT-AMP-Kanäle und zwei Vier-Kanal-Multiplexer für die vier IBTEST-Pins notwendig gewesen, die nach Abbildung 4.3 verschaltet werden müssten.

⁸Abgekürzt als ADC

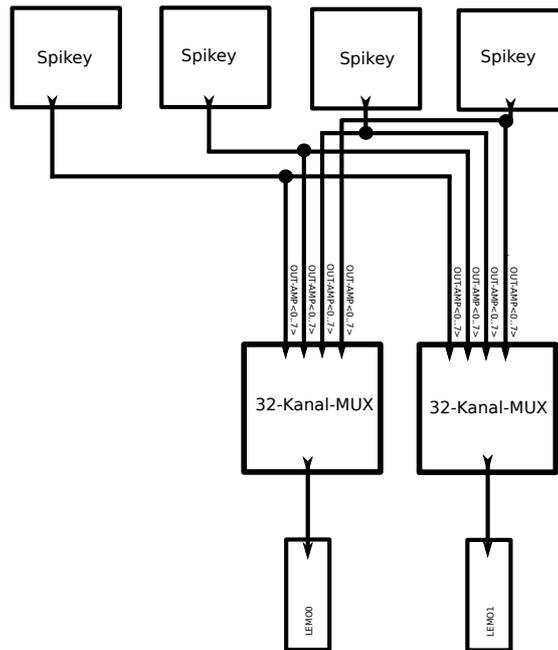


Abbildung 4.3.: Schaubild des Systems zum Multiplexen der OUT-AMP-Kanäle mit zwei 32-Kanal-Multiplexern.

Bauteile mit dieser Anzahl an Kanälen allerdings erreichen nicht die notwendige Übertragungsfrequenz (für die endgültige Verschaltung siehe Abschnitt 5.4).

Das ausgewählte Bauteil (ADG658 von Analog Devices, siehe [Analog Devices, e]) besitzt acht Eingänge für jeden OUT-AMP-Kanal. Im Gegenteil zum Multiplexer auf *Recha* ist in dem diesem Bauteil kein Operationsverstärker integriert, der Operationsverstärker muss als extra Bauteil nachgeschaltet werden.

Um die IBTEST-Kanäle zu multiplexen wird ein weiterer Multiplexer verwendet (ADG604 von Analog Devices siehe [Analog Devices, d]) mit vier Kanälen. Da IBTEST ein fast statisches Signal überträgt, ist ein schneller Multiplexer für diese Signale nicht notwendig. Beide Bauteile besitzen CMOS-Eingänge (siehe [Paul Horowitz, 2006, Seite 969]), sodass die zu messende analoge Spannung nicht durch Basisströme gestört wird.

Eine Möglichkeit auf dem Board zwei HF-Buchsen zu verbauen, besteht darin einen Multiplexer zu finden, der mehr als acht Kanäle besitzt.

Falls die 5V-Versorgung des FPGA-Boards (über Blockkondensatoren, etc.) ausreichend von Rauschen zu befreien ist, ist es möglich Multiplexer mit eingebautem Operationsverstärker (wie der MAX4312EEE bei *Recha*) zu verwenden. Der Vorteil, denselben Multiplexer wie auf *Recha* zu verwenden, besteht darin, dass der zusätzliche Operationsverstärker obsolet wird. Auf *DajaSittahV2* wurde die Anzahl der aufgebrauchten Multiplexer zu verdoppeln, wobei jeder OUT-AMP-Kanal je an zwei Multiplexer laufen.

4.10. Temperatursensor

Die Aufgabe des Temperatursensors besteht darin die Temperatur des *Spikey* zu messen. Ein interner Sensor ermittelt die Temperatur und wandelt sie in ein 13-Bit-Wort um. Die Aufgabe des Sensor besteht darin, auf eine mögliche Überhitzung beim *Spikey*-Chip hinzuweisen.

Der auf *Recha* verwendete Sensor (MAX1617 von Maxim) konnte nicht übernommen werden konnte, da er einen externen Sensor verwendet.

Es wurde der TC77 von Microwire aus Kostengründen ausgesucht (siehe [*Microchip Technology*, 2002]).

Eine Eigenschaft dieses Bauteils besteht darin, dass er über einen Drei-Kanal-SPI (siehe 6) angesprochen wird. Eines der Pins dient sowohl als Dateneingang als auch -ausgang. Der Dateneingang ist dafür gedacht den Sensor zu konfigurieren, eine Fähigkeit, von der im Normalfall kein Gebrauch gemacht wird⁹.

4.11. CPLD

Das Problem des FPGA-Board für unsere Zwecke besteht darin, dass in der IO-Leiste nicht genügend Pins zur Verfügung standen. Daher wird der CPLD verwendet um ein serielles Signal in parallele Signals umzuwandeln.

CPLD sind Logikbausteine, die mit beliebigen Programmen programmiert werden können. Die Komplexität des Programms ist dabei abhängig von der Anzahl der im CPLD verwendeten Makrozellen: Eine Makrozelle konstituiert dabei eine programmierbare AND/OR-Matrix oder Rückkopplung, einen Eingabe- oder Ausgabeblock. In dem Fall von *DajaSittah* wurde ein CPLD ausgesucht, dessen Anzahl Makrozellen der Anzahl der zu setzenden Signale, internen Bits, internen Rechenschritte entspricht.

Der CPLD erzeugt keinen eigenen Takt, aber verfügt über ein Takt-Netz, welches einen Takt aufnehmen und an im UCF-File indizierte IO-Pins ausgeben kann. Drei Pins GCK0, GCK1 und GCK2¹⁰ liefern den Takt an das Takt-Netz; IO-Pins, deren Zweck darin besteht einen Takt auszugeben, werden im UCF-File des CPLD mit COOL-Takt gekennzeichnet. Ein intern implementierter Takt-Teiler erlaubt es den Takt durch 2, 4, 6, 8, 10, 12, 14 und 16 zu teilen.

Der wesentliche Vorteil der CPLD besteht darin, dass er viele Logikpins besitzt, die unter anderem Signale mit 3,3V Logikpegel treiben; dieser Umstand erlaubt es uns eine hohe Anzahl an Signalen zu setzen. Der CPLD ermöglicht eine interne Terminierung unter anderem nach SSTL2. Dazu benötigt der CPLD eine externe Terminierungsspannung, welche ihm an einem beliebigen IO-Pin geliefert werden muss.

Die Auswahl des CPLD hängt ab von der Aufwendigkeit der verwendeten Programmierung, der Anzahl an bereitgestellten IO-Pins und der erreichbaren Taktfrequenz.

Zur Energieversorgung dienen diverse Spannungspins V_{CC} an drei Kanten des Chips.

⁹Es wird empfohlen dieses Bauteil auszutauschen gegen ein noch einfacheres Modell, welches nur über zwei Kanäle verfügt.

¹⁰GCK2 ist bevorzugt auszuwählen

Zur Einstellung der Spannungsversorgung der Logikpegel stehen VIO_{CC} an jeder Seite des Chips zur Verfügung.

Des weiteren stellt der CPLD diverse spezielle IO-Pins wie Global-Reset oder Data-Gate-Enable zur Verfügung, die für diesen Fall dauerhaft auf MASSE bzw. auf 3.3V gesetzt sind.

5. Board Layout

Beim Layout eines Boards sind die Position der elektronischen Bauteile, der Lagenaufbau und die Verlegung der Datenleitungen von besonderer Bedeutung. Des weiteren ist zu beachten, dass die Bauteile nach Fertigung der Platine auf das Board gelötet werden müssen; bei eng beieinander liegenden Bauteilen ist der Spielraum beim Löteten gering. Beim Design der Leiterplatte wurden die Regeln von Paragraph 5 beachtet:

Faustregeln zum Verlegen von Datenleitungen Generell gelten beim Verlegen von Datenleitungen folgende Richtlinien¹:

1. Datenleitungen benötigen eine festgelegte Impedanz. Jeder Impedanzsprung sorgt für Reflexionen des Signals (siehe Paragraph 3.2). Die Impedanz sorgt dafür, dass diese Reflexionen zu einem gewissen Teil gedämpft werden.
2. Datenleitungen sollten kurz gehalten werden. Umso länger die Datenleitung, desto mehr Störungen koppeln sich in die Leitung.
3. Datenleitungen sollten wenn möglich auf einer einzigen Lage verlaufen. Der Wechsel von einer Boardlage auf eine andere ist in der Regel mit einem Impedanzsprung verbunden. Des weiteren durchlöchern Vias die Lagen, ein Umstand, der besonders ins Gewicht fällt, da eine MASSE- oder Energie-Leitfläche an allen Stellen eine notwendige Breite aufweisen muss, um genügend Strom transportieren zu können.
4. Datenleitungen sollten wenn möglich gerade verlaufen; ein Knick in Datenleitungen bedeutet eine Scherung der Leiterlänge. Wird eine Richtungsänderung einer Leitung notwendig, so sollte sie um einen Winkel von 45 Grad geschehen.
5. Es ist auf eine saubere Energieversorgung des Senders bzw. Empfängers zu achten. Störungen an der Strom- oder Spannungsversorgung schlagen sich auf die mit den Bauteilen verbundenen Datenleitungen nieder.
6. Leiterplatten sollten mindestens vier Lagen aufweisen, wenn sowohl auf Ober- als auch Unterseite Signalleitungen verlaufen sollen. Dadurch besitzt sowohl Ober- als auch Unterseite seine eigene Referenzlage, welche Impedanz-kontrollierte Strukturen ermöglicht. Diese Referenzlagen sollten wenn möglich nicht durch Leitungen unterbrochen werden und auf konstantem Spannungswert gehalten werden.

¹Regeln zum Verlegen von differentiellen Leitungen stammen aus dem [National Semiconductor, 2004, Kapitel 3-1]

7. Der Abstand einer zu einer weiteren Datenleitung sollte mindestens der doppelten eigenen Leiterbahnbreite entsprechen. Der Abstand eines differentiellen Paares zu einer anderen Leitung sollte mindestens dem dreifachen des Abstands zwischen seiner positiven zum negativen differentiellen Leitung entsprechen. Ein hoher Abstand sorgt dafür, dass eine Leitung nicht mit dem elektromagnetischen Feld einer anderen Leitung koppelt, dass eine differentielle Leitung den Einfluss einer anderen Leitung stärker spürt als die der anderen Leitung.
8. Bei mehreren Datenleitungen sind regelmässige Strukturen zu bevorzugen. Regelmässige Strukturen deuten darauf hin, dass Datenleitungen die gleiche Länge aufweisen.

Die Gerberdaten der Platinen finden sich unter [Kuruvilla, 2010a].

5.1. Board Dimensionen

Bei der Dimensionierung von *DajaSittah* musste Vorsicht auf die Ausmasse des Boards gelegt werden, da *Sittah* auf ML505 aufgesteckt wird und somit Randbedingungen vorgegeben sind.

5.1.1. Aufteilung in IC-Board und *Spikey*-Board

Im sehr frühen Stadium des Boarddesigns wurde festgelegt zwei Leiterplatten statt einer herzustellen: Auf einer Platine werden die *Spikey*-Chips gebondet, die andere Platine trägt die verwendeten elektronischen Bauteile; das Board, welches die Chips trägt, wird mit *Daja*², das Elektronikboard mit *Sittah* bezeichnet. *Daja* lässt sich auf *Sittah* aufstecken.

Diese Aufteilung hat mehrere Vorteile:

1. *Spikey*-Chips sind aufwendig zu bondene Bauteile. Es ist notwendig, dass der Chip vor allen anderen Bauteilen auf einer Platine aufgebracht wird. Ein gebondeter Chip allerdings erschwert das Löten anderer Bauteile auf der Platine. Wird die Platine selbst beschädigt, so kann der Chip nicht entfernt werden von der Platine. Des weiteren lassen sich die elektronischen Bauteile nicht einzeln testen; wenn die richtige Arbeitsweise der Spannungsregler nicht gewährleistet ist, ist ein Durchbrennen des Chips möglich.
2. Es ist nützlich die Anzahl an hintereinander geschalteten *Spikey*-Chips variieren zu können, besonders unter dem Umstand, dass die Line Receiver nicht mit der gewünschten Geschwindigkeit zu erwerben sind (siehe Abschnitt 4.6). Ein separates *Spikey*-Board kann mit einem, zwei oder vier Footprints ausgestattet werden, während die Platine mit den elektronischen Bauteilen beibehalten werden kann.

²*Daja1* bezeichnet dabei *Daja* mit einem *Spikey*-Footprint, *Daja2* mit zwei *Spikey*-Footprints, mit vier *Spikey*-Footprints.

5. Board Layout

3. Durch das Verwenden zweier Boards, die übereinander gesteckt werden, lässt sich der Platzbedarf verringern.

Die Aufteilung des Systems in zwei Platinen erhöht allerdings die Kosten.

5.1.2. Nutzen

MultiPCB erlaubt die Zusammenlegung verschiedener Platinen auf einen Nutzen, wenn beim Design der Platinen bestimmte Regeln eingehalten werden: Platinen in einem Nutzen kosten weniger in der Herstellung als diese Platinen einzeln herzustellen zu lassen (Die Regeln zum Erstellen eines Nutzen finden sich auf der Webseite unter den Designhilfen von MultiPCB). Um die Platinen zu trennen, sind Frässtege angebracht, an denen die einzelnen Boards ausgebrochen werden können.

Ein Abstand von Platinenrand zu Bauteilen, Lagen und Leiterbahnen von mindestens 0,200mm und zwischen den Platinen ein Mindestabstand von 10mm muss eingehalten werden. Der Lagenaufbau aller beteiligten Platinen muss identisch sein.

5.2. Versorgungsspannung und Lagenaufbau

In diesem Abschnitt werden der Lagenaufbau als auch die Spannungs-/Stromversorgung beschrieben.

Der Lagenaufbau ist für die Spannungs- und Stromversorgung von Bedeutung, denn die Versorgungsleistung wird hauptsächlich über die beiden inneren Lagen von *DajaSittah* übertragen.

Daja und *Sittah* sind beiden Vier-Lagen-Platinen, wobei sich vornehmlich auf Ober- und Unterseite Bauteile und Leitungen befinden, eine Lage dient generell als Energie und die weitere als MASSE-Leitfläche: Diese beiden inneren Lagen sind dabei wieder unterteilt in digitale und analoge Bereiche, bei der Lage für die Energieversorgung sind die Leiterflächen unterteilt in Leitflächen für die einzelnen Spannungswerte.

Die Hauptverbraucher in diesem System stellen die vier *Spikey*-Chips sowie die LVDS-Bauteile und deren Terminierung da; ihr Verbrauch alleine beträgt bis zu 4A. Der Verbrauch der gesamten Elektronik ohne die LVDS-Bauteile dagegen verbraucht einige mA.

In Tabelle B.1 findet sich der Lagenaufbau.

Die Dicke der Leitfläche hängt von der Leiterbahnbreite ab, denn dünne Leiterbahnen lassen sich nicht in dicke Leitflächen ätzen: Das Fräsen der Leiterbahn führt bei dicken Leitflächen zu Ungenauigkeiten, die bei dünnen Leitern kritisch werden.

Die dünnsten Leiterbahn sind die LVDS-Leitungen in der Nähe des *Spikey*-Footprints: Sie dürfen 75 μ m nicht unterschreiten; daher sind für solche Strukturen bei Fertigung der Platine bei MultiPCB eine Lagendicke von 18 μ m notwendig (siehe MultiPCB Designhilfe).

Lagen für MASSE Die MASSE-Lage teilt sich auf in eine Leitfläche für die digitale und eine Leitfläche für die analoge MASSE, wobei die analoge Leitfläche am Rand verläuft und sich die digitale Leitfläche in der Mitte und damit unterhalb der LVDS-Bauteile

befindet. Die digitale Leitfläche ist direkt an die MASSE-Pins der IO-Leiste angeschlossen. Die analoge Leitfläche von *DajaSittah* ist nicht direkt an die IO-Leiste angeschlossen, denn MASSE, welche vom ML505 geliefert wird, ist nicht genügend rauschfrei. Stattdessen laufen die analoge und digitale Leitfläche unter dem digitalen 1,8V-Spannungsregler zusammen und werden dort über den grossflächigen MASSE-Pin verbunden. Die MASSE-Leitflächen müssen breit genug sein, damit darüber alle digitalen und analogen Ströme fließen können: Die analoge und digitale Leitfläche transportieren bis zu 2A bzw. 3A. Laut MultiPCB-Constraints ist eine minimale Breite von 4mm notwendig, welche auf dem Board realisiert wurde.

Lagen für die Energieversorgung Die Energie-Lage ist aufgeteilt in fünf Bereiche für fünf verschiedene Spannungsversorgungen:

1. In der Mitte befindet sich die digitale 3,3V-Leitfläche, welche bis zu 2A transportiert. Der Vorteil speziell dieser Leitfläche besteht darin, dass sie direkt an die Spannungspins der IO-Leiste angeschlossen ist. Das Problem, dass Stellen dieser Leitfläche zu eng sind, tritt bei dieser Leitfläche nicht auf, da das Gros an Strom von den LVDS-Bauteilen verbraucht wird. Die digitale 3,3V-Leitfläche ist die einzige Versorgung, die anstatt von der 5V-Versorgung über einen Spannungsregler erzeugt zu werden direkt an die Spannungsversorgungspins der IO-Leiste angeschlossen ist, da das starke Rauschen auf dieser Leitfläche weniger bedenklich ist.
2. Um die 3,3V-Leitfläche läuft die digitale 1,8V-Leitfläche, welche ebenfalls bis zu 2A transportiert.
3. Daneben folgt die analoge 1,8V-Leitfläche, welche ebenfalls bis zu 2A transportiert.
4. Den äussersten Rand bildet die analoge 3,3V-Spannung. Sie überträgt maximal einige 100mA und benötigt nicht mehr als einige 100µm an Breite.

Aus Platzproblemen musste allerdings drei Energie-Leitflächen auf die Oberseite verlegt werden.

1. Auf der Oberseite der Platine befindet sich V_{TERM} , welche bis zu 1A transportiert.
2. Eine kleine Leitfläche verbindet die 5V-Pins mit dem Spannungsregler, welcher die Terminierungsspannung erzeugt. Diese Leitfläche transportiert 1A. Wie sich die Terminierungs
3. Eine lange und dünne Leitfläche verbindet die 2,5V-Pins der IO-Leiste mit den beiden 1,8V-Spannungsreglern. Diese Leitfläche transportiert bis zu 4A.

Terminierungsspannung Die Terminierungsspannung wird von einem justierbaren Spannungsregler (siehe Abschnitt 4.2) erzeugt. Sie liegt auf der halben Spannung des Logikpegels, das bedeutet bei einem 3,3V-Logikpegel eine Spannung von 1,65V. Um den Stromverbrauch zu berechnen ist folgendes zu beachten: Strom fließt von der Spannungs-Leitfläche in die Datenleitung, wenn an der jeweiligen Datenleitung ein LOW anliegt,

5. Board Layout

denn die Terminierungsspannung ist im Pegel höher als die MASSE, die an der Datenleitung anliegt. Auf Empfängerseite fließt der Strom durch einen 50Ω -Widerstand pro LVDS-Empfänger. Auf Senderseite dagegen fließt er, da die Senderleitung serien-terminiert ist, zusätzlich noch durch den 25Ω -Widerstand pro LVDS-Empfänger. Nach der Ohm'schen Regel ergibt sich dadurch:

$$I_{Termination} = U_{Termination} * \left(\frac{1}{R_{Empfänger}} + \frac{1}{R_{Sender}} \right) * n = 1,65V * \left(\frac{1}{50\Omega} + \frac{1}{75\Omega} \right) * 18 = 1000mA \quad (5.1)$$

Es werden nach Formel 5.1 maximal 1A verbraucht. Der Aspekt der Terminierungsspannung wurde in der *DajaSittahV2* überarbeitet. Bei einem LOW-Pegel einer Datenleitung fließt Strom von der Terminierungsspannungs-Leitfläche in die Datenleitung; bei einem HIGH-Pegel fließt Strom von der Datenleitung zur Terminierungsspannungs-Leitfläche. Befinden sich mehr Datenleitungen im LOW-Zustand, so verbleibt die Terminierungsspannungs-Leitfläche im eingestellten Pegel, da Strom vom Spannungsregler nachgeliefert wird; befinden sich aber mehr Datenleitungen im HIGH-Pegel als im LOW-Pegel, so wird die Terminierungsspannungs-Leitfläche durch den zusätzlichen Strom aus den Datenleitungen im Pegel angehoben, da jener Strom an keiner Stelle abfließen kann und in der Leitfläche verbleiben. In der *SittahV2* wurden die Leitflächen neu geordnet unter anderem, da die Erzeugung der Terminierungsspannung so nicht funktionieren kann (siehe Paragraph 5.2). Die digitale 5V-, 3,3V-, digitale und analoge 1,8V-Leitfläche bleiben gleich. Die 2,5V-Leitfläche -angebunden an die IO-Leiste- wurde von der Oberseite auf die Energie-Leitfläche verschoben. In Konsequenz zur Neuausrichtung der LVDS-Bauteile wurde ebenfalls die Terminierungsspannungs-Leitfläche neu gezeichnet: Sie ist breiter und kürzer und füllt den Platz zwischen IO-Leiste und SMT-Steckern besser aus.

Daja wurde bezüglich der MASSE-Leitflächen in eine obere und untere Hälfte unterteilt. Die analoge MASSE-Leitfläche ist auf der rechten Seite mit dem SMT-Steckern verbunden, die digitale Leitfläche mit den SMT-Steckern auf beiden Seiten verbunden. Dabei beläuft der Rand der beiden Hälften entlang des *Spikey*-Footprints.

Bezüglich der Energie-Leitfläche ist *Daja* in vier Teile eingeteilt.

1. Die analoge 1,8V-Leitfläche befindet sich an der Oberseite und reicht bis unter den Teil des *Spikey*-Footprints, welche die analoge 1,8V-Spannungsversorgung des Chips herstellt.
2. Die digitale 1,8V-Leitfläche befindet sich in der Mitte des *Daja*-Boards. Damit verläuft sie unterhalb der digitalen 1,8V-Spannungsversorgung. Das Problem dieser Leitfläche besteht darin, dass sie von einer Reihe von Vias durchlöchert wird, welche dazu dienen die LVDS-Leitungen von der Unter- auf die Oberseite verlagern.
3. Die digitale 3,3V-Leitfläche befindet sich an der unteren Seite von *Daja*. Diese reicht unter die untere Kante der 3,3V-Spannungsversorgung. Diese Spannungsversorgung ist durch zwei Vias mit der Leitfläche verbunden.

Die analoge 3,3V-Versorgung für *Spikey* auf *Daja* läuft über normale Leitungen und nicht über eine Leitfläche; eine Leitfläche für diesen Spannungswert wurde für die *DajaSittahV2* von *Daja* eingeführt um die Anzahl an Leitungen zu minimieren.

Die Leitflächen wurden für die *DajaSittahV2* teilweise überarbeitet. Da die Vias, welche durch die LVDS-Leitungen verursacht werden, wegfallen, ist die MASSE-Leitfläche auf der Empfänger-Seite des *Spikeys* nicht mehr durchlöchert. Die analoge 3,3V-Leitfläche wurde auf die Oberseite von *Daja* verlegt.

Pins sind auf allen Boards angebracht, damit sämtliche Spannungen gemessen werden können.

5.3. Membranspannungen

In diesem Kapitel wird erläutert, wie die *Spikey*-Chips nach Abbildung 5.1 mit den analogen Membranspannungen V_{REST} , V_{CASDAC} , V_{START} und V_M und dem Referenzstrom I_{REFDAC} versorgt werden.

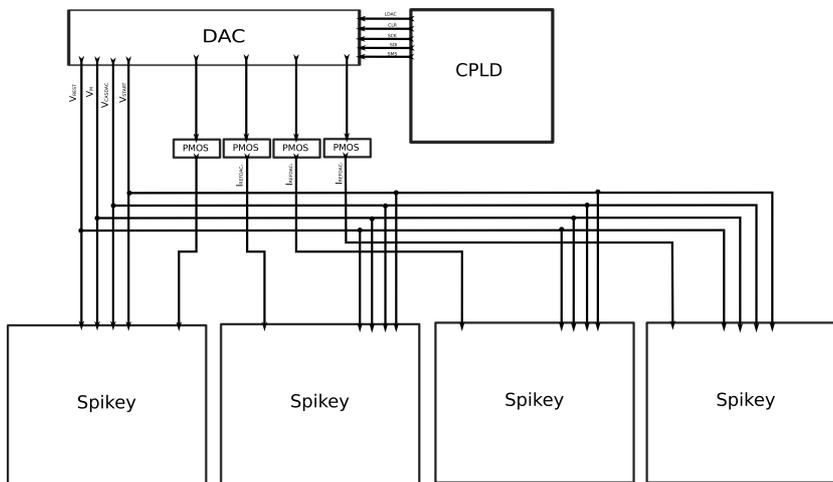


Abbildung 5.1.: Genereller Aufbau zum Erzeugen der analogen Spannungen.

Die für die Erzeugung der analogen Spannungen verantwortlichen Bauteile befinden sich an der rechten oberen Kante von *Sittah* über den SMT-Steckern. Diese Position erlaubt kurze Leiterbahnen hin zu den SMT-Steckern.

Dabei werden an alle *Spikey*-Chips diese fünf analogen Parameter angelegt. Die maximal mögliche analoge Spannung ist 1,8V, der maximal lieferbare Referenzstrom ist 25μ A.

I_{REFDAC} muss für jeden *Spikey* einzeln erzeugt und angelegt werden. Zwar leistet ein einzelner Transistor ein Vielfaches des maximal von einem Chip benötigten Referenzstroms von $2,5\mu$ A, es ist allerdings nicht garantiert, dass jeder Chip exakt die gleiche Menge an Strom verbraucht, sondern einen höheren Anteil.

Die Konfiguration dieser Bauteile wurde für *DajaSittahV2* beibehalten.

Für *DajaSittahV2* werden die Spannungen nicht mehr direkt an die Chips geleitet, da die Konfiguration der Leitungen zu kompliziert würde. Stattdessen befindet sich über den Chips eine Plane für je eine analoge Spannung, an der alle Chips angeschlossen werden.

5. Board Layout

Es wird bei diesem Board angenommen, dass die analogen Spannungen sich innerhalb eines Experiments nicht ändern: V_{CASDAC} , V_{START} und V_M sind durch ein RC-Glied terminiert. V_{REST} ist durch einen Blockkondensator stabilisiert wird³.

Auf *Daja* kann jede Spannung an Pins und alle vier Referenzströme an Jumpfern gemessen werden.

5.4. Neuronenspannungen

Wie *RechaV2* erlaubt *DajaSittah* das Messen der Neuronenspannungen.

Die OUT-AMP-Kanäle aller *Spikey*-Chips werden von *Daja* über die SMT auf *Sittah* geführt; dort wird über ein System von Multiplexern ein Kanal durchgeschaltet.

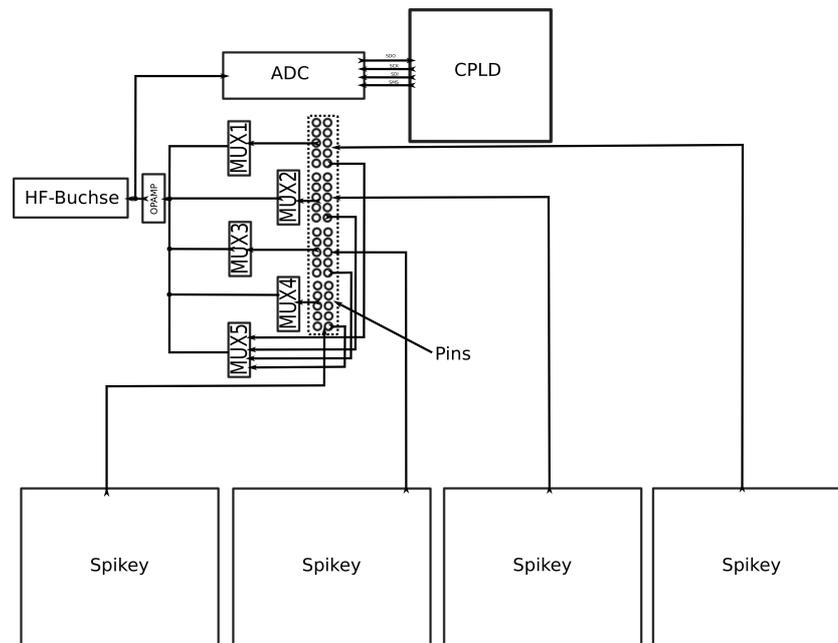


Abbildung 5.2.: Schaubild des Systems zum Multiplexen der OUT-AMP-Kanäle.

Die acht OUT-AMP-Kanäle jedes *Spikey*-Chips laufen je in den Acht-Kanal-Multiplexer (siehe Abschnitt 4.9), insgesamt werden vier Multiplexer für alle vier Chips benötigt.

Es existiert ein weiterer Multiplexer für die IBTEST-Kanäle. Die Multiplexer befinden sich neben dem SMT-Stecker links oben auf *Sittah*.

Alle fünf Multiplexer laufen auf einen TRISTATE-Kanal zusammen. Dieser TRISTATE-Kanal läuft in einen Operationsverstärker, der wiederum sowohl einen

³Es ist denkbar, dass Experimente aufgebaut werden, bei denen die Membranspannungen kontinuierlich variiert werden; dazu müssen die RC-Glieder abgelötet werden.

Leiter zur HF-Buchse als auch einen ADC besitzt⁴.

Wie Abbildung 5.2 dargestellt, können die OUT-AMP-Kanäle allesamt mittels eines Pins gemessen werden.

Für *DajaSittahV2* wurde die Anzahl an Multiplexer verdoppelt und eine zweite HF-Buchse angebracht, um zwei Kanäle von OUT-AMP messen zu können. Durch diese Konstruktion lassen sich zwei beliebige der 36 OUT-AMP- und IBTEST-Kanäle gleichzeitig auf den zwei HF-Buchsen ausgeben. Die Pinreihen wurden direkt neben die SMT-Stecker verschoben. Von dort laufen die 36 Leitungen zu den Multiplexern, sowohl auf der Ober- als auch Unterseite der Platine. Diese Konstruktion ist komplex, da sich die Strukturen auf sehr kleinen Flächen befinden.

5.5. Complex Programmable Logic Device

Dieser Platz wurde gewählt, da die 20 Kontroll-Signale des *Spikey*-Chips am unteren Rand von *Sittah* unter den SMT-Steckern entlang laufen. Der CPLD liegt der Art, dass sowohl eine digitale 1,8V- als auch 3,3V-Plane unter dem Footprint des CPLD verläuft.

⁴Damit ist dieses System nicht sehr verschieden von dem System, dass auf *RechaV2* aufgebaut ist (siehe Abschnitt 2.2).

6. Kommunikation zwischen FPGA und CPLD

Die Kommunikation zwischen *DajaSittah* und dem PC besteht aus drei Komponenten. Der Quellcode findet sich unter [Kuruwilla, 2010b].

- Die Software stellt ein Python-basiertes User-Interface namens *PyNN* (siehe [Davison et al., 2008]) zur Verfügung, welches dem Nutzer ermöglicht neuronale Experimente aufzusetzen und auf der Hardware laufen zu lassen. Der Code zu diesem User-Interface liegt oberhalb der hardware-spezifischen Schicht und muss daher für *DajaSittah* nicht angepasst werden.
- Unter dem User-Interface arbeitet die Hardwareabstraktionsschicht. Dieser Softwareteil benötigt für den Betrieb von *DajaSittah* Anpassungen.
- Der PC kommuniziert seine Daten zum FPGA mit dem Gigabit-Ethernet-Anschluss¹.
- Der FPGA kommuniziert sowohl mit dem CPLD als auch dem *Spikey*-Chip. Mit *Spikey* wird kommuniziert mittels der differentiellen Leitungen, der CPLD wird mittels serieller Kommunikation angesteuert.
- Der CPLD kommuniziert mit den einzelnen Bauteilen, den Slaves. Im CPLD sind VHDL-Entities² implementiert, welche diese Slaves kontrollieren.

Abbildung 6.1 stellt das Schema da, nachdem FPGA und CPLD kommunizieren:

¹Die Ansteuerung konnte bis zur Abgabe dieser Arbeit nicht vollendet werden

²Ein Slave wird jeweils von einer VHDL-Entity bedient, daneben gibt es noch Entities, die Takte kontrollieren.

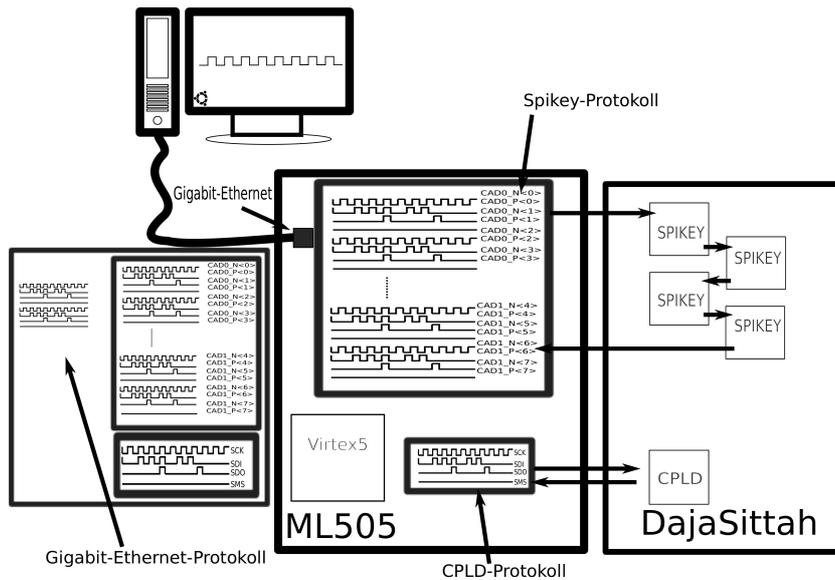


Abbildung 6.1.: Kommunikationschema zwischen CPLD und FPGA

Abbildung 6.2 zeigt die Arbeitsweise der FPGA-Seite bei der Kommunikation mit dem CPLD.

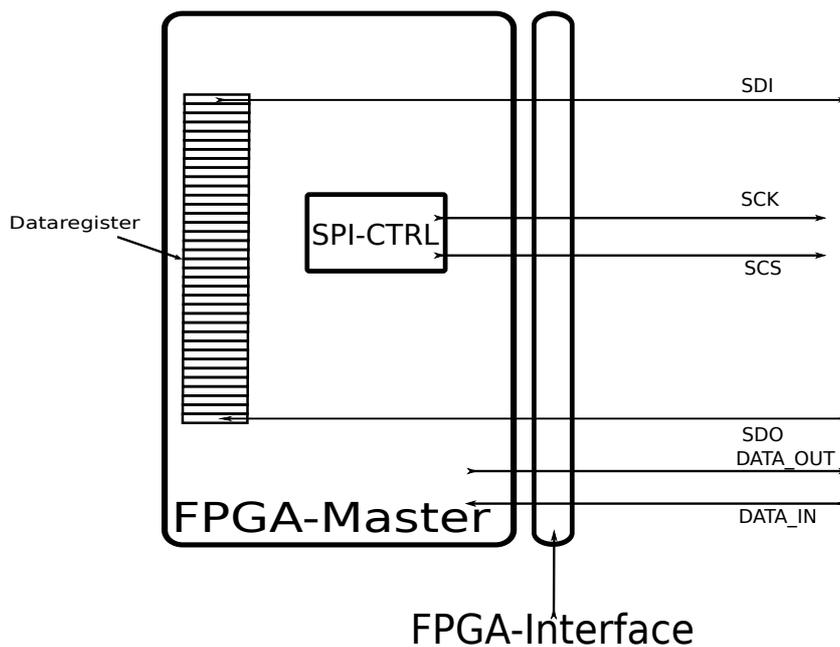


Abbildung 6.2.: SPI-Schnittstelle der FPGA-Seite.

FPGA-seitig arbeitet ein 36-Bit-Schieberegister: Die ersten fünf Bits speichern dabei die Adresse, die weiteren Register speichern jene Daten, welche an den jeweiligen Slave

6. Kommunikation zwischen FPGA und CPLD

übertragen werden sollen. Die Anzahl n an Bits für jedes Datum variiert je nach angelegtem Adressaten des Datums, maximal daher 33 Bits an Daten; daher variiert die Anzahl an Datenregistern, die beim SHIFT durchgereicht werden. SDO liegt am Datenregister 0 an; SDI liegt am Datenregister $4 + n$ an.

Abbildung 6.3 zeigt die SPI-Kette auf der CPLD-Seite.

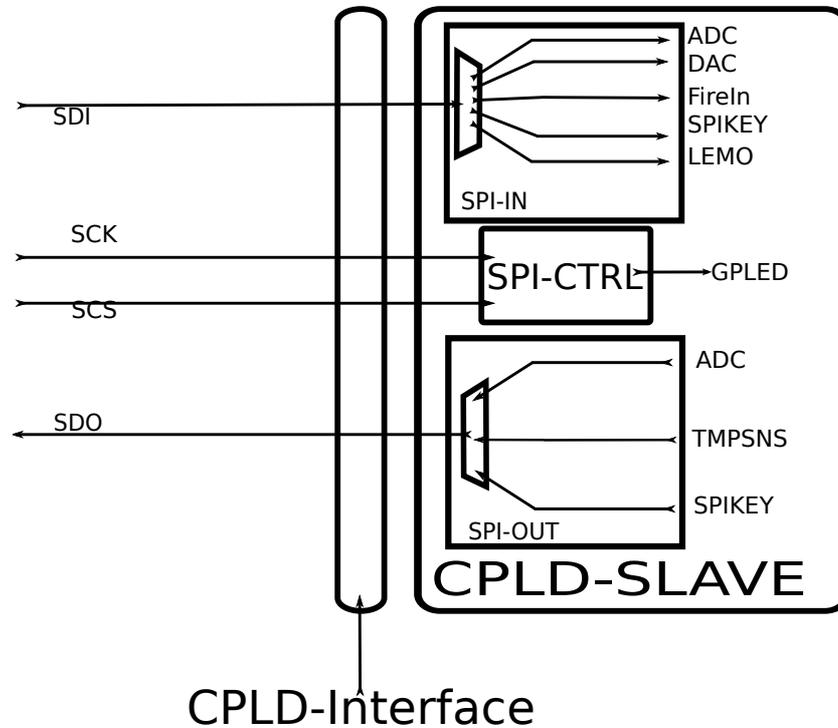


Abbildung 6.3.: SPI-Schnittstelle der CPLD-Seite.

Jede Entity ist unter anderem aufgebaut durch ein Schieberegister, ein SHIFT-Signal, einen SDI-Eingang und einen SDO-Ausgang. Sobald ein SHIFT-Signal auf HIGH gesetzt wird, beginnt das Schieberegister durchzulaufen, dabei liegt das letzte Registerbit an SDO und SDI am ersten Registerbit an. Die SHIFT-Signale aller Entitys liegt an einem Demultiplexer an, welcher das SHIFT-Signal der entsprechenden Entity auslöst. Tabelle 6 zeigt eine Übersicht der Entitys und der von ihnen angesteuerten Signale.

Spikey kontrolliert die Kontrollsignale für den *Spikey*. Es handelt sich um 18 Ausgangs- und vier Eingangssignale.

Multiplexer-Kontrolle kontrolliert alle vier Acht-Kanal-Multiplexer und dem Vier-Kanal-Multiplexer.

LEDs kontrolliert die LEDs.

Analog-Digital-Converter kontrolliert den ADC.

Digital-Analog-Converter kommuniziert mit dem DAC.

Temperatursensor liest die Temperaturdaten des Sensors aus, konvertiert sie und sendet sie an den FPGA.

Kommunikation mit *Spikey* Die 21 Parameter für *Spikey* werden in der Regel am Anfang eines Experiments gesetzt und dann nicht mehr geändert, RST wird in der Regel nicht verwendet. Ein Register mit 21Bits nimmt alle Parameter auf und legt sie an die jeweiligen Signale an. Daher existiert ein weiterer Register mit 21 Bits, dessen Aufgabe es ist, die alten Daten solange beizubehalten, bis die neuen Daten vollständig angekommen sind. Damit kann die vorhergehende Programmierung beibehalten werden, bis die neue Programmierung vollständig an die Entity übertragen wurde.

HF-Buchse Die Entity zur Kontrolle der HF-buchse besteht aus Kontrolle der Multiplexer-Adressen- und -ENABLE-Signale. Dazu werden im CPLD zehn Register verwendet. Die Adressen liegen an allen OUT-AMP-Multiplexern an, das ENABLE-Signal bestimmt, welcher Multiplexer dabei sein Signal sendet. Des weiteren wird

DAC Die Entity zur Kontrolle des DAC beschränkt sich darauf, Daten vom FPGA zum DAC weiterzuleiten. Das wird dadurch erleichtert, dass der DAC selbst keine Daten zum FPGA sendet. Allerdings setzt die DAC-Entity die Signale *ldac* und *clr*; *clr* wird vor der Konfiguration gesetzt, um alle analogen Spannungen simultan auf 0V zu setzen, *ldac* wird nach der Konfiguration hochgezogen, um die analogen Spannungen simultan zu setzen.

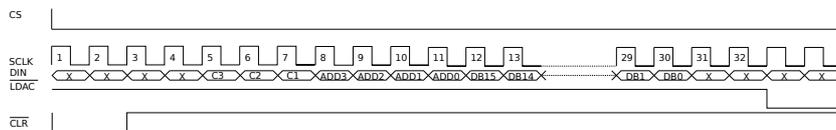


Abbildung 6.4.: Kommunikationsprotokoll des DAC mit fünf Eingängen (siehe [Analog Devices, a, Seite 8] und [Analog Devices, a, Seite 23])

Abbildung 6.4 zeigt das Kommunikationsprotokoll via 32Bits³.

ADC Die ADC-Entity leitet die Daten von dem FPGA an den ADC weiter und umgekehrt. Der ADC besitzt genau vier Pins nach dem typischen SPI-Standard, weswegen sich in dieser Entity keine Logik befindet. Der weitere Vorteil besteht darin, dass der Takt der ADC mit dem CPLD-Takt übereinstimmt; daher nimmt der ADC nur wenige Ressourcen des CPLD ein.

³Die genaue Beschaffenheit findet sich in [Analog Devices, a, Seite 23]

6. Kommunikation zwischen FPGA und CPLD

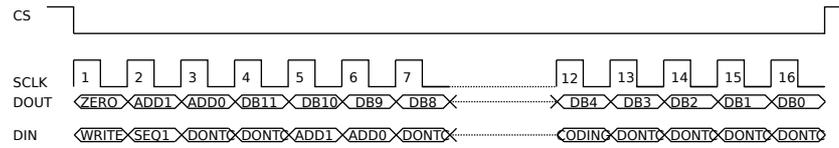


Abbildung 6.5.: Kommunikations-Protokoll des ADC (siehe [Analog Devices, b, Seite 19] und [Analog Devices, b, Seite 21]).

Abbildung 6.5 zeigt das Kommunikations-Protokoll des ADCs mit 32 Takten (die genaue Beschaffenheit der Daten wird beschrieben in [Analog Devices, b, Seite 21]).

LED-Leuchten Die LED-Leuchten sind aufgeteilt in vier *Spikey*-LEDs und zwei CPLD-LEDs. Die CPLD-LEDs zeigen an, dass sich der CPLD in Betrieb befindet und mit einem Programm programmiert ist. Diese LEDs leuchten abwechselnd.

Die *Spikey*-LED-Leuchten werden durch ein Bitarray gesteuert, welches den Namen CODE trägt.

- Normales Verhalten: Blinken der LEDs in einer bestimmten Frequenz, CODE ist gesetzt auf 00.
- Schnelles Blinken: Rekonfiguration der *Spikeys*, CODE ist gesetzt auf 01.
- Die LEDs bleiben dauerhaft auf HIGH: Temperatur Alarm, CODE ist gesetzt auf 10.

CODE wird gesteuert durch das Verhalten des Temperatursensors und des *Spikey*-Chips. Die Temperatursensor-Entity signalisiert der LED-Entity, ob ein Temperaturalarm vorliegt. Die *Spikey*-Entity signalisiert, ob eines der Chips einen PLL_LOCKED-Zustand vorweist. Liegen gleichzeitig ein Temperaturalarm und ein PLL_LOCKED-Signal an, so ignoriert die LED-Entity das PLL_LOCKED-Signal. In der verbesserten Version von *DajaSittah* entspricht der Anzahl an aufleuchtenden LEDs im normalen Betrieb der Anzahl der tatsächlich auf dem *Daja*-Board aufgebondenen Chips; bei Temperaturalarm leuchten alle LEDs auf. Die LED-Entity erlaubt Erweiterungen für den Fall, dass die LEDs dem Nutzer weitere Informationen anzeigen sollen.

Temperatursensor Der Temperatursensor läuft mit 5MHz statt mit 20MHz. Die Umwandlung der Frequenz verläuft durch einen Takt-Teiler, der die 20MHz durch Vier teilt. Die Daten, welche der Temperatursensor mit dem FPGA austauscht, müssen mit 20MHz zwischen FPGA und CPLD und mit 5MHz zwischen CPLD und Temperatursensor ausgetauscht werden. Die Daten, die vom Temperatursensor über den CPLD an den FPGA gesendet werden, müssen zunächst gespeichert werden. Ein Datenregister wird mit 5MHz mit diesen Temperaturdaten beschrieben, ein weiterer Datenspeicher mit 20MHz vom FPGA gelesen; beide Datenregister werden synchronisiert, wenn weder gelesen noch geschrieben wird. Die Entity arbeitet der Art, dass es regelmässig die Daten aus dem Temperatursensor mit 5MHz ausliest und in seinem Datenregister speichert. Diese Daten werden

dann, falls angefordert, mit 20MHz an den FPGA weitergeleitet. Ein grösseres Problem stellt die Tatsache da, dass der Temperatursensor einen Pin hat, der sowohl Dateneingang als auch Datenausgang darstellt. Abbildung 6.6 zeigt das SPI-Kommunikationsprotokoll für den Temperatursensor.

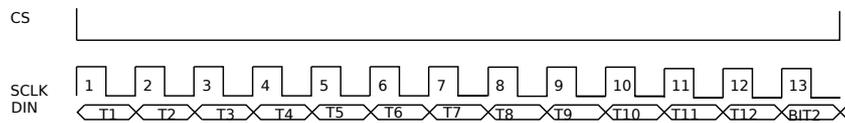


Abbildung 6.6.: SPI-Kommunikationsprotokoll für den Temperatursensor (siehe [*Microchip Technology*, 2002, Seite 10])

Das Beschreiben des Temperatursensors mit Kontrolldaten ist im normalen Gebrauch nicht notwendig, daher wurde die Funktionalität nicht in der Entity implementiert. Ein SMS-Signal dauert daher maximal acht Takte⁴. Zwischen FPGA und CPLD werden die Daten nach seriell ausgetauscht.

⁴Zum Beschreiben des Temperatursensors muss das SMS-Signal 16 Takte dauern.

7. Zusammenfassung

Diese Arbeit beschäftigte sich mit der Entwicklung des Systems *DajaSittah* von dem Design der Platine über die Implementierung der Kommunikation zwischen CPLD, FPGA und dem PC. Die primäre Aufgabe von *DajaSittah* besteht darin, die derzeitige Hardware zu vereinfachen. Dem Experimentator wird es möglich ein handliches System bei Demonstrationen oder zu Lehrzwecken zu präsentieren, welches alle Eigenschaften der bisher verwendeten Hardware aufweist. Es ist möglich, dass *DajaSittah* das vorhandene System zu einem Grossteil ersetzt. Das Multiplexersystem erlauben dem Nutzer die neuronalen Spannungen des Chips zu messen und auf einer HF-Buchse auszugeben. Auf der zweiten Version ist es möglich zwei beliebige neuronale Spannungen nebeneinander zu messen, egal ob sie von dem gleichen oder unterschiedlichen Chips stammen. Im Gegensatz zu *Recha* erlaubt *DajaSittah* dem Nutzer bei der Kalibrierung des *Spikey*-Chips auch den Kalibrationsstrom zu messen.

Der Zeitplan zur Entwicklung konnte allerdings nicht eingehalten werden, sodass das System nicht fertiggestellt ist.

Die Version 1 befindet sich in der Testphase. Um das System endgültig in Betrieb zu nehmen sind folgende Schritte notwendig:

- Bei dem CPLD muss kontrolliert werden, ob das implementierte serielle Interface korrekt funktioniert.
- Bei dem DAC ist die Überprüfung der analogen Spannungen von Nöten.
- Der ADC muss daraufhin überprüft werden, ob er die angelegten Spannungen korrekt digitalisiert.
- Bei den Multiplexern muss sichergestellt werden, dass jeder Multiplexerkanal korrekt ausgewählt wird.
- Die differentiellen Leitungen müssen dahingehend untersucht werden, ob sie Daten korrekt weiterleiten.

Allerdings ist es auf Grund des einfachen Aufbaus und der relativen Verwandtschaft zu *Recha* nicht unwahrscheinlich, dass das System ohne weitere Probleme funktionieren wird. Für die zukünftige Fertigung von *DajaSittah* wird empfohlen die Version 2 zu verwenden. Da die neue Version die gleichen Bauteile verwendet, sind aufwendige Tests auf diesem System nicht mehr notwendig.

Da nicht sichergestellt werden kann, dass die verwendeten LVDS-Bauteile die Datenrate von 800MBit/s erreichen, ist die Verwendung von *Daja4* nicht zu empfehlen: Die für jeden der vier Chips verfügbaren Bandbreite wäre nicht ausreichend. Es macht daher

Sinn die bisher verwendeten LVDS-Bauteile in naher Zukunft auszutauschen gegen Bauteile, welche die Datenrate 800MBit/s erreicht. Der ML505 besitzt einen Video-Analog-Digital-Converter: Sollte dieser ADC in Betrieb genommen werden, so lassen sich die neuronalen Membranspannungen sowohl auf der Hochfrequenz-Buchse als auch über den ML505 beobachten. *Sittah* bietet Platz einen zweiten CPLD einzubauen, welcher verwendet werden kann Experimente im Bereich der Robotik auszuführen. Die Vernetzung verschiedener Virtex5-FPGAs bietet die Möglichkeit die Anzahl an zusammengesetzten *Spikey*-Chips weiter zu erhöhen.

A. Spikey Chip

A.0.1. Footprints

Die folgenden Footprints wurden hergestellt für die Fertigung von *Daja1V2*, *Daja2V2* und *Daja4V2*. Diese Modifikation ist notwendig um mehrere Chips via Bonddrähten zu verbinden, anstatt differentiellen Leitungen zwischen den Chips zu ziehen. Der Footprint verwendet auf *Daja1* entspricht dem Footprint auf *Daja1V2*.

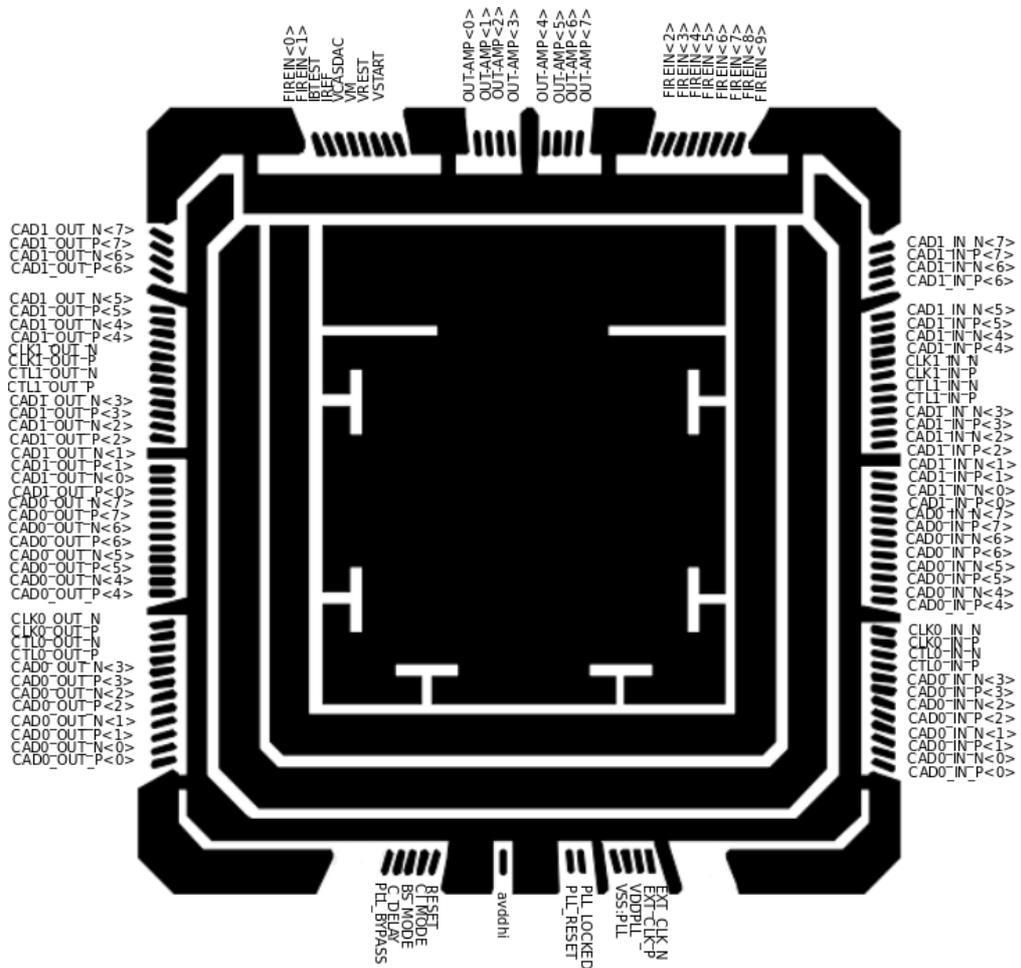


Abbildung A.1.: Der Footprint von einem *Spikey* auf *Daja1V2*.

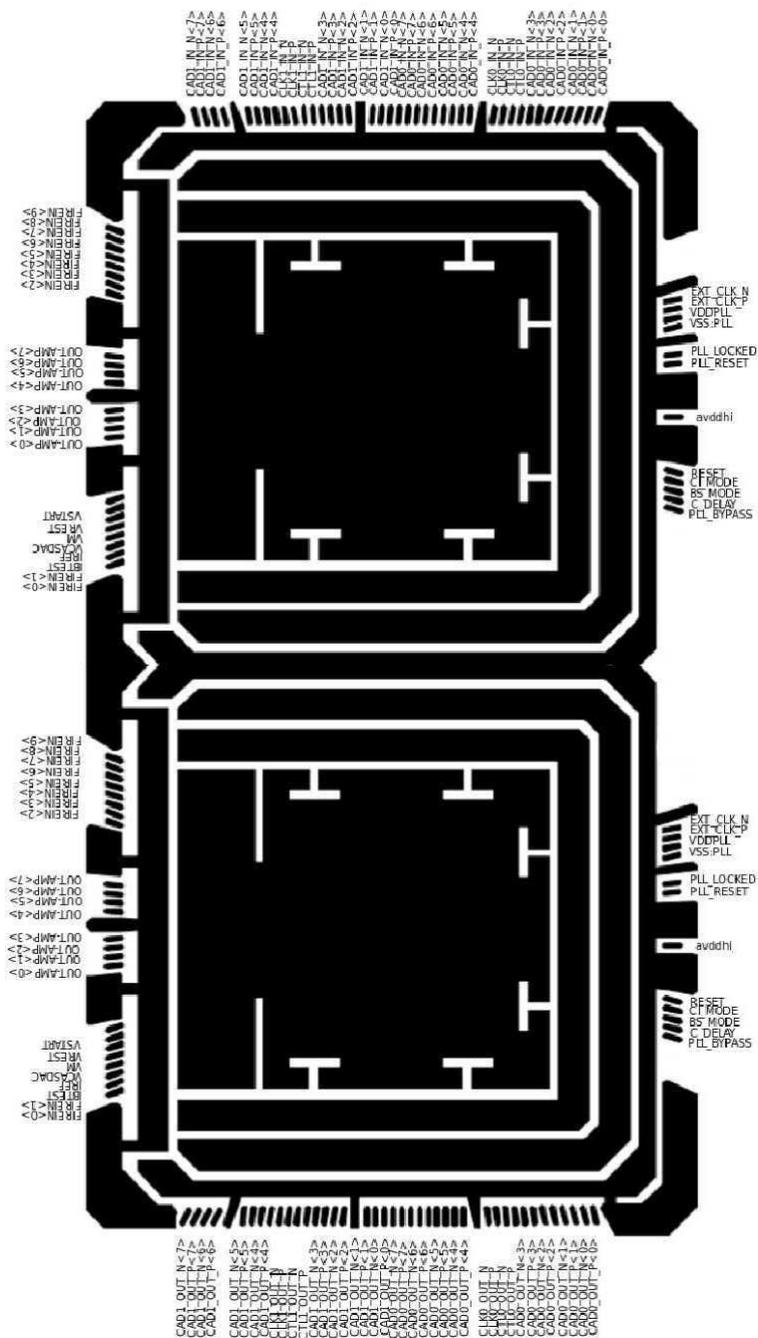


Abbildung A.2.: Der Footprint von zwei *Spikey* auf *Daja2*.

A.0.2. Bondingpläne

Die folgenden Bondingpläne zeigen das Bonding für einen einzelnen Chip auf *Daja1V2*, aber auch für das Bonden von Chips, die auf *Daja2* und *Daja4* verwendet werden. Der Unterschied zwischen *Daja1V2* und *Daja1* besteht darin, dass die JTAG-Verbindungen entfernt wurden.

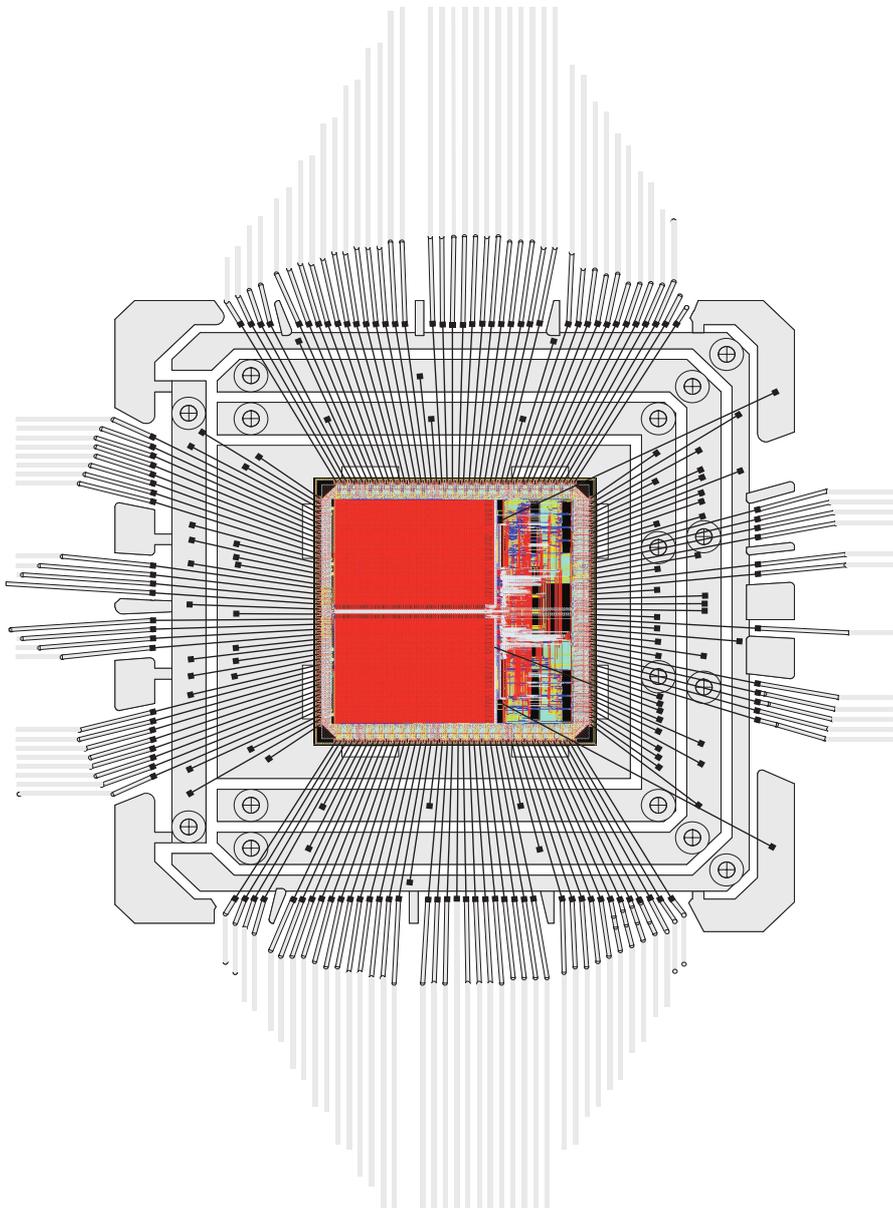


Abbildung A.4.: Der Bondingplan des Spikey-Footprints mit einem Spikey auf *Daja1V2*.

A. Spikey Chip

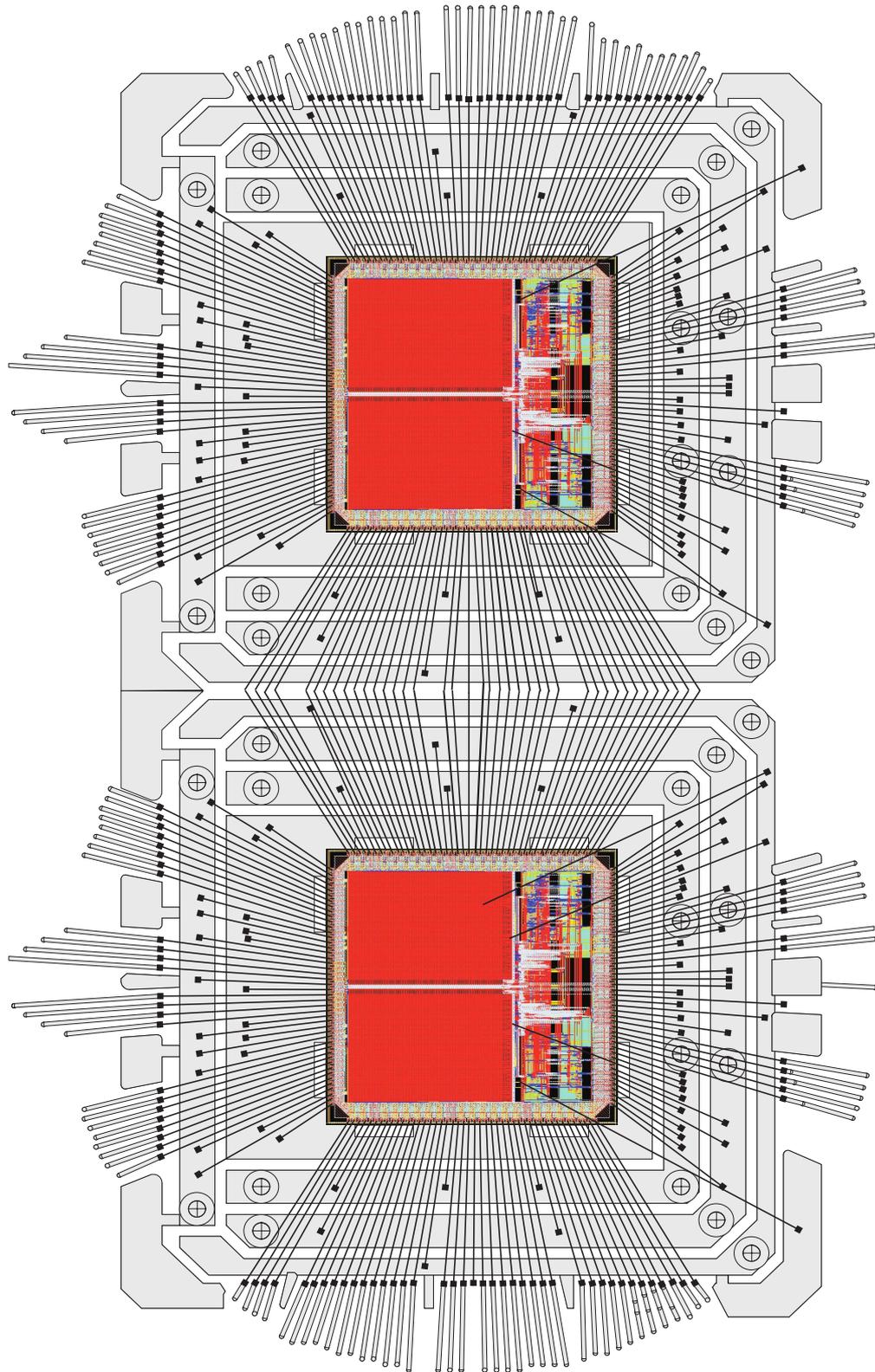


Abbildung A.5.: Der Bondingplan des Spikey-Footprints mit zwei Spikeys auf *Daja2*.

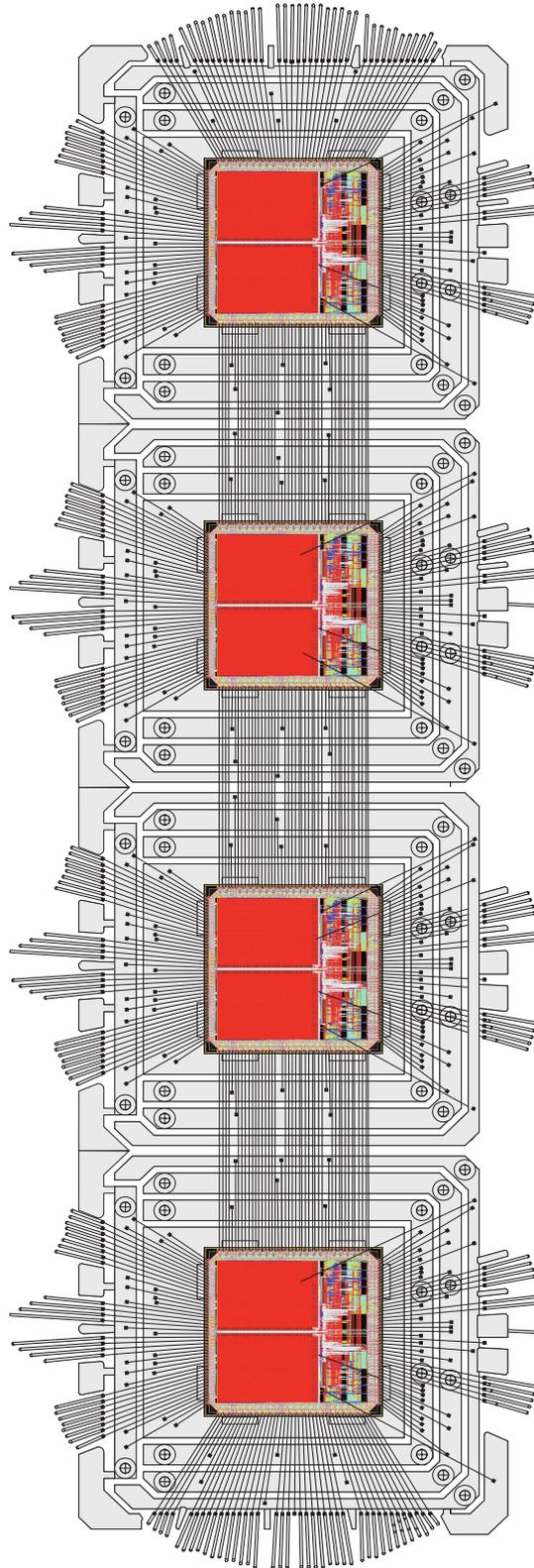
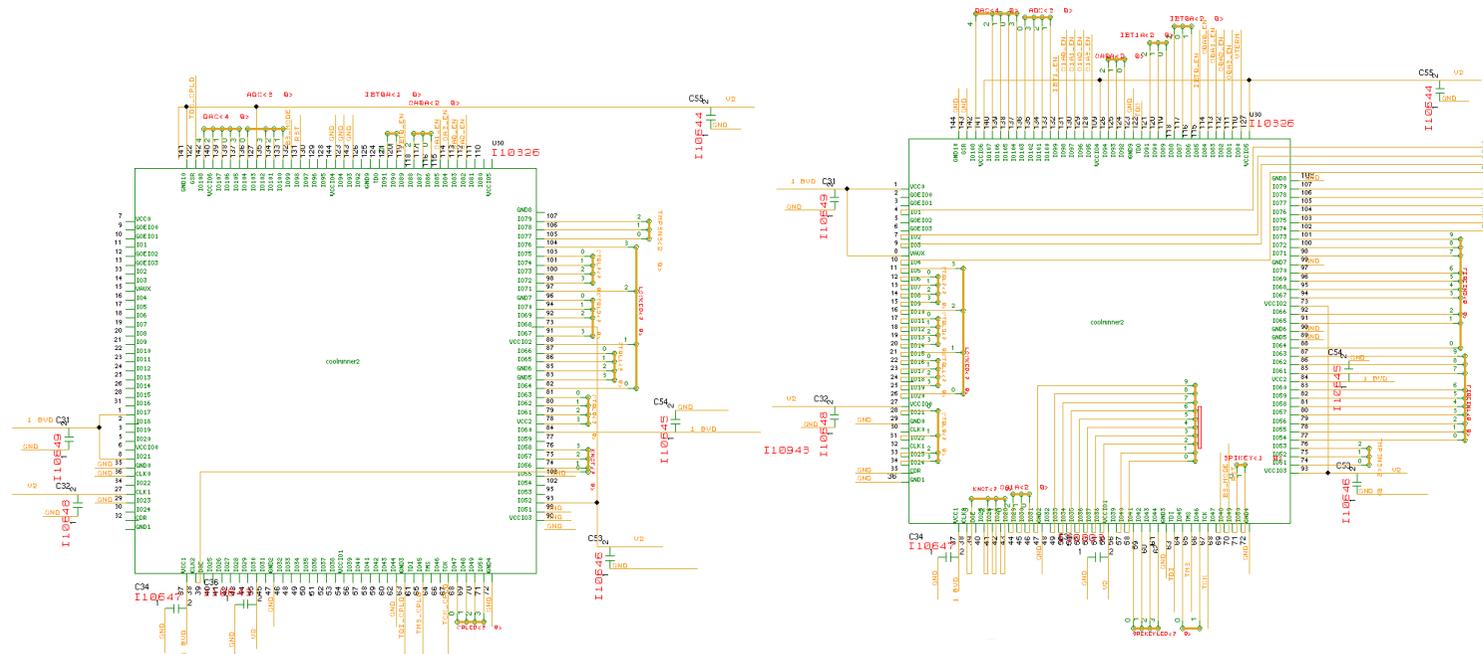


Abbildung A.6.: Der Bondingplan des Spikekey-Footprints mit vier Spikes auf *Daja4*.

B. Schaltungen

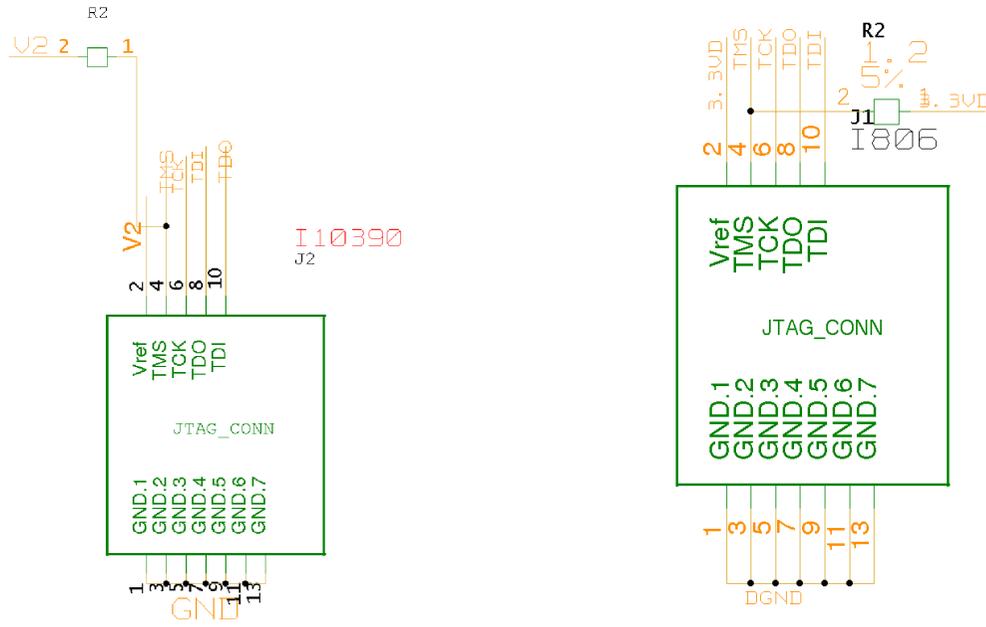
Die folgenden Schaltungen finden sich auf *Sittah* und *Daja*. Nicht dargestellt sind die Pins, die zur Messung der 36 Membranpotentiale oder zur Messung der Spannungen verwendet werden



(a) Der CPLD auf der alten Version von *Sittah*.

(b) Der CPLD auf der neuen Version von *Sittah*.

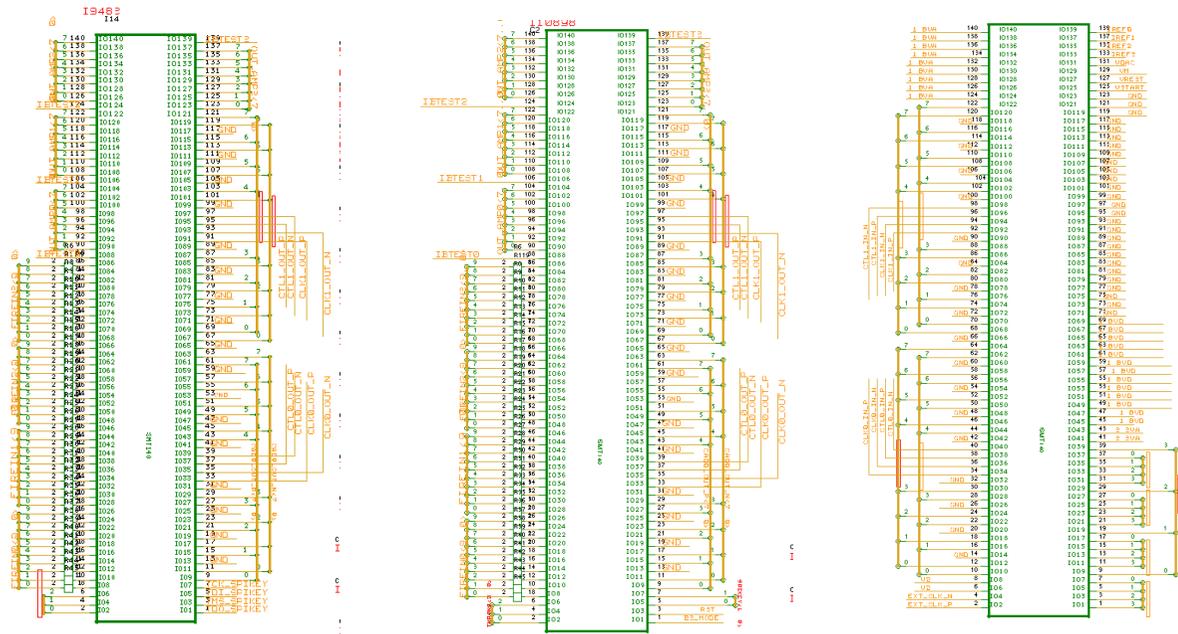
Abbildung B.1.: Der CPLD auf beiden Versionen von *Sittah*. Die Änderung besteht darin, dass Datenleitungen verschieden angeschlossen wurden.



(a) Die JTAG-Verbindung, wie sie verwendet wurde auf *Sittah*.

(b) Die JTAG-Verbindung auf der neuen Version von *Sittah*.

Abbildung B.2.: Die JTAG-Verbindung für den CPLD und den *Spikey*. Für *DajaSittahV2* wurde die JTAG-Verbindung für den *Spikey* entfernt, da er für *Spikey* kaum verwendet werden.

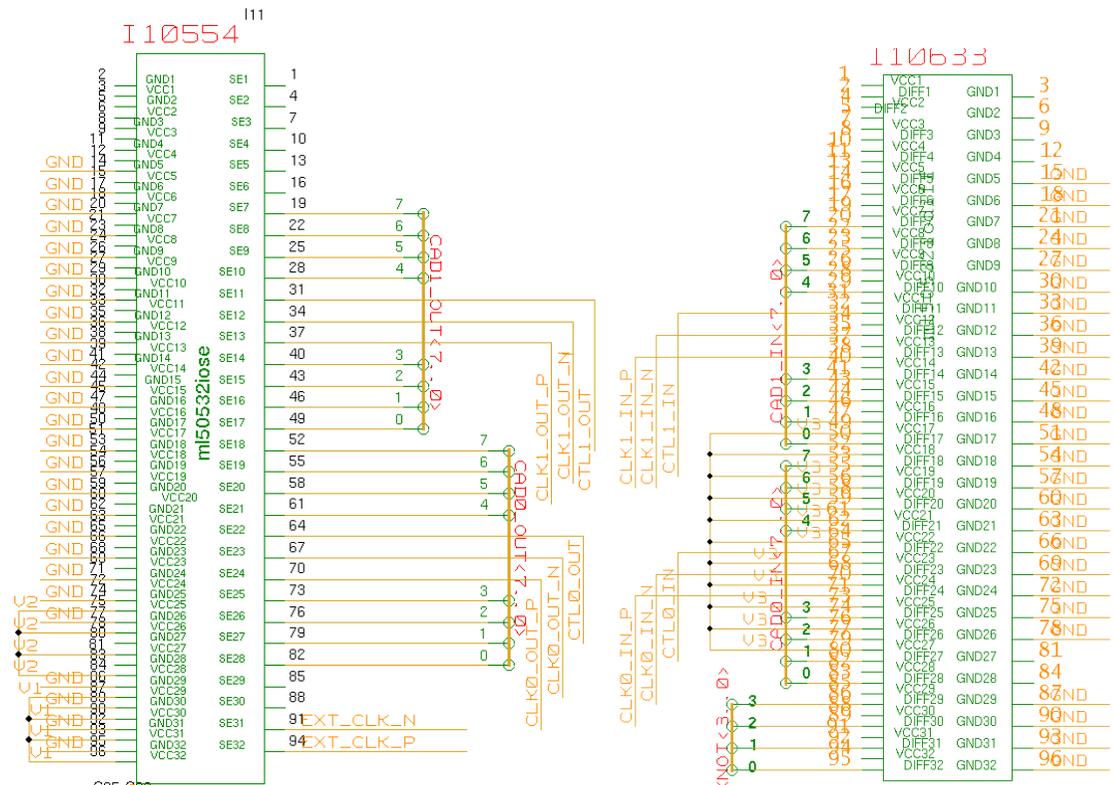


(a) Rechter SMT-Stecker, aus denen die Daten von *Sittah* zu *Daja* laufen.

(b) Linker SMT-Stecker, aus denen die Daten von *Daja* zu *Sittah* laufen.

(c) Linker SMT-Stecker, aus denen die Daten von *Daja* zu *Sittah* laufen.

Abbildung B.6.: SMT-Stecker auf beiden Seiten der Boards. Auf der rechten Seite laufen die Daten von *Sittah* nach *Daja*, auf der linken von *Daja* nach *Sittah* laufen.



(a) IO-Pinreihe, in denen die Daten vom *Sittah* zu (b) IO-Pinreihe, in denen die Daten vom FPGA-Board zu *Sittah* laufen

Abbildung B.7.: IO-Pinreihe auf *Sittah*, auf der die Daten zum und vom FPGA-Board laufen.

B. Schaltungen

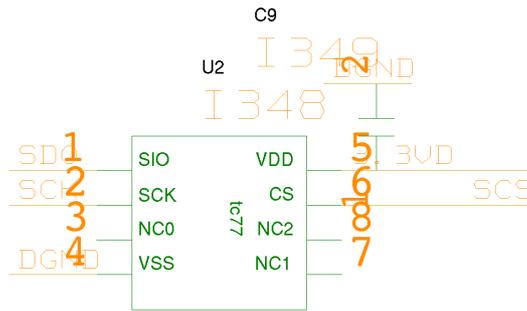


Abbildung B.8.: Temperatursensor auf beiden Versionen von *Daja*.

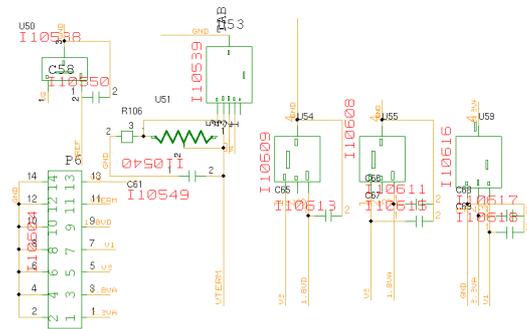


Abbildung B.9.: Die Spannungsregler auf *Sittah*.

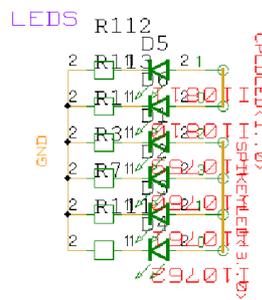
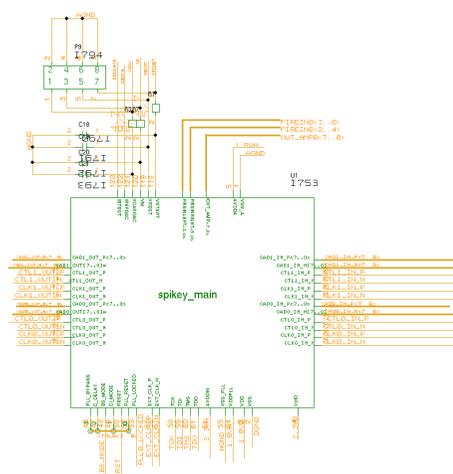
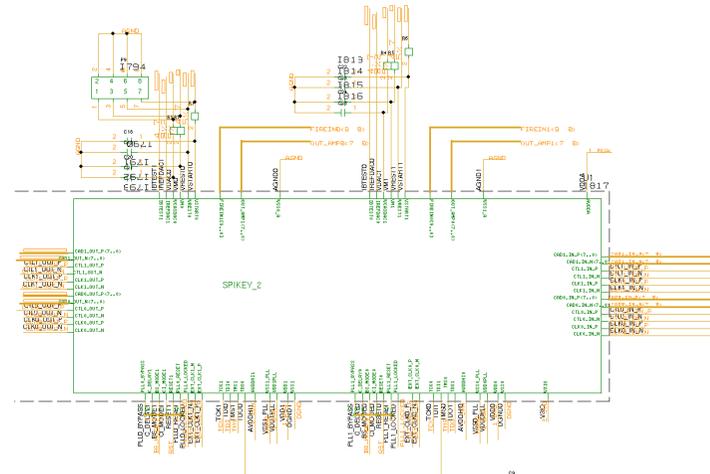


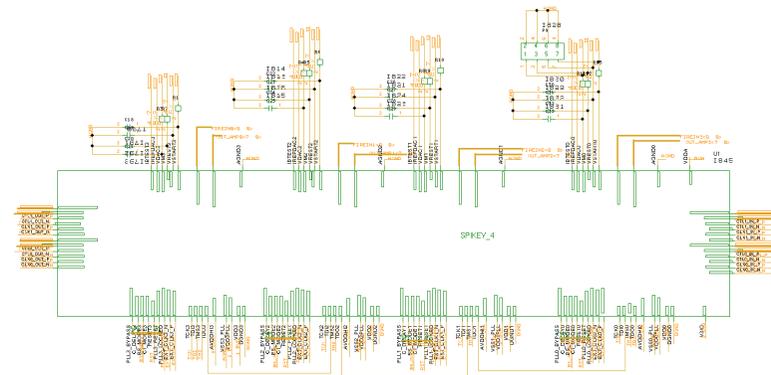
Abbildung B.10.: Die LEDs auf *Sittah*.



(a) Schaltung von einem *Spikey* auf *Daja*.



(b) Schaltung von zwei *Spikey* in einer Reihe.



(c) Schaltung von vier *Spikey* in einer Reihe.

Abbildung B.11.: Schaltungen für die *Spikey*-Chips auf allen Versionen von *Daja*.

B. Schaltungen

B.1. Lagenaufbau und wichtige Stromwerte

Die folgende Tabelle gibt den Lagenaufbau von *DajaSittah* an.

Lage	Dicke in <i>mm</i>	Material	Dielektizitätswert
TOP	0,035	Kupfer	
Zwischenlage	0,3	FR4	4,5
LAYER2	0,035	Kupfer	
Zwischenlage	0,3	FR4	4,5
LAYER3	0,035	Kupfer	
Zwischenlage	0,3	FR4	4,5
BOTTOM	0,035	Kupfer	

Tabelle B.1.: Lagenaufbau

LAYER2 wurde bei *Sittah* als Spannungs-Leitfläche verwendet, LAYER3 als MASSE. Bei *Daja* dagegen wird LAYER3 als Spannungs-Leitfläche, LAYER2 als MASSE verwendet.

C. Materialliste

Tabelle C.2 zeigt die notwendigen Bauteile zum Bau von *DajaSittah*; der Spikey ist nicht inbegriffen.

IC	Name	# ^a	Anwendung	Footprint	Hersteller	Preis ^b	Distributor	Bestellcode
Line Receiver	SN65LVDS2	18	Umwandlung der diff. Leitungen in Masse-bezogene	SOT235	TI	1.39	Farnell	1575586
Line Driver	SN65LVDS2	18	Umwandlung der Masse-bezogene Leitungen in diff.	SOT235	TI	1.11	Farnell	8452482
DAC	AD5668	1	Erzeugung der analogen Spannungen	TSSOP-16	AD	35.79	Farnell	1438925
ADC	AD7924	1	Umwandlung der Membranspannungen	TSSOP16	AD	9.05	Farnell	8397759
OP-AMP 2 Kanäle	OPA2355	1	Verstärkung der Membranspannungen	MSOP-10	TI	4.20	Farnell	1535740
OP-AMP 4 Kanäle	MAX4234	1	Verstärkung der Referenzstroms	TSSOP/SO	AD	2.00	MAXIM	1776145
Regulator 3.3V	REG1117	1	Erzeugung analoge 3.3V-Spannung	SOT223-3	TI	2.15	RS-Components	1212355
Regulator 1.8V	LD29300	2	Erzeugung analoge und digatale 1.8V-Spannung	TO-263	ST	1.14	RS-Components	686-9170
Referenz 1.8V	REF3318	1	Referenzspannung für ADC und DAC	TO-263	TI	2.32	Farnell	1755082
Regulator adj.	AIDBZT	1	Referenzspannung für ADC und DAC	TO-263	TI	2.32	Farnell	1755082
Regulator adj.	NCP565	1	Erzeugung der Terminierungsspannung	TO-263-5	ON	2.59	Farnell	1460682
MUX 4 Kanäle	ADG604	1	Multiplexen der IBTEST-Leitungen	TSSOP-10	AD	2.75	Farnell	9425756
MUX 8 Kanäle	ADG658	4	Multiplexen der OUT-AMP-Leitungen	TSSOP-16	AD	2.49	Farnell	1438947
CPLD	XC2C256	1	Setzen von Signalen	TQ144	Xilinx	25.29	Farnell	1776145
PMOS	BSS84	4	Erzeugung des Referenzstroms	SOT23		0,32	Farnell	1094997
LEDs		6	Anzeige	SMD0805			E-Werkstatt	1776145
Temperatursensor	TC77	1	Sensor für die Chips	SOIC	MICRO-WIRE	1.01	Farnell	1439513

Tabelle C.2.: Liste aller elektronischen Bauteile.

^aAnzahl^bin Euro pro Bauteil

IC	Name	Anzahl	Anwendung	Hersteller	Preis je Bauteil	Distributor	Bestellcode
Potentiometer	3296	1	Einstellung des Spannungsteilers	Bourns	2.47	E-Werkstatt	9316094
Pins 3x32 reihig	ESQ-132-12-G-T	2	Verbindung ML505 und <i>DajaSittah</i>	SAMTEC	16.00	Farnell	1769478
Pins 2 reihig		60	Messung von aller OUT-AMP-Kanäle				
JTAG-Connector		2	CPLD und <i>Spikey</i>				
Coaxial-Buchse	EPL.00.250.NTN	2	Messung OUT-AMP-Kanäle und IBTEST	LEMO	12.00	Farnell	1569230
Stiftleiste 140-polig		2	Verbindung <i>Sittah</i>	SMT	5.84	Farnell	361-6614
Buchsenleiste 140-polig		2	Verbindung <i>Daja</i>	SMT	4.84	Farnell	361-6838

Tabelle C.4.: Liste aller nicht elektronischen Bauteile.

D. Fehler auf der Platine

Während der Arbeit wurden folgende Fehler auf *DajaSittah* festgestellt.

Position	Fehler auf der Platine	Fehlerbehebung
HF-Buchse	Serienterminierung ist am Empfänger statt am Sender terminiert	Fehler ist nicht behoben Es wurde ein spezielles Programmierkabel verwendet, bei dem TDI und TDO vertauscht werden können
JTAG-CPLD-Pins	Die Pins für TDO und TDI sind vertauscht	V_{REF} wurde getrennt und mittel Fädeldraht an 1,8V angeschlossen
JTAG-CPLD V_{REF} Terminierungsspannung der LVDS-Leitungen	V_{REF} ist mit 3,3V statt 1,8V verbunden Terminierung zur Hälfte des Logikpegels funktioniert aus Prinzip nicht	Fehler kann nicht behoben werden Terminierung durch ein Fädeldraht von der Terminierungsspannung der LVDS-Bauteile
CPLD	Terminierungsspannung wurde nicht definiert	

Tabelle D.1.: Liste der Fehler auf *DajaSittah*.

Literaturverzeichnis

Analog Devices, *AD5628/AD5648/AD5668 Datasheet*, Wilhelm-Wagenfeld-Strasse 6, Munich, Bav 80807, Germany, a.

Analog Devices, *ADG7924 Datasheet*, Wilhelm-Wagenfeld-Strasse 6, Munich, Bav 80807, Germany, b.

Analog Devices, *AD8063 Datasheet*, 3 Technology Way, MA 02062, USA, c.

Analog Devices, *ADG604 Datasheet*, Wilhelm-Wagenfeld-Strasse 6, Munich, Bav 80807, Germany, d.

Analog Devices, *ADG658/ADG659 Datasheet*, Wilhelm-Wagenfeld-Strasse 6, Munich, Bav 80807, Germany, e.

Association, J. S. S. T., *Stub Series Terminated Logic for 2.5V*, 2002.

Davison, A. P., D. Brüderle, J. Eppler, J. Kremkow, E. Müller, D. Pecevski, L. Perinet, and P. Yger, PyNN: a common interface for neuronal network simulators, *Front. Neuroinform.*, 2(11), 2008.

Grübl, A., Eine FPGA-basierte Plattform für neuronale Netze, Diploma thesis (German), Ruprecht-Karls-Universität Heidelberg, HD-KIP-03-2, 2003.

Grübl, A., VLSI implementation of a spiking neural network, Ph.D. thesis, Ruprecht-Karls-Universität Heidelberg, document No. HD-KIP 07-10, 2007.

Kuruvilla, Gerberdaten zu dajasittah, 2010a.

Kuruvilla, Sourcecode zu dajasittah, 2010b.

Maxim Integrated Products, *MAX4312EEE Datasheet*, 120 San Gabriel Drive, Sunnyvale, CA 94086, USA, a.

Maxim Integrated Products, *MAX5253ACAP Datasheet*, 120 San Gabriel Drive, Sunnyvale, CA 94086, USA, b.

Microchip Technology, *TC77 Datasheet*, 2355 West Chandler Blvd., Chandler, AZ 85008, USA, 2002.

National Semiconductor, LVDS owner's manual, LVDS.national.com, 2004.

NXP Semiconductors N.V., *BSS84 Datasheet*, High Tech Campus 60, Eindhoven, AG 5656, Netherlands, 2008.

- ON Semiconductor, *NCP565 Datasheet*, 5005 East McDowell Road, Phoenix, AZ 85008, USA, a.
- ON Semiconductor, *REF3312, REF3318, REF3320, REF3325, REF3330, REF3333 Datasheet*, 12500 TI Boulevard, Dallas, Texas 75243, USA, b.
- Ostendorf, B., Charakterisierung eines Neuronalen Netzwerk-Chips, Diploma thesis (German), Ruprecht-Karls-Universität Heidelberg, HD-KIP 07-12, 2007.
- Paul Horowitz, W. H., *The Art of Electronics*, 20th ed., Cambridge University Press, New York, NY, USA, 2006.
- ST Microelectronics, *LD29300xx Datasheet*, 39 Chemin du Champ des Filles, Geneva, CA 95124, Switzerland, 2008.
- Texas Instruments, *REG1117 Datasheet*, 12500 TI Boulevard, Dallas, Texas 75243, USA, a.
- Texas Instruments, *SN65LVDS1, SN65LVDS2, SN65LVDT2 Datasheet*, 12500 TI Boulevard, Dallas, Texas 75243, USA, b.
- Xilinx, *Virtex-5 FPGA User Guide*, Xilinx, www.xilinx.com, 2009.

Danksagungen

In der Electronic Vision(s) Gruppe durfte ich mein letztes Jahr an der Ruperta Carola mit einem interessanten Thema beschäftigen. Es waren Mihai Petrovici und Daniel Brüderle, die mich dem Johannes Schemmel vorstellten. Dieser wiederum bot mir eine passende und spannende Diplomarbeit an.

Daneben konnte ich mich bei den zahlreichen aufkommenden Fragen, Verständnisschwierigkeiten, Verbesserungsvorschlägen und Ratschlägen ganz auf die Hardware-Gruppe verlassen. Insgesamt genoss ich die Unterstützung, Stimmung und den Humor der gesamten Electronic Vision(s) Gruppe.

Neben vielen anderen Mitgliedern in der Gruppe wäre ich wohl ratlos ohne die technische Unterstützung durch Eric Müller.

Ralf Achenbach übernahm das Bonden des Chips, sodass mir ein grosser Aufwand erspart blieb.

Vor allen anderen aber gilt meine Dankbarkeit Matthias Hock und Simon Friedmann für ihre aufopfernde Hilfe und Geduld. Tatsächlich könnte ich mir keine besseren Betreuer wünschen.

Statement of Originality (Erklärung):

I certify that this thesis, and the research to which it refers, are the product of my own work. Any ideas or quotations from the work of other people, published or otherwise, are fully acknowledged in accordance with the standard referencing practices of the discipline.

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

Heidelberg, April 28, 2011

.....
(signature)