

# Masterarbeit

René Pascal Fontes

Entwicklung eines autarken Präzisionsmessgerätes in FPGA Technologie zur Messung des Festkörperanteils in Mehrphasensystemen



René Pascal Fontes

Entwicklung eines autarken Präzisionsmess-  
gerätes in FPGA Technologie zur Messung des  
Festkörperanteils in Mehrphasensystemen

Masterarbeit eingereicht im Rahmen der Masterprüfung  
im gemeinsamen Studiengang Mikroelektronische Systeme  
am Fachbereich Technik  
der Fachhochschule Westküste  
und  
am Department Informations- und Elektrotechnik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr.-Ing. Robert Fitz  
Zweitgutachter : Prof. Dr.-Ing. Alfred Ebberg

Abgegeben am 30. April 2009



**René Pascal Fontes**

**Thema der Masterarbeit**

Entwicklung eines autarken Präzisionsmessgerätes in FPGA Technologie zur Messung des Festkörperanteils in Mehrphasensystemen

**Stichworte**

FPGA, ADC, DAC,  $\mu$ Controller, Microblaze, Klimawandel, Arktis, Meereis

**Kurzzusammenfassung**

Meereis stellt mit seinen veränderlichen Anteilen von festem Frischwassereis und eingeschlossener flüssiger Salzlake ein Zwei-Komponenten, Zwei-Phasen System dar. In dieser Masterarbeit wird die Entwicklung eines in-situ Messgerätes zur Messung des Anteils der jeweiligen Komponente beschrieben. Als Grundlage dient ein FPGA, über welchen ein digitales Messsignal erzeugt wird. Über Digital-Analog-Umsetzer und Analog-Digital-Umsetzer mit SPI-Schnittstelle sind eine entwickelte Hardware an den FPGA angebunden. Die entwickelte Hardware realisiert die analoge Messung zur Bestimmung des Festkörperanteils von Meereis. Hierzu werden bei Anlegen des Messsignals der resultierende Strom und die resultierende Spannung zwischen zwei im Meereis eingeschlossenen Platindrähten ermittelt. Ein  $\mu$ Controller auf der Hardware dient zur Steuerung von mehreren Messsensoren. Die entwickelte Hardware ist für den autarken Einsatz in arktischen Regionen konzipiert und auf die Messung der Phase und des Effektivwertes des resultierenden Stroms bzw. der resultierenden Spannung optimiert.

**René Pascal Fontes**

**Title of the master thesis**

On the Development of an Independent and Accurate Measuring Instrument Based on a FPGA for Measuring the Solid Fraction in Multiphase Systems

**Keywords**

FPGA, ADC, DAC,  $\mu$ Controller, Microblaze, Climate Change, Arctic, sea ice

**Abstract**

Sea ice with its changing fractions of solid freshwater-ice crystals and interstitial liquid saline brine is a two component, two-phase medium. This master thesis describes the development of an in-situ measuring instrument capable for measuring both fractions in sea ice. The measuring instrument is based on a FPGA to generate and analyse a digital measuring signal. To determine the solid fraction one digital-analog-converter and two analog-digital-converter measures the phase and the rms-value between two platinum wires. Additionally, there is a microcontroller on the newly developed hardware that controls multiple measurement sensors. The developed hardware is designed for the independent use in Arctic regions.



# Inhaltsverzeichnis

<b>Tabellenverzeichnis</b>	<b>III</b>
<b>Abbildungsverzeichnis</b>	<b>1</b>
<b>1 Einleitung</b>	<b>3</b>
<b>2 Vorüberlegungen für den Aufbau der Harp3d</b>	<b>9</b>
<b>3 Aufbau der Harp3d</b>	<b>17</b>
3.1 Gesamtaufbau . . . . .	17
3.2 Hardwareaufbau der einzelnen Module . . . . .	22
3.2.1 ML403 Development Board und Adapterplatine . . . . .	23
3.2.2 Basis Modul . . . . .	28
3.2.2.1 Spannungsversorgungsplatine . . . . .	30
3.2.2.2 Steuerungsplatine . . . . .	33
3.2.2.3 ADC/DAC-Platine . . . . .	35
3.2.3 Mess Modul . . . . .	39
3.3 FPGA und Custom-IP-Core . . . . .	42
3.3.1 FPGA . . . . .	43
3.3.2 Custom-IP-Core . . . . .	47
3.3.2.1 Schnittstelle zum IPIF-Interface . . . . .	48
3.3.2.2 SPI Interface . . . . .	52
3.3.2.3 Clock . . . . .	56
3.3.2.4 Signalerzeugung . . . . .	57
3.4 Programmablauf im FPGA und im $\mu$ Controller . . . . .	64
3.4.1 Interkommunikation zwischen Haupt- und Basismodul . . . . .	64
3.4.2 Programmablauf im $\mu$ Controller des Basismoduls . . . . .	66
3.4.3 Programmablauf im Microblaze des Hauptmoduls . . . . .	69
<b>4 Inbetriebnahme der Harp3d und erste Messergebnisse</b>	<b>73</b>
<b>5 Fazit und Ausblick</b>	<b>79</b>

<b>Literaturverzeichnis</b>	<b>80</b>
<b>A Anhang für Kapitel 4</b>	<b>87</b>
A.1 Schaltpläne und Layouts der realisierten Hardware . . . . .	87
A.2 Dimensionierung des Interpolations-Tiefpasses . . . . .	101
A.3 Dimensionierung des Antialiasing-Filters . . . . .	104
A.4 Custom-IP-Core . . . . .	108
<b>Versicherung über Selbstständigkeit</b>	<b>111</b>

# Tabellenverzeichnis

3.1	Bereitgestellte Spannungen der Spannungsversorgung. . . . .	30
3.2	Aus dem FPGA herausgeführte Anschlüsse. . . . .	46
3.3	Signal harp3d_data_select des Custom-IP-Cores. . . . .	50
3.4	Ausgangsregisterwerte der user_logic. . . . .	51
4.1	Stromverbrauch der einzelnen Komponenten. . . . .	78
A.1	Zuordnung der we-Signale zu den Eingangswerten des harp3d_data_select-Signals. . . . .	108
A.2	Verwendete Hardwareressourcen innerhalb des FPGA für den Custom-IP-Core. . . . .	108



# Abbildungsverzeichnis

1.1	Rückgang des Meereises in den Sommermonaten von 1987 (links) bis 2007 (rechts) [1]. . . . .	4
1.2	Vergleich Modelle und Messungen der Ausdehnung des Meereises [2]. . . . .	5
1.3	Messmodul der Harp3d. . . . .	6
2.1	Ersatzschaltbild des Messverfahrens. . . . .	10
2.2	Das bestehende System. . . . .	12
3.1	Schematische Übersicht des Gesamtaufbaus der Harp3d. . . . .	18
3.2	Gesamtaufbau der Harp3d (1: Hauptmodul, 2: Adapterplatine, 3: Basismodul, 4: Messmodule, 5: Messsensoren). . . . .	22
3.3	ML 403 Development Board ([3] Seite 12). . . . .	23
3.4	Schematischer Aufbau des Hauptmoduls. . . . .	24
3.5	Schematischer Aufbau der Adapterplatine . . . . .	26
3.6	Schematischer Aufbau des Basismoduls. . . . .	29
3.7	Aufbau der +5 V (oben) und -5 V (unten) Spannungsversorgung. . . . .	31
3.8	Anschlüsse am $\mu$ Controller mit zugehöriger Temperaturmessung. . . . .	34
3.9	Aufbau des DAC mit zugehörigem Interpolations-Tiefpass. . . . .	35
3.10	Frequenzgang (links) und Phasengang (links) des Interpolations-Tiefpasses [4].	36
3.11	Aufbau der Addiererstufe (rechts) und ein Teil des Antialiasing-Tiefpasses (links) der ADCs. . . . .	37
3.12	Aufbau des ADCs mit zugehörigem Antialiasing-Tiefpass. . . . .	38
3.13	Frequenzgang (links) und Phasengang (links) des Antialiasing-Tiefpasses [4]. .	38
3.14	Schematischer Aufbau des Messmoduls. . . . .	39
3.15	Aufbau der Messvorverstärker. . . . .	41
3.16	Microblaze Prozessor [5]. . . . .	42
3.17	Aufbau der Hardwarestruktur innerhalb des FPGAs. . . . .	44
3.18	Schematischer Aufbau des Custom-IP-Cores. . . . .	47
3.19	Signale des Custom-IP-Cores zum IPIF. . . . .	49
3.20	Signal harp3d_data_select des Custom-IP-Cores. . . . .	50
3.21	Ausgangsregisterwerte der user_logic. . . . .	51

3.22 Funktionsweise des ADCs [6]. . . . .	52
3.23 Signale am SPI-Interface. . . . .	53
3.24 Anschlüsse am SPI-Interface. . . . .	53
3.25 SPI-Ausgangsschnittstelle innerhalb des SPI-Interfaces. . . . .	54
3.26 SPI-Eingangsschnittstelle innerhalb des SPI-Interfaces. . . . .	55
3.27 Signale innerhalb der SPI-Eingangsschnittstelle. . . . .	56
3.28 Funktionsblock der Clock-Generation . . . . .	57
3.29 Ein- und Ausgangssignale der Signal-Generation. . . . .	57
3.30 Schematischer Aufbau des Signal-Generators. . . . .	58
3.31 Schematischer Aufbau der Signalerzeugung in der Signal-Generation. . . . .	59
3.32 Schematischer Aufbau der Sinussignalerzeugung. . . . .	60
3.33 Schematischer Aufbau der Rechteck- und Dreiecksignalerzeugung. . . . .	62
3.34 Schematischer Aufbau der Addiererstufe in der Signal-Generation. . . . .	63
3.35 Aufbau des Übertragungsprotokolls zwischen Haupt- und Basismodul. . . . .	65
3.36 Aufbau der Sensor ID. . . . .	65
3.37 Programmablauf im $\mu$ Controller. . . . .	67
3.38 Programmablauf im Microblaze. . . . .	69
3.39 Ablauf der Zwischenspeicherung der Messwerte. . . . .	71
4.1 Signal (links) und Spektrum (rechts) der Sinussignalerzeugung des Custom-IP-Cores. . . . .	74
4.2 Dreiecksignal (links) und Rechtecksignal (rechts) der Signalerzeugung des Custom-IP-Cores. . . . .	75
4.3 Sinusausgangssignal an der Spannungsmessung des Messmoduls; links: Sinussignal, mitte: Spektrum mit 2,9 kHz Nutzsinal (1) und 13 kHz Störsignal (2), recht: Spektrum mit 2,9 kHz Nutzsinal (1), 13 kHz Störsignal (2), 105 kHz Störsignal (3) und 205 kHz Störsignal (4). . . . .	75
4.4 Messsignal am differentiellen Eingang eines ADCs; oben: positiver Eingang des ADCs, unten: negativer Eingang des ADCs. . . . .	76
4.5 Störsignalauswirkung auf der Spannungsversorgung; links: +5 V Spannungsversorgung (oben) und Referenzspannung (unten) ohne Signale auf dem SPI-Interface, rechts: +5 V Spannungsversorgung (oben) und Referenzspannung (unten) mit Störsignalbeeinflussung durch Signale auf dem SPI-Interface. . . .	76

---

A.1	Schaltplan der Adapterplatine. . . . .	88
A.2	Layout der Adapterplatine von der Oberseite (oben) und von der Unterseite (un- ten). . . . .	89
A.3	Schaltplan der Spannungsversorgung des Basismoduls. . . . .	90
A.4	Schaltplan der Steuerplatine des Basismoduls. . . . .	91
A.5	Schaltplan der DAC/ADC Platine des Basismoduls. . . . .	92
A.6	Layout der Basismodul Platinen von der Oberseite (oben Spannungsversor- gungsplatine, mitte DAC/ADC Platine und unten Steuerungsplatine) . . . . .	93
A.7	Layout der Basismodul Platinen von der Unterseite (oben Spannungsversor- gungsplatine, mitte DAC/ADC Platine und unten Steuerungsplatine) . . . . .	94
A.8	Schaltplan der Platine für die Messsensoren der Platindrähte des Messmoduls. . . . .	95
A.9	Layout der Platine für die Messsensoren der Platindrähte des Messmoduls (rechts die Oberseite und links die Unterseite). . . . .	95
A.10	Schaltplan der Platine für die Adapterplatine der Messsensoren der Platindrähte des Messmoduls. . . . .	96
A.11	Layout der Platine für die Adapterplatine der Platindrähte des Messmoduls (rechts die Oberseite und links die Unterseite). . . . .	97
A.12	Schaltplan der Platine für die Messsensoren der Thermistoren des Messmoduls. . . . .	97
A.13	Layout der Platine für die Messsensoren der Thermistoren des Messmoduls (rechts die Oberseite und links die Unterseite). . . . .	98
A.14	Schaltplan der Platine für die Adapterplatine der Thermistoren der Platindrähte des Messmoduls. . . . .	99
A.15	Layout der Platine für die Adapterplatine der Thermistoren des Messmoduls (rechts die Oberseite und links die Unterseite). . . . .	100
A.16	Dimensionierung des Interpolations-Tiefpasses 1 [4]. . . . .	101
A.17	Dimensionierung des Interpolations-Tiefpasses 2 [4]. . . . .	102
A.18	Dimensionierung des Interpolations-Tiefpasses 3 [4]. . . . .	103
A.19	Dimensionierung des Antialiasing-Tiefpasses 1 [4]. . . . .	104
A.20	Dimensionierung des Antialiasing-Tiefpasses 2 [4]. . . . .	105
A.21	Dimensionierung des Antialiasing-Tiefpasses 3 [4]. . . . .	106
A.22	Dimensionierung des Antialiasing-Tiefpasses 4 [4]. . . . .	107



---

## KAPITEL 1

# Einleitung

---

Die hier vorgestellte Masterarbeit beschäftigt sich mit der Entwicklung eines Messgerätes, welches für die Messung des Festkörperanteils in Meereis gedacht ist. Da Meereis aus Frischwassereis besteht, in welchem Salzlake eingeschlossen ist, bildet Meereis ein Zwei-Komponenten System, wobei die zwei Komponenten wiederum in zwei unterschiedlichen Phasen (fest und flüssig) auftreten. Meereis ist somit ein Mehrphasensystem [7]. Das Messgerät ist auf der Grundlage eines Field-Programmable-Gate-Arrays (FPGA) aufgebaut und verfügt über eine variable Anzahl von Messsensoren zur Messung des Festkörperanteils.

Obwohl Meereis in den polaren Regionen nur etwa 1% der weltweiten Eismasse einnimmt, bedeckt es im Maximum doch 5% der nördlichen und 8% der südlichen Hemisphäre [8]. Es ist mit 1 m bis 3 m relativ dünn [9]. Grundsätzlich existieren drei Arten von Eis in der nördlichen Hemisphäre: langjähriges Eis, dreijähriges Eis und einjähriges Eis. Die Bedeckung der Wasserfläche mit Meereis schwankt dabei saisonal sehr stark. Dabei schwankt die Fläche von einjährigem Eis, welches sich in den Wintermonaten bildet, am stärksten, gefolgt vom dreijährigen Eis und dem langjährigen Eis.

Im globalen Klimasystem spielt Meereis unter anderem eine entscheidende Rolle, da es die einfallende Sonneneinstrahlung stärker reflektiert als offenes Wasser. Je weniger Meereis vorhanden ist, desto mehr Sonneneinstrahlung wird durch die offenen Wasserflächen absorbiert.

Aus diesem Grund kann das Fehlen von Meereis als ein Verstärkungsfaktor des Klimawandels angesehen werden [10]. Man spricht hier vom sogenannten Eis-Albedo Feedback. Eine Berücksichtigung des Meereises, insbesondere der saisonal schwankenden Fläche des einjährigen Meereises, ist in aktuellen Klimamodellen von entscheidender Wichtigkeit.

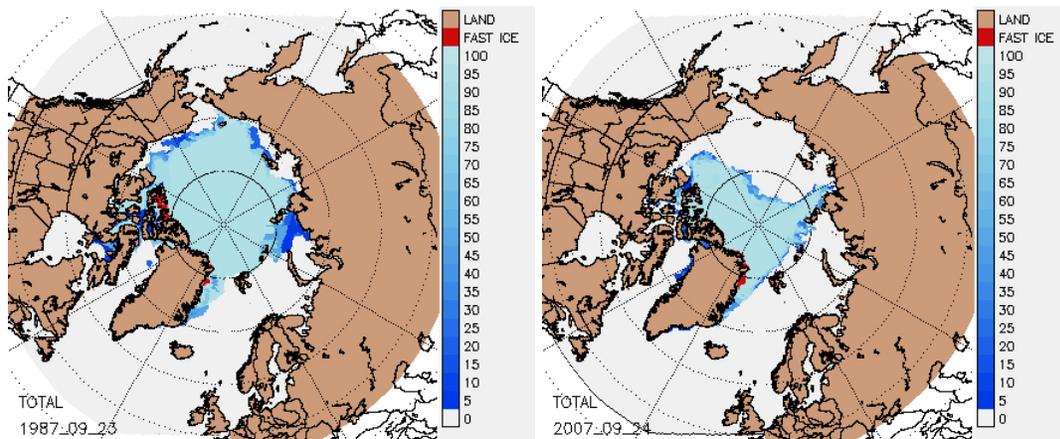


Abbildung 1.1: Rückgang des Meereises in den Sommermonaten von 1987 (links) bis 2007 (rechts) [1].

In den vergangenen Jahren ist das Meereis in der Arktis im Mittel immer weiter zurück gegangen. Im Jahr 2007 erreichte der Rückgang in der Arktis ein nicht vorhergesehenes Maximum (siehe Abb. 1.1). Zusätzlich wird das in den Wintermonaten neu entstehende Meereis immer dünner [9]. Dieser Rückgang wird sich nach heutiger Erkenntnis wahrscheinlich fortsetzen. In welchem Ausmaß, welcher Geschwindigkeit und ob sich dieser Rückgang überhaupt fortsetzen wird, lässt sich aber noch nicht hinreichend genau sagen.

Die durch den voraussichtlichen Rückgang des arktischen Meereises hervorgerufenen Veränderungen des Klimasystems Arktis - insbesondere auch als Lebensraum für viele Tierarten, sowie in der Arktis lebende Völker - sind bislang noch nicht absehbar.

Absehbar jedoch ist, dass die wirtschaftlich immer interessanter werdende Förderung von Bodenschätzen und die Nutzung der Nord-West-Passage für den Schiffsverkehr gravierende Auswirkungen auf das empfindliche Öko- und Klimasystem Arktis haben wird.

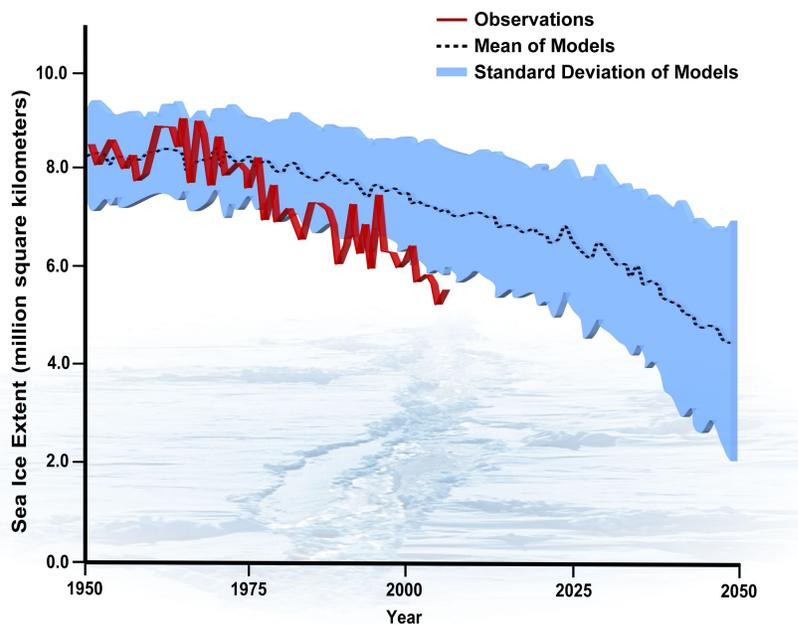


Abbildung 1.2: Vergleich Modelle und Messungen der Ausdehnung des Meereises [2].

Der rapide Rückgang des Meereises im Jahre 2007 konnte von keinem aktuellen Klimamodell vorhergesagt werden. Zwar sagen alle Modelle voraus, dass die Meereisausdehnung zurückgeht, weichen dabei aber von den reell gemessenen Daten noch stark ab (siehe Abb. 1.2). Dies liegt an der noch nicht ausreichend realistischen numerischen Darstellung der physikalischen Eigenschaften des Meereises in heutigen Klimamodellen. Aus diesem Grund ist die grundlegende Erforschung der physikalischen Eigenschaften von Meereis von Nöten, wobei insbesondere der Entstehungsprozess als auch der Schmelzprozess eine entscheidende Rolle spielt.

Eine Methode zur Messung der Entstehung von Meereis ist die Scheinwiderstandsmessung zwischen zwei im Meereis eingeschlossenen Platindrähten [11]. Durch die Platindrähte entsteht hierbei ein Elektrode-Elektrolyt-Elektrode Übergang. Bildet sich Meereis, so steigt der Scheinwiderstand mit dem Anteil des sich um die Platindrähte bildenden Frischwassereises. Hintergrund hierfür sind die vollkommen unterschiedlichen elektrischen und elektrochemischen Eigenschaften von Frischwassereis und Salzlake. So ist reines Eis z.B. elektrisch nicht leitend, Salzwasser aber ein guter Leiter.

Durch Anlegen einer kleinen Wechselspannung lässt sich der Scheinwiderstand aus der Spannung und dem resultierenden Strom ableiten. Diese Methode wird auch als AC-Methode

in der Elektrochemie [12], oder als AC-Elektroden-Polarisation in der Medizin bezeichnet [13]. Der resultierende Scheinwiderstand ist hierbei abhängig von der Temperatur, dem Salzgehalt, dem Abstand und der Länge der Platindrähte, der angelegten Spannung und der Frequenz der angelegten Spannung [12; 14]. Dieses resultiert aus dem komplexen Ersatzschaltbild zwischen den Platindrähten, welches mehrere Kapazitäten und Widerstände umfasst. Unter anderem wird diese Methode auch in der Medizin zur Salzgehaltsbestimmung im Blut eingesetzt [13]. In der Elektrochemie ist dieses Messverfahren eine gängige Methode zur Bestimmung des Salzgehaltes in Flüssigkeiten.

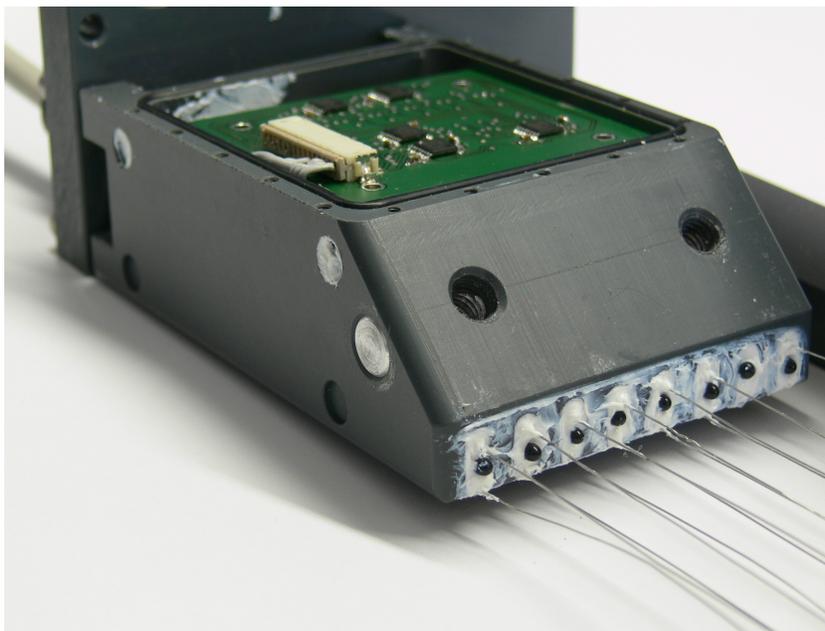


Abbildung 1.3: Messmodul der Harp3d.

Zur Quantifizierung der Entstehung von Meereis ist die AC-Methode bereits erfolgreich getestet worden [15]. Das verwendete in-situ Messgerät wurde am Max-Planck-Institut für Meteorologie in der selbstständigen Nachwuchsgruppe „Meereis im Erdsystem“ unter der Leitung von Dirk Notz weiter entwickelt (Harp3d). Die Harp3d wurde so entwickelt, dass mehrere Bausteine eine den jeweiligen Umständen angepasste flexible räumliche Anordnung mit einer unterschiedlichen Anzahl von Messsensoren ermöglichen [16].

Im Rahmen dieser Masterarbeit wurde hierzu der elektronische Teil der Harp3d neu entwickelt. Ziel des Neuaufbaus war hierbei, die Messgenauigkeit des bestehenden Systems zu verbessern, die Möglichkeit der Phasenmessung zu integrieren, um die Impedanz messen zu können und den Aufbau möglichst kompakt zu halten. Das Messgerät soll in Zukunft autark zu

betreiben und für arktische Temperaturen ausgelegt sein.

Der Aufbau der elektrotechnischen Komponenten ist auf Basis eines FPGAs mit zugehöriger Peripherie realisiert. Durch die Änderung auf Basis eines FPGAs soll erreicht werden, dass neben der AC-Methode in der Zukunft auch weitere Messmethoden eingesetzt werden können.

Kapitel 2 dieser Masterarbeit gibt einen kurzen Überblick über die theoretischen Grundlagen der verwendeten Messmethode. Das bereits existierende Messgerät wird vorgestellt und die Anforderungen an den Neuentwurf dargestellt.

Kapitel 3 gibt einen ausführlichen Überblick des neu entwickelten, elektronischen Bereichs der Harp3d. Hierzu gehört die entworfene Hardware zur Anbindung an den FPGA. Die entworfene Hardware ist in einzelne Module unterteilt. Hierdurch ist es möglich, eine variable Anzahl an Messsensoren zu nutzen. Es wird der Aufbau innerhalb des FPGAs betrachtet, welcher die Signalerzeugung und Aufzeichnung der Messung beinhaltet. Im Anschluss wird die realisierte Software auf dem FPGA und einen genutzten  $\mu$ Controller vorgestellt. Die realisierte Software steuert den Messablauf.

Kapitel 4 befasst sich mit ersten Messergebnissen des realisierten Aufbaus. Hierzu gehört auch die Fehleranalyse der realisierten Hardware. Auf der realisierten Hardware sind noch Störsignale vorhanden, welche im Anschluss an diese Masterarbeit beseitigt werden.

Kapitel 5 liefert eine Zusammenfassung dieser Masterarbeit und gibt einen Ausblick auf den Einsatz und mögliche Erweiterungen der Harp3d.



# Vorüberlegungen für den Aufbau der Harp3d

---

Dieses Kapitel befasst sich mit Vorüberlegungen für den Aufbau der Harp3d. Hierzu gehören ein Überblick über das verwendete Messverfahren zur Bestimmung der Impedanz. Dieses wird im ersten Abschnitt dargestellt. Als Grundlage für den Neuentwurf diente ein bereits vorhandenes Messgerät. Die Funktionsweise und die möglichen Verbesserungen werden in einem zweiten Abschnitt behandelt. Aus den möglichen Verbesserungen ergeben sich direkt die Anforderungen an einen Neuentwurf, welche im dritten Abschnitt behandelt werden und Gegenstand dieser Masterarbeit sind.

Der Unterschied zwischen Meereis und Frischwassereis, wie man es auf Seen kennt, liegt in dem Eis eingeschlossenen Salz, was man als Salzlake oder Brine bezeichnet. Meerwasser hat im Mittel einen Salzanteil von 35 ‰. Bei der Entstehung von Meereis entsteht reines Frischwassereis. In diesem Frischwassereis eingeschlossen, befinden sich Salzlaken. Beim Entstehungsprozess des Frischwassereises, wird das im Salzwasser befindliche Salz an die umgebenden Salzlaken abgegeben. Hiermit steigt dessen Salzgehalt. Meereis stellt somit ein poröses Medium aus Eis und Einschlüssen aus Salzlake dar. Mit sinkender Temperatur bildet sich immer mehr Frischwassereis und es wird immer mehr Salz an die Salzlaken abgegeben. Hierdurch sinkt dessen Gefrierpunkt, weshalb diese auch bei Temperaturen weit unter 0°C

noch flüssig sind. Im Laufe der Zeit sickern die Salzlaken immer weiter nach unten, so dass bei mehrjährigem Eis die oberen Schichten aus reinem Frischwassereis bestehen.

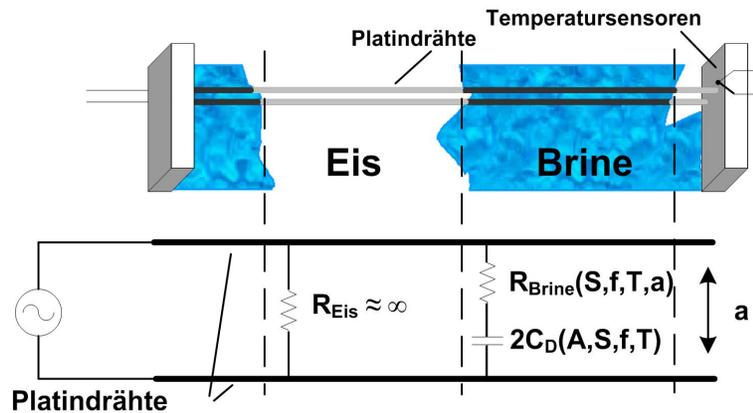


Abbildung 2.1: Ersatzschaltbild des Messverfahrens.

Die Bestimmung des Anteils von festem zu flüssigem Anteil im Meereis, also von reinem Frischwassereis zu eingeschlossener Salzlake, kann über eine Impedanzmessung zwischen zwei im Meereis befindlichen Platindrähten erfolgen (siehe Abb. 2.1). Platindrähte werden unter anderem deswegen verwendet, weil Platin nicht reaktiv ist. Andere Metalle würden durch das Salzwasser innerhalb kürzester Zeit oxidieren. Bei Anlegen einer kleinen Wechselspannung an die Platindrähte bildet sich um die Platindrähte eine so genannte Double-Layer-Kapazität ( $C_D$ ). Diese Double-Layer-Kapazität bildet sich nur innerhalb der Salzlaken. Durch die angelegte Spannung an die Platindrähte entsteht eine geladene Oberfläche mit positiven Ionen. Negative, freie Ionen des Salzes richten sich hierdurch an der Oberfläche der Platindrähte aus. Es entsteht eine geordnete Ionenausrichtung, welche sich direkt an der unmittelbaren Oberfläche der Platindrähte befindet und nur einige  $\mu\text{m}$  breit ist [17]. Die Größe der Double-Layer-Kapazität ist abhängig von der Temperatur ( $T$ ), dem Salzgehalt ( $S$ ), sowie der Amplitude ( $A$ ) und der Frequenz ( $f$ ) der angelegten Spannung [14]. Die angelegte Frequenz und Amplitude der Wechselspannung beeinflussen die Bildung der geladenen Oberfläche der Platindrähte. Der Salzgehalt und die Temperatur beeinflussen die im Salz befindlichen Ionen in ihrer Ausrichtung. Neben der Double-Layer-Kapazität befindet sich zwischen den Platindrähten noch der endliche Zellenwiderstand. Durch die Salzlake entsteht ein leitendes Medium. Dieser rein ohmsche Widerstand ist ebenfalls abhängig von dem Salzgehalt ( $S$ ), der Frequenz ( $f$ ), der Temperatur ( $T$ ) und zusätzlich vom Abstand zwischen den Platindrähten ( $a$ ). Dieses Ersatzschaltbild gilt für Frequenzen im kHz Bereich und für angelegte Spannungen

---

zwischen 20 mV - 150 mV und wird als Conductivity-Cell bezeichnet [12]. Die Bestimmung der Größen der Double-Layer-Kapazität und des Zellenwiderstandes wird dabei durch eine Kalibrationslösung mit definierter Leitfähigkeit bestimmt. Eine numerische Berechnung ist nicht möglich. Bei dem hier vorgestellten Messgerät mit Platindrähten von 15 cm Länge und  $0,3 \text{ mm}^2$  Durchmesser befinden sich die Absolutwerte der Impedanz zwischen  $5 \Omega$  in reinem Salzwasser und bis zu  $100 \Omega$  in entstehendem Eis. Der maximale Wert des Absolutwertes ist theoretisch unendlich, da reines Frischwassereis kein Salz enthält. Durch Fehlen des Salzes kann sich keine Double-Layer-Kapazität oder Zellenwiderstand bilden. Diese Gegebenheit wird genutzt, um den Festkörperanteil, sprich den Anteil des Frischwassereises im Meereis zu bestimmen [18]. Das angewendete Messverfahren durch Anlegen einer kleinen Wechselspannung wird als AC-Methode bezeichnet. Neben diesem Messverfahren existieren andere elektrochemische Messverfahren. So wird z.B. mittels der A.C. Polarography durch Anlegen einer Dreiecksspannung mit einer Amplitude von einigen Volt und überlagerter kleiner Wechselspannung die chemische Zusammensetzung des umgebenden Mediums bestimmt [17]. In Zukunft soll dieses Verfahren unter anderem mit dem Neuaufbau des Messgerätes auch für Meereis getestet werden.

Als Grundlage für den Neuentwurf der Harp3d diene ein bestehendes Messsystem [19; 15], welches hier kurz vorgestellt werden soll. Der elektronische Teil des bestehenden Messsystems ist im Rahmen dieser Arbeit vollkommen neu aufgebaut worden. Als Anforderungen an den Neuaufbau dienten hierzu teilweise Erkenntnisse aus dem Design des bestehenden Systems.

Die bestehende Harfe ist ein Messgerät, welches den Betrag der Impedanz (Scheinwiderstand) zwischen zwei parallelen sehr dünnen Platindrähten misst. Hieraus lässt sich aufgrund der verschiedenen Leitfähigkeiten der im Meereis vorhandenen Salzlaken und des umgebenden Süßwassereises der Feststoffanteil von Meereis erstmals auch in Feldexperimenten in-situ bestimmen. Mittels der gleichzeitig gemessenen Temperatur folgt hieraus direkt der Salzgehalt von Meereis. Es existieren mehrere dieser Platindrahtpaare in horizontaler und vertikaler Ausrichtung. Insgesamt ist es somit möglich, ein dreidimensionales Profil des Salzgehaltes vom umgebenden Meereis zu berechnen.

Der elektrische Aufbau des bestehenden Systems (siehe Abb. 2.2) beruht auf der Scheinwiderstandsmessung zwischen insgesamt 16 Platindrahtpaaren (unterer Bereich). Der Scheinwiderstand wird über die Messung des Stroms und der Spannung von analog aufgebauten

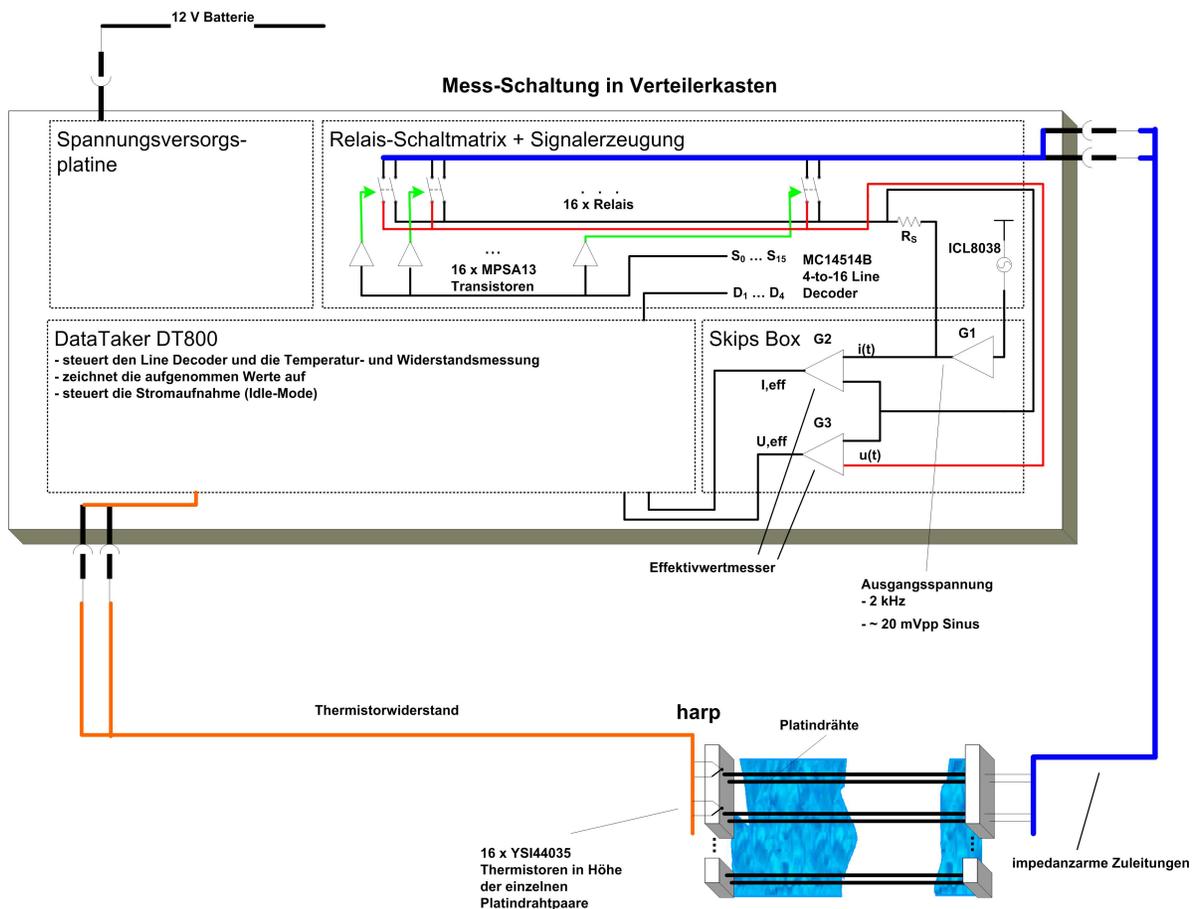


Abbildung 2.2: Das bestehende System.

Effektivwertmessern (innerhalb der Skips Box) berechnet. Ein Sinusgenerator (ICL8038, linker oberer Bereich) erzeugt hierzu eine Sinusspannung von 2 kHz. Über einen Vorverstärker in der Skips Box wird das Signal auf etwa  $20 \text{ mV}_{pp}$  verstärkt. Das so erzeugte Messsignal wird über einen Messwiderstand ( $R_S$ ) auf eine Schaltmatrix aus 16 Relais geleitet. Die Relais leiten das Messsignal über impedanzarme Zuleitungen an die 16 Platindrahtpaare. Der rückführende Pfad des Messsignals wird ebenfalls über die Relais gesteuert an den Effektivwertmesser der Spannung geführt. Der Strom wird über den Messwiderstand bestimmt. Die gemessenen Werte des Stroms und der Spannung werden durch einen DataTaker DT800<sup>1</sup> aufgezeichnet. Die Steuerung der Relais erfolgt über digitale Ausgänge des Datataker DT800 und einem 4-to-16 Line-Decoder. Der Line-Decoder wandelt vier digitalen Ausgänge des Datataker DT800 in 16

<sup>1</sup>Bei dem DataTaker DT800 handelt es sich um einen programmierbaren Messrechner in kompakter Bauform mit mehreren analogen Messeingängen. Über die Messeingänge können unter anderem Spannungen, Widerstände und Ströme gemessen werden. Über digitale Ausgänge kann externe Peripherie gesteuert werden. Die Aufzeichnung erfolgt auf einer Compact-Flash-Karte.

Steuersignale zur Ansteuerung von 16 Transistoren um. Die Transistoren dienen als Schalter zur Ansteuerung der Relais. Aus den gemessenen Werten für Strom und Spannung lässt sich der Betrag der Impedanz errechnen

$$|\underline{Z}| = \frac{U_{eff}}{I_{eff}}. \quad (2.1)$$

Die Widerstandswerte von insgesamt 16 Thermistoren werden ebenfalls über den Datataker DT800 gemessen und aufgezeichnet. Hierzu sind für jeden Thermistor eine eigene Zuleitung am Datataker DT800 aufgelegt. Über eine Spannungsversorgungsplatine werden alle benötigten Spannungen erzeugt. Um Strom zu sparen, kann der Datataker DT800 in einen Idle-Mode wechseln. Dies ist von Vorteil, da Messungen nicht kontinuierlich durchgeführt werden, sondern durch Pausen unterbrochen sind.

Ansatzpunkte für eine Verbesserung des bestehenden Systems waren, dass das System mit den Zuleitungen für die Thermistoren und Platindrahtpaaren statisch aufgebaut ist und hiermit zu unflexibel für geplante Aufgaben ist. Eine Erweiterung der Messsensoren ist nur schwer zu realisieren bzw. insbesondere die Anbindung der Thermistoren auch durch die Anzahl der Eingänge am Datataker DT800 beschränkt. Das bestehende System ist zudem dafür ausgelegt, dass nur ein Modul mit Messsensoren an ihm angeschlossen werden kann. In Zukunft sollen mehrere solcher Module eingesetzt werden. Dies ist mit dem bestehenden System nicht möglich. Eine Fehlerquelle bei dem Aufbau sind die Zuleitungen zu den Platindrahtpaaren. Der Scheinwiderstand zwischen den Platindrähten erreicht in Salzwasser bei Temperaturen über 0°C Werte von lediglich etwa 5 Ω [16]. Bei solchen niedrigen Scheinwiderstandswerten tragen die Zuleitungen und die Verteilung der Messsignale über Relais entscheidend zur Messgenauigkeit bei. Dies betrifft insbesondere die Temperaturabhängigkeit der Zuleitungen und Relais. Durch den Einsatzbereich in arktischen Regionen sind Temperaturschwankungen von bis zu 30°C nicht ausgeschlossen. Neueste Ergebnisse [16; 20] haben darüber hinaus gezeigt, dass die Messung der wirklichen Impedanz von entscheidender Wichtigkeit sind. Es hat sich gezeigt, dass sich der Wirkwiderstand und der Blindwiderstand bei der Entstehung von Meereis unterschiedlich verhalten. Es ist zu vermuten, dass eine Unterscheidung das grundlegende Verständnis über die Entstehung von Meereis entscheidend verbessert. Darüber hinaus sollen in Zukunft andere Messmethoden, wie die AC-Polarography, getestet werden. Hierfür ist die Erzeugung eines anderen Messsignals notwendig. Für geplante Messkampagnen sollen weitere Messsensoren, wie Strömungsmesser oder eine Anbindung an eine Satellitenkommunikation eingesetzt werden. In dieses Gesamtsystem muss der Neuentwurf integrierbar sein bzw. soll der der Neuentwurf so erweiterbar sein, dass es weitere

Messsensoren aufnehmen und eine Anbindung an andere Peripherie gewährleisten kann. Neben diesen elektrotechnischen relevanten Änderungen sind umfangreiche mechanische Änderungen am Aufbau notwendig. Diese sind nicht Bestandteil dieser Masterarbeit und werden durch Dritte ausgeführt [16].

Aus den genannten Ansatzpunkten für eine Verbesserung ergeben sich die Zielsetzung des Neuentwurfs des elektronischen Teils der Harp3d. Zu den Grundanforderungen gehören hierbei die Realisierung einer

- Wechsellspannungssignalerzeugung,
- Digitalisierung des Messsignals,
- Messsignalvorverarbeitung und
- Speicherung des Messsignals auf einem wechselbaren Datenträger.

Neben diesen Grundanforderungen sind weitere Randbedingungen des Neuentwurfs, dass die

- Messgenauigkeit verbessert wird. Die Harp3d soll das bestehende System in der Messgenauigkeit verbessern. Hierzu gehören, dass der Neuentwurf die Messung der Phasenverschiebung mit berücksichtigen soll, sowie die Messbeeinflussung durch die Zuleitungen zu den einzelnen Platindrähten minimiert werden. Zusätzlich soll die Messgenauigkeit über einen Temperaturbereich bis  $-20^{\circ}\text{C}$  möglichst genau sein.
- optimalen Messparameter (wie Frequenz, Signalform und Amplitude) der Wechsellspannungsquelle frei bestimmbar sind. Hierzu soll bei dem Neuentwurf das Ausgangssignal der Wechsellspannungsquelle möglichst flexibel gestaltet werden. Darüber hinaus soll für weitere Messmethoden die Wechsellspannungsquelle durch andere Messsignale erweiterbar sein.
- Harp3d autark über einen längeren Zeitraum mit Hilfe einer Batterie in Feldexperimenten in Polarregionen betrieben werden kann. Aus diesem Grund muss die aufgenommene Leistung so gering wie möglich sein. Zudem müssen die elektronischen Komponenten für Temperaturen bis  $-40^{\circ}\text{C}$  ausgelegt sein.
- Einbindung weiterer Peripheriegeräte, wie Wetterstation, GPS-Empfänger und satellitengestützte Datenübertragung zu einem späteren Zeitpunkt möglich sein soll. Hierzu müs-

sen bei dem Neuentwurf diverse externe Schnittstellen, wie RS-232, USB und Ethernet, bereits mit integriert sein.

- Einbindung eines austauschbaren Speichermediums möglich ist, um die Messdaten aufzuzeichnen.
- Anzahl der später eingesetzten Platindrahtpaare mit zugehörigen Temperaturfühlern variabel gehalten werden kann. Aus diesem Grund, muss die Messung der Impedanz eines Platindrahtpaares bzw. die Messung eines Temperaturfühlers ebenfalls individuell und programmierbar erfolgen.

Zur Realisierung dieser Anforderungen wurde der Neuentwurf auf Basis eines FPGAs gewählt. Dies gewährleistet ein höchstes Maß an Flexibilität. Zur Anbindung an den FPGA wurde eine Hardware entworfen, welche modular aufgebaut ist. Hierdurch ist eine flexible Anzahl von Modulen, welche die Messsensoren enthalten, möglich. Dieser so realisierte Neuentwurf wird im nachfolgenden Kapitel ausführlich beschrieben.



# Aufbau der Harp3d

---

Das folgende Kapitel beschäftigt sich mit dem realisierten Aufbau der Harp3d. Der Gesamtaufbau ist in einzelne Module unterteilt, um eine bestmögliche Performance für das Gesamtsystem zu erreichen. Im ersten Unterkapitel wird hierzu der Gesamtaufbau des realisierten Systems mit den einzelnen Modulen vorgestellt. Die beiden folgenden Unterkapitel beschäftigen sich mit dem Aufbau der einzelnen Module und dem FPGA der Harp3d. Das letzte Unterkapitel beschäftigt sich mit dem Programmablauf der realisierten Software innerhalb der einzelnen Module und auf dem FPGA.

## 3.1 Gesamtaufbau

Aufgrund der Anforderungen für die Realisierung aus Kapitel 2 ist der Gesamtaufbau der Harp3d durch einzelne Module realisiert (siehe Abb. 3.1). Diese sind

- das Hauptmodul,
- die Basismodule und
- die Messmodule.

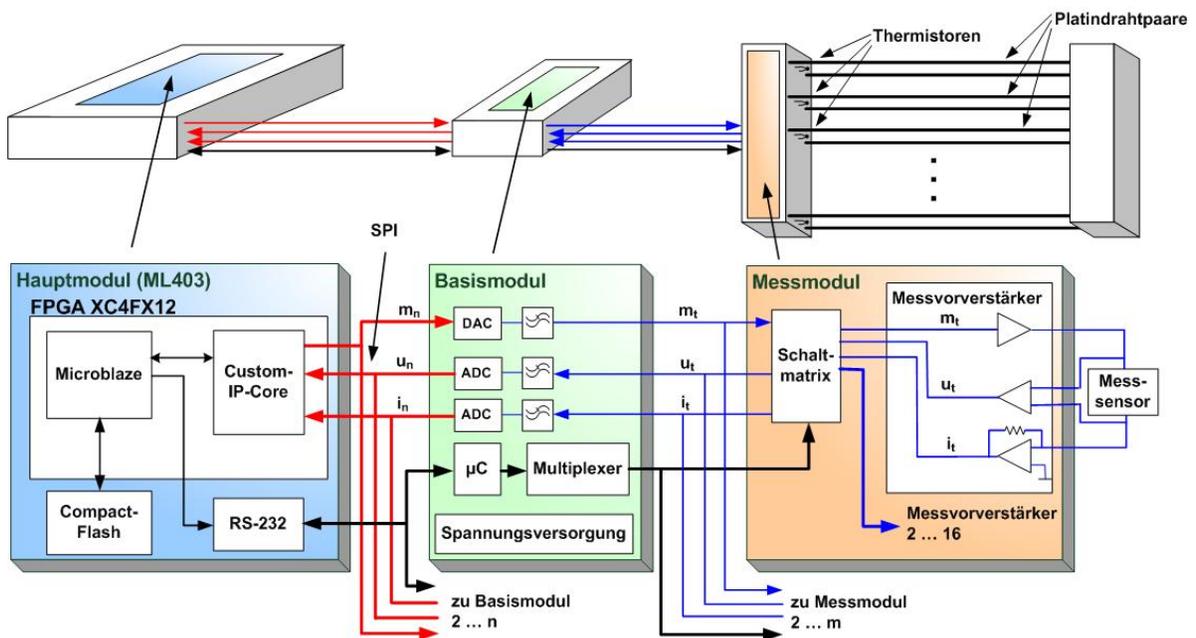


Abbildung 3.1: Schematische Übersicht des Gesamtaufbaus der Harp3d.

Die Aufteilung ist so vorgenommen, dass innerhalb des Hauptmoduls ein digitales Messsignal erzeugt, über das Basismodul analogisiert und an das Messmodul weitergegeben wird. Innerhalb des Messmoduls werden über einzelne Messsensoren, welche ein Thermistor oder ein Platindrahtpaar sein können, Strom und Spannung gemessen und an das Basismodul zurückgeführt. Dieses digitalisiert die analogen Messsignale und führt es an das Hauptmodul zurück, in dem es ausgewertet und abgespeichert wird.

Der modulare Aufbau ist so konzipiert, dass die Anzahl der jeweiligen Module pro Messaufbau variabel gehalten werden kann. Somit ist es möglich, sich an die konkreten Messaufgaben möglichst gut anzupassen.

Es können an jedes Basismodul bis zu acht Messmodule mit jeweils 16 Messsensoren angeschlossen werden. Dies ermöglicht es, pro Basismodul 128 Messsensoren anzuschließen. Je Hauptmodul ist es theoretisch möglich beliebig viele Basismodule anzuschließen. Praktisch limitiert die Busanbindung zwischen Haupt- und Basismodul den Einsatz auf 32 Basismodule. Diese Beschränkung kann durch den Einsatz von Zwischenverstärkern umgangen werden. In der Praxis werden jedoch voraussichtlich lediglich bis zu vier Basismodule mit zugehörigen Messmodulen eingesetzt. Durch den modularen Aufbau können die Messmodule, in denen die Messsensoren untergebracht sind, mechanisch klein gehalten werden. Dies stellt sicher, dass das Messmodul das ihn umgebene Medium (Meereis) so wenig wie möglich bei der Entste-

hung beeinflusst. Die analoge Signalübertragung zwischen Basis- und Messmodul ist durch Leitung realisiert, dies stellt einen notwendigen Kompromiss dar. Eine direkte Digitalisierung auf den einzelnen Messmodulen wäre zu teuer. Zudem würde sich hierdurch der mechanisch kleine Aufbau der Basismodule wesentlich vergrößern. Aus diesem Grund müssen impedanzarme und geschirmte Leitungen zwischen Basis- und Messmodul verwendet werden.

Das Hauptmodul, welches ein ML403 Development Board der Firma Xilinx<sup>®</sup> ist [21], beinhaltet einen Virtex<sup>®</sup>-4 XC4VFX12 FPGA Chip [22], welcher die Hauptkomponente des Hauptmoduls bildet. Der Aufbau des Gesamtsystems auf Grundlage eines FPGA Chips wurde gewählt, um eine höchstmögliche Flexibilität für spätere Erweiterungen zu haben. Hiermit ist es möglich, an das bestehende System andere, noch nicht weiter spezifizierte, externe Sensoren anzuschließen. Zum Beispiel weitere Temperatursensoren mit einer I<sup>2</sup>C Schnittstelle. Dies erfordert einen zusätzlichen Custom-IP-Core<sup>1</sup>, welcher über eine Hardware Description Language (HDL) programmiert werden muss. Einige Schnittstellen, wie VGA, Ethernet, USB und andere sind auf dem Development Board darüber hinaus als Hardware Schnittstelle bereits vorhanden.

Die Anbindung der Basismodule an das Hauptmodul geschieht über zwei Bussysteme. Der erste Bus dient zur Anbindung eines  $\mu$ Controllers auf dem Basismodul an das Hauptmodul, welcher als Feldbus<sup>2</sup> ausgelegt ist. Hierzu wird die serielle Schnittstelle des  $\mu$ Controllers genutzt. Die serielle Schnittstelle der einzelnen  $\mu$ Controller auf den Basismodulen ist direkt an einen Uartlite-IP-Core innerhalb des FPGAs angeschlossen. Über den so realisierten Feldbus werden Kontrollsignale an die einzelnen Basismodule übertragen, welche den Messablauf steuern. Um die Basismodule getrennt voneinander anzusprechen, ist ein eigenes Kommunikationsprotokoll entworfen worden, welches über eine individuelle ID und Checksummenerzeugung gewährleistet, dass immer nur ein Basismodul mit dem Hauptmodul kommunizieren kann (siehe Kapitel 3.4.1). Der zweite Bus (SPI-Bus) dient zur Übertragung der digitalen Messsignale ( $m_n$ ,  $u_n$  und  $i_n$ ). Die Messsignale werden von dem auf dem Basismodul untergebrachten Digital zu Analog Konverter (DAC) und Analog zu Digital Konvertern (ADC) über einer Serial-Peripheral-Interface (SPI<sup>3</sup>) Schnittstelle übertragen. Als Übertragungsprotokoll

---

<sup>1</sup>Ein IP-Core (engl. Intellectual Property Core) stellt das Chipdesign einer abgeschlossenen Baugruppe innerhalb eines FPGA dar.

<sup>2</sup>Ein Feldbus stellt unter anderem ein Kommunikationssystem zu mehreren unabhängigen Steuerungsgeräten für Messsensoren dar.

<sup>3</sup>SPI ist ein von der Firma Motorola entwickeltes dreileitiges Bussystem.

ist der EIA-485<sup>4</sup> Standard gewählt, da er über lange Leitungswege (>1200 m) eine relativ hohe Übertragungsgeschwindigkeit (> 10 Mbps) gewährleistet. Falls die Leitungslänge verkürzt wird, sind auch noch höhere Übertragungsgeschwindigkeiten möglich, was für diese Anwendung (siehe Kapitel 3.3.2.2) notwendig ist. Der SPI-Bus besteht insgesamt aus sieben differentiellen Leitungspaaren. Es werden für  $m_n$  drei differentielle Leitungspaare (clk, cs und data) übertragen. Für  $u_n$  und  $i_n$  werden vier differentielle Leitungspaare (clk, cs, adc\_data1 und adc\_data2) genutzt. Das clk bzw. cs Signal wird für die beiden ADCs zusammen übertragen. Dies ist möglich, da die beiden ADCs synchron zueinander die digitalen Daten (adc\_data1 bzw. adc\_data2) ausgeben.

Der im FPGA untergebrachte Custom-IP-Core beinhaltet die digitale Messsignalerzeugung, sowie die Anbindung des Hauptmoduls an den SPI-Bus. Als HDL ist der Custom-IP-Core mittels VHDL (Very High Speed Integrated Circuit Hardware Description Language) programmiert.

Auf einer Compact-Flash-Karte werden die ausgewerteten digitalen Messsignale abgespeichert. Weiter befindet sich auf der Compact-Flash-Karte der Programmcode für den FPGA, welcher beim Systemstart (Einschalten der Spannung am Hauptmodul) von der Compact-Flash-Karte in den FPGA geladen wird.

Um das System in einem Temperaturbereich bis  $-40^{\circ}\text{C}$  einsetzen zu können, muss das Hauptmodul zu einem späteren Zeitpunkt (siehe Kapitel 5) mit Bauteilen nach *Industrie Norm* neu aufgebaut werden. Dies ist durch die ausführliche Dokumentation des Development Boards, die unter anderem Schaltpläne und Board-Layouts enthält, möglich.

Das Basismodul dient als Verbindungsmodul zwischen Hauptmodul und Messmodul. Die Beweggründe, das Basismodul als Verbindungsmodul einzufügen sind,

- dass die Leitungswege zwischen Messmodulen, welche analoge Messsignale ausgeben, und der Digitalisierung der Messsignale möglichst kurz gehalten werden. Dies ist notwendig um Störsignale über die Zuleitungen zu vermeiden. Da in Zukunft verschiedene Messsensoren in einem Abstand von mehreren Metern eingesetzt werden, sind direkte Zuleitungen zum Hauptmodul zu lang.

---

<sup>4</sup>EIA-485 ist ein differentielles Übertragungsprotokoll, was im Gegensatz zu EIA-232 (auch als RS-232 bezeichnet) längere Übertragungsstrecken und höhere Geschwindigkeiten ermöglicht. Ein integrierter Widerstand im Sender gewährleistet einen kurzschlussfesten Ausgang, was den Betrieb an einem Bus ermöglicht [23].

- dass ein direkter Anschluss der Messsensoren an das Hauptmodul zu unflexibel in der Anzahl der zu verwendenden Messmodule ist. Für jedes Messmodul müsste ein separater Anschluss am Hauptmodul vorhanden sein. Ein variabler und damit an die jeweilige Messaufgabe angepasster Aufbau wäre somit nicht möglich. Eine Konsequenz dieser Forderung ist auch, dass die Basismodule über ein Bussystem an das Hauptmodul angebunden sind.

Das realisierte Basismodul bildet somit eine Schnittstelle zwischen Hauptmodul und mehreren Messmodulen. Je Basismodul können bei dem realisierten System bis zu acht Messmodule angeschlossen werden. Die Auswahl des zu verwendenden Messmoduls geschieht über den auf dem Basismodul untergebrachten  $\mu$ Controller. Dieser steuert über digitale Ausgänge mehrere Multiplexer ICs. Mit Hilfe der Multiplexer ICs und einer Schaltmatrix auf den Messmodulen wird der Messsensor ausgewählt. Um bei Anbindung mehrerer Basismodule mit zugehörigen Messmodulen den Energiebedarf möglichst gering zu halten, kann die auf dem Basismodul untergebrachte Spannungsversorgung durch den  $\mu$ Controller abgeschaltet werden. Der  $\mu$ Controller verfügt hierzu über eine eigene Spannungsversorgung.

Über die Messmodule werden die eigentlichen Messsensoren angeschlossen. Diese können ein Thermistor oder ein Platindrahtpaar sein. Auf dem Messmodul sind für jeden Messsensor Messvorverstärker zur Entkopplung des Messsensors von dem analogen Messsignal ( $m_t$ ) untergebracht. Zwei weitere Sensorverstärker dienen zur Messung der Sensorspannung ( $u_t$ ) und des Sensorstroms ( $i_t$ ) am Messsensor. Die Leitungsführung innerhalb des Messmoduls ist dabei so konzipiert, dass die Sensormessung in 4-Leiteranschlusstechnik erfolgt. Über die Schaltmatrix wird der zu messende Messsensor ausgewählt.

## 3.2 Hardwareaufbau der einzelnen Module

Das folgende Unterkapitel befasst sich mit dem Aufbau der einzelnen Hardwarekomponenten der Harp3d. Die entwickelten Module sind mit der Software „Target 3001“ entworfen.

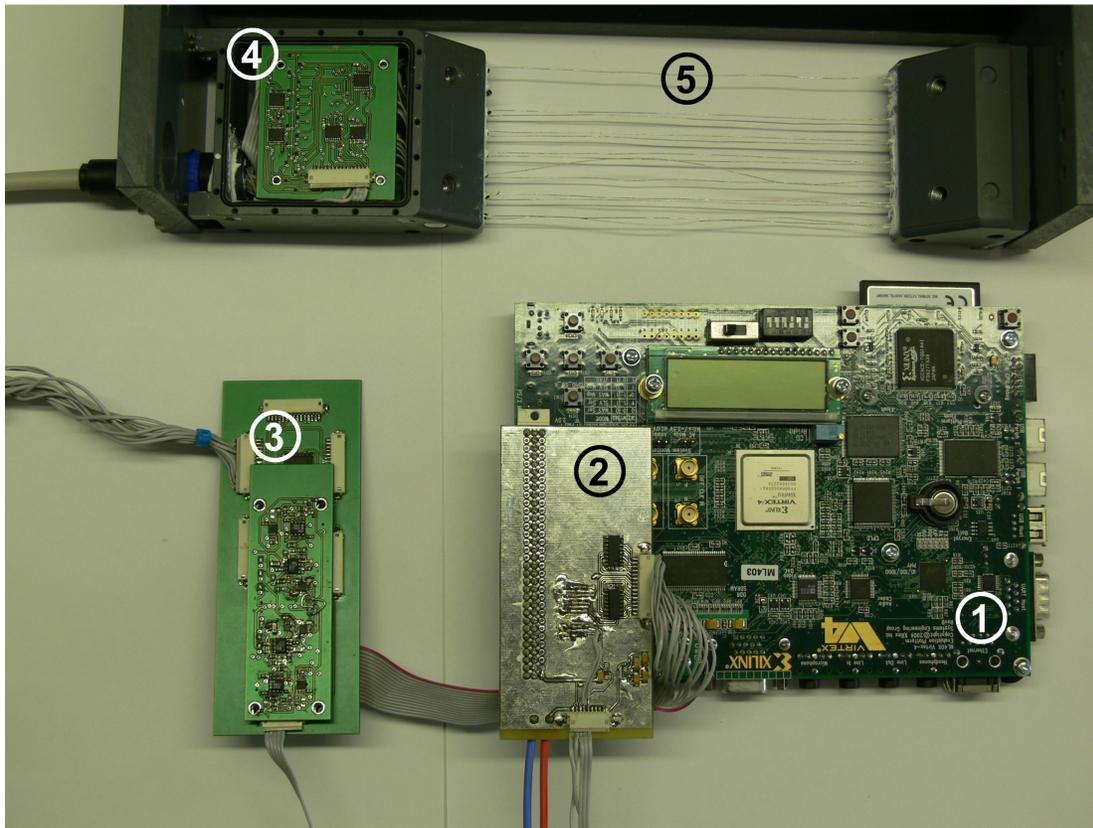


Abbildung 3.2: Gesamtaufbau der Harp3d (1: Hauptmodul, 2: Adapterplatine, 3: Basismodul, 4: Messmodule, 5: Messsensoren).

Der realisierte Aufbau der Harp3d ist in einzelne Module unterteilt (siehe Abb. 3.2). An einem ML403 Development Board (1) befindet sich eine Adapterplatine (2) zur Anbindung des SPI- und seriellen Busses, sowie zur Anbindung der Spannungsversorgung für die Basismodule. Über zwei separate Zuleitungen für die Spannungsversorgung inklusive seriellen Bus und dem SPI-Bus ist die Adapterplatine mit dem Basismodul (3) verbunden. Das Basismodul ist über impedanzarme Zuleitungen zur Übertragung der analogen Messsignale und der Spannungsversorgung mit den Messmodulen (4) verbunden. Die Messmodule dienen zur Anbindung der Platindrahtpaare (5) und der Thermistoren.

### 3.2.1 ML403 Development Board und Adapterplatine

Die Hauptkomponente der Harp3d bildet das ML403 Development Board (siehe Abb. 3.3) der Firma Xilinx®. Mittelpunkt des Development Boards ist ein Virtex®-4 XC4VFX12 FPGA Chip, an welchem diverse weitere Hardwarekomponenten angeschlossen sind.

Das ML403 Development Board wird unter anderem genutzt, da es einen Virtex®-4 FPGA

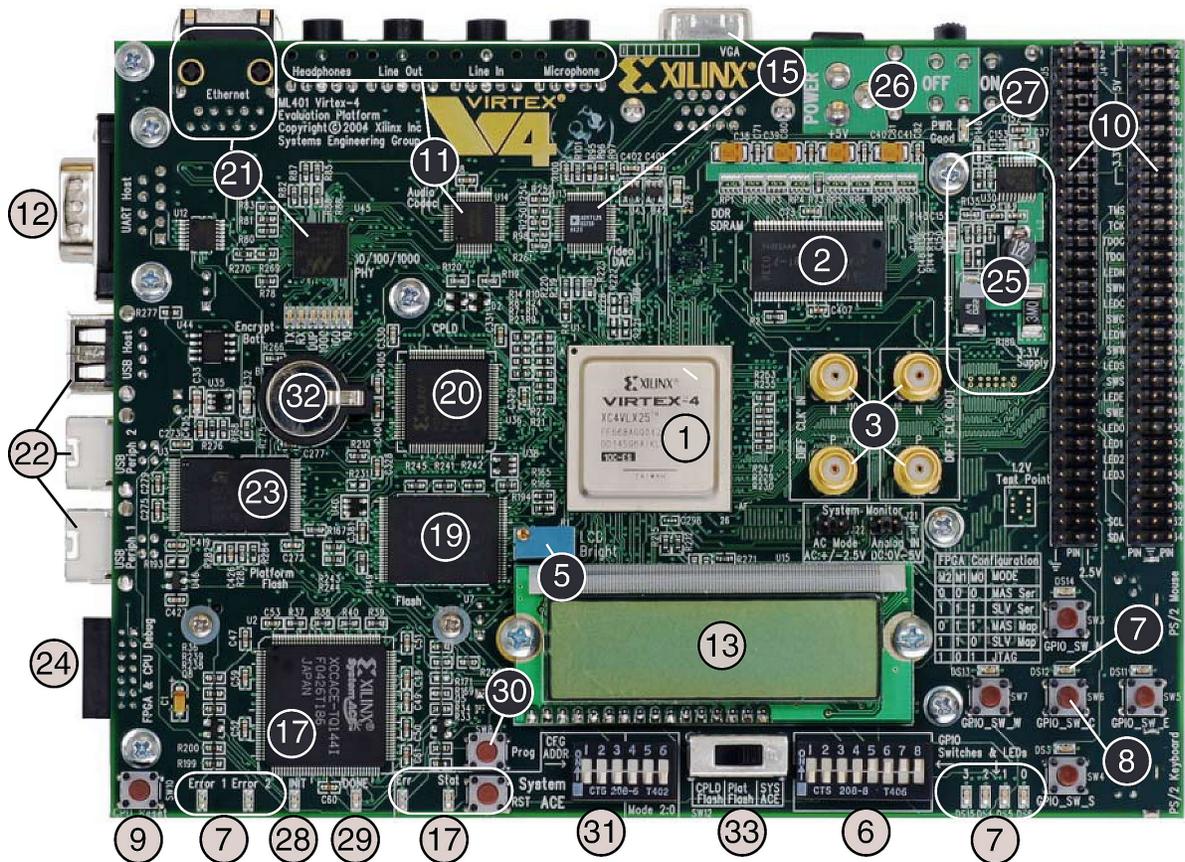


Abbildung 3.3: ML 403 Development Board ([3] Seite 12).

Chip der Serie FX beinhaltet. Die FX Serie zeichnet sich durch eine hohe Performance aus und ist für den Einsatz in Embedded Plattform Applikationen optimiert [22]. Dies ist wichtig, da zur Anbindung der Basismodule an das Hauptmodul eigene IP-Cores innerhalb des FPGA implementiert sind. Andere Serien der Virtex®-4 FPGA Chips zeichnen sich durch hohe Performance zur Erstellung logischer Applikationen (LX Serie) bzw. durch hohe Performance in Bezug auf digitale Signalverarbeitung (SX Serie) aus. Weiter verfügt der Virtex®-4 XC4VFX12 über einen implementierten PowerPC<sup>5</sup> und über zwei implementierte Ethernetchnittstellen,

<sup>5</sup>PowerPC Performance optimization with enhanced RISC Performance Chip

welche für spätere Applikationen genutzt werden können. Mit etwa 500€ ist das ML403 Development Board inkl. benötigter Software, wie das EDK<sup>6</sup> mit zugehörigem SDK<sup>7</sup> zur Erstellung der IP-Cores innerhalb des FPGAs, recht kostengünstig.

Zur Nutzung des Development Boards im späteren Einsatz in extremen Umgebungsbedingungen bis -40°C muss das Development Board neu aufgebaut werden. Durch die ausführliche Dokumentation des Development Boards, welche unter anderem auch die Gerber<sup>8</sup> Dateien der Leiterplatte enthalten, ist dies mit geringem Aufwand möglich. Die elektronischen Komponenten auf dem Development Board müssen hierzu mit identischen Komponenten nach *Industrie Norm* ausgetauscht werden. Um den Strombedarf des Development Boards zu verringern werden hierbei nur die Komponenten neu aufgebaut, welche für die Hard3d verwendet werden. Der Aufbau des Boards erfolgt im Anschluss an diese Masterarbeit.

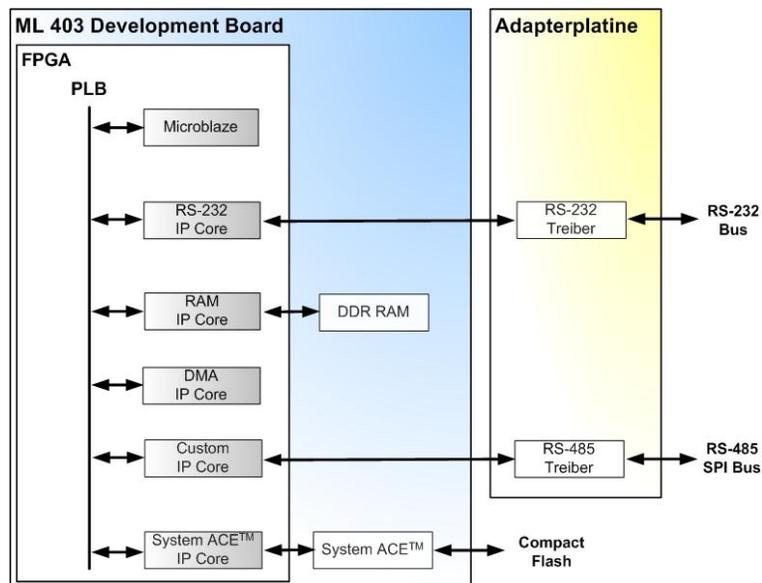


Abbildung 3.4: Schematischer Aufbau des Hauptmoduls.

Für den Aufbau des Hauptmoduls (siehe Abb. 3.4) der Harp3d werden, wie bereits angedeutet, nur einige der Komponenten des Development Boards genutzt. Um die Basismodule an das ML403 Development Board anzubinden, wird eine entwickelte Adapterplatine genutzt.

<sup>6</sup>EDK: Embedded-Development-Kit

<sup>7</sup>SDK: Software-Development-Kit

<sup>8</sup>Gerber ist ein standardisiertes Format zur Speicherung von Layoutinformationen und dient als Eingabeformat für Fotoplottter, welche hieraus Leiterplatten erstellen.

Auf dem Virtex<sup>®</sup>-4 XC4VFX12 FPGA ist ein Microblaze<sup>9</sup> implementiert, welcher als Prozessor des Hauptmoduls dient. Über einen Processor-Local-Bus (PLB)<sup>10</sup> sind an den Microblaze weitere IP-Cores angebunden.

Ein *XPS UART Lite* IP-Core (RS-232 IP-Core) dient zur Anbindung des seriellen Busses der Basismodule. Ein nachgeschalteter RS-232 Treiber auf der Adapterplatine wandelt das vom RS-232 IP-Core erzeugte 3,3 V CMOS Signal in ein RS-232 Signal. Hierbei entspricht eine logische 0 einem Pegel von +12V und eine logische 1 einem Pegel von -12V. Der RS-232 Treiber wird benötigt, um die Übertragungstrecke zu den Basismodulen und die anschließbaren Basismodule zu erhöhen. Bei Anschluss von nur einem Basismodul mit relativ kurzer Zuleitung ist der Einsatz des Treibers optional. In diesem Fall kommuniziert der RS-232 IP-Core direkt mit dem auf dem Basismodul befindlichen  $\mu$ Controller.

Ein *PLB Double Data Rate (DDR) Synchronous DRAM (SDRAM<sup>11</sup>) Controller* (RAM IP Core) dient zur Anbindung von auf dem Development Board untergebrachten 64MB DDR SDRAM Speicher (siehe Abb. 3.3 (2)). Der DDR RAM wird zur Zwischenspeicherung der aufgenommenen digitalen Messsignale  $u_n$  und  $i_n$  (siehe Abb. 3.1) genutzt. Hierzu werden die digitale Messsignale über einen *XPS Central DMA<sup>12</sup> Controller* von dem Custom-IP-Core direkt in den DDR Speicher übertragen. Weiter befindet sich im DDR RAM der Programmablauf für den Microblaze Prozessor.

Der Custom-IP-Core (siehe auch Kapitel 3.3.2), welcher in VHDL erstellt ist, dient als Schnittstelle zwischen RS-485 SPI Bus und PLB. Zwei RS-485 Treiber wandeln das 3,3 V CMOS Signal in ein RS-485 Signal, über welches das SPI Signal für die Messsignale  $m_n$ ,  $u_n$  und  $i_n$  übertragen wird.

Über ein *XPS SYSACE (System ACE) Interface Controller* (System ACE IP-Core) wird der System ACE<sup>™</sup> CF Controller XCCACE-TQG144I (siehe Abb. 3.3 (17)) in das System eingebunden. Über den System ACE<sup>™</sup>CF Controller ist eine Compact-Flash-Karte in das System eingebunden (auf der Unterseite des Development Boards). Die Compact-Flash-Karte dient zur Abspeicherung der gemessenen und ausgewerteten Messsignale, als auch zur Speicherung des Programmcodes für den Microblaze. Der Programmcode (.elf Datei) wird beim Einschalten des Development Boards in den FPGA bzw. in den DDR-RAM geladen. Er enthält die Informationen über den zu realisierenden Hardwareaufbau innerhalb des FPGA und den

<sup>9</sup>Der Microblaze ist ein von der Firma Xilinx<sup>®</sup>entworfener Softcore RISC Prozessor. Softcore bedeutet, dass der Prozessor in HDL entworfen ist.

<sup>10</sup>der PLB ist Teil der IBM CoreConnect<sup>™</sup> Busstandards.

<sup>11</sup>SDRAM engl. Synchronous Dynamic Random Access Memory

<sup>12</sup>DMA engl. Direct Memmory Access

Programmablauf für den Microblaze. Außerdem kann über den System ACE™ CF Controller mit Hilfe der JTAG<sup>13</sup>-Schnittstelle der FPGA auch von extern programmiert werden.

Zur Anbindung der Adapterplatine wird der auf dem ML403 Development Boards zur Verfügung gestellte 0,1 inch Erweiterungsstecker genutzt (siehe Abb. 3.3 (10) rechter Stecker). Zusammengefasst werden auf dem ML403 Development Board als Hardwarekomponenten somit lediglich der DDR RAM, der Erweiterungsstecker, die Spannungsversorgung (siehe Abb. 3.3 (26)) und der System ACE™ CF Controller mit zugehöriger Compact-Flash-Karte genutzt. Alle weiteren Komponenten können in einem neu aufzubauenden Hauptmodul weggelassen werden. Für spätere eventuelle Erweiterungen stehen sie aber weiter zur Verfügung und müssen dann nachträglich bestückt werden.

Zur Anbindung der Basismodule wird eine Adapterplatine (siehe Abb. 3.5) an dem Erweiterungsstecker des ML403 Development Boards genutzt.

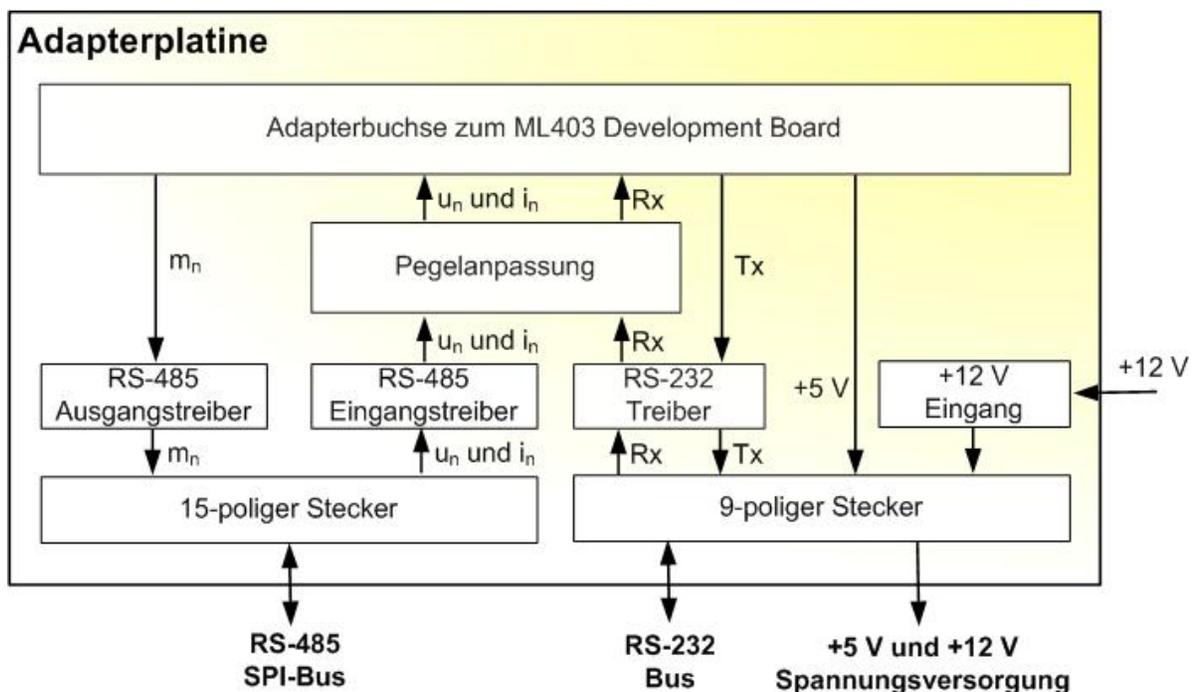


Abbildung 3.5: Schematischer Aufbau der Adapterplatine

Zum Anschluss an das ML403 Development Board dient eine 96-polige, 3-reihige und nach 0,1 inch Standard konzipierte Adapterbuchse. Die Adapterbuchse dient zur Anbindung an die digitalen Ein- und Ausgangssignale des Custom-IP-Cores und des RS-232 IP Cores

<sup>13</sup>JTAG Joint Test Action Group; IEEE-Standard 1149.1 zum Testen und Debuggen von Hardware

(siehe Abb. 3.4). Mit Hilfe der EDK Software sind diese Signale mittels Eintaktausgänge nach 3,3 V CMOS Standard aus dem FPGA zum rechten Erweiterungsstecker des ML403 Development Boards herausgeführt. An der Adapterbuchse wird zudem eine vom ML403 Development Board bereitgestellte +5 V Spannung abgegriffen. Diese dient zur Spannungsversorgung der  $\mu$ Controller auf den Basismodulen.

Die digitalen Ein- und Ausgangssignale des FPGAs sind über eine Pegelanpassung an drei Treiber-ICs für die Anbindung an die beiden Bussysteme angeschlossen. Eine Pegelanpassung ist notwendig, da die Treiber-ICs nach TTL Standard arbeiten. Damit die Signalspannung in den FPGA +3,3 V nicht übersteigt, werden die +5 V Signale der Treiber-ICs dem FPGA über die Pegelanpassung zugeführt. Hierzu gehören die vier Eingangssignale des SPI-Busses ( $u_n$  und  $i_n$ ) und Rx des RS-232 Busses. Das Rx Eingangssignal ist über einen 1 k $\Omega$  zu 3 k $\Omega$  Spannungsteiler realisiert. Die  $u_n$  und  $i_n$  Signale sind über 30  $\Omega$  Widerstände gegen Masse an den FPGA angebunden. Sie sind somit direkt angeschlossen, die Pegelwandlung wird durch den 30  $\Omega$  Abschlusswiderstand realisiert. Dies ist notwendig, da die Eingangskapazität der FPGA Anschlüsse zusammen mit dem Spannungsteiler einen Tiefpassfilter bilden. Eine Übertragung der recht hochfrequenten (das clk Signal wird mit einigen MHz übertragen) SPI-Signale wäre hiermit nicht möglich. Die verwendeten Abschlusswiderstände, sowie der Spannungsteiler fungieren weiter als Pull-Up Widerstand gegen Masse für den FPGA. Dies unterdrückt undefinierte Eingangspegel am FPGA.

Die digitalen SPI Ausgangssignale ( $m_n$ ) liegen direkt an einem RS-485 Ausgangstreiber an. Über einen 15-poligen Stecker sind die differentiellen Ausgangssignale des RS-485 Ausgangstreibers an den SPI-Bus angeschlossen. Die differentiellen Eingangssignale von  $u_n$  und  $i_n$  sind ebenfalls über diesen 15 poligen Stecker, an die RS-485 Eingangstreiber angeschlossen.

Die beiden Signale (Tx und Rx) des RS-232 Busses sind über einen 9-poligen Stecker mit einem RS-232 Treiber verbunden. Der RS-232 Treiber ist optional und wird nur bei längeren Leitungswegen genutzt. In einem ersten Aufbau der Hardware ist dieser Treiber auf der Adapterplatine und dem Basisboard im Layout implementiert, wird aber nicht genutzt. Stattdessen sind die Signale von Tx und Rx direkt an den RS-232 Bus angeschlossen. Über den 9-poligen Stecker werden den Basismodulen weiter eine +5 V und eine von extern anschließbare +12 V Spannungsversorgung zugeführt.

Bedingt durch bereits vorgesehene Erweiterungen des Gesamtsystems wird die Adapterplatine im Anschluss an diese Arbeit weiter modifiziert. Unter anderem soll auf ihr ein IC als Zeitgeber mit Anschluss an einen GPS Empfänger untergebracht werden. Dieser soll gewähr-

leisten, dass das System mit einer definierten Uhrzeit betrieben werden kann. Zudem soll er die Gesamtspannungsversorgung (auch für das ML403 Development Board) überwachen und abschaltbar machen, um den Energieverbrauch zwischen einzelnen Messintervallen weiter zu senken.

### 3.2.2 Basis Modul

Das Basismodul dient als Schnittstelle zwischen Hauptmodul und den einzelnen Messmodulen. Über zwei Bussysteme ist das Basismodul mit dem Hauptmodul verbunden. Über dieses Bussystem ist es möglich mehrere Basismodule an das Hauptmodul anzubinden. Die Messmodule werden über mehrere Stecker mit dem Basismodul verbunden.

Die Grundfunktion des Basismoduls besteht darin, das digitale Messsignal  $m_n$  vom Hauptmodul zu analogisieren und an die Messmodule weiter zu leiten, sowie die von den Messmodulen kommenden analogen Messsignale ( $u_t$  und  $i_t$ ) zu digitalisieren und an das Hauptmodul zurück zu senden.

Das Layout des Basismoduls ist so konzipiert, dass die Abmessungen des Basismoduls möglichst klein gehalten sind. Hierzu ist das Basismodul (siehe Abb. 3.6) in drei einzelne Platinen aufgeteilt, welche über Steckverbinder miteinander verbunden sind.

- Die Spannungsversorgungsplatine dient zur Bereitstellung aller benötigten Spannungen des Basismoduls. Zu diesen gehören zwei durch den  $\mu$ Controller abschaltbare Grundspannungsquellen ( $\pm 5V$ ). Über die  $+5V$  Gleichspannungsquelle wird mit Hilfe einer Referenzspannungsquelle eine  $+2,048V$  Spannung erzeugt, welche als Referenz für den DAC bzw. die ADCs dient. Über zwei Spannungsteiler werden eine  $\pm 1,024V$  Spannung erzeugt. Diese Spannungen werden für die Addiererstufen auf der ADC/DAC-Platine benötigt.
- Die Steuerungsplatine mit einem ATmega8  $\mu$ Controller dient zur Steuerung. Er steuert eine abschaltbare Spannungsversorgung, einen Multiplexers und die Verstärkerstufen auf der ADC/DAC-Platine. Der Multiplexer wählt den zu verwendenden Messsensor auf den einzelnen Messmodulen aus. Zwei RS-485 Treiber-ICs wandeln die digitalen Messsignale ( $m_n$ ,  $u_n$  und  $i_n$ ) in differentielle Signale und binden sie an den RS-485 Bus an.

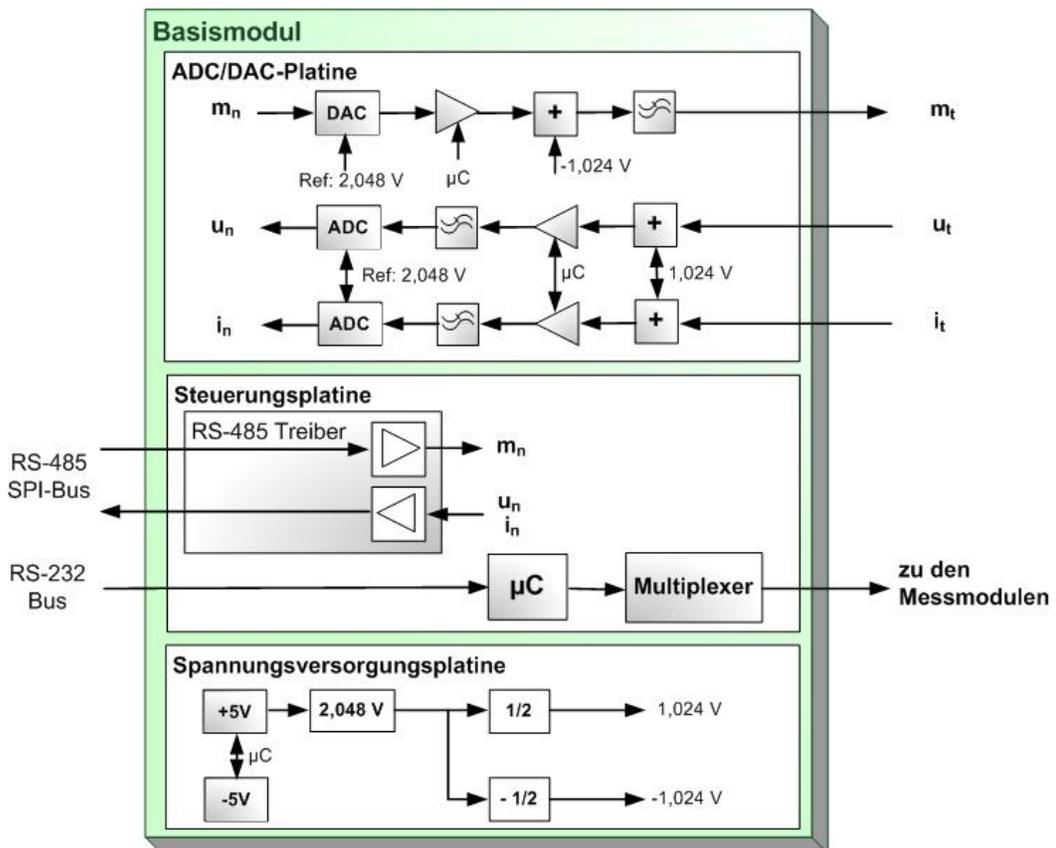


Abbildung 3.6: Schematischer Aufbau des Basismoduls.

- die ADC/DAC-Platine. Auf ihr wird das digitale Messsignal vom Hauptmodul ( $m_n$ ) mit Hilfe eines DACs, einer nachgeschalteten Verstärkerstufe, einer Addierstufe und einem Interpolations-Tiefpasses analogisiert und an die Messmodule weiter geleitet ( $m_t$ ). Die Addierstufe dient dazu, das durch den DAC produzierte analoge Messsignal von 0 - 2,048 V auf einen Spannungswert zwischen +/- 1,024 V zu transformieren. Die analogen Messsignale der Messmodule ( $u_t$  und  $i_t$ ) werden jeweils über eine Addierstufe, welche das analoge Messsignal auf einen Spannungswert von 0 - 2,048 V rücktransformiert, eine Verstärkerstufe, einen Antialiasing-Tiefpass und einem ADC digitalisiert ( $u_n$  und  $i_n$ ). Die Verstärkerstufen werden hierbei durch den  $\mu$ Controller auf der Steuerungsplatine gesteuert.

Die folgenden Unterkapitel zeigen den detaillierten Aufbau der einzelnen Platinen des Basismoduls.

### 3.2.2.1 Spannungsversorgungsplatine

Die Spannungsversorgungsplatine stellt alle benötigten Spannungen für das Basismodul zur Verfügung (Tabelle 3.1).

Tabelle 3.1: Bereitgestellte Spannungen der Spannungsversorgung.

Spannung	Zuführung zu	Funktion
+5 V	Steuerungs- und ADC/DAC-Platine	+5 V Grundspannung
-5 V	Steuerungs- und ADC/DAC-Platine	-5 V Grundspannung
ADC 1 +2,048 V	ADC/DAC-Platine	+2,048 V Referenzspannung für ADC 1
ADC 2 +2,048 V	ADC/DAC-Platine	+2,048 V Referenzspannung für ADC 2
DAC +2,048 V	ADC/DAC-Platine	+2,048 V Referenzspannung für den DAC
+1,024 V	ADC/DAC-Platine	+1,048 V Spannung für die Addierstufen der ADCs
-1,024 V	ADC/DAC-Platine	-1,048 V Spannung für die Addierstufe des DACs

Die +/-5 V Grundspannungen werden mit Hilfe zweier Abwärtswandler<sup>14</sup> erzeugt. Abwärtswandler sind gewählt worden, da sie mit bis zu 90% eine relativ hohe Effektivität in der Gleichspannungswandlung haben. Abb. 3.7 zeigt den Aufbau der +/-5 V Spannungsversorgung. Die Spannungswandlung ist jeweils durch einen MIC4680 *1A 200kHz SuperSwitcher™ Buck Regulator* realisiert. Die Dimensionierung der Schaltung wurde der Literatur entnommen [31; 32]. L1, L2, C1, C4, C6, C5, D1 und D2 sind die für den MIC4680 benötigten externen Bauteile. C6 bzw. C1 stellen die benötigte Eingangskapazität dar. D1, L1 und C4 bzw. D2, L2 und C5 stellen den Ausgangstiefpass mit zugehöriger Diode dar. Die Schaltung ist so dimensioniert, dass sie bis jeweils 250mA Ausgangsstrom betrieben werden kann. Hierbei sind die Spulen L1 und L2 die begrenzenden Bauteile, welche für einen maximalen Durchlassstrom von 250mA ausgelegt sind. Über das Teilverhältniss von R3 und R1 bzw. von R4 und R5 lässt sich die

<sup>14</sup>Ein Abwärtswandler wandelt über eine Puls-Weiten-Modulierte Spannung mit Hilfe einer Diode und eines nachfolgenden LC-Tiefpass eine im Betrag höhere Eingangsspannung in eine kleinere Ausgangsspannung.

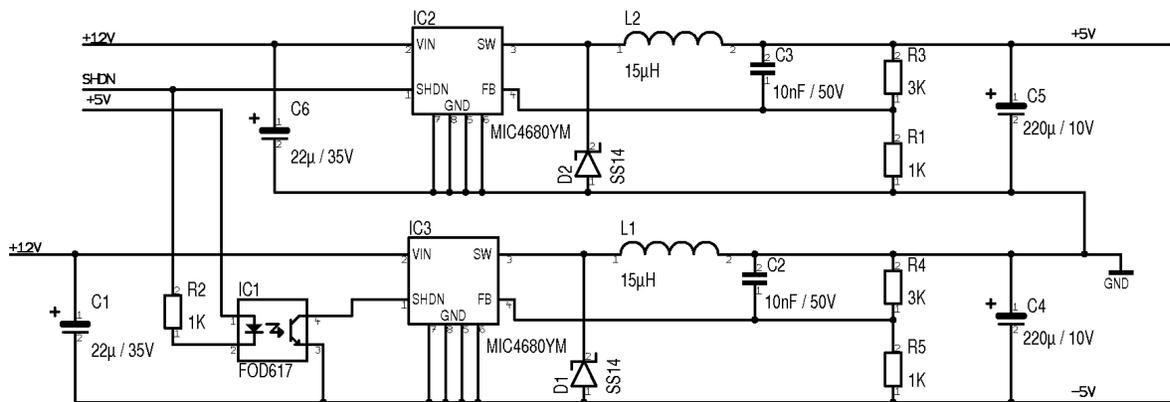


Abbildung 3.7: Aufbau der +5 V (oben) und -5 V (unten) Spannungsversorgung.

Ausgangsspannung einstellen. Hierbei gilt [32]:

$$u_{out,+5V} = 1,23V \left( \frac{R_3}{R_1} + 1 \right) \text{ bzw. } u_{out,-5V} = 1,23V \left( \frac{R_4}{R_5} + 1 \right). \quad (3.1)$$

Somit ergeben sich für die verwendeten Widerstände eine Ausgangsspannung von +/-4,92 V. Da keine der nachfolgenden Bauteile auf eine genaue +/- 5V Spannungsversorgung angewiesen ist, sind mit Berücksichtigung der Bauteiltoleranzen eine ausreichend genaue Grundspannungsversorgung garantiert.

Das Layout der Spannungsversorgung ist so aufgebaut, dass die kritischen Pfade der MIC4680 Abwärtsrichter möglichst kurz gehalten werden. Zu den kritischen Pfaden gehören die Verbindungen zwischen MIC4680 und den Ein- und Ausgangskapazitäten (C1 und C4 bzw. C6 und C5), sowie den beiden Dioden (D1 bzw. D2).

Die verwendeten MIC4680 Abwärtsrichter lassen sich über den SHDN-Anschluss ausschalten. Hierzu muss der SHDN-Anschluss auf das gleiche Potential wie die GND-Anschlüsse gelegt werden. Der SHDN Eingang der +5V Spannungsversorgung ist mit einem, vom  $\mu$ Controller auf der Steuerungsplatine produzierten, SHDN-Signal verbunden. Über dieses SHDN-Signal ist die +5V Spannungsversorgung abschaltbar. Bei der -5V Spannungsversorgung liegt am GND-Anschluss die -5V Ausgangsspannung an. Ein direkter Anschluss des SHDN-Signals an den SHDN-Anschluss des MIC4680 ist somit nicht möglich. Aus diesem Grund wird über einen Optokoppler die schaltbare Verbindung zwischen dem SHDN-Anschluss auf den GND-Anschluss gewährleistet. Der Optokoppler wird dabei ebenfalls von

dem SHDN-Signal des  $\mu$ Controllers betrieben.

Zum Betreiben der ADCs und des DACs wird eine Referenzspannung benötigt. Zum späteren Betrieb werden relativ kleine Ausgangsspannungen benötigt, so dass mit 2,048 V eine relativ kleine Referenzspannungsquelle gewählt wurde. Bei der Referenzspannungsquelle handelt es sich um einen ADR430B Chip, welcher für den Betrieb mit den verwendeten DAC bzw. ADCs ausgelegt ist [33]. Die Genauigkeit der DAC bzw. ADCs ist maßgeblich von der verwendeten Referenzspannung abhängig. Aus diesem Grund ist die B-Variante der Referenzspannungsquelle mit  $3 \text{ ppm}/^\circ\text{C}$  gewählt. Am Ausgang der Referenzspannungsquelle befindet sich ein Impedanzwandler zur Entkopplung des Referenzspannungsausgangs von dem Rest der Schaltung. Weitere drei Impedanzwandler dienen zur Ankopplung der Referenzspannung an die einzelnen DAC bzw. ADCs. Weiter wird mittels eines Spannungsteilers aus zwei  $1\text{k}\Omega$  Präzisionswiderständen aus der Referenzspannung die 1,024 V Spannung erzeugt [34]. Ein nachgeschalteter invertierender Verstärker mit  $v=1$  erzeugt die zusätzlich benötigte -1,024 V Spannung. Hierbei ist der invertierende Verstärker ebenfalls mittels  $1\text{k}\Omega$  Präzisionswiderständen aufgebaut.

Das Layout der Platine sieht eine Temperaturkompensation der Ausgangsimpedanzwandler für den DAC bzw. ADCs vor. Hierzu ist das Layout so konzipiert, dass die Impedanzwandler als nicht invertierender Verstärker oder als Verstärker mit vorgeschaltetem Spannungsteiler betrieben werden können. Hierdurch ist es möglich mit Hilfe eines temperaturabhängigen Widerstandes ( $R_T$ ) zu einem späteren Zeitpunkt eine temperaturabhängige Kompensation der Referenzspannungen aufzubauen.

Bei Betrieb als nicht invertierender Verstärker ergibt sich für die Ausgangsspannung

$$u_{out}(T) = u_{Ref} + u_{Ref} \left( \frac{R_T}{R} \right), \quad (3.2)$$

bei Betrieb mit Hilfe des Spannungsteilers

$$u_{out}(T) = u_{Ref} \left( \frac{R}{R_T + R} \right). \quad (3.3)$$

Somit ist sowohl eine positive als auch negative Temperaturkompensation möglich.

### 3.2.2.2 Steuerungsplatine

Die Steuerungsplatine dient zur Steuerung der Signale und Bauteilkomponenten des Basismoduls. Sie gewährleistet die Anbindung an die Busse zum Hauptmodul und zu den Messmodulen.

Zur Anbindung an die Busse zum Hauptmodul existieren eine 15-polige Adapterbuchsen. Eine 9-polige Adapterbuchse verbindet die Steuerungsplatine mit dem RS-232 Bus, sowie mit der Spannungsversorgung.

Zur Anbindung der Messmodule an die Steuerungsplatine existieren fünf 15-polige Adapterbuchsen. Über diese werden die analogen Messsignale und die Spannungsversorgung für die Messmodule übertragen. Mittels sechs digitaler Signale werden die jeweiligen Messsensoren auf den Messmodulen ausgewählt. Im Speziellen werden,

- die analogen Messsignale  $m_t$ ,  $u_t$  und  $i_t$ ,
- die Spannungsversorgung mit +/- 5V und Masse,
- drei digitale Signale (CNT\_MMn) zur Auswahl der acht Messsensoren und
- drei digitale Signale (CNT\_EN\_WTR) als Enable-Signale für die einzelnen Platinen in den Messmodulen

an das Messmodul übertragen.

Mittelpunkt der Steuerungsplatine ist ein ATmega8  $\mu$ Controller (siehe Abb. 3.8), welcher über eine separate +5V Spannungsversorgung vom Hauptmodul gespeist wird. Zusätzlich ist an den  $\mu$ Controller eine temperaturabhängige OP-Verstärkerschaltung angeschlossen, um die Temperatur auf dem Basismodul zu ermitteln.

Die digitalen, vom  $\mu$ Controller generierten, CNT-Steuersignale des Basismodul sind im Einzelnen,

- das SHDN-Signal für die Spannungsversorgung.
- die  $v_0$ - bzw.  $v_1$ -Signale. Sie dienen zur Auswahl der Vorverstärkung auf der DAC/ADC-Platine.
- die drei digitalen Signale zur Auswahl des jeweiligen Messsensors auf den Messmodulen (CNT\_MMn\_x).

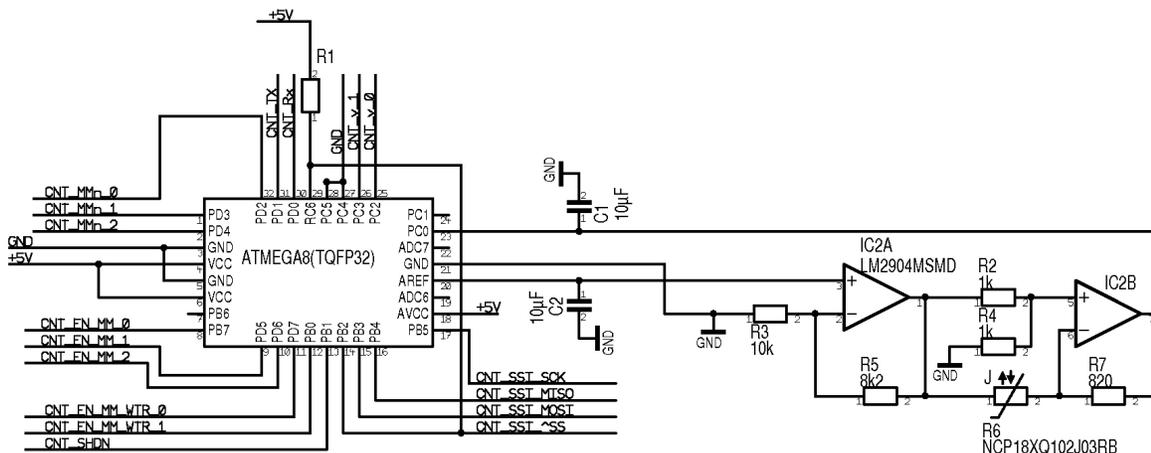


Abbildung 3.8: Anschlüsse am  $\mu$ Controller mit zugehöriger Temperaturmessung.

- drei digitalen Signale (CNT\_EN\_MM) zur Auswahl der jeweiligen Messmodule.
- zwei digitale Signale (CNT\_EN\_MM\_WTR) zur Auswahl der Platindraht-, Temperatur- oder Referenzsensoren in den Messmodulen. Der Referenzsensor brückt das Eingangssignal direkt auf das Ausgangssignal.

Über die fünf Anschlüsse zu den Messmodulen werden aus CNT\_EN\_MM und CNT\_EN\_MM\_WTR mit Hilfe von drei Multiplexern die drei benötigten CNT\_EN\_WTR-Signale erzeugt. Über CNT\_EN\_MM wird eine Auswahl des Messmoduls getroffen. Die CNT\_EN\_MM\_WTR-Signale wählen einen der drei Multiplexer aus. Insgesamt werden so 15 Enable-Signale erzeugt, welche über die fünf Anschlüsse zu den Messmodulen die einzelnen Messsensoren (Platindraht-, Temperatur- oder Referenzsensor) auswählen.

Die CNT\_SST Anschlüsse stellen eine SPI-Schnittstelle dar und dienen zur Programmierung des  $\mu$ Controllers. R1 dient als Pull-Up Widerstand für den Reset Eingang (RC6) des  $\mu$ Controllers. CNT\_TX und CNT\_RX sind die Anschlüsse der seriellen Schnittstelle. Die temperaturabhängige Spannung am Eingang PC0 des  $\mu$ Controllers kann über einen im  $\mu$ Controller untergebrachten 12 Bit ADC vom Hauptmodul abgefragt werden (siehe 3.4.1). Die Temperaturinformation soll später für eine zusätzliche Kompensierung der durch die Temperatur verursachten Messfehler dienen. Durch die 12 Bit beschränkt sich die Genauigkeit der gemessenen Werte auf etwa  $\pm 1^\circ\text{C}$ . Dies ist für eine Information über den temperaturabhängigen Fehler der analogen Signalverarbeitung auf dem Basismodul aber ausreichend. Die Funktionsweise der Schaltung ist, dass mittels eines nicht invertierenden Verstärkers die,

vom  $\mu$ Controller zur Verfügung gestellte, 2,56 V Referenzspannung (AREF) auf 4,60 verstärkt wird. Diese dient als Referenzspannung für einen Brückenverstärker mit integriertem NTC-Temperatursensor (R6). Durch die Wahl von R7 liegt die resultierende Ausgangsspannung zwischen +4 V bei  $-40^{\circ}\text{C}$  und 0 V bei  $30^{\circ}\text{C}$ .

Weiter befinden sich auf der Steuerungsplatine zwei RS-485 Treiber, ein Sende- und ein Empfangstreiber. Der Empfangstreiber, wandelt die drei differentiell übertragenen Eingangssignale ( $\overline{\text{CS}}$ , DIN und SCLK) für den DAC in TTL Pegel. Die TTL Pegel werden über den Adapteranschluss zwischen den einzelnen Platinen des Basismoduls der ADC/DAC-Platine zugeführt. Der Sendetreiber wandelt die 4 von den ADCs kommenden Signale in differentielle Signale.

Ein RS-232 Treiber dient für spätere Anwendungen zur Anbindung des  $\mu$ Controllers an lange Leitungswege. Wie in Kapitel 3.2.1 bereits erwähnt, kommt der RS-232 Treiber noch nicht zum Einsatz.

### 3.2.2.3 ADC/DAC-Platine

Auf der ADC/DAC-Platine sind der DAC und die ADCs untergebracht. Bei dem DAC handelt es sich um einen 16-Bit AD5541. Bei den ADCs (AD7688) handelt es sich jeweils um 16-Bit ADCs mit differentiellem Eingang. DAC und ADCs sind für eine maximale Abtastfrequenz von 500 kHz konzipiert. Ausschlaggebend für die Wahl dieser Komponenten war die hohe Genauigkeit mit wahren 16-Bit und der geringe Stromverbrauch.

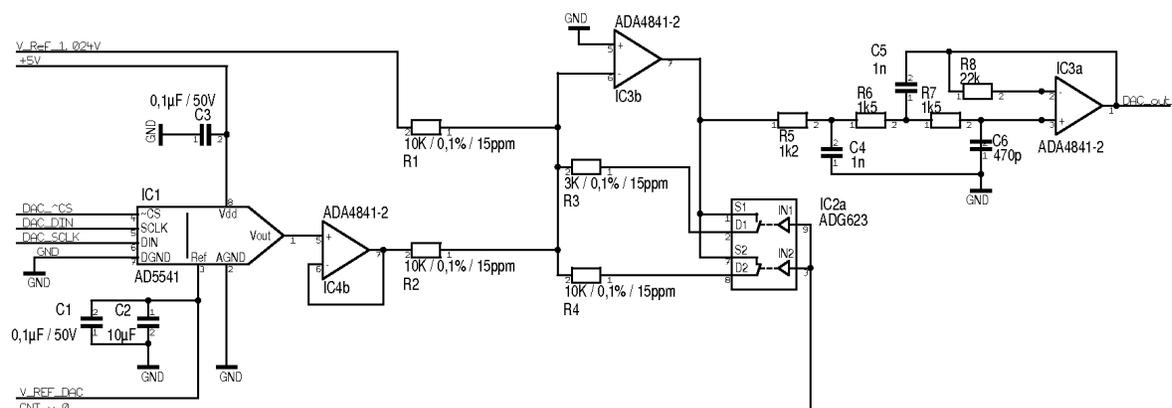


Abbildung 3.9: Aufbau des DAC mit zugehörigem Interpolations-Tiefpass.

Abb. 3.9 zeigt den Aufbau des DACs mit zugehöriger Addiererstufe und Interpolations-Tiefpass. IC4b dient als Impedanzwandler für das durch den DAC (IC1) produzierte Signal. Dies ist notwendig, da der DAC keinen Ausgangstreiber besitzt. Grundsätzlich kann der DAC auch ohne Ausgangstreiber betrieben werden, was in einem ersten Aufbau auch geschehen ist. Da der DAC mit  $6,25 \text{ k}\Omega$  aber einen recht hohen Ausgangswiderstand besitzt, wurde die nachfolgende Addiererstufe zu sehr beeinflusst.

Bei der Addiererstufe mit IC2a, IC3b, R1, R2, R3 und R4 handelt es sich um eine OP Addiererschaltung. Die Addiererstufe ist notwendig, um die Ausgangsspannung des DACs auf einen Spannungspegel um 0V zu transformieren. Mit Hilfe von  $V_{REF\_1.024V}$  wird die Ausgangsspannung des DACs, welche zwischen 0 V und  $V_{REF\_DAC} = 2,048 \text{ V}$  liegt, auf eine um 0V symmetrische Spannung transformiert. IC2a dient zur umschaltbaren Verstärkung mittels CNT\_v\_0. Die Ausgangsspannungen ergeben sich zu

$$u_{out} = -u_{DAC} \frac{R_3 \text{ bzw. } R_4}{R_2} - u_{V\_REF\_1.024V} \frac{R_3 \text{ bzw. } R_4}{R_1}. \quad (3.4)$$

Da der Innenwiderstand der Schalter von IC2b als zusätzlicher Reihenwiderstand zu R3 und R4 liegt, ist IC2b mit einem ADG623 Baustein so gewählt, dass der Schalterwiderstand mit maximal  $5 \text{ }\Omega$  sehr klein ist. Noch wichtiger ist die Temperaturabhängigkeit des Schalterwiderstandes, welcher mit einer Schwankung von  $2 \text{ }\Omega/100^\circ\text{C}$  ebenfalls sehr klein ist. Durch den Schalter ergibt sich somit, bei der Wahl von  $3 \text{ k}\Omega$  als Widerstand der Addiererstufe, ein maximaler Fehler durch die Temperatur von unter 1%.

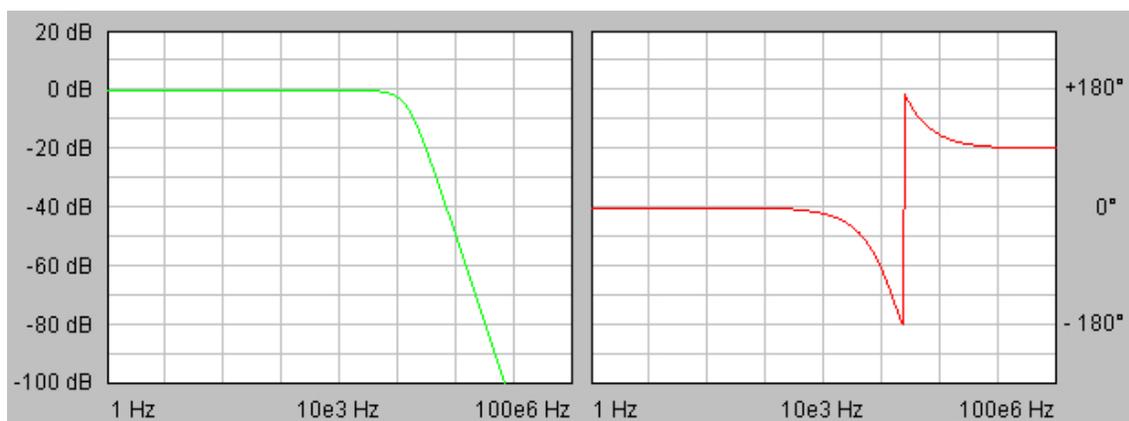


Abbildung 3.10: Frequenzgang (links) und Phasengang (rechts) des Interpolations-Tiefpasses [4].

Der Addiererstufe nachgeschaltet ist ein Besseltiefpass dritter Ordnung, welcher als Interpolationstiefpass dient. Ein Besselfilter ist notwendig, um die Phaseninformationen bestmöglichst zu erhalten. Die Dimensionierung des Besselfilters wurde mit Hilfe des *Analog Filter Wizard™* [4] vorgenommen. Der Interpolations-Tiefpass besitzt eine Grenzfrequenz von 100 kHz. Die Grenzfrequenz resultiert aus der Wandelfrequenz des DACs von 460 kHz (siehe auch Kapitel 3.3.2). Hiermit ist sichergestellt, dass das Ausgangssignal des DACs optimal interpoliert wird.

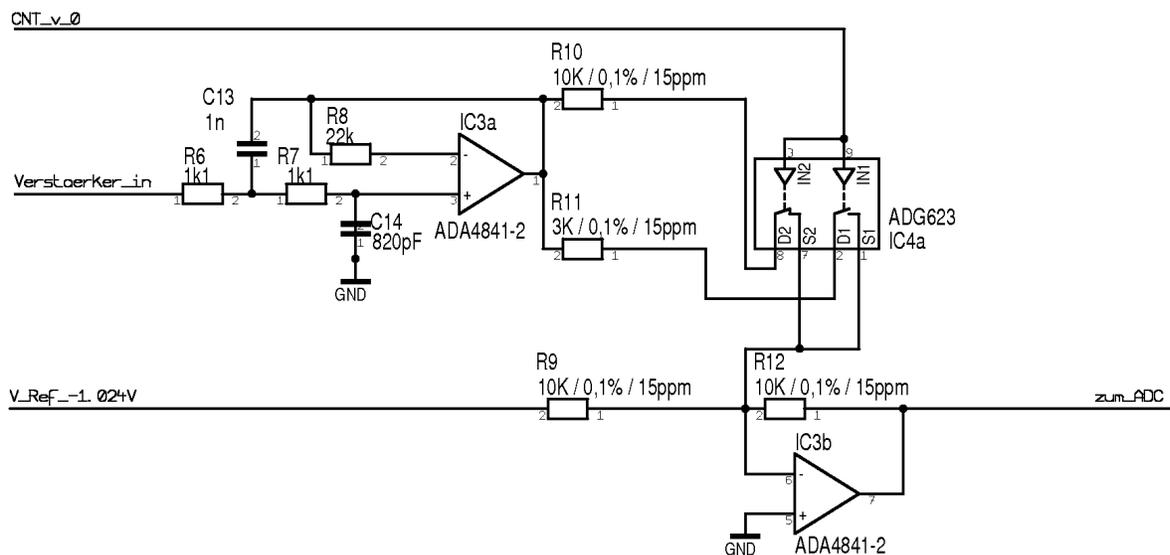


Abbildung 3.11: Aufbau der Addiererstufe (rechts) und ein Teil des Antialiasing-Tiefpasses (links) der ADCs.

Der Aufbau der ADCs-Stufen (siehe Abb. 3.11) ist in eine Addiererstufe und einen Antialiasing-Tiefpasses aufgeteilt. Bei der Addiererstufe mit IC3b, IC4a, R9, R10, R11 und R12 handelt es sich, wie bei der DAC-Schaltung, um eine OP Addiererschaltung. Die Addiererstufe transformiert das Eingangssignal, welches zwischen  $\pm 1,024$  V liegt, auf einen Spannungspegel von 0 V bis 2,048 V. IC4a dient zur umschaltbaren Verstärkung mittels CNT\_v\_0 bzw. CNT\_v\_1 bei der Addiererstufe des zweiten ADCs. Die Problematik der Schalterwiderstände sind identisch denen beim DAC.

Zur Aliasing Unterdrückung des Eingangssignal dient ein Besseltiefpass fünfter Ordnung. Der Antialiasing-Tiefpass besitzt eine Grenzfrequenz von 100 kHz. Der Antialiasing-Tiefpass befindet sich im linken Teil von Abb. 3.11 und Abb. 3.12. Einen Teil des Besselfilters wird durch einen ADA4941 realisiert. Dieser dient weiter als differentieller Eingangsreiber für die differen-

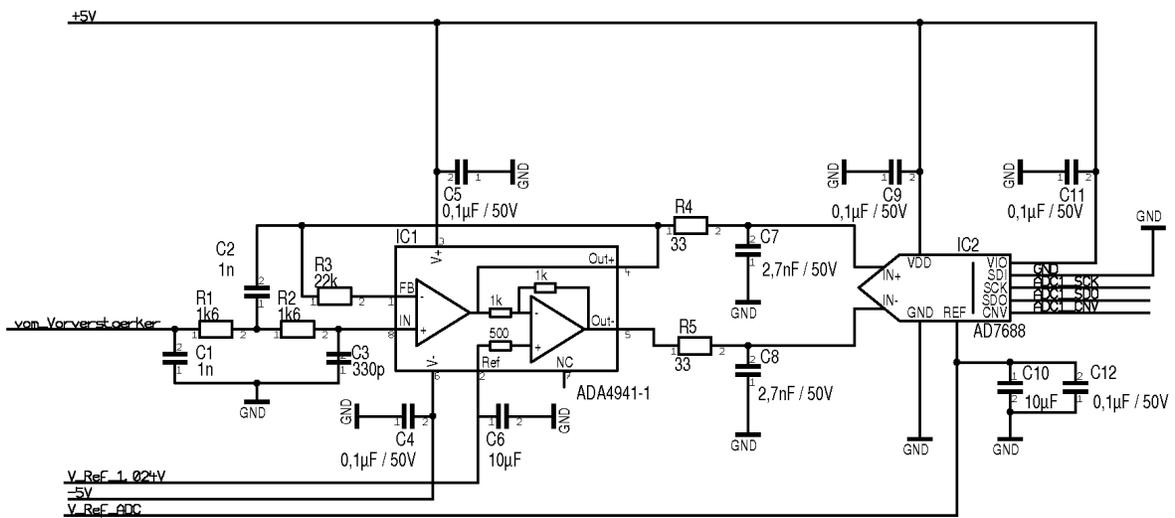


Abbildung 3.12: Aufbau des ADCs mit zugehörigem Antialiasing-Tiefpass.

teilen der ADCs. Der ADA4941 ist speziell für diese Anwendungen konzipiert und eigentlich für 18-Bit ADCs ausgelegt [34] worden. Mit Hilfe von  $V_{Ref\_1.024V}$  erzeugt der ADA4941 zwei um  $+1,024 V$  symmetrische Ausgangsspannungen, welche über zwei weitere Tiefpässe (R4 und C7 bzw. R5 und C8) die differentiellen Eingänge der ADCs speisen.

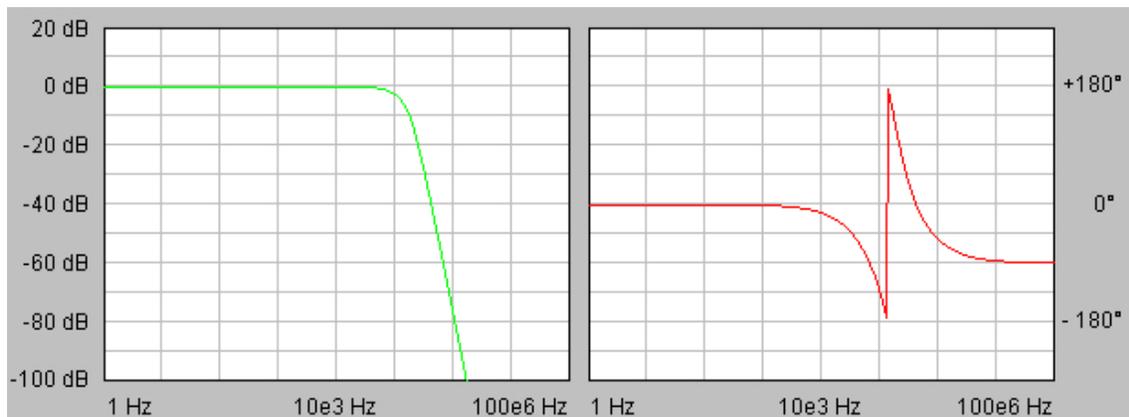


Abbildung 3.13: Frequenzgang (links) und Phasengang (rechts) des Antialiasing-Tiefpasses [4].

Beim Layout wurde darauf geachtet, dass der Digitalteil, sprich die digitalen Eingangssignale der ADCs und des DACs von dem restlichen analogen Teil der Schaltung getrennt ist. Hierzu wurde die Layoutempfehlung aus den Datenblättern übernommen [6]. Erste Ergebnisse (siehe Kapitel 4) haben gezeigt, dass diese Trennung nicht ausreichend war und ein neues Layout für die ADC/DAC-Platine notwendig ist.

### 3.2.3 Mess Modul

Das Messmodul dient zur Anbindung der einzelnen Messsensoren. Das Messmodul ist über Steckverbinder mit dem Basismodul verbunden. Die Zuleitungen sind impedanzarm und kurz gehalten, da über sie die analogen Messsignale übertragen werden.

Das Layout des Messmoduls ist so konzipiert, dass die Abmessung des Basismoduls möglichst klein gehalten sind. Hierzu ist das Messmodul in vier einzelne Platinen aufgeteilt (siehe Abb. 3.14). Untergebracht ist das Messmodul in einem hierfür konzipierten Gehäuse [16]. Das Gehäuse ist so aufgebaut, dass es das umgebende Medium möglichst nicht beeinflusst, wodurch die kompakte Bauform der einzelnen Platinen resultiert.

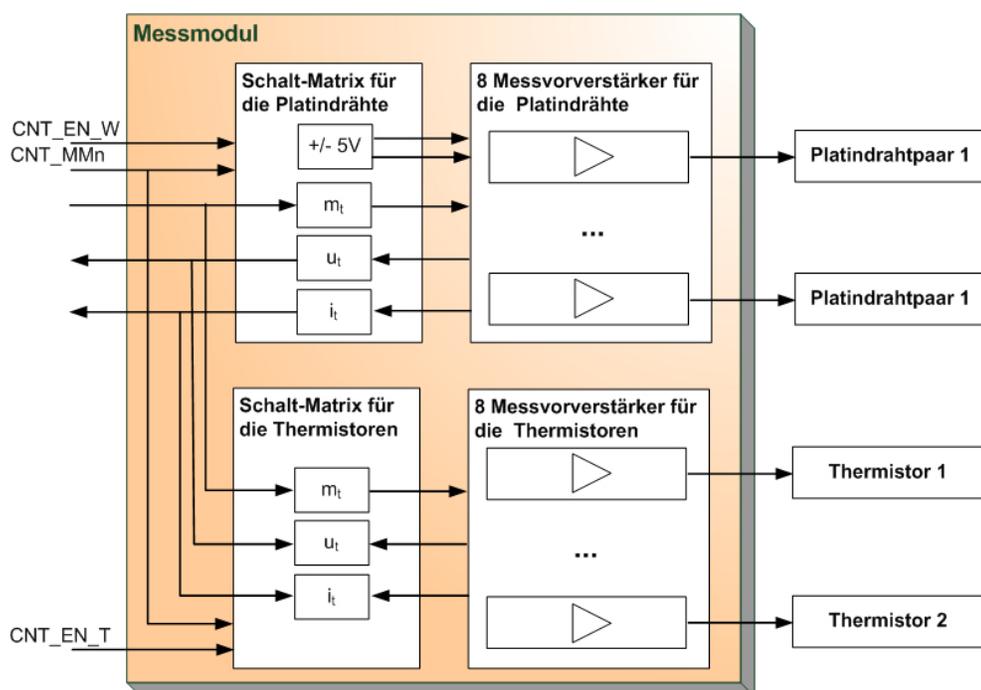


Abbildung 3.14: Schematischer Aufbau des Messmoduls.

Pro Messmodul können acht Platindrahtpaare und acht Thermistoren angeschlossen werden. Die Messsensoren sind durch Messvorverstärker in 4-Leiter Anschlusstechnik von dem Rest der Schaltung entkoppelt.

Die einzelnen Platinen haben dabei folgende Funktion:

- Über die Platine *Schalt-Matrix für die Platindrähte* werden mit Hilfe von fünf ADG658 analog Multiplexen die jeweiligen Messsensoren für die Platindrähte ausgewählt. Es

werden dabei die analogen Messsignale  $m_t$ ,  $u_t$  und  $i_t$  an einen der acht Messsensoren durchgeschaltet. Weiter werden die +/- 5V Versorgungsspannungen für die jeweiligen Messsensoren geschaltet. Ein Schalten der Versorgungsspannung ist notwendig, da durch das umgebende Salzwasser die einzelnen Platindrahtpaare untereinander elektrisch leitend verbunden sind. Durch die Strommessung liegt ein Platindraht während der Messung auf Masse. Dieser Masseschluss würde die Messung verfälschen, weshalb alle anderen Platindrahtpaare während einer Messung potentialfrei sein müssen. Über CNT\_EN\_W wird eine Messung von den Platindrähten initialisiert. Mittels der drei digitalen Signale CNT\_MMn werden die acht Messsensoren ausgewählt.

- Über die Platine *Schalt-Matrix für die Thermistoren* werden die jeweiligen zu messenden Thermistoren ausgewählt. Die Funktionsweise ist identisch mit denen der *Schalt-Matrix für die Platindrähte*. Einzig die Spannungsversorgung und das Messsignal ( $m_t$ ) wird nicht geschaltet, da dies bei den Thermistoren nicht notwendig ist. Über CNT\_EN\_T wird eine Messung von den Thermistoren initialisiert.
- Auf der Platine *Messvorverstärker für die Platindrähte* sind acht einzelne Messvorverstärker für die Platindrähte untergebracht.
- Auf der Platine *Messvorverstärker für die Thermistoren* sind acht einzelne Messvorverstärker für die Thermistoren untergebracht.

Die jeweiligen Platinen sind über Platinensteckverbinder direkt miteinander verbunden.

Abb. 3.15 zeigt den Aufbau der Messvorverstärker. IC1a dient zur Entkopplung des Messsignals von den Messsensoren, welche an K1 und K2 angeschlossen werden. Da im Speziellen die Kapazität zwischen den Platindrähten mit einigen  $\mu\text{F}$  recht hoch werden kann, ist ein Operationsverstärker notwendig, welcher große Kapazitäten treiben kann. Aus diesem Grund ist ein AD8698 gewählt, welcher zum Treiben großer Kapazitäten geeignet ist [39]. Weiter besitzt er eine kleine Offset-Spannung und ist sehr rauscharm.

Die beiden Messvorverstärker zur Messung der Spannung und des Stroms sind AD8510 Operationsverstärker, welche ebenfalls sehr rauscharm sind [40]. Die Spannung wird mittels einer subtrahierenden Operationsverstärkerschaltung gemessen. Die Ausgangsspannung ergibt sich bei gleichem Widerstandswert aller Widerstände zu

$$u_{\text{Mess\_out}} = u_{K1} - u_{K2}. \quad (3.5)$$

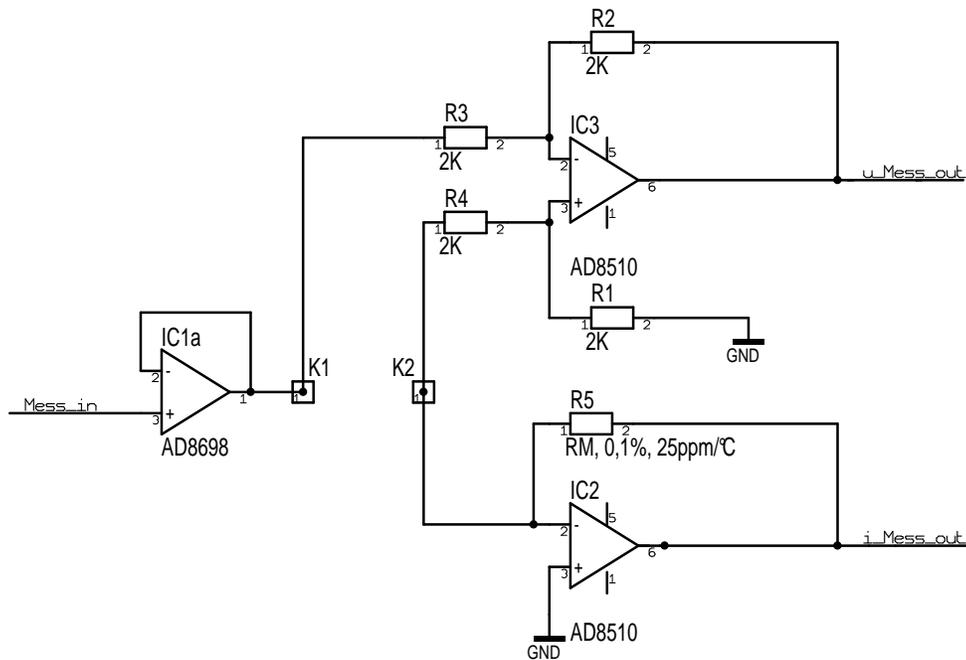


Abbildung 3.15: Aufbau der Messvorverstärker.

Die Widerstände sind dabei vom Layout her so angeordnet, dass sie alle nahe beieinander liegen. Weiter sind sie aus einer Charge. Dies gewährleistet eine geringe Drift in Abhängigkeit von der Temperatur.

Der Strom wird über einen *Current Feedback Amplifier* gemessen, welcher einen U/I-Wandler darstellt. Der Vorteil dieser Variante ist der geringe Fehler beim Messen von kleinen Strömen [41; 42]. Bei Strömen unter 20mA sollte nicht die Standardvariante über einen Messwiderstand gewählt werden, da der sich ergebende Messfehler zu hoch wäre. Die Ausgangsspannung ergibt sich zu

$$u_{\text{Mess\_out}} = i_{K1} R5. \quad (3.6)$$

Um den Fehler durch R5 gering zu halten, wird ein Präzisionswiderstand mit geringer Temperaturdrift verwendet. Für die Messung des Stroms an den Platindrähten wird ein  $100 \Omega$  und bei Messung des Thermistorstroms ein  $2 \text{ k}\Omega$  Widerstand verwendet. Die Werte sind so dimensioniert, dass ein möglichst hoher Ausgangspegel erreicht wird.

Die Thermistoren besitzen Widerstandswerte zwischen  $2,5 \text{ k}\Omega$  (bei  $25^\circ\text{C}$ ) und  $40 \text{ k}\Omega$  (bei  $-30^\circ\text{C}$ ). Bei Anlegen einer  $1 \text{ V}$  Messspannung ergeben sich mit dem verwendeten  $2 \text{ k}\Omega$  Widerstand Spannungen am Ausgang der Strommessschaltung von  $800 \text{ mV}$  bis  $5 \text{ mV}$ . Bei den Platindrähten ergeben sich bei Anlegen einer  $20 \text{ mV}$  Messspannung und typischen Wider-

standswerten zwischen den Platindrähten von  $5 \Omega$  (reines Salzwasser) bis  $100 \Omega$  (Platindrähte eingeschlossen im Eis) Ausgangsspannungen von 400 mV bis 2 mV.

### 3.3 FPGA und Custom-IP-Core

Das folgende Unterkapitel befasst sich mit dem Aufbau innerhalb des FPGAs auf dem ML403 Development Boards. Mit Hilfe der EDK Software ist innerhalb des FPGAs ein Microblaze Softcore Prozessor (siehe Abb. 3.16) implementiert.

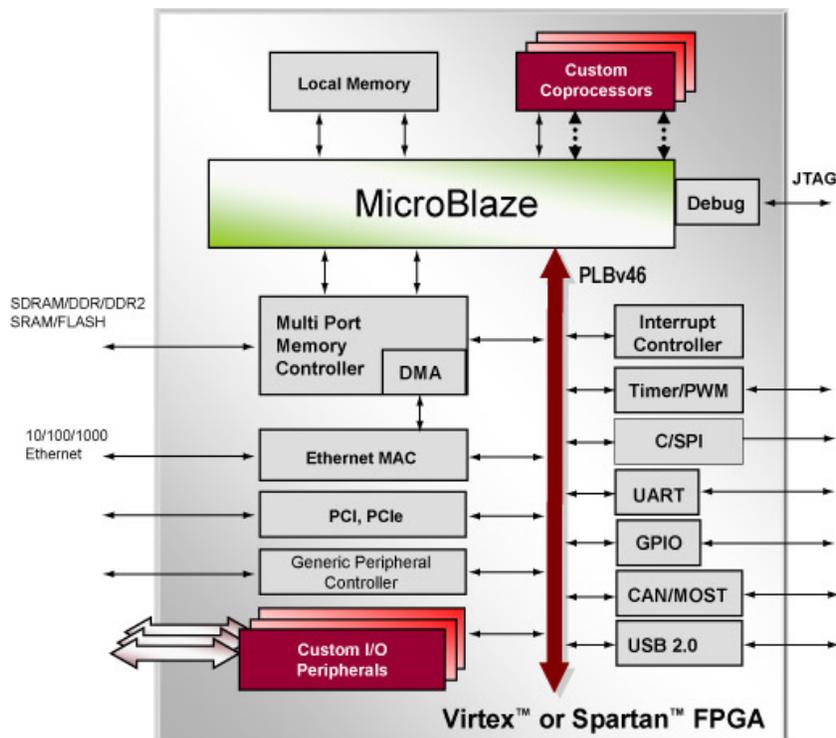


Abbildung 3.16: Microblaze Prozessor [5].

Bei dem Microblaze handelt es sich um einen 32-Bit RISK Prozessor, welcher über ein Bus-system (PLB) mit anderen Peripheriebausteinen (IP-Cores) innerhalb des FPGAs angebunden ist. Über eine JTAG-Schnittstelle lässt sich der Microblaze programmieren, bzw. dient es auch zur Fehlersuche (Debug). Mit Hilfe des EDKs lassen sich weitere Peripheriebausteine, wie z.B. ein Interrupt-Controller oder ein Timer, in den FPGA integrieren. Diese sind ebenfalls an den PLB angeschlossen. Als ein spezieller Peripheriebaustein lässt sich ein Custom-IP-Core (Cu-

stom I/O Peripherals) implementieren, mit dessen Hilfe eigene Hardware-Applikationen in das System eingebunden werden.

Für die Anbindung der Hardware der Harp3d ist ein eigener Custom-IP-Core erstellt worden. Er beinhaltet die Messsignalerzeugung und die Anbindung an den SPI-Bus.

Die folgenden beiden Unterkapitel befassen sich mit dem Aufbau, insbesondere der verwendeten Peripherie innerhalb des FPGAs und mit dem erstellten Custom-IP-Core.

### 3.3.1 FPGA

Die Erstellung der Hardware innerhalb des FPGA ist mit Hilfe der EDK Software in der Version 9.2.02 erstellt worden. Die Software für den Microblaze ist mittels dem SDK in der Version 9.2.02 erstellt.

Für den realisierten Aufbau innerhalb des FPGAs (siehe Abb. 3.17) wurden mehrere IP-Cores, im Speziellen LogiCores™<sup>15</sup>, in das System eingebunden, zu ihnen gehören

- der Microblaze Prozessor (microblaze\_0, im oberen rechten Bereich), welcher als eigentlicher Prozessor innerhalb des FPGA dient. Grundsätzlich hätte auch ein Power-PC, welcher einen Hardware-Prozessor darstellt, als Prozessor genutzt werden können, da er ebenfalls auf dem verwendeten FPGA enthalten ist. Versuche mit einem implementierten Power-PC zeigten allerdings ein fehlerhaftes Verhalten beim Laden der Initialisierungsdateien von der Compact-Flash-Karte. Der Power-PC startet nach dem Laden der Dateien nicht selbstständig und musste über die Reset-Taste am ML403 Development Board von Hand gestartet werden.
- ein Block-RAM (bram\_block, im oberen rechten Bereich), welcher als RAM für den Microblaze dient. Der Block-RAM ist über zwei Busse (DLMB und ILMB) mit dem Microblaze verbunden. Grundsätzlich dient der Block-RAM zur Speicherung und Ablaufsteuerung der auf dem Microblaze ausgeführten Software. Da der mit 32 kB große Block-RAM zu klein für den erstellten Programmcode (siehe Kapitel 3.4.3) ist, wurde der Programmcode auf dem externen DDR-RAM untergebracht. Hierzu ist das Linker-Script des SDKs so modifiziert, dass alle Sektionen des Programms auf dem DDR-RAM untergebracht sind. Einzig die Heap- und Stack-Sektionen<sup>16</sup> sind im Block-RAM implementiert. Eine

<sup>15</sup>LogiCores™ sind spezielle IP-Cores der Firma Xilinx®, welche kompatibel mit dem IBM CoreConnect™ (PLB) sind.

<sup>16</sup>diese dienen als Speicher von Umgebungsvariablen wenn innerhalb eines Programmablaufs im Programm „gesprungen“ wird, zum Beispiel bei Funktionsaufrufen.

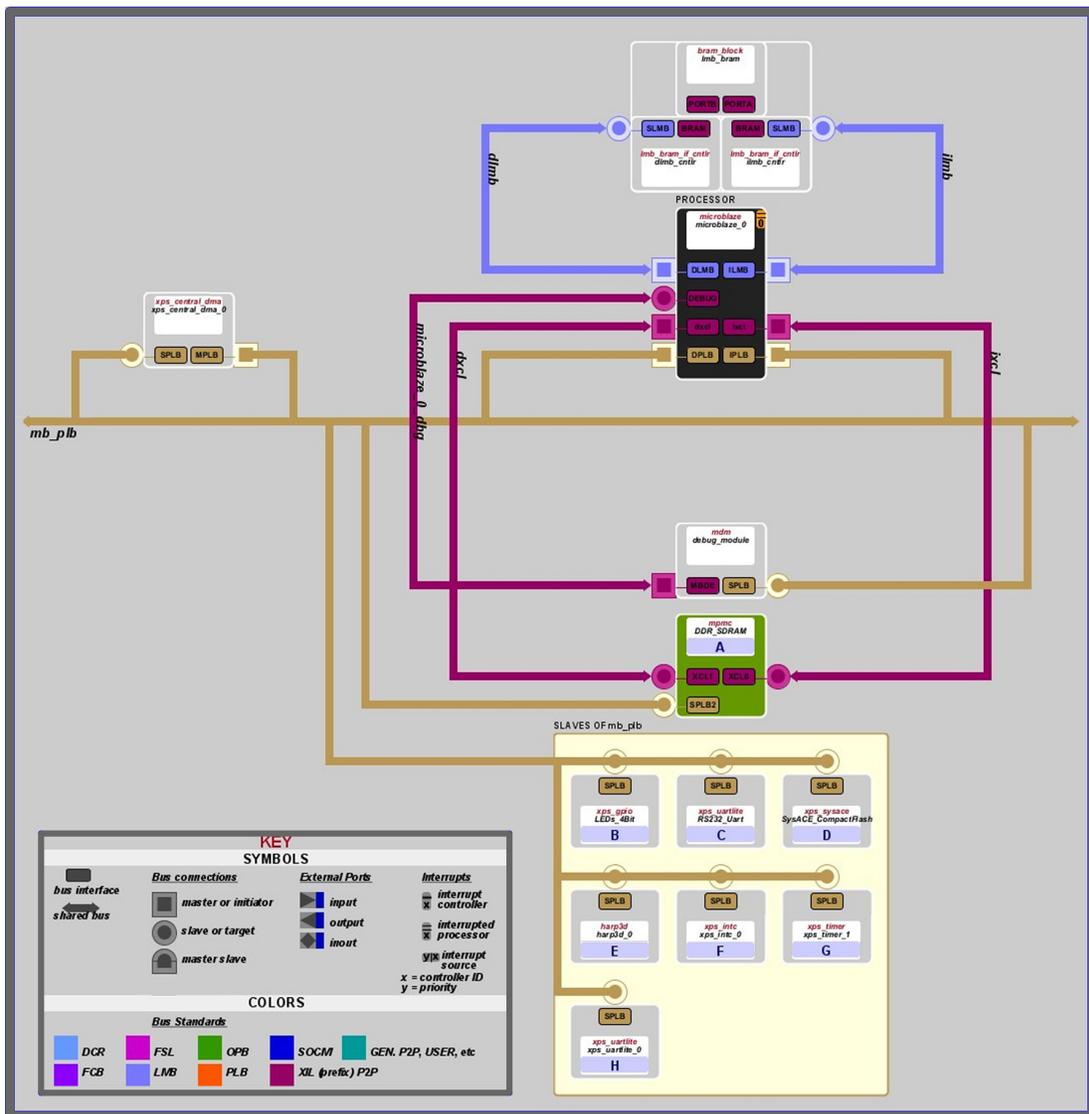


Abbildung 3.17: Aufbau der Hardwarestruktur innerhalb des FPGAs.

Konsequenz hiervon ist, dass der Programmablauf innerhalb des Microblazes etwa um den Faktor 4 langsamer ist, als wenn er aus dem Block-RAM ausgeführt wird.

- ein IP-Core zur Fehlersuche (debug\_module, im mittleren rechten Bereich). Über diesen IP-Core ist es möglich den Programmablauf im Microblaze über die JTAG-Schnittstelle direkt zu steuern.
- ein IP-Core zur Anbindung des externen DDR-RAMs (DDR\_SDRAM, im mittleren rechten Bereich). Der DDR-RAM ist über drei Busse ( PLB, DXCL und IXCL) an den Microblaze angebunden. Die XCL-Busse dienen zur Verwendung des ebenfalls implemen-

tierten Cache-Speichers. Dieser dient neben der Speicherung des Programmablaufs auch zur Zwischenspeicherung der aufgenommenen Messwerte, welche über den DMA-Controller direkt von dem Custom-IP-Core in den DDR-RAM übertragen werden.

- ein DMA-Controller (`xps_central_dma_0`), welcher einen direkten Speicherzugriff über den PLB auf den DDR-RAM ermöglicht.
- ein IP-Core zur Ansteuerung von vier LEDs auf dem ML403 Development Board (`LEDs_4Bit`, unterer rechter Bereich). Die LEDs werden zur schnellen optischen Überprüfung des Programmablaufs genutzt.
- ein IP-Core zur Anbindung der seriellen Schnittstelle auf dem ML403 Development Board (`RS232_Uart`, unterer rechter Bereich). Diese RS232-Schnittstelle wird zur Überprüfung des Programmablaufs genutzt.
- ein IP-Core zur Anbindung des System ACE™ auf dem ML403 Development Board (`SysACE_Compactflash`, unterer rechter Bereich). Hierüber wird die Compact-Flash-Karte und die JTAG-Schnittstelle in das System eingebunden.
- der Custom-IP-Core (`harp3d_0`, unterer rechter Bereich).
- ein IP-Core mit der Funktion eines Interrupt-Controllers (`xps_intc_0`). Da bei der Software für den Microblaze weitestgehend auf Interrupts verzichtet wurde, ist an dem Interrupt-Controller nur der Interrupt des Zeitgebers angeschlossen.
- ein IP-Core als Zeitgeber (`xps_timer_1`, unterer rechter Bereich). Der IP-Core wird als Zeitgeber für den Programmablauf genutzt. Hierzu produziert er zyklisch in 100 ms Schritten einen Interrupt. Im Programmablauf wird über eine Interrupt-Service-Routine (ISR) hiermit die Zeit seit dem Hochfahren des Programms ermittelt. Zusätzlich werden mit Hilfe des Zeitgebers Timeouts im Programmablauf ermittelt, wie z.B. eine zu lange Messung.
- eine zweite serielle Schnittstelle (`xps_uartlite_0`, unterer rechter Bereich). Diese dient zur Anbindung des seriellen Busses der Basismodule.

Damit der Initialisierungscode des FPGAs beim Starten des FPGAs von der Compact-Flash-Karte geladen werden kann, muss ein eine so genannte .ace-Datei erstellt werden. Die .ace-Datei vereinigt die

- .bit-Datei, welche den strukturellen Aufbau innerhalb des FPGAs beinhaltet und
- die .elf-Datei, welche den gelinkten Programmcode für die Software enthält.

Beim Starten des FPGAs werden mittels der .ace-Datei durch den System ACE™ zuerst die Struktur in den FPGA geladen, anschließend wird der Block-RAM und der DDR-RAM initialisiert und das Programm gestartet.

Tabelle 3.2: Aus dem FPGA herausgeführte Anschlüsse.

Bezeichnung	Richtung	Signal	Anschluss an
RX	in	fpga_0_RS232_Uart_RX	xps_uartlite_0
TX	out	fpga_0_RS232_Uart_TX	xps_uartlite_0
adc_spi_clk_in	in	harp3d_0_adc_spi_clk_in	harp3d_0 Custom-IP-Core
adc2_in	in	harp3d_0_adc2_in	harp3d_0 Custom-IP-Core
adc1_in	in	harp3d_0_adc1_in	harp3d_0 Custom-IP-Core
adc_cs_cnv_in	in	harp3d_0_adc_cs_cnv_in	harp3d_0 Custom-IP-Core
dac_out	out	harp3d_0_dac_out	harp3d_0 Custom-IP-Core
cs_cnv_out	out	harp3d_0_cs_cnv_out	harp3d_0 Custom-IP-Core
spi_clk_out	out	harp3d_0_spi_clk_out	harp3d_0 Custom-IP-Core

Zur Anbindung an das Basismodul sind aus dem FPGA die in Tabelle 3.2 aufgeführten Signale an die Adapterplatine für das ML 403 Development Board herausgeführt. Bei ihnen handelt es sich um die zwei Signalleitungen der seriellen Schnittstelle (RX und TX), sowie die Signale für die Anbindung des SPI-Busses. Weitere aus dem FPGA herausgeführte Anschlüsse verbleiben auf dem ML403 Development Board.

### 3.3.2 Custom-IP-Core

Zur Anbindung des Basismoduls über den SPI-Bus und zur Signalerzeugung ist mit Hilfe des ISE® WebPACK™ in der Version 10.1 ein Custom-IP-Core entwickelt worden. Der Custom-IP-Core ist über den PLB an den Microblaze angeschlossen und kann über die Software im Microblaze gesteuert werden. Der Custom-IP-Core (siehe Abb. 3.18) besteht aus dem

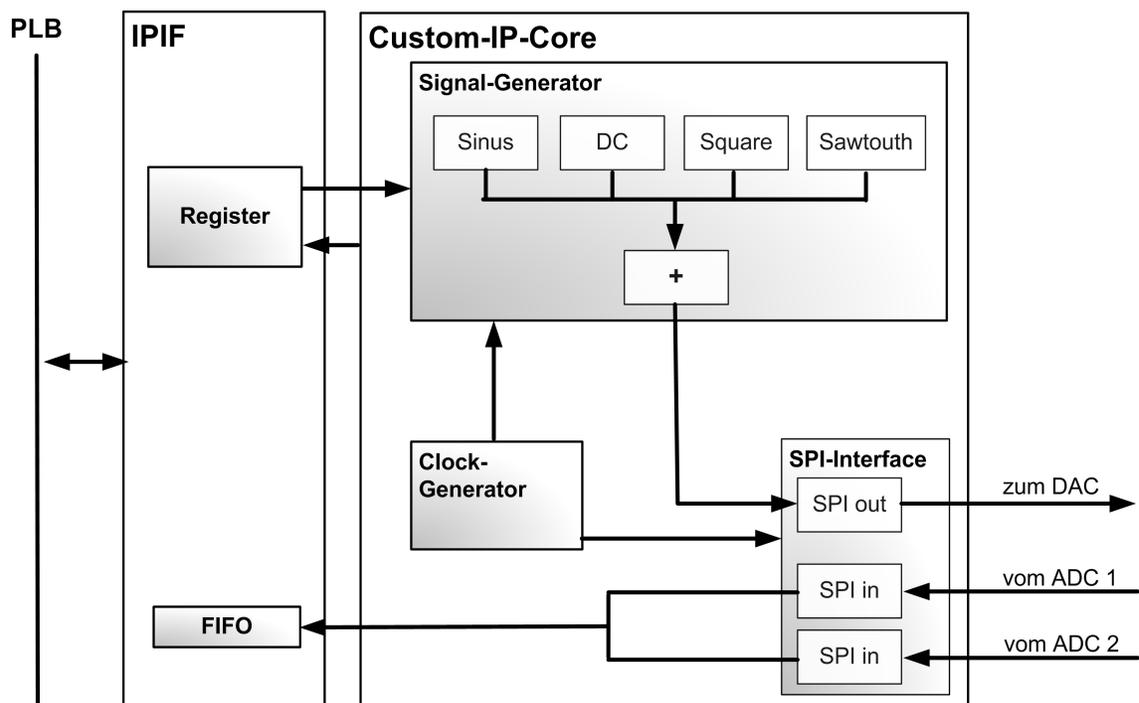


Abbildung 3.18: Schematischer Aufbau des Custom-IP-Cores.

Signal-Generator, dem Clock-Generator und dem SPI-Interface. Über eine IPIF<sup>17</sup>-Schnittstelle ist der Custom-IP-Core an den PLB angebunden. Zur Übertragung der Messsignale in den DDR-RAM ist innerhalb des IPIF ein FIFO implementiert. Dies gewährleistet die störungsfreie Übertragung über den DMA-Controller. Um die einzelnen Komponenten des Custom-IP-Cores steuern zu können, verfügt das IPIF weiter über vier 32-Bit Register, welche an den Custom-IP-Core angebunden sind.

<sup>17</sup>IPIF ist ein IP-Core der Firma Xilinx und steht für IP Interconnect. Womit ausgedrückt werden soll, dass der IPIF eine Schnittstelle zwischen PLB und einem Custom-IP-Core darstellt.

Der Custom-IP-Core ist aufgrund seiner Komplexität in die drei Teilbereiche aufgeteilt.

- Der Clock-Generator stellt alle für den Betrieb des Custom-IP-Cores benötigten Clock-Signale zur Verfügung.
- Der Signal-Generator erzeugt über eine Addiererstufe durch mehrere Signalerzeugungskomponenten, wie einer Sinusspannungsquelle (Sinus), einer Gleichspannungsquelle (DC), einer Rechteckspannungsquelle (Square) und einer Dreieckspannungsquelle (Sawtooth), das digitale Messsignal. Das Messsignal wird durch das SPI-Interface über den SPI-Bus an den DAC auf den Basismodulen übertragen. Die Einstellung der benötigten Parameter, wie die Ausgangsfrequenz und die Amplitude, der einzelnen Signalerzeugungskomponenten erfolgt über Registerwerte innerhalb der IPIF. Über Registerwerte ist auch der Addierer einstellbar.
- Das SPI-Interface stellt die Schnittstelle zum SPI-Bus dar. Hierzu wird einerseits das vom Signal-Generator erzeugte Signal in ein serielles SPI-Signale für die DACs gewandelt. Andererseits werden die beiden seriellen SPI-Signale von den ADCs in ein paralleles 16-Bit Signal für den FIFO rückgewandelt.

Zum unabhängigen Testen des Custom-IP-Cores können die Ausgänge des SPI-Interfaces über ein Registerwert auf die Eingänge des SPI-Interfaces gelegt werden (loopback).

Über den im EDK enthaltenen *Create and Import Peripheral Wizard* ist die Umgebung für den Custom-IP-Core erstellt worden. Das EDK erstellt hierzu eine Musterumgebung (user\_logic), welche über das IPIF die Anbindung an den PLB darstellt. Innerhalb der user\_logic wird die Anbindung an die Komponenten des IPIF implementiert. Bei der Erstellung der Musterumgebung durch den *Create and Import Peripheral Wizard* ist das IPIF so ausgelegt worden, dass es einen Ausgangs-FIFO mit Burst-Anbindung an den PLB, je zwei 32-Bit über Software steuerbare Ein- und Ausgangsregister und einen softwaregesteuerten Reset-Eingang besitzt. In die so erstellte user\_logic ist anschließend der Custom-IP-Core implementiert worden.

### 3.3.2.1 Schnittstelle zum IPIF-Interface

Die Ein- und Ausgangssignale (siehe Abb. 3.19) des Custom-IP-Cores sind über die user\_logic und das IPIF an den PLB, bzw. an die Ausgänge des FPGAs angebunden.

Zu den Ein- und Ausgangssignalen gehören

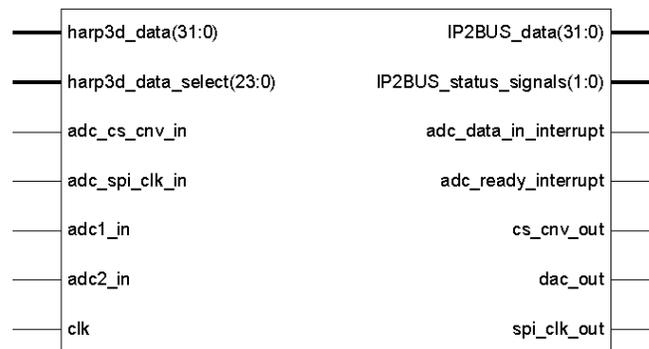
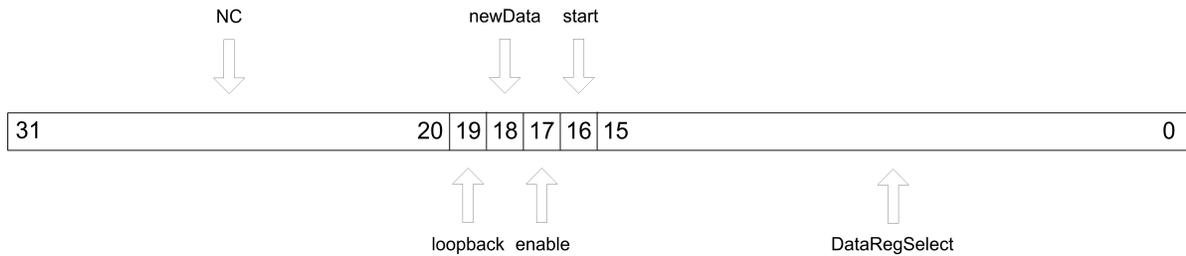


Abbildung 3.19: Signale des Custom-IP-Cores zum IPIF.

- zwei Datenbusse harp3d\_data und harp3d\_data\_select zur Anbindung an die Eingangsregister der IPIF.
- der Datenbus IP2BUS\_status\_signals zur Anbindung an ein Ausgangsregister der IPIF.
- der Datenbus IP2BUS\_data zur Anbindung an den FIFO des IPIF.
- die beiden direkt herausgeführten Interruptquellen adc\_data\_in\_interrupt und adc\_ready\_interrupt.
- die Ein- und Ausgangssignale zur Anbindung an den SPI-Bus, welche durch das IPIF an die Ausgänge des FPGA geführt sind.
- das 100 MHz clk-Signal des FPGAs.

Alle Einstellungen innerhalb des Custom-IP-Cores werden über harp3d\_data und harp3d\_data\_select gesteuert. Über das 32-Bit breite harp3d\_data-Signal werden die Werte für die Einstellungen der verschiedenen Komponenten, wie z.B. die Frequenz der Sinusspannungserzeugung, innerhalb des Custom-IP-Cores übertragen. Mittels harp3d\_data\_select erfolgt eine Auswahl für welche Komponente die Datenwerte bestimmt sind. Die Steuerung des Messablaufs, wie der Start der Messung, erfolgt ebenfalls über harp3d\_data\_select (siehe Abb. 3.20 und Tabelle 3.3). Die beiden Signale sind über die softwaregesteuerten Eingangsregister der user\_logic änderbar.

Abbildung 3.20: Signal `harp3d_data_select` des Custom-IP-Cores.Tabelle 3.3: Signal `harp3d_data_select` des Custom-IP-Cores.

Bits	Name	Beschreibung
0 - 15	<code>component_select</code>	Auswahl über die zu verwendende Komponente für die Daten an <code>harp3d_data</code> .
16	<code>start</code>	Startsignal für eine Messung.
17	<code>enable</code>	Enable-Signal für den Custom-IP-Core. Ein Low-Signal an <code>enable</code> versetzt alle Komponenten innerhalb des Custom-IP-Cores in einen definierten Ausgangszustand (Reset). Das Enable-Signal wird innerhalb der <code>user_logic</code> unter Berücksichtigung der verschiedenen Interruptquellen generiert.
18	<code>newData</code>	Zeigt für zwei <code>clk</code> -Zyklen (High-Signal) an, dass sich <code>component_select</code> geändert hat. Es wird innerhalb des <code>user_logic</code> erzeugt.
19	<code>loopback</code>	Signal zum Testen der SPI-Schnittstelle des Custom-IP-Cores. Bei einem High-Signal sind die Ausgangssignale der SPI-Schnittstelle direkt an die Eingänge der SPI-Schnittstelle geführt.
20 - 31		nicht angeschlossen

Über das Signal IP2BUS\_status\_signals werden Statussignale des Custom-IP-Cores an die user\_logic übermittelt. Mit den innerhalb der user\_logic generierten Statussignalen werden diese in eines der Ausgangsregister (siehe Abb. 3.21 und 3.4) übertragen. Hierdurch ist es möglich über die Software im Microblaze den aktuellen Status des Custom-IP-Cores und der user\_logic abzufragen.

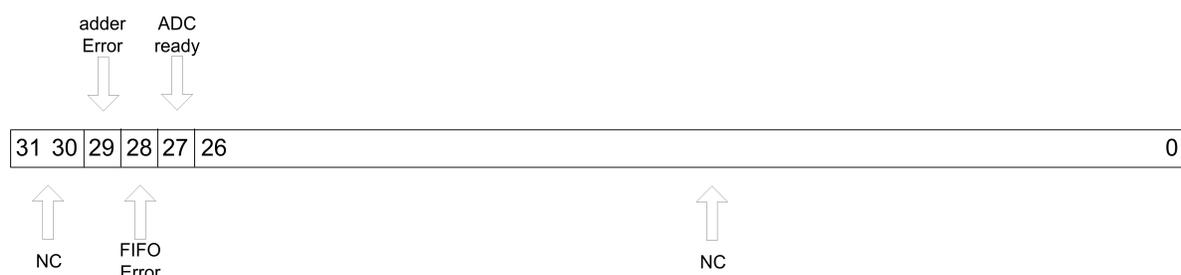


Abbildung 3.21: Ausgangsregisterwerte der user\_logic.

Tabelle 3.4: Ausgangsregisterwerte der user\_logic.

Bits	Name	Beschreibung	Quelle
0 - 26, 30, 31		nicht verwendet	
27	ADCready	High-Signal wenn eine Messung beendet ist.	Custom-IP-Core
28	FIFOError	High-Signal wenn der FIFO im IPIF voll ist. Dies geschieht wenn der DMA die Daten nicht schnell genug in den DDR-RAM schreiben konnte.	user_logic
29	adderError	High-Signal wenn der Addierer im Custom-IP-Core „überläuft“, sprich wenn der Ausgangswert den definierten Zahlenbereich von 16-Bit übersteigt.	Custom-IP-Core

Das 32-Bit Signal IP2BUS\_data beinhaltet die aufgenommenen Messsignale von den ADCs am SPI-Bus. Das Signal wird bei der positiven Taktflanke des adc\_data\_in\_interrupt-Signals in den FIFO innerhalb des IPIFs übertragen.

Zur Generierung des Enable-Signals (enable) für den Custom-IP-Core sind innerhalb der user\_logic vier Interruptquellen (adc\_ready\_interrupt, adc\_data\_in\_interrupt, adderError und FIFOerror) zusammengefasst. Zusätzlich ist das Enable-Signal vom Software-Reset des IPIFs abhängig.

### 3.3.2.2 SPI Interface

Das SPI-Interface enthält eine SPI-Ausgangsschnittstelle zur Anbindung der DACs und zwei SPI-Eingangsschnittstellen zur Anbindung der ADCs. Grundsätzlich existieren vorhandene SPI-Cores im CORE Generator™ innerhalb des ISE®, allerdings sind diese nur für Datenbreiten bis 8-Bit ausgelegt, so dass diese nicht verwendet werden konnten. Aus diesem Grund sind eigene SPI-Schnittstellen entworfen worden. Grundlage hierfür sind die Spezifikationen der ADCs und DACs.

Die Funktionsweise des ADCs (siehe Abb. 3.22) ist, dass am Eingang anliegende analoge

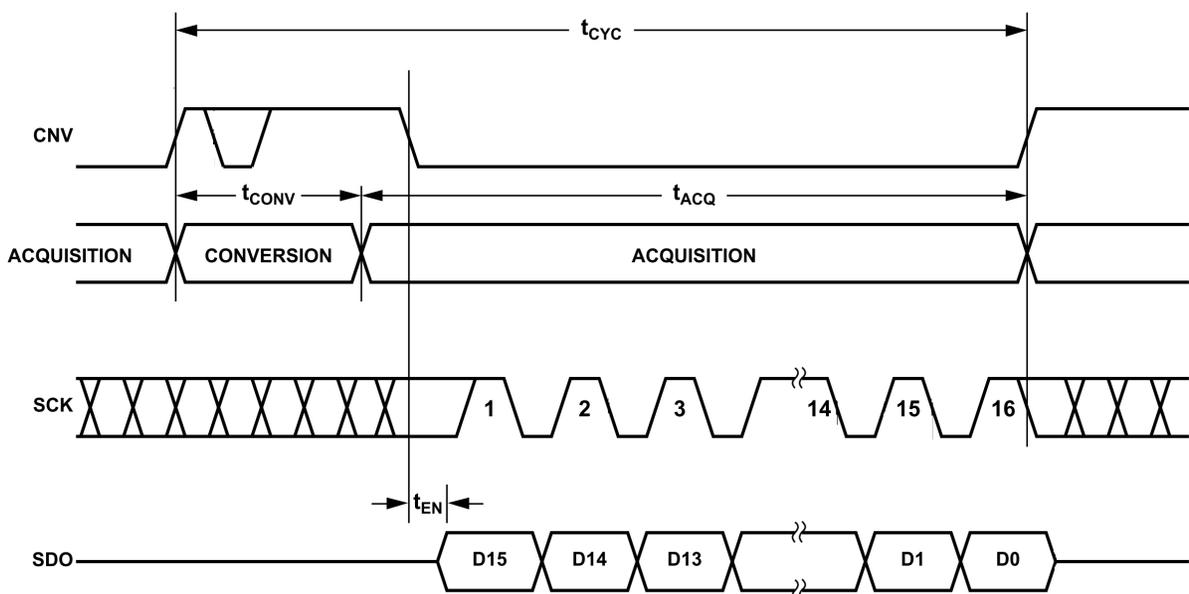


Abbildung 3.22: Funktionsweise des ADCs [6].

Signal zuerst über eine Halteschaltung mit Kondensatoren aufzunehmen (ACQUISITION) und anschließend in ein digitales 16-Bit Signal zu wandeln (CONVERSION). Eine Wandlung des aufgenommenen Eingangssignals erfolgt bei der positiven Flanke des CNV-Signals. Bei der negativen Flanke des CNV-Signals wird das digitale Ausgangssignal am SDO-Anschluss ausgegeben. Hierzu wird bei jeder negativen Flanke des SPI-Clock-Signals (SCK) je ein Bit am SDO-Anschluss ausgegeben. Für eine Signalaufzeichnung benötigt der ADC mindestens  $1,6 \mu\text{s}$  ( $t_{ACQ}$ ). Für eine Wandlung mindestens  $0,4 \mu\text{s}$  ( $t_{CONV}$ ). Hieraus ergibt sich eine Gesamtzeit pro Zyklus ( $t_{CYC}$ ) von  $2 \mu\text{s}$ , bzw. eine maximale Wandelrate von 500 kHz.

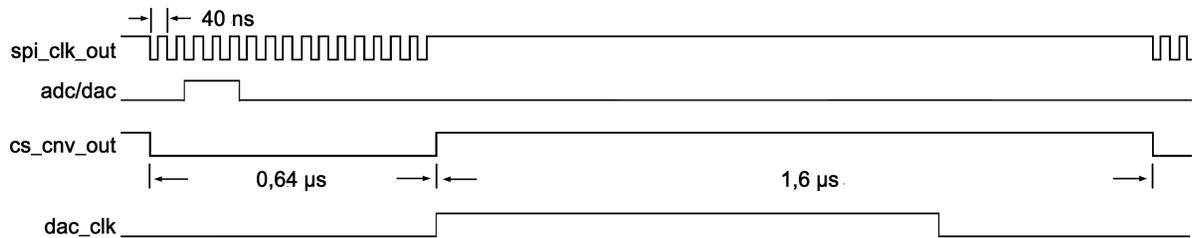


Abbildung 3.23: Signale am SPI-Interface.

Das SPI-Clock-Signal und CNV-Signal muss durch das SPI-Interface zur Verfügung gestellt werden. Über die SPI-Ausgangsschnittstelle sind hierzu die beiden Signale `spi_clk_out` und `cs_cnv_out` (siehe auch Abb. 3.23) an die ADCs bzw. DACs angebunden. Die Periodendauer (40 ns) des SPI-Clock-Signals ergibt sich aus den Spezifikationen des DACs. Für eine 16-Bit Übertragung werden demnach  $640 \mu\text{s}$  benötigt. In dieser Zeit hat das `cs_cnv_out`-Signal einen Low-Pegel. Im Anschluss an die Datenübertragung folgt eine Pausenzeit von 1,6 ns, damit die ADCs genügend Zeit für die Aufnahme des analogen Eingangssignal haben. In dieser Zeit hat das `spi_clk_out`- und `cs_cnv_out`-Signal einen High-Pegel.

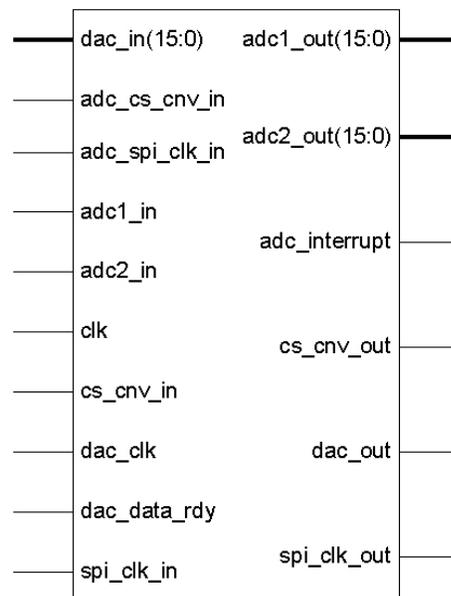


Abbildung 3.24: Anschlüsse am SPI-Interface.

Das `dac_clk`-Signal wird nicht an die ADCs bzw. DACs geleitet, es wird intern im SPI-Interface genutzt. Die Signale `cs_cnv_out`, `spi_clk_out` und `dac_clk` werden in der Clock-

Generation erzeugt und liegen als Eingangssignale am SPI-Interface (siehe Abb. 3.24) an.

Weitere Eingangssignale sind

- das digitale 16-Bit breite Messsignal (`dac_in`) von der Signal-Generation,
- ein Statussignal (`dac_data_rdy`) von der Signal-Generation. Bei einem High-Pegel von `dac_data_rdy` liegen gültige Daten am `dac_in`-Eingang an.
- das 100 Mhz Clock-Signal des FPGAs.
- die Signale der ADCs vom SPI-Bus.

Als Ausgangssignale erzeugt das SPI-Interface die beiden in 16-Bit gewandelten ADC-Signale (`adc1_out` und `adc2_out`), die SPI-Signale für den DAC und ein Interrupt-Signal. Das Interrupt-Signal hat für genau zwei `clk`-Zyklus einen High-Pegel bei neu empfangenen Daten an den ADC-Eingängen.

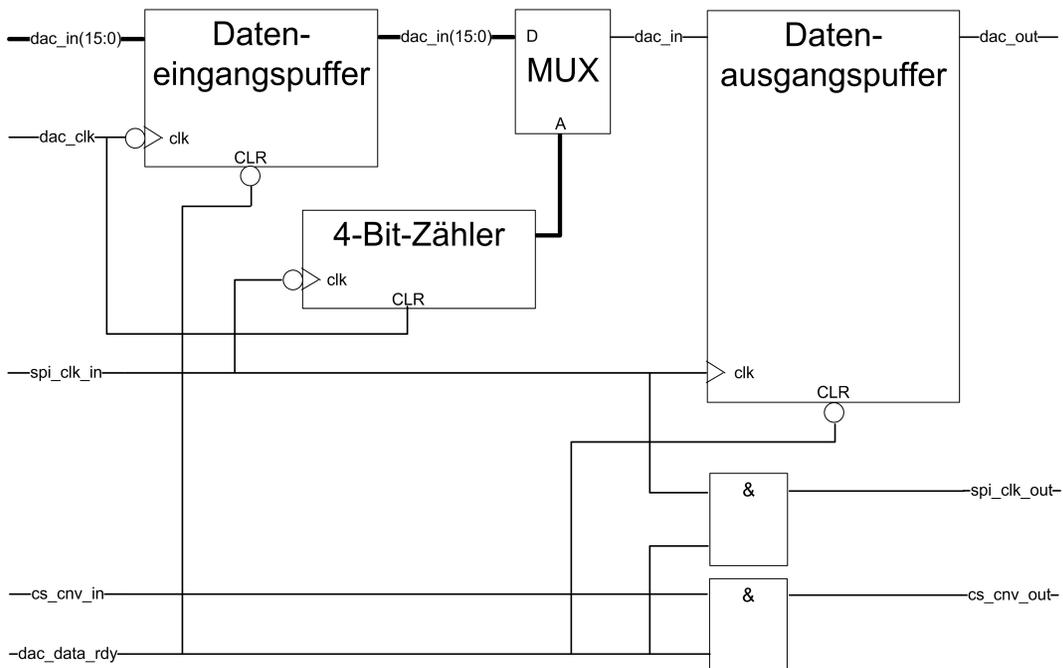


Abbildung 3.25: SPI-Ausgangsschnittstelle innerhalb des SPI-Interfaces.

Die SPI-Ausgangsschnittstelle (siehe Abb. 3.25) zur Anbindung der DACs beinhaltet einen Dateneingangspuffer in dem das `dac_in`-Signal von der Signal-Generation bei der negativen

Flanke des `dac_clk`-Signals zwischengespeichert wird. Der Inhalt des Dateneingangspuffers kann über ein Low-Pegel des `dac_data_rdy`-Signals zurück gesetzt werden. Ein 4-Bit-Zähler zählt bei jeder negativen Taktflanke des `spi_clk_in`-Signals von 15 auf 0 runter. Ein High-Pegel des `dac_clk`-Signals setzt den 4-Bit-Zähler auf seinen Startwert von 15 zurück. Über einen Multiplexer werden so Bit für Bit aus dem zwischengespeicherten `dac_in`-Signal an den Datenausgangspuffer übertragen. Hierbei liegt nach der negativen Flanke des `dac_clk`-Signals und somit vor der Datenübertragung immer Bit 15 am Eingang des Datenausgangspuffers an. Der Datenausgangspuffer wird über einen Low-Pegel vom `dac_data_rdy`-Signal zurück gesetzt. Die DACs auf den Basismodulen lesen bei jeder positiven Flanke des `spi_clk_out`-Signals den am Dateneingang liegenden Signalpegel aus. Aus diesem Grund werden die Werte des Ausgangspuffers bei jeder negativen Flanke des `spi_clk_in`-Signals ausgegeben. Der Datenausgangspuffer wird bei einem Low-Pegel des `dac_data_rdy`-Signals zurück gesetzt. Die beiden Signale `spi_clk_out` und `cs_cnv_out` werden bei einem High-Pegel des `dac_data_rdy`-Signals aus `spi_clk_in` und `cs_cnv_in` erzeugt.

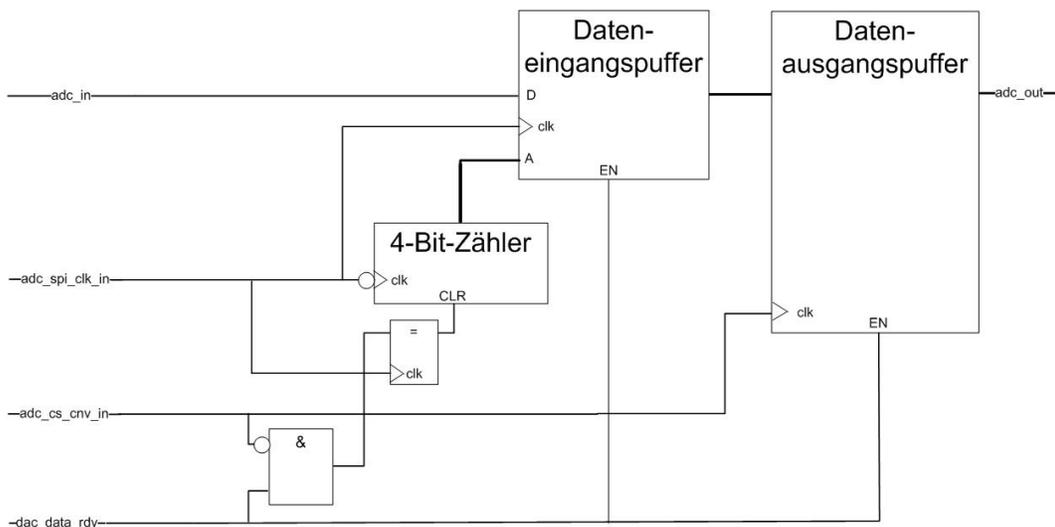


Abbildung 3.26: SPI-Eingangsschnittstelle innerhalb des SPI-Interfaces.

Die beiden SPI-Eingangsschnittstellen (siehe Abb. 3.26) zur Anbindung der DACs sind baugleich aufgebaut. An sie werden jeweils die SPI-Signale `adc_in`, `adc_spi_clk_in` und `adc_cs_cnv_in` übergeben. Der 4-Bit-Zähler zählt von 15 auf 0 runter. Wobei er jeweils um eins bei der negativen Taktflanke des `adc_spi_clk_in`-Signals dekrementiert. Bei der ersten negativen Taktflanke liegen am `adc_in`-Signal noch keine gültigen Werte an. Aus diesem Grund darf der 4-Bit-Zähler nicht bei der ersten negativen Taktflanke dekrementieren. Über das `clr`-

Signal des 4-Bit-Zählers wird mit Hilfe der Signale `adc_spi_clk_in` und `adc_cs_cnv_in` ein um einen halben Takt von `adc_spi_clk_in` verschobenes Enable-Signal erzeugt (siehe Abb. 3.27). Dieses gewährleistet, dass der 4-Bit-Zähler erst bei der zweiten negativen Taktflanke des `adc_spi_clk_in`-Signals den Ausgangswert von 15 auf 14 setzt.

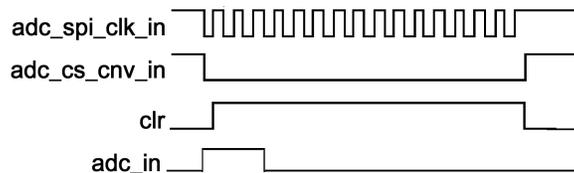


Abbildung 3.27: Signale innerhalb der SPI-Eingangsschnittstelle.

Bei jeder negativen Taktflanke des `adc_spi_clk_in`-Signals geben die ADCs die Daten an `adc_in` aus. Aus diesem Grund speichert der Eingangspuffer bei jeder positiven Taktflanke des `adc_spi_clk_in`-Signals das `adc_in`-Signal in Abhängigkeit der vom 4-Bit-Zähler angelegten Adresse in einem 16-Bit breiten internen Speicher ab. Bei der positiven Flanke des `adc_cs_cnv_in`-Signals, was das Ende der Datenübertragung anzeigt, werden die Daten des Eingangspuffers in einen 16-Bit breiten Ausgangspuffer übernommen. Im selben Augenblick wird das Interrupt-Signal (`adc_interrupt`), welches außerhalb der SPI-Eingangsschnittstelle erzeugt wird, für zwei Taktzyklen des `clk`-Signals erzeugt. Es zeigt an, dass neue Daten von den ADCs vorhanden sind.

### 3.3.2.3 Clock

Die Clock-Generation (siehe Abb. 3.28) dient zur Erstellung aller im Custom-IP-Core benötigter Clock-Signale. Zu ihnen gehören das `dac_clk`-Signal und die vom SPI-Interface benötigten Signale der SPI-Ausgangsschnittstelle (`cs_cnv_out` und `spi_clk_out`). Für den Signal-Generator wird weiter ein Signal mit einer Periodendauer von 1 ms erzeugt (`one_ms_clk`).

Bis auf `one_ms_clk` werden alle Ausgangssignale bei einem High-Pegel von `enable` generiert. Das Ausgangssignal `one_ms_clk` wird durch einen High-Pegel des Eingangssignal `start_signal_generation` gestartet. Die Signale werden mittels Teiler-Bausteinen und Zähler-Bausteinen aus dem 100 MHz `clk`-Signal erzeugt. Bei einem Low-Pegel von `enable` werden die Ausgangssignale `one_ms_clk`, `cs_cnv_out` und `spi_clk_out` auf einen Low-Pegel und `dac_clk` auf einen High-Pegel gesetzt.

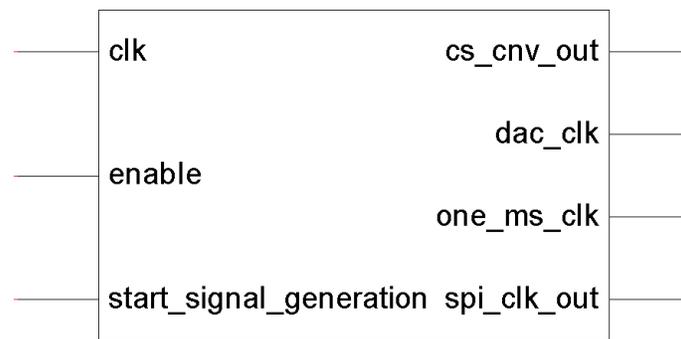


Abbildung 3.28: Funktionsblock der Clock-Generation

### 3.3.2.4 Signalerzeugung

Die Komponente Signal-Generation erzeugt das für den DAC benötigte digitale Messsignal. Innerhalb der Signal-Generation werden durch die vier Signalerzeugungskomponenten (Sinus, DC, Square und Sawtooth) das 16-Bit breite Ausgangssignal `dac_data_out` (siehe Abb. 3.29) erstellt. Hierzu werden mehrere Eingangssignale benötigt. Die Eingangssignale `dac_clk` und `one_ms_clk` werden von der Clock-Generation zur Verfügung gestellt. Das `clk`-Signal wird ebenfalls von der Clock-Generation zur Verfügung gestellt und entspricht dem `spi_clk_out`-Signal. Das Signal `harp3d_data(31:0)` entspricht dem ersten Eingangsregister der `user_logic`. Das Signal `harp3d_data_select` stellt die unteren 16-Bit des zweiten Eingangsregisters dar. Die Eingangssignale (`status_signals_in(2:0)`) entsprechen den Bits 16 (`start`), 17 (`enable`) und 18 (`newData`) des zweiten Eingangsregisters. Über `status_signal_out(1:0)` werden durch die Signal-Generation Statussignale ausgegeben. Diese sind `dac_adder_failure`, welches an den Ausgangsregisterwert 29 der `user_logic` geführt wird und `dac_data_rdy`, welches als Eingangssignal für das SPI-Interface dient.

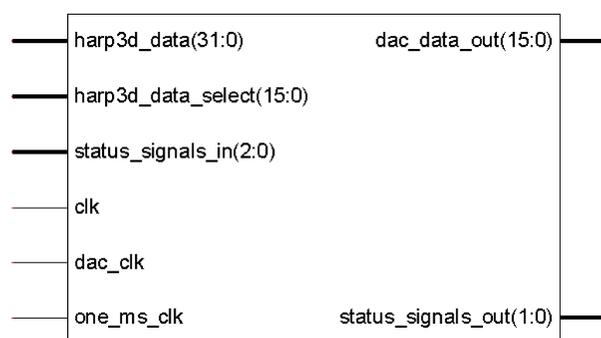


Abbildung 3.29: Ein- und Ausgangssignale der Signal-Generation.

Die Signal-Generation ist aus drei Teilbereichen, einer Datenauswahl, einer Messzeitsteuerung und der eigentlichen Signalerzeugung (siehe Abb. 3.30) zusammengesetzt.

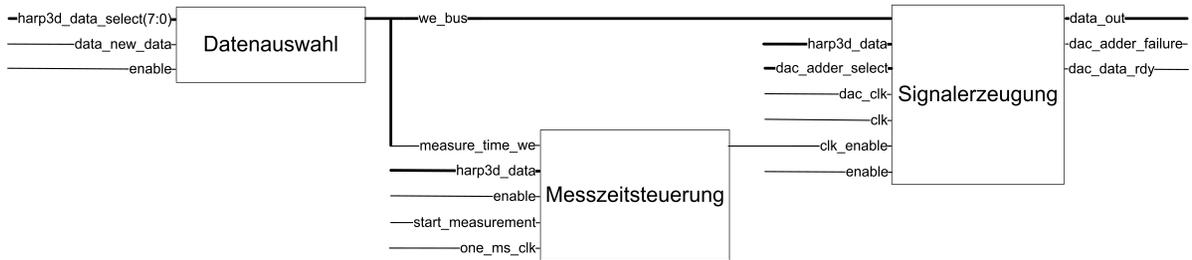


Abbildung 3.30: Schematischer Aufbau des Signal-Generators.

Über die Datenauswahl werden aus dem 8-Bit breiten Eingangssignal `harp3d_data_select` Enable-Signale für die einzustellenden Parameter erzeugt. Hierzu wertet die Datenauswahl über eine Switch-Anweisung das Signal `harp3d_data_select` bei der positiven Taktflanke von `data_new_data` aus. Bei der positiven Taktflanke der Enable-Signale auf den `we_bus` lesen die einzelnen Komponenten in der Signalerzeugung und der Messzeitsteuerung die Daten an `harp3d_data` ein. Hierdurch können mehrere Parameter über zwei Signale seriell übertragen werden. Über den `we_Bus` werden insgesamt acht Enable-Signale übertragen, zu ihnen gehören

- `measure_time_we` zum Einlesen der einstellbaren Messzeit in der Messzeitsteuerung.
- drei `freq_we`-Signale zum Einlesen der einzustellenden Ausgangsfrequenz der Sinus-, Square- und Sawtooth-Komponente in der Signalerzeugung.
- drei `mult_we`-Signale zum Einlesen der einstellbaren Ausgangsamplitude der Sinus-, Square- und Sawtooth-Komponente in der Signalerzeugung.
- `we_DC` zum Einlesen des Gleichspannungswertes in der Gleichspannungsquelle (DC).

Die Dauer der Messzeit kann in 1 ms Schritten über die Messzeitsteuerung eingestellt werden. Der Messzeitwert wird an dem Signal `harp3d_data` bei der positiven Flanke von `measure_time_we` eingelesen. Das Eingangssignal `start_measurement`, welches über die Software im Microblaze im zweiten Eingangsregister gesetzt wird, startet einen 32-Bit Zähler innerhalb der Messzeitsteuerung. Der 32-Bit Zähler ist dabei über das Eingangssignal `one_ms_clk` getaktet. Beim Starten des Zählers wird das Ausgangssignal `clk_enable` auf

einen High-Pegel gesetzt. Erreicht der Zählerwert den vorher gesetzten Messzeitwert wird das `clk_enable`-Signal wieder auf einen Low-Pegel gesetzt. In der Signalerzeugung wird für die Dauer des High-Pegels am `clk_enable`-Signal das Ausgangssignal `data_out` erzeugt.

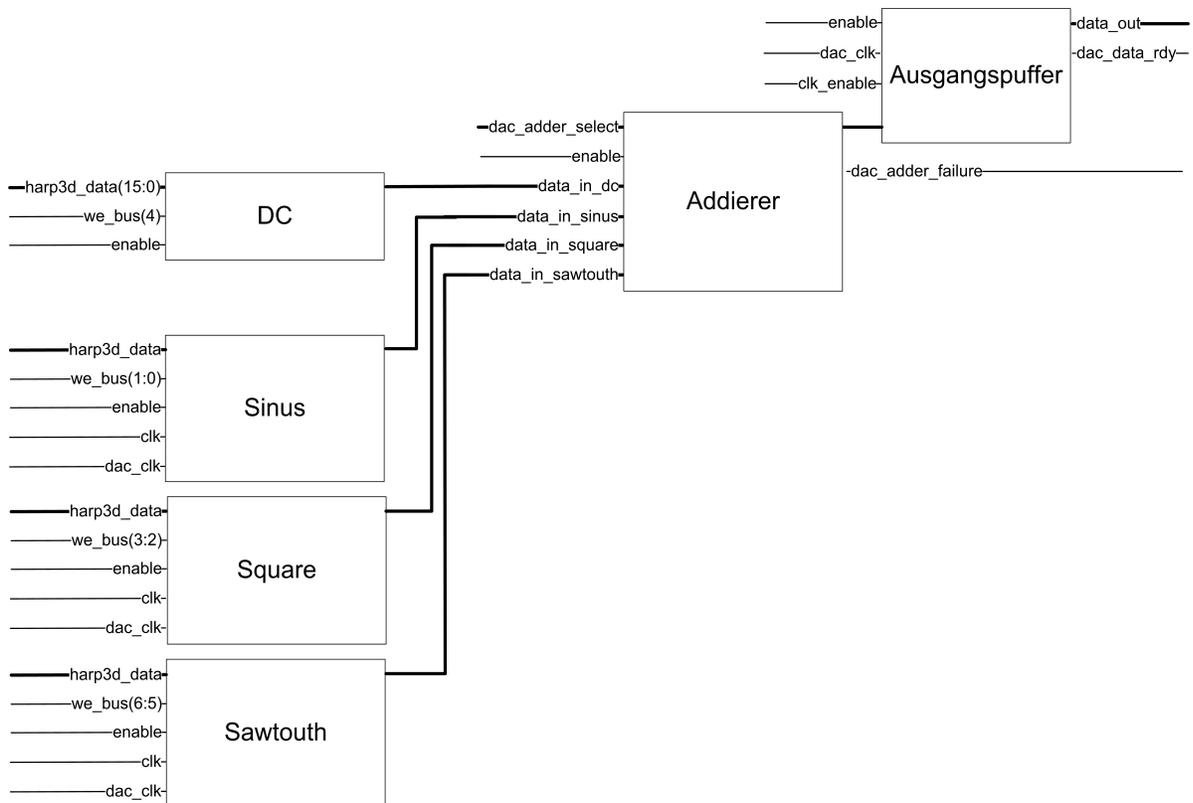


Abbildung 3.31: Schematischer Aufbau der Signalerzeugung in der Signal-Generation.

Die Komponente Signalerzeugung erzeugt das eigentliche Messsignal. Hierzu sind innerhalb dieses Teilbereichs (siehe Abb. 3.31) vier Signalerzeugungskomponenten, ein Addierer und ein Ausgangspuffer untergebracht. Die Parametrisierung für die Signalerzeugungskomponenten erfolgt über die Signale `we_bus` von der Datenauswahl und `harp3d_data`. Die erzeugten Messsignale der Signalerzeugungskomponenten Sinus, Square und Sawtooth werden bei der negativen Flanke des `dac_clk`-Signals an den jeweiligen Ausgängen ausgegeben. Die Signalerzeugungskomponente DC besteht lediglich aus einem Zwischenspeicher, der den Wert von `harp3d_data(15:0)` bei der positiven Flanke des Signals `we_bus(4)` abspeichert. Aus diesem Grund liegt hier das Ausgangssignal kontinuierlich an.

Die vier Ausgangssignale der Signalerzeugungskomponenten werden in einem 4-stufigen Addierer zusammengefasst. Eine Auswahl über die zu addierenden Signale der Signalerzeugung

gungskomponenten kann mit dem Signal `dac_adder_select` vorgenommen werden. Das Signal `dac_adder_select` entspricht dem Eingangssignal `harp3d_data_select(11:8)`, welches über die Software im Microblaze eingestellt werden kann. Das Ausgangssignal `dac_adder_failure` des Addierers dient als Interruptquelle.

Der Ausgangspuffer speichert das vom Addierer erzeugte Messsignal und gibt es bei der positiven Flanke des `dac_clk`-Signals und einem High-Pegel von `clk_enable` am Ausgang `data_out` aus. Das `data_out`-Signal wird durch das SPI-Interface bei der negativen Flanke des `dac_clk`-Signals eingelesen und an die DACs auf den Basismodulen übermittelt. Das `enable`-Signal setzt den Ausgangspuffer zurück. Hierbei werden das `data_out`-Signal und das `dac_data_rdy`-Signal auf einen Low-Pegel gesetzt. Das `dac_data_rdy`-Signal wird zusätzlich über ein Low-Pegel von `clk_enable` zurückgesetzt. Das `dac_data_rdy`-Signal zeigt an, ob gültige Datenwerte am `data_out`-Signal anliegen. Hierzu wird das `dac_data_rdy`-Signal nach einem Rücksetzen bei der ersten positiven Flanke von `dac_clk` auf einen High-Pegel gebracht. Durch die Signalerzeugung innerhalb der Clock-Generation ist hiermit sichergestellt, dass die Signalerzeugungskomponenten Sinus, Square und Sawtooth bereits bei der vorherigen negativen Flanke des `dac_clk`-Signals ihre Daten an den Addierer übermittelt haben. Innerhalb der Komponente Signal-Generation wird die negative Flanke des `dac_data_rdy`-Signal darüber hinaus zur Generierung des Interruptsignals `ADCready` genutzt.

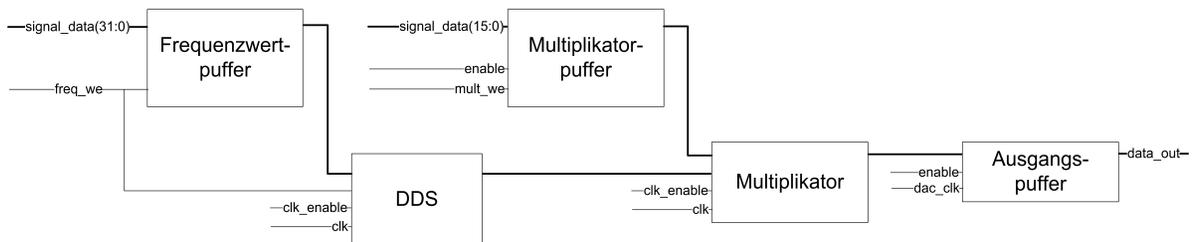


Abbildung 3.32: Schematischer Aufbau der Sinussignalerzeugung.

Die Sinussignalerzeugung (siehe Abb. 3.32) besteht ebenfalls aus mehreren Bausteinen. Über einen Direct-Digital-Synthesizer (DDS), welcher mit dem im ISE enthaltenen CORE Generator™ erstellt ist, wird eine 16-Bit breite Sinusspannung mit einer Taktfrequenz des `clk`-Signals erzeugt. Die Ausgangsfrequenz des DDS wird über das `freq_we`-Signal vom `we_bus` der Datenauswertung bei einem High-Pegel konfiguriert. Hierzu werden durch einen Frequenzwertpuffer die am `harp3d_data`-Signal anliegenden Frequenzdaten bei der positiven Flanke des `freq_we`-Signals zwischengespeichert. Dies ist notwendig, da der DDS nicht Taktflankengesteuert, sondern Pegel abhängig Datenwerte einliest. Es kann durch den Aufbau der Daten-

auswertung nicht sicher gestellt werden, dass während eines High-Pegels am freq\_we-Signal immer gültige Daten am harp3d\_data-Signal anliegen. Dies ist nur für die positive Taktflanke des freq\_we-Signals gegeben. Die Ausgangsfrequenz des DDS ist gegeben durch [45]

$$f_{out} = \frac{f_{clk} \Delta\Theta}{2^{B_{\Theta(n)}}}. \quad (3.7)$$

$B_{\Theta(n)}$  stellt die interne Wortbreite zur Berechnung des Sinussignals des DDS dar. Sie beträgt 32-Bit und wurde so gewählt, dass am Ausgang des DDS ein SNR von 96 dB erreicht wird.  $\Delta\Theta$  stellt den mittels dem harp3d\_data-Signal übertragenen 32-Bit breiten Frequenzwert dar. Das Ausgangssignal des DDS ist ein vorzeichenbehafteter 16-Bit Wert.  $f_{clk}$  stellt die 25 MHz Frequenz des spi\_clk\_out-Signals dar.

Der getaktete Multiplikator, welcher ebenfalls mit dem im ISE enthaltenen CORE Generator™ erstellt worden ist [46], ist so dimensioniert, dass es einen vorzeichenbehafteten Faktor und einen vorzeichenlosen Faktor multipliziert. Hiermit wird das vorzeichenbehaftete DDS Ausgangssignal mit einem im Multiplikatorpuffer gespeicherten vorzeichenlosen Amplitudenwert multipliziert. Von dem 32-Bit breiten Ausgangsprodukt werden die oberen 16 Bit an den Ausgangspuffer gegeben.

Über den Multiplikatorpuffer wird der einstellbare Verstärkungsfaktor ( $v_{out}$ ) des Ausgangs zwischengespeichert. Hierzu wird bei der positiven Flanke des mult\_we-Signals der einzustellende Amplitudenwert ( $A_{in}$ ) am harp3d\_data-Signal eingelesen. Der Verstärkungsfaktor ergibt sich zu

$$v_{out} = \frac{A_{in}}{2^{16}}. \quad (3.8)$$

Da  $A_{in}$  niemals größer als  $2^{16}$  werden kann, ergibt sich immer ein Verstärkungsfaktor kleiner eins. Dies ist gewollt, da über den Addierer die einzelnen Signale noch zusammengefasst werden.

Der Ausgangspuffer speichert den aktuellen Wert des Multiplikators bei der negativen Flanke des dac\_clk-Signals ab und stellt es am Ausgang für den Addierer zur Verfügung.

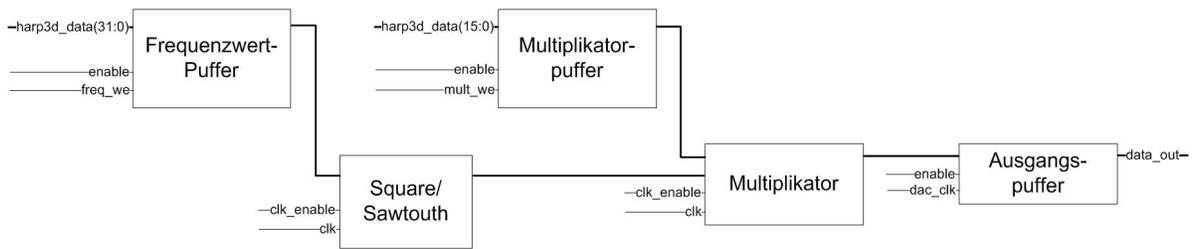


Abbildung 3.33: Schematischer Aufbau der Rechteck- und Dreiecksignalerzeugung.

Die beiden vorzeichenbehafteten Signalerzeugungskomponenten der Dreiecks- und Rechteckspannung (siehe Abb. 3.33) sind ähnlich aufgebaut wie die der Sinussignalerzeugung. Die Ausgangsfrequenz ergibt sich über den im Frequenzwertpuffer abgespeicherten 32-Bit Wert

$$f_{out} = \frac{f_{clk} D_{Frequenzwertpuffer}}{2^{32}}. \quad (3.9)$$

Die Verstärkung des erstellten Signals ergibt sich zu

$$V_{out} = \frac{D_{Multiplikatorpuffer}}{2^{16}}. \quad (3.10)$$

Die Dreiecksignalerzeugung erfolgt über einen Zähler, welcher von 0 über  $+2^{15}$  auf  $-2^{15}$  hochzählt. Die Schrittweite des Inkrementierens ist identisch mit dem im Frequenzwertpuffer abgespeicherten Wert.

Die Rechtecksignalerzeugung erfolgt ebenfalls über einen Zähler, welcher mit dem im Frequenzwertpuffer abgespeicherten Wert inkrementiert. Der Zähler zählt hierzu von 0 auf  $+2^{32}$  hoch. Die Rechteckspannung wird durch Auswertung von Bit 32 des Zählers bestimmt. Beträgt Bit 32 eins wird ein Ausgangswert von  $-2^{15}$  erzeugt. Ist Bit 32 null wird ein Ausgangswert von  $+2^{15}$  generiert.

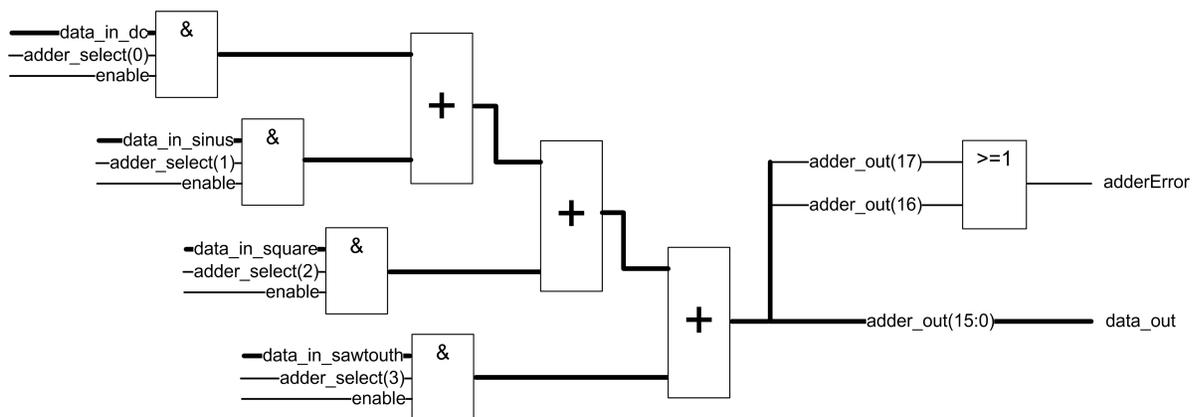


Abbildung 3.34: Schematischer Aufbau der Addiererstufe in der Signal-Generation.

Der Addierer addiert die vier erzeugten Signale der Signalerzeugungskomponenten (siehe Abb. 3.34). Welche Eingangssignale zu dem Ausgangssignal `data_out` dazu addiert werden sollen, ist dabei über die Software im Microblaze steuerbar. Hierzu können die einzelnen Eingangssignale `data_in_dc`, `data_in_sinus`, `data_in_square` und `data_in_sawtouth` über die vom Eingangsregister der `user_logic` anliegenden `adder_select`-Werte ausgewählt werden. Die Addierer sind 18-Bit Addierer. Hierdurch ist es möglich Überläufe bei dem Addieren zu detektieren (siehe `adderError`). Das so erzeugte Signal dient als Interruptquelle in der `user_logic` und wird ebenfalls im Ausgangsregister der `user_logic` hinterlegt. Zur Ansteuerung der DACs dürfen nur positive vorzeichenlose Werte übertragen werden. Aus diesem Grund muss über das `data_in_dc`-Signal dafür gesorgt werden, dass immer ein ausreichender Offset von den um den Wert Null erzeugten Signalen `data_in_sinus`, `data_in_square` und `data_in_sawtouth` vorhanden ist. Da die Addierer 18-Bit breit sind, werden von den Eingangssignalen `data_in_sinus`, `data_in_square` und `data_in_sawtouth` jeweils das Vorzeichenbit (Bit Nummer 16) auf die Bits 17 und 18 gespiegelt. Bei dem `data_in_dc`-Signal betragen Bit 17 und 18 immer null. Hierdurch wird erreicht, dass der DC-Wert als vorzeichenloser Wert interpretiert wird und der DC-Offset über den gesamten 16-Bit breiten Bereich des Ausgangssignals gewährleistet ist.

## 3.4 Programmablauf im FPGA und im $\mu$ Controller

Nachdem in den drei vorangegangenen Unterkapiteln der Gesamtaufbau, der Hardwareaufbau der erstellten Module und der Aufbau innerhalb des FPGAs betrachtet worden ist, beschäftigt sich dieses Unterkapitel mit dem Programmablauf der erstellten Software im Microblaze und im  $\mu$ Controller auf den Basismodulen. Der  $\mu$ Controller und der Microblaze können über den seriellen Bus miteinander kommunizieren. Hierzu ist ein Kommunikationsprotokoll entwickelt worden, welches die Kommunikation von mehreren am seriellen Bus angeschlossenen Basismodulen gewährleistet. Mit diesem Protokoll beschäftigt sich das folgende Unterkapitel. Die beiden weiteren Unterkapitel beschäftigen sich mit dem eigentlichen Programmablauf im  $\mu$ Controller und im Microblaze.

### 3.4.1 Interkommunikation zwischen Haupt- und Basismodul

Zur Steuerung der  $\mu$ Controller auf den Basismodulen durch den Microblaze auf dem Hauptmodul ist ein Kommunikationsprotokoll entwickelt worden. Das Kommunikationsprotokoll sieht vor, dass zwischen dem Microblaze und den  $\mu$ Controllern Daten in einzelnen Segmenten ausgetauscht werden. Die Anforderungen an das Protokoll sind, dass

- mehrere  $\mu$ Controller auf den Basismodulen an den seriellen Bus angeschlossen werden können.
- die  $\mu$ Controller individuell angesprochen werden können.
- die Länge der übertragenen Pakete variabel gehalten werden kann, um unterschiedliche Daten übertragen zu können. Gleichzeitig sollen die individuellen Messsensoren an den Basismodulen einfach angesprochen werden können.
- die übertragenen Pakete auf ihre Gültigkeit überprüft werden können. Dies soll verhindern, dass durch ein Fehler in der Paketübermittlung zwei oder mehrere  $\mu$ Controller gleichzeitig Daten auf dem Bus senden.

Damit mehrere Module gleichzeitig an den seriellen Bus angeschlossen werden können und die  $\mu$ Controller individuell ansprechbar sind, besitzen die übertragenen Segmente einen Header mit nachfolgenden Datenpaketen variabler Breite (siehe Abb. 3.35).

Innerhalb des Headers sind die einzelnen  $\mu$ Controller über eine individuelle Empfänger ID ansprechbar. Der Microblaze besitzt die Sender ID 0x0000.

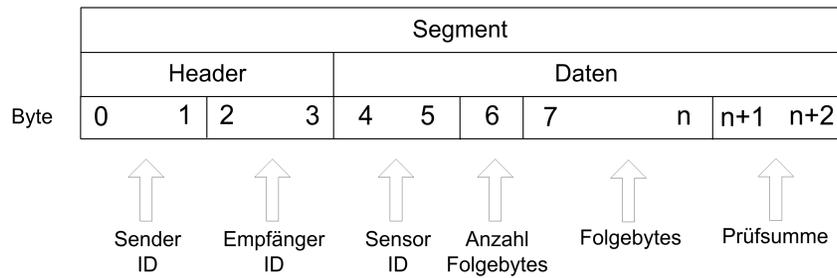


Abbildung 3.35: Aufbau des Übertragungsprotokolls zwischen Haupt- und Basismodul.

Der nachfolgende Datenbereich besteht aus einer 16 Bit breiten Sensor ID. Hierüber werden die einzelnen Messsensoren auf den Messmodulen, sowie weitere zu übergebende Parameter ausgewählt. Das sechste Byte dient dazu Daten mit einer variablen Segmentgröße zu übermitteln. Hierzu ist im sechsten Byte hinterlegt wie viele Folgebytes der Datenbereich besitzt. Im nachfolgenden variablen Bereich (Folgebytes) können 1 bis n Bytes übertragen werden. Die gesamte Segmentgröße ist auf 128 Byte beschränkt, womit n maximal 125 sein kann. Insgesamt können somit bis zu 118 Bytes im variablen Datenbereich übertragen werden. In den Bytes n+1 und n+2 wird eine Prüfsumme übertragen, welche die Gültigkeit des übertragenen Segmentes prüft. Die 16 Bit breite Prüfsumme (CRC) wird mittels einer Addition aller Bytes des Segmentes erzeugt

$$CRC = \sum_{i=0}^n \text{Byte}_i. \tag{3.11}$$

Die 16 Bit breite Sensor ID ist zur Auswahl der einzelnen Sensoren in vier 8 Bit breite Bereiche unterteilt. Über die letzten vier Bits (Bit 12 bis 15) wird eine allgemeine Sensorauswahl

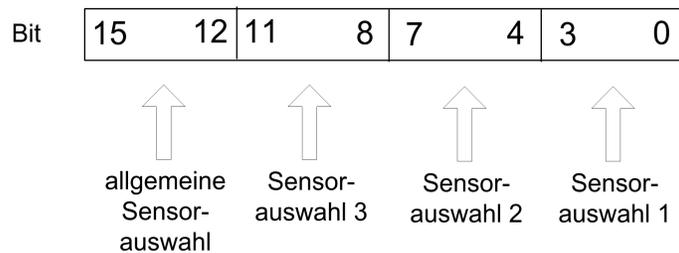


Abbildung 3.36: Aufbau der Sensor ID.

getroffen. Die Bereiche Sensorauswahl 1 bis Sensorauswahl 3 enthalten genauere Informationen über den im Bereich „allgemeine Sensorauswahl“ getroffenen Sensor.

Zu der allgemeine Sensorauswahl gehören bei einem Wert von

- 0: die Auswahl der einzelnen Messsensoren auf den Basismodulen. Hierzu werden durch den  $\mu$ Controller über die Bits der „Sensorauswahl 1“ bis „Sensorauswahl 3“ die Ausgänge CNT\_EN\_MM\_WTR, CNT\_EN\_MMn und CNT\_MMn des  $\mu$ Controllers entsprechend der übermittelten Werte gesetzt.
- 1: die Auswahl der variablen Verstärkung der ADCs und DACs. Der  $\mu$ Controller setzt hierzu die beiden Ausgänge CNT\_v\_0 und CNT\_v\_1 in Abhängigkeit des Wertes in dem Bereich „Sensorauswahl 1“.
- 15: mehrere weitere Funktionen, welche durch den  $\mu$ Controller ausgegeben werden können. Die einzelnen Funktionen werden über die „Sensorauswahl 1“ getroffen. Zu ihnen gehören unter anderem das Schalten der Spannungsversorgung, die Temperatur der Temperaturmessung am  $\mu$ Controller abzufragen, die Firmware des  $\mu$ Controllers abzufragen oder die individuelle Empfänger ID des  $\mu$ Controllers zu ändern.

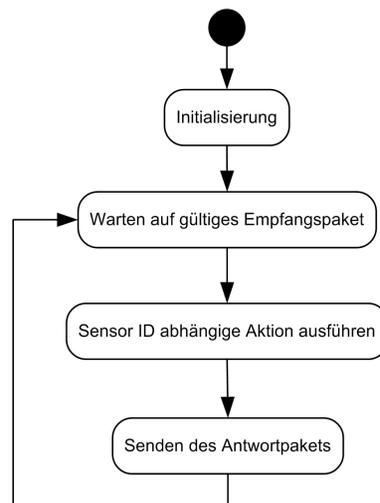
Zur Bestätigung der vom Microblaze gesendeten Anfrage wird vom  $\mu$ Controller bei erfolgreicher Ausführung der Anfrage das gesendete Datenpaket an den Microblaze zurück gesendet. Hierbei wird gegebenenfalls der Folgebytebereich genutzt um angeforderte Informationen zu senden.

Die Kommunikation zwischen Microblaze und den  $\mu$ Controllern stellt ein Client-Server-Modell dar. Hierbei agiert der Microblaze als Client, welcher von den einzelnen Servern ( $\mu$ Controllern) Daten abfragt. Der  $\mu$ Controller initialisiert hierbei niemals eine Verbindung.

### 3.4.2 Programmablauf im $\mu$ Controller des Basismoduls

Die Software des  $\mu$ Controllers wurde mit Hilfe der freien Software AVR Studio<sup>®</sup> 4 von der Firma Atmel erstellt. Das AVR Studio<sup>®</sup> benötigt weiter die Open-Source Software WinAVR<sup>™</sup>. Die eigentliche Programmierung des ATmega8  $\mu$ Controllers erfolgt über deren SPI-Schnittstelle mit Hilfe eines AVRISP mkII Programmiergerätes.

Der Programmablauf (siehe Abb. 3.37) innerhalb des  $\mu$ Controllers ist so aufgebaut, dass der  $\mu$ Controller nach dem Programmstart eine Initialisierung vornimmt. Im Anschluss daran wartet der  $\mu$ Controller auf ein gültiges Paket auf der seriellen Schnittstelle. Wird ein gültiges Paket empfangen wird dieses ausgewertet und eine von der Sensor ID abhängige Aktion

Abbildung 3.37: Programmablauf im  $\mu$ Controller.

ausgeführt. Nach erfolgreicher Ausführung der Aktion wird ein Antwortpaket auf der seriellen Schnittstelle zurück gesendet.

Der Programmcode ist dabei so aufgebaut, dass der Programmablauf über eine Ereignisstruktur gesteuert wird. Die Ereignisstruktur ist als Ringspeicher aufgebaut. Innerhalb der Hauptroutine wird auf eine Änderung des aktuellen Ereignisses gewartet und dieses im Anschluss über eine Switch-Anweisung abgearbeitet. Innerhalb der Ereignisstruktur können bis zu acht Ereignisse gespeichert werden, welche sequentiell abgearbeitet werden. Somit ist es möglich, mehrere über Interrupt initialisierte Aktionen nacheinander auszuführen ohne in der ISR zu verbleiben. Innerhalb der ISR wird hierzu ein neues Ereignis in das Ereignisarray eingetragen. Nach Verlassen der ISR wird das Ereignis direkt abgearbeitet oder es wird so lange gewartet bis andere Ereignisse abgearbeitet wurden. Hierdurch wird langer Programmcode in der ISR vermieden. Außerdem ist es hierdurch möglich den Programmcode mit den zugehörigen Funktionen sehr flexibel zu halten.

Um Timeouts zu vermeiden ist eine Software gesteuerte Timerstruktur im Programmcode implementiert. Innerhalb der Struktur können bis zu acht Softwaretimer mit einem Timeoutwert gesetzt werden. Über einen auf dem  $\mu$ Controller vorhandenen Hardwaretimer werden alle 10 ms ein Interrupt ausgelöst. Innerhalb der ISR werden die einzelnen Softwaretimer, soweit sie gesetzt sind, jeweils um eins dekrementiert. Wird ein Softwaretimer durch die ISR auf null gesetzt, wird innerhalb der ISR ein Timeoutereignis des entsprechenden Softwaretimers in das Ereignisarray angehängt.

Die Initialisierungsphase dient zur Grundkonfiguration des  $\mu$ Controllers. Innerhalb dieser werden die Anschlüsse des  $\mu$ Controllers als Ein- bzw. Ausgänge deklariert und die serielle Schnittstelle, das Ereignisarray und Timeoutarray initialisiert. Außerdem werden global Interrupts erlaubt.

Nach der Initialisierung wird auf ein gültiges Empfangspaket gewartet. Hierzu werden über eine ISR alle am seriellen Eingang des  $\mu$ Controllers empfangenen Daten Byte für Byte in einen Empfangszwischenspeicher abgespeichert. Nach Empfangen eines kompletten Segmentes, was innerhalb der ISR überprüft wird, fügt die ISR das Ereignis in das Ereignisarray ein. Werden für mehr als 20 ms keine Bytes empfangen wird der Zwischenspeicher verworfen. Ist ein neues Empfangspaket vorhanden, wird zuerst geprüft ob die Prüfsumme stimmt. Ist die Prüfsumme korrekt wird überprüft ob die Empfänger ID mit der individuellen ID übereinstimmt. Ist dies auch der Fall liegt ein gültiges Empfangspaket vor.

Nach dem Empfangen eines gültigen Empfangspaketes wird über eine Switch-Anweisung die Sensor ID ausgewertet. Je nach Inhalt der allgemeinen Sensorauswahl wird eine individuelle Funktion ausgeführt. So werden z.B. die einzelnen Anschlüsse zur Auswahl der Messsensoren gesetzt. Im Anschluss wird in das Ereignisarray das Ereignis eingefügt, dass ein Antwortpaket gesendet werden soll.

Nach dem Abarbeiten der Anfrage des Microblaze wird ein Antwortpaket an den Microblaze gesendet. Hierzu wird aus der individuellen ID, der Empfängeradresse des Microblaze, der angefragten Sensor ID und der eventuellen Antwort auf die Anfrage (Folgebytes) die Prüfsumme bestimmt und alle Komponenten in ein Segment zusammengeführt. Anschließend wird das Segment Byte für Byte über eine ISR auf der seriellen Schnittstelle an den Microblaze gesendet.

### 3.4.3 Programmablauf im Microblaze des Hauptmoduls

Der Programmablauf (siehe Abb. 3.38) innerhalb des Microblaze ist hierarchisch aufgebaut und wurde mit Hilfe des SDKs, welches im EDK enthalten ist, erstellt.

Nach der Initialisierung werden die zur Messung nötigen Messwerte ermittelt. Im Anschluss wird der  $\mu$ Controller des Basismoduls über die serielle Schnittstelle initialisiert. Ist die Initialisierung erfolgreich, werden die Parameter des Custom-IP-Cores über dessen Eingangsregister gesetzt. Anschließend wird die eigentliche Messung durchgeführt, falls die Initialisierung des Custom-IP-Cores erfolgreich war. Während der Messung werden die vom Custom-IP-Core empfangenen Messsignale über einen DMA-Controller direkt in den DDR-RAM übertragen. Nach der Messung werden die Messdaten im DDR-RAM ausgewertet und der Messwertzähler, welcher die zu ladenden Messwerte bestimmt, inkrementiert.

Eine typische Messung sieht so aus, dass im ersten Durchlauf das Basismodul eingeschal-

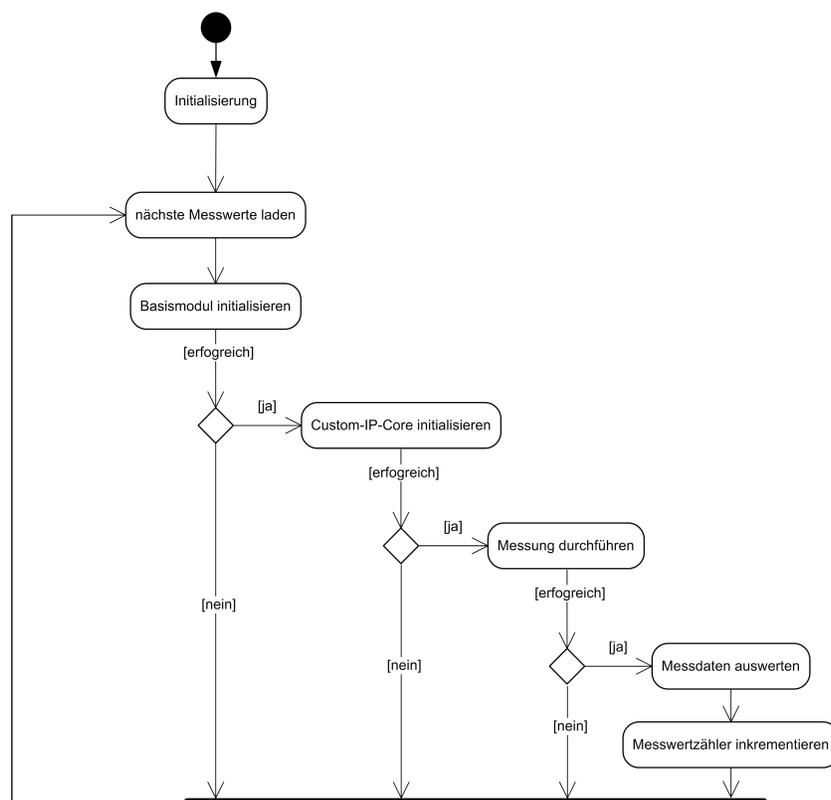


Abbildung 3.38: Programmablauf im Microblaze.

tet wird. In diesem Fall wird der Custom-IP-Core nicht initialisiert und auch keine Messung durchgeführt. Im zweiten Durchlauf wird über das Basismodul der zu messende Messsensor ausgewählt und die Parameter, wie Frequenz und Amplitude, des Custom-IP-Cores gesetzt.

Im Anschluss wird die Messung am jeweiligen Messsensor durchgeführt. Im dritten Durchlauf wird das Basismodul wieder ausgeschaltet.

Um Timeouts während des Messvorgangs zu verhindern, ist im FPGA ein Hardwaretimer implementiert. Über eine ISR dekrementiert dieser alle 100 ms das Timeoutsignal für eine laufende Messung. Hiermit wird sichergestellt, dass die Messung nicht länger als durch die vorgegebene Messzeit dauert. Über den Timer wird zusätzlich überprüft, ob bei der Initialisierung des Basismoduls das Antwortpaket innerhalb von 200 ms zurück gesendet wird. Dies ist notwendig, da der  $\mu$ Controller bei einem fehlerhaften Empfangspaket kein Antwortpaket zurück sendet.

In der Initialisierungsphase wird die serielle Schnittstelle für die Anbindung der Basismodule, der Timer mit einer zyklischen Interruptgenerierung von 100 ms, der DMA-Controller und ein Interrupt-Controller initialisiert. Der Interrupt-Controller wird für den Timer benötigt, welcher als einziger IP-Core mittels Interrupt mit dem Microblaze kommuniziert. Außerdem wird eine `harp3d_measure` Struktur mit Standardwerten initialisiert.

Zum Laden der Messwerte werden alle benötigten Messwerte in der `harp3d_measure` Struktur abgespeichert. Hierzu werden über eine Switch-Anweisung die Messwerte anhand des jeweiligen Messwertzählers ermittelt. Zu den Messwerten gehören alle Parameter, welche die Einstellung des Basismoduls, des Custom-IP-Cores und den weiteren Programmablauf betreffen.

Für die Basismodule werden das auszuwählende Basismodul mittels der individuellen ID und der zu messende Sensor über die Sensor ID abgespeichert.

Für den Custom-IP-Core werden die Messzeit und die Parameter wie Frequenz und Amplitude für die einzelnen Signalerzeugungskomponenten in `harp3d_measure` eingetragen. Hierbei kann auch pauschal eine Signalerzeugungskomponenten über ein Enable-Signal ausgeschaltet werden. Außerdem kann eingestellt werden, ob der Custom-IP-Core im Loopback-Betrieb betrieben werden soll.

Für den weiteren Programmablauf kann eingestellt werden ob eine Messung durchgeführt werden soll. Dies ist zum Beispiel nicht der Fall, wenn nur das Basismodul eingeschaltet wird.

Die Initialisierung des Basismoduls erfolgt über die im FPGA implementierte serielle Schnittstelle. Hierzu werden aus den in der `harp3d_measure` Struktur gespeicherten Werten

ein Sendepaket erstellt und an der seriellen Schnittstelle ausgegeben. Nach dem Senden werden für 200 ms die Daten am Eingang der seriellen Schnittstelle zyklisch abgefragt. Die empfangenen Daten werden im Anschluss auf ihre Gültigkeit überprüft. Hierzu gehört, ob die Prüfsumme stimmt und ob die übertragene Sensor ID, Empfängeradresse und Sendeadresse übereinstimmen.

Zur Initialisierung des Custom-IP-Cores wird der Custom-IP-Core zuerst über den in der user\_logic implementierten Softwarereset-Funktion zurückgesetzt. Anschließend wird die Messzeit übermittelt. Hierzu wird zuerst die 32-Bit breite Messzeit in das erste Eingangsregister des Custom-IP-Cores geschrieben. Als zweiter Schritt wird in das zweite Eingangsregister der Registerwert für das Setzen der Messzeit (measure\_time\_we) geschrieben. Im Anschluss werden die einzelnen Parameter zum Setzen der Frequenz und der Amplitude der ausgewählten Signalerzeugungskomponenten an den Custom-IP-Core übermittelt. Hierzu werden ebenfalls zuerst die Werte für das erste Eingangsregister und dann die Werte für das zweite Eingangsregister übermittelt.

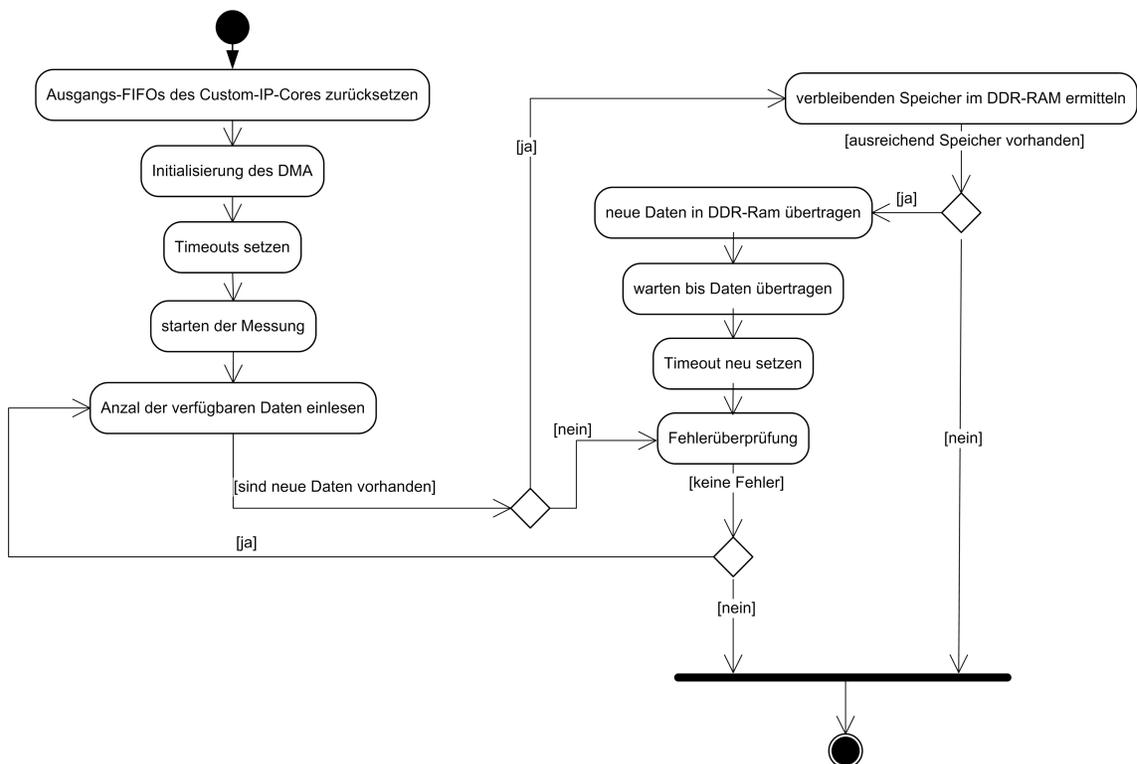


Abbildung 3.39: Ablauf der Zwischenspeicherung der Messwerte.

Zur Messung werden zuerst die Ausgangs-FIFOs des Custom-IP-Cores zurückgesetzt (siehe Abb. 3.39). Anschließend wird an den DMA-Controller die Adresse des Ausgangs-FIFOs und die Adresse der Speichervariablen im DDR-RAM übergeben. Außerdem wird der DMA-Controller so initialisiert, dass die Adresse des Ausgangs-FIFOs statisch ist und die Adresse der Variablen im DDR-RAM bei der Datenübertragung dekrementiert werden soll. Nach der Initialisierung und dem Setzen der Timeouts wird die Messung gestartet. Hierzu wird in das zweite Eingangsregister des Custom-IP-Cores der Registerwert zum Starten geschrieben.

Über das Ausgangsregister der Ausgangs-FIFOs wird die Anzahl der zur Verfügung stehenden Daten ermittelt. Wenn neue Daten vorhanden sind, wird der verbleibende Speicher für die Speichervariable im DDR-RAM ermittelt. Hierzu wird die Anzahl der bereits übertragenen Daten mit der Anzahl der neuen addiert und mit der Größe des Speicherbereichs der Speichervariablen verglichen. Andernfalls würde der DMA-Controller in einen Speicherbereich des DDR-RAMs der anderen Variablen oder dem Programmcode zugeordnet ist schreiben. Wenn ausreichend Speicher vorhanden ist, werden die neuen Daten von den Ausgangs-FIFOs des Custom-IP-Cores über den DMA-Controller in den DDR-RAM übertragen. Nach erfolgreicher Übertragung wird der Timeout neu gesetzt und der DMA-Controller über ein Register auf Fehler bei der Übertragung abgefragt. Anschließend wird das Ausgangsregister des Custom-IP-Cores ausgelesen und auf aufgetretene Interrupts überprüft. Sind keine Fehler vom DMA-Controller und keine Interrupts vom Custom-IP-Core aufgetreten, wird die Messung mit der Anzahl der verfügbaren Daten in den Ausgangs-FIFOs fortgesetzt. Sind Fehler aufgetreten oder nicht genügend Arbeitsspeicher vorhanden, wird die Messung beendet. Ein Timeout, welcher durch die ISR des Timers ermittelt wird, kann ebenfalls die Messung beenden.

Nach erfolgreicher Messung erfolgt eine Datenauswertung. Diese ist noch nicht im Microblaze implementiert. Grund hierfür ist die schlechte Qualität der von den ADCs auf den Basismodulen übermittelten Signale. Dieser Umstand, welcher sich bei der Inbetriebnahme der Harp3d gezeigt hat, wird unter anderem im nächsten Kapitel dargestellt.

# Inbetriebnahme der Harp3d und erste Messergebnisse

---

Nachdem in dem vorangegangenen Kapiteln der Aufbau der Harp3d ausführlich besprochen wurde, beschäftigt sich dieses Kapitel mit der Inbetriebnahme und ersten Messergebnissen der Harp3d. Bei der Inbetriebnahme und einer anschließenden Verifizierung hat sich herausgestellt, dass das Layout des Basismoduls nochmals überarbeitet werden muss. Eine Inbetriebnahme des Gesamtsystems war möglich, allerdings werden durch Fehler im Layout des Basismoduls die Messsignale durch diverse Faktoren so gestört, dass eine Messung über die ADCs nicht möglich ist. Aus diesem Grund beschäftigt sich dieses Kapitel auch ausführlich mit der Fehleranalyse und den notwendigen Verbesserungen am Layout. Der Digitalteil konnte hiervon unabhängig in Betrieb genommen werden. Über die Loopback-Funktion innerhalb des Custom-IP-Cores ist es möglich, das digitale Messsignale näher zu betrachten. Der  $\mu$ Controller konnte ebenfalls unabhängig vom analogen Teil erfolgreich in Betrieb genommen werden. Neben der Fehleranalyse des digitalen Messsignals wird außerdem der Stromverbrauch der realisierten Komponenten in diesem Kapitel behandelt.

Zur Funktionsprüfung des Hauptmoduls wurde mit Hilfe der internen Loopback-Funktion des Custom-IP-Cores das erzeugte digitale Messsignal direkt auf den Eingang des SPI-Interfaces zurück geleitet. Die im DDR-RAM abgespeicherten Daten wurden anschließend auf

der seriellen Schnittstelle des Development Boards in Klartext ausgegeben. Die so gewonnenen Daten wurden dann mit Matlab<sup>®</sup> ausgewertet.

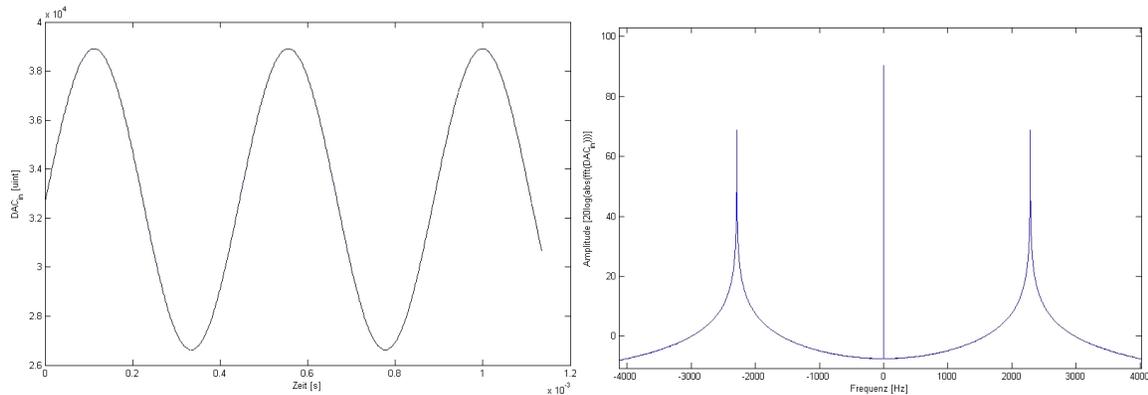


Abbildung 4.1: Signal (links) und Spektrum (rechts) der Sinussignalerzeugung des Custom-IP-Cores.

Zur Verifizierung der Funktionsfähigkeit sind über die Sinussignalerzeugung ein Sinussignal mit DC-Offset erzeugt worden (siehe Abb. 4.1). Die verwendeten Parameter für den Custom-IP-Core sind  $\Delta\Theta = 0x00006000$  ( $\Rightarrow f = 2,888\text{ kHz}$ ) und  $A_{in} = 0x2FFF$  ( $\Rightarrow v_{out} = 0.187$  bzw.  $-14,56\text{ dB}$ ). Diese Werte entsprechen vergleichbaren Werten des späteren Einsatzes von etwa 2 kHz und einer Sinusspannung von  $\approx 130\text{ mV}_{eff}$ . Die Offset-Spannung ist so gewählt, dass der Gleichspannungsanteil der Hälfte des maximalen Ausschlags entspricht ( $DC = 0x8000 \Rightarrow v_{out} = 0.5$ ). Dies entspricht auf dem Basismodul einer Spannung von 0 V nach dem Interpolationstiefpass mit Addiererschaltung. Aus dem Spektrum ergibt sich ein SNR von etwa 70 dB in Bezug auf den Gleichspannungsanteil. Dies ist für die spätere Anwendung mehr als ausreichend. Bei dem in Matlab<sup>®</sup> erstellten Spektrum muss der SNR des Sinussignal in Bezug zum Gleichspannungsanteil ermittelt werden. Praktisch kann bei einer 16-bitigen Auflösung der SNR nie größer als 96 dB sein. Matlab<sup>®</sup> berechnet das Spektrum allerdings mit float-Werten.

Die Signalerzeugungskomponenten der Rechteck- und Dreiecksspannungserzeugung konnten ebenfalls erfolgreich getestet werden (siehe Abb. 4.2). Die verwendeten Parameter sind identisch denen der Sinussignalerzeugung.

Bei der Inbetriebnahme des Basismoduls konnte der  $\mu$ Controller mit angeschlossener Peripherie erfolgreich getestet werden. Die Messmodule konnten in Verbindung mit dem Basismodul ebenfalls erfolgreich in Betrieb genommen werden. Darüber hinaus konnten Teile der Messmodule unter Temperaturbedingungen bis  $-25^\circ\text{C}$  für erste Ergebnisse in Laborexperi-

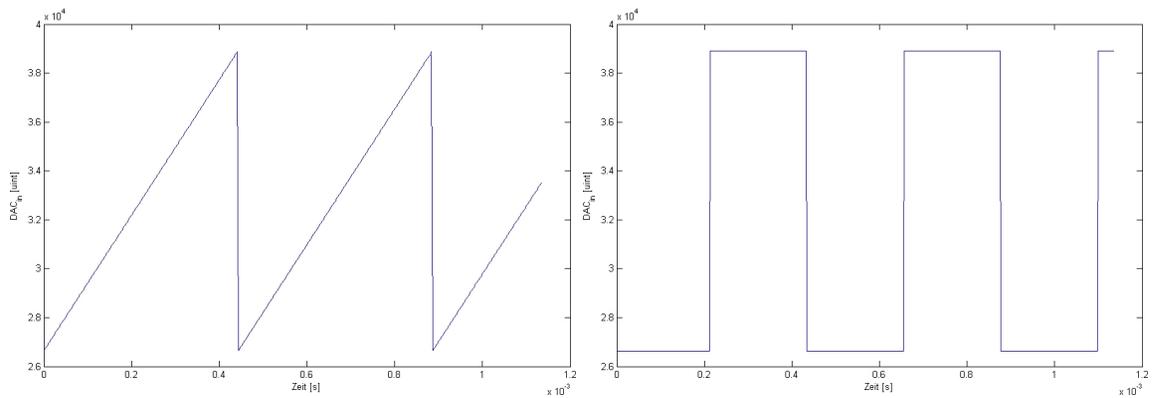


Abbildung 4.2: Dreieckssignal (links) und Rechtecksignal (rechts) der Signalgenerierung des Custom-IP-Cores.

menten genutzt werden [16]. Hierbei wurden Teile des bestehenden Systems mit den neu entworfenen Messmodulen ersetzt, sowie eine separate Phasenmessschaltung eingesetzt.

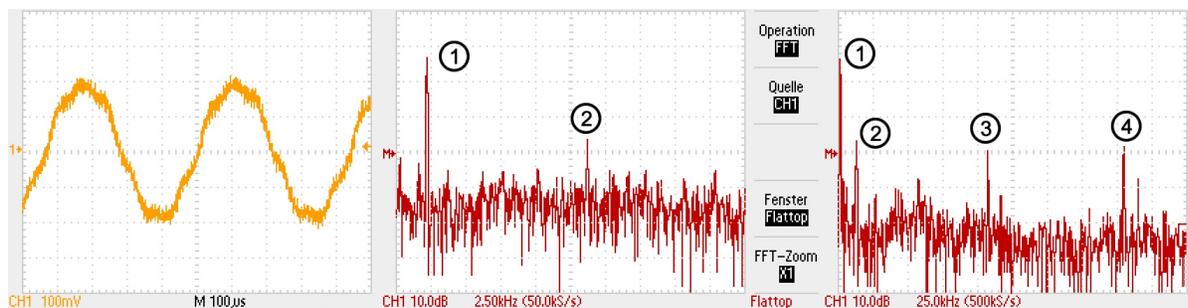


Abbildung 4.3: Sinusausgangssignal an der Spannungsmessung des Messmoduls; links: Sinussignal, mitte: Spektrum mit 2,9 kHz Nutzsignal (1) und 13 kHz Störsignal (2), recht: Spektrum mit 2,9 kHz Nutzsignal (1), 13 kHz Störsignal (2), 105 kHz Störsignal (3) und 205 kHz Störsignal (4).

Zum Testen des Gesamtsystems wurde das System mit Hauptmodul, Basismodul und Messmodul komplett in Betrieb genommen. Als Messsensor diente ein  $20 \Omega$  Widerstand, welcher eine realistische Widerstandsgröße für den späteren Einsatz darstellt. Zum Testen des Basismoduls wurde über den Custom-IP-Core das selbe Sinussignal wie zur Verifizierung des Custom-IP-Cores verwendet. Das über den DAC erzeugte Messsignal wurde hinter dem Messvorverstärker der Spannungsmessung auf dem Messmodul abgegriffen (siehe Abb. 4.3). Die Signale wurden mit einem Tektronix TDS 2022B aufgezeichnet. Für das Ausgangssignal ergibt sich ein gut zu erkennendes Sinussignal mit einem SNR von etwa 20 dB. Dies ist für einen späteren Einsatz noch nicht ausreichend. Insbesondere die Störsignale (2 bis 4) sind nicht

vertretbar. Erste Tests haben gezeigt, dass die Störsignale von der Spannungsversorgung verursacht werden.

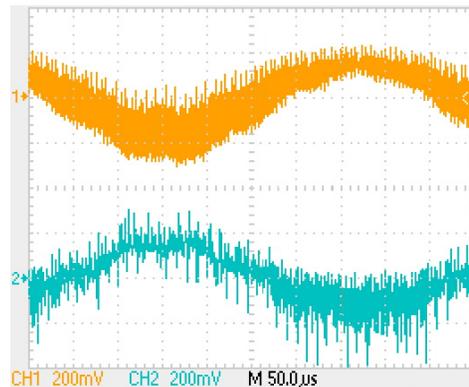


Abbildung 4.4: Messsignal am differentiellen Eingang eines ADCs; oben: positiver Eingang des ADCs, unten: negativer Eingang des ADCs.

Gravierender ist die Störbeeinflussung an den differentiellen Eingängen der ADCs (siehe Abb. 4.4). Zu den Störungen durch die Spannungsversorgung kommt hier eine hochfrequente Störung mit der Frequenz der Abtastrate der ADCs hinzu. Diese, durch die mit 446 kHz gepulste digitale Signalübertragung auf dem SPI-Bus hervorgerufene, Störung beeinflusst den Anti-Aliasing-Filter, womit das differentielle Eingangssignal des ADCs gestört ist.

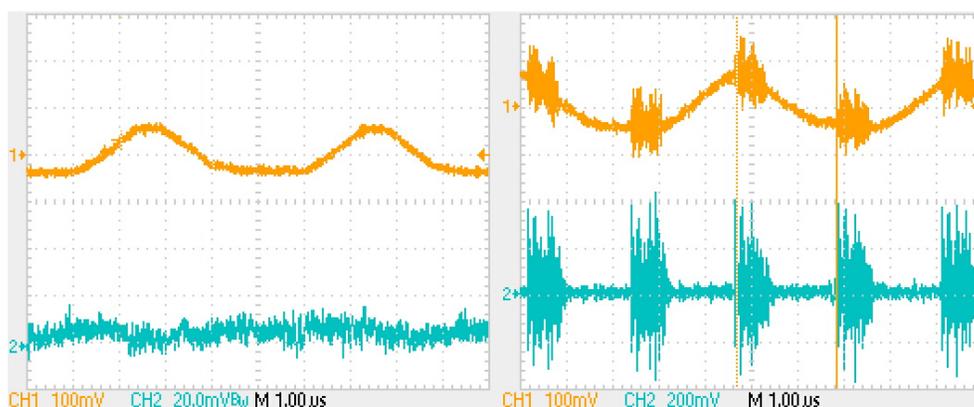


Abbildung 4.5: Störsignalauswirkung auf der Spannungsversorgung; links: +5 V Spannungsversorgung (oben) und Referenzspannung (unten) ohne Signale auf dem SPI-Interface, rechts: +5 V Spannungsversorgung (oben) und Referenzspannung (unten) mit Störsignalbeeinflussung durch Signale auf dem SPI-Interface.

Außerdem werden durch die digitale Übertragung auf dem SPI-Bus und der Beeinflussung durch die Spannungsversorgung die Versorgungsspannungen der ADCs (siehe Abb. 4.5) stark beeinflusst. Durch diese starke Beeinflussung ist eine Digitalisierung durch die ADCs nicht

---

mehr möglich. Insbesondere die Störungen auf dem Referenzsignal von bis zu 800 mV sorgen dafür, dass das digitalisierte Signal der ADCs nicht verwendet werden kann.

Im Anschluss an diese Masterarbeit muss daher das Layout des Basismoduls nochmals überarbeitet werden. In einem ersten Schritt werden dazu die digitalen Komponenten, wie die Ausgangstreiber des SPI-Busses, außer Betrieb genommen und auf einer separaten Platine mit eigener Spannungsversorgung außerhalb des Basismoduls neu aufgebaut. Anschließend werden die digitalen Signale inklusive der digitalen Masseleitung direkt an die DACs und ADCs geführt. Hiermit soll eine Störbeeinflussung der Spannungsversorgung und der Masse auf den Platinen des Basismoduls minimiert werden. Die eigentliche Spannungsversorgung des Basismoduls wird ebenfalls außer Betrieb genommen und in einem ersten Schritt durch eine Versorgung durch ein Labornetzteil ersetzt. Dabei werden die Versorgungsspannungen und das Referenzsignal direkt an die ADCs geführt. Hiermit wird sichergestellt, dass die Zuleitungen der Spannungsversorgung ausreichend dimensioniert sind. Insgesamt werden sich hiervon Erkenntnisse für ein Re-Design der Platine erhofft. Ein mögliches Re-Design sieht so aus, dass die Spannungsversorgung durch eine fertige +/- 5V Versorgung ersetzt wird. Die Impedanzwandler für die Referenzspannung, welche sich jetzt noch auf der Spannungsversorgungsplatine befinden, auf die ADC/DAC-Platine zu verlagern, um die Zuleitung möglichst kurz zu halten. Die Ausgangstreiber des SPI-Busses auf eine komplett separate Platine zu verlagern, so dass lediglich die digitalen Zuleitungen an die ADCs und den DAC auf der ADC/DAC-Platine verbleiben. Außerdem die ADCs und DACs in größeren Abstand vom restlichen analogen Bereich der Platine aufzubauen, um eine Beeinflussung des digitalen Teils vom analogen zu verringern.

Der Stromverbrauch ist ein entscheidender Faktor für den späteren Betrieb. Aus diesem Grund wurden lediglich Bauteile mit geringem Stromverbrauch (falls möglich) verwendet. Zur Bestimmung des Gesamtverbrauchs wurde der Stromverbrauch aller Komponenten separat bestimmt (siehe Tabelle 4.1). Es zeigt sich, dass das Development Board mit 73 % den Gesamtstromverbrauch am meisten beeinflusst. Das größte Einsparpotential besteht somit in dem geplanten Neuaufbau des Development Boards. Hierbei werden nur die Bauteile implementiert, welche auch wirklich benötigt werden. Ausserdem ist geplant eine abschaltbare Spannungsversorgung für das gesamte System zu realisieren. Diese soll dafür sorgen, dass zwischen einzelnen Messintervallen lediglich der Strombedarf der geplanten  $\mu$ Controller-Steuerung ver-

braucht wird. Insgesamt entspricht der Verbrauch im Messbetrieb (5 W) dem des im bestehenden System verwendeten Datatakers DT800 [50].

Tabelle 4.1: Stromverbrauch der einzelnen Komponenten.

Beschreibung	Spannungsquelle	Strom	Verbrauch	Anteil am Gesamtverbrauch
Development Board ohne Adapterplatine und Basismodul	5 V	730 mA	3,65 W	73 %
Adapterplatine	5 V	20 mA	100 mW	2 %
Basismodul nur $\mu$ Controller	5 V	6 mA	30 mW	0,6 %
Basismodul in ausgeschaltetem Zustand	12 V	2 mA	24 mW	-
Basismodul in eingeschaltetem Zustand	12 V	80 mA	960 mW	19,2 %
Messmodul	12 V	21,5 mA	258 mW	5,2 %
Gesamtverbrauch während einer Messung	-	-	5 W	100 %

## Fazit und Ausblick

---

Im Rahmen der vorliegenden Masterarbeit wurde der Neuentwurf eines bestehenden Messsystems zur Festkörperbestimmung in Meereis vorgestellt.

Eine Verbesserung der Messgenauigkeit konnte bisher noch nicht ausreichend verifiziert werden. Hierzu wird im Anschluss an diese Masterarbeit mit den gewonnenen Erkenntnissen das Layout des Basismoduls weiter verbessert. Außerdem wird eine genaue Analyse der Temperaturbeeinflussung des realisierten Aufbaues vorgenommen. Eine Verbesserung ist aber grundsätzlich durch den Aufbau der Harp3d gegeben. Die Beeinflussung der Zuleitungen des bestehenden Systems konnte durch die Messvorverstärker innerhalb der Messmodule minimiert werden. Eine Messung der Phasenverschiebung zwischen der gemessenen Spannung und des Stromes ist grundsätzlich möglich. Durch Wahl der Bauteile ist ein Einsatz der Harp3d bis  $-40^{\circ}\text{C}$  gegeben. Für die Messvorverstärker wurde dies darüber hinaus in Laborexperimenten bereits praktisch verifiziert.

Durch die Wahl eines bestehenden Development Boards mit einem FPGA als Grundlage für den Neuentwurf ist die flexible Erweiterung des Gesamtsystems realisiert worden.

Durch Wahl eines FPGAs und der Entwicklung eines Custom-IP-Cores konnte eine flexible Signalerzeugung realisiert werden. Hierdurch ist der Neuentwurf in der Lage für zukünftige Laborexperimente, als auch Feldexperimente, ein Messsignal zur Verfügung zu stellen mit der andere Messmethoden analysiert werden können.

Die Wahl der Bauteile ermöglicht einen autarken Betrieb über einen längeren Zeitraum. Um das System weiter zu verbessern, wird im Anschluss an diese Arbeit eine abschaltbare Spannungsversorgung für das System aufgebaut. Sie soll, gesteuert von nur einem  $\mu$ Controller, gewährleisten, dass zwischen einzelnen Messintervallen die Spannungsversorgung für die Basismodule und dem ML403 Development Board unterbrochen wird. Eine weitere Verbesserung des Betriebes über einen längeren Zeitraum wird durch den geplanten Neuaufbau des Development Boards erreicht. Hierbei werden nur die Komponenten implementiert die auch wirklich benötigt werden.

Der modulare Aufbau des Neuentwurfs ist eine der entscheidendsten Verbesserung. Hierdurch ist es möglich eine flexible Anzahl von Messsensoren einzusetzen. Durch die Wahl die Anbindung der Basismodule durch digitale Bussysteme zu realisieren, ist eine Beeinflussung der Messgenauigkeit ausgeschlossen. Mit dem so realisierten Design ist es erstmals möglich wirkliche dreidimensionale Messungen über eine Entfernung von einigen Metern durchzuführen.

Im Anschluss an diese Masterarbeit soll das System, nach den notwendigen Änderungen am Layout des Basismoduls, weiter erweitert werden. Hierzu soll eine Anbindung an einen GPS-Empfänger eine genaue Zeitangabe zur Datenabspeicherung gewährleisten. Eine abschaltbare Spannungsversorgung für das gesamte System soll die Stromaufnahme weiter minimieren. Der Programmablauf des Microblaze soll so geändert werden, dass der Ablauf einer Messung aus einer Konfigurationsdatei auf der Compact-Flash Karte eingelesen wird. Dies erspart, dass noch notwendige kompilieren des Quellcodes des Microblaze bei einer Änderung des Messablaufs. Zusätzlich sollen Strömungsmesser in das System integriert werden.

Insgesamt konnte gezeigt werden, dass das bestehende System elektrotechnisch grundlegend neu aufgebaut worden ist. Die vorgenommenen Änderungen erlauben es in Zukunft vollkommen neue Messaufbauten zu realisieren. Hiervon werden sich neue Erkenntnisse in den physikalischen Eigenschaften von Meereis versprochen, welche für eine Erweiterung der bestehenden Klimamodelle unbedingt notwendig sind.

# Literaturverzeichnis

- [1] NSIDC: *NSIDC BIST: Compare data: National Ice Center: Weekly Chart Products*. [http://nsidc.org/cgi-bin/bist/bist.pl?config=nic\\_weekly](http://nsidc.org/cgi-bin/bist/bist.pl?config=nic_weekly), 2009. – The Browse Image Spreadsheet Tool (BIST) was originally developed as the Web Image Spreadsheet Tool (WIST) by the National Geophysical Data Center's Geospatial Services group, T. Habermann, lead. NSIDC Scientific Programmer J. Collins generalized the tool for use with additional NSIDC data in 2007.
- [2] DEYO, Steve: *Arctic September sea ice extent (DI01898) Illustration by Steve Deyo*. <http://www.fin.ucar.edu/netpub/server.np?find&catalog=catalog&template=detail.np&field=itemid&op=matches&value=2677&site=imagelibrary>, April 2009. – ©UCAR; based on research by NSIDC and NCAR; Download am 15.04.2009
- [3] XILINX® INC. (Hrsg.): *ML401/ML402/ML403 Evaluation Platform User Guide UG080 (v2.5)*. [http://www.xilinx.com/support/documentation/boards\\_and\\_kits/ug080.pdf](http://www.xilinx.com/support/documentation/boards_and_kits/ug080.pdf): Xilinx® Inc., Mai 2006. – Download am 20.03.2008
- [4] ANALOG DEVICES (Hrsg.): *Analog Filter Wizard™ v1.0*. [http://www.analog.com/en/design-tools/dt-adisim-design-sim-tool/design-center/dt-adisim-design-sim-tool/Filter\\_Wizard/resources/fca.html](http://www.analog.com/en/design-tools/dt-adisim-design-sim-tool/design-center/dt-adisim-design-sim-tool/Filter_Wizard/resources/fca.html): Analog Devices, 2008. – Download am 17.07.2008
- [5] XILINX® INC. (Hrsg.): *MicroBlaze Processor*. [http://www.xilinx.com/products/design\\_resources/proc\\_central/microblaze.htm](http://www.xilinx.com/products/design_resources/proc_central/microblaze.htm): Xilinx® Inc., 2009. – Download am 04.04.2009
- [6] ANALOG DEVICES (Hrsg.): *16-Bit, 1.5 LSB INL, 500 kSPS PuLSAR™ Differential ADC in MSOP/QFN; AD7688*. [http://www.analog.com/static/imported-files/data\\_sheets/AD7688.pdf](http://www.analog.com/static/imported-files/data_sheets/AD7688.pdf): Analog Devices, 2008. – Download am 13.07.2008
- [7] FELTHAM, D. L. ; UNTERSTEINER, N. ; WETTLAUER, J. S. ; WORSTER, M. G.: Sea ice is a mushy layer. In: *GEOPHYSICAL RESEARCH LETTERS* 33 (2006). – L14501, doi:10.1029/2006GL026290
- [8] MAYKUT, G. A. ; UNTERSTEINER, N.: Some results from a time-dependent, thermodynamic model of sea ice. In: *J. Geophys. R.* 76 (1971), S. 1550–1575

- [9] ROTHROCK, D.A. ; YU, Y. ; MAYKUT, G.A.: Thinning of the Arctic Sea-Ice Cover. In: *GEOPHYSICAL RESEARCH LETTERS* 26 (1999), Nr. 23, S. 3469–3472
- [10] HOLLAND, M. M. ; BITZ, C. M.: Polar amplification of climate change in coupled models. In: *Climate Dynamics* 21 (2003), Nr. 3-4, S. 221–232
- [11] SHIRTCLIFFE, T.G.L. ; KERR, Ross C.: On the use of electrical resistance and temperature as measures of the solid fraction in a mushy layer. In: *Journal of Crystal Growth* 125 (1992), S. 495–501
- [12] KISSINGER, Peter T. ; HEINEMAN, William R.: *Laboratory Techniques in Electroanalytical Chemistry*. Monticello, NY., 1996. – ISBN 0-8247-9445-1
- [13] SCHWAN, H. P.: Alternating Current Electrode Polarization. In: *Biophysik* 3 (1966), S. 181–201
- [14] RUIZ, G. ; FELICE, C.J.: Non-linear response of an electrode-electrolyte interface impedance with the frequency. In: *Chaos, Solitons and Fractals* 31 (2007), S. 327–335
- [15] NOTZ, Dirk ; WETTLAUFER, John S. ; WORSTER, M. G.: A non-destructive method for measuring the salinity and solid fraction of growing sea ice in situ. In: *Journal of Glaciology* 51 (2005), Nr. 172
- [16] EHLERT, I. ; FONTES, R. ; NOTZ, D.: Improved in-situ measurements of sea-ice salinity. In: *Geophysical Research Abstracts* Bd. 11, EGU General Assembly 2009, April 2009. – EGU2009-12405
- [17] RIEGER, Phillip H.: *Electrochemistry*. Chapman & Hall, 1994. – ISBN: 0-412-04391-2
- [18] CHIARELI, A. O. P. ; WORSTER, M. G.: On measurement and prediction of the solid fraction within mushy layers. In: *Journal of Crystal Growth* 125 (1992), S. 487–494
- [19] NOTZ, Dirk: *Thermodynamic and Fluid-Dynamical Processes in Sea Ice*, University of Cambridge, Diss., 2005
- [20] YAMAGISHI, Daisuke ; LANGHORNE, Pat J.: Experimental study of phase change in rapidly growing sea ice. In: *19th IAHR International Symposium on Ice*, IAHR, 2008
- [21] XILINX® INC. (Hrsg.): *ML403 Development Board*. <http://www.xilinx.com/products/devkits/HW-V4-ML403-UNI-G.htm>: Xilinx® Inc., 2008. – Download am 14.07.2008

- [22] XILINX® INC. (Hrsg.): *Virtex-4 Family Overview (v3.0)*. [http://www.xilinx.com/support/documentation/data\\_sheets/ds112.pdf](http://www.xilinx.com/support/documentation/data_sheets/ds112.pdf): Xilinx® Inc., September 2007. – Download am 23.07.2008
- [23] TIA/EIA: *Application Guidelines for TIA/EIA-485-A (TSB-89-A-2006)*. January 2006
- [24] FRIEDRICH, Ing.-Büro: *Target 3001 Platinen CAD Software*. <http://server.ibfriedrich.com/wiki/ibfwikide/index.php?title=Hauptseite>, 2008. – Version 11
- [25] XILINX® INC. (Hrsg.): *MicroBlaze Processor Reference Guide UG081 (v9.0)*. [http://www.xilinx.com/support/documentation/sw\\_manuals/mb\\_ref\\_guide.pdf](http://www.xilinx.com/support/documentation/sw_manuals/mb_ref_guide.pdf): Xilinx® Inc., January 2008. – Download am 23.09.2008
- [26] XILINX® INC. (Hrsg.): *XPS UART Lite (v1.00a)*. [http://www.xilinx.com/support/documentation/ip\\_documentation/xps\\_uartlite.pdf](http://www.xilinx.com/support/documentation/ip_documentation/xps_uartlite.pdf): Xilinx® Inc., April 2007. – Download am 02.11.2008
- [27] XILINX® INC. (Hrsg.): *PLB Double Data Rate (DDR) Synchronous DRAM (SDRAM) Controller*. [http://www.xilinx.com/support/documentation/ip\\_documentation/plb\\_ddr.pdf](http://www.xilinx.com/support/documentation/ip_documentation/plb_ddr.pdf): Xilinx® Inc., Oktober 2003. – Download am 15.11.2008
- [28] XILINX® INC. (Hrsg.): *XPS Central DMA Controller (v1.00a)*. [http://www.xilinx.com/support/documentation/ip\\_documentation/xps\\_central\\_dma.pdf](http://www.xilinx.com/support/documentation/ip_documentation/xps_central_dma.pdf): Xilinx® Inc., April 2007. – Download am 25.03.2009
- [29] XILINX® INC. (Hrsg.): *XPS SYSACE (System ACE) Interface Controller (v1.00a)*. [http://www.xilinx.com/support/documentation/ip\\_documentation/xps\\_sysace.pdf](http://www.xilinx.com/support/documentation/ip_documentation/xps_sysace.pdf): Xilinx® Inc., März 2007. – Download am 23.07.2008
- [30] XILINX® INC. (Hrsg.): *System ACE CompactFlash Solution (v2.0)*. [http://www.xilinx.com/support/documentation/data\\_sheets/ds080.pdf](http://www.xilinx.com/support/documentation/data_sheets/ds080.pdf): Xilinx® Inc., 2008. – Download am 15.12.2008
- [31] DIXON, Jeff: *Application Hint -5V/0.3A Buck-Boost Regulator by Jeff Dixon*. [http://www.micrel.com/\\_PDF/App-Hints/ah-37k.pdf](http://www.micrel.com/_PDF/App-Hints/ah-37k.pdf): MICREL®, März 2009. – Download am 17.09.2009

- [32] MICREL<sup>®</sup> (Hrsg.): *MIC4680 1A 200kHz SuperSwitcher™ Buck Regulator*. [http://www.micrel.com/\\_PDF/mic4680.pdf](http://www.micrel.com/_PDF/mic4680.pdf): MICREL<sup>®</sup>, März 2000. – Download am 17.09.2009
- [33] ANAOLG DEVICES (Hrsg.): *Ultralow Noise XFET Voltage References with Current Sink and Source Capability ADR430/ADR431/ADR433/ADR434/ADR435/ADR439*. [http://www.analog.com/static/imported-files/data\\_sheets/ADR430\\_431\\_433\\_434\\_435\\_439.pdf](http://www.analog.com/static/imported-files/data_sheets/ADR430_431_433_434_435_439.pdf): Anaolg Devices, 2008. – Download am 13.07.2008
- [34] ANAOLG DEVICES (Hrsg.): *Single-Supply, Differential 18-Bit ADC Driver ADA4941-1*. [http://www.analog.com/static/imported-files/data\\_sheets/ADA4941-1.pdf](http://www.analog.com/static/imported-files/data_sheets/ADA4941-1.pdf): Anaolg Devices, 2008. – Download am 13.07.2008
- [35] ATMEL<sup>®</sup> (Hrsg.): *AVRISP mkII User Guide*. [www.atmel.com/dyn/resources/prod\\_documents/avrismkii\\_ug.pdf](http://www.atmel.com/dyn/resources/prod_documents/avrismkii_ug.pdf): Atmel<sup>®</sup>, April 2004. – Download am 25.07.2008
- [36] ATMEL<sup>®</sup> (Hrsg.): *ATmega8 8-bit AVR<sup>®</sup> with 8K Bytes In-System Programmable Flash*. [www.atmel.com/dyn/resources/prod\\_documents/doc2486.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc2486.pdf): Atmel<sup>®</sup>, Mai 2008. – Download am 25.07.2008
- [37] ANAOLG DEVICES (Hrsg.): *5 V, Serial-Input, Voltage-Output, 16-Bit DACs; AD5541/AD5542*. [http://www.analog.com/static/imported-files/data\\_sheets/AD5541\\_5542.pdf](http://www.analog.com/static/imported-files/data_sheets/AD5541_5542.pdf): Anaolg Devices, 2008. – Download am 13.07.2008
- [38] ANAOLG DEVICES (Hrsg.): *+3 V/+5 V/+/-5 V CMOS 4- and 8-Channel Analog Multiplexers ADG658/ADG659*. [http://www.analog.com/static/imported-files/data\\_sheets/ADG658\\_659.pdf](http://www.analog.com/static/imported-files/data_sheets/ADG658_659.pdf): Anaolg Devices, 2009. – Download am 04.02.2009
- [39] ANAOLG DEVICES (Hrsg.): *Dual Precision, Rail-to-Rail Output Operational Amplifier AD8698*. [http://www.analog.com/static/imported-files/data\\_sheets/AD8698.pdf](http://www.analog.com/static/imported-files/data_sheets/AD8698.pdf): Anaolg Devices, 2004. – Download am 13.07.2008
- [40] ANAOLG DEVICES (Hrsg.): *Precision, Very Low Noise, Low Input Bias Current, Wide Bandwidth JFET Operational Amplifiers AD8510/AD8512/AD8513*. [http://www.analog.com/static/imported-files/data\\_sheets/AD8510\\_8512\\_8513.pdf](http://www.analog.com/static/imported-files/data_sheets/AD8510_8512_8513.pdf): Anaolg Devices, 2009. – Download am 3.04.2009

- [41] FRENZEL, Louis E.: Accurately measure nanoampere and picoampere currents.(Design basics of). In: *Electronic Design* (2007), Februar
- [42] LOEVINGER, R.: Precision Measurement with the Total\_Feedback Electrometer. In: *Phys. Med. Biol.* 11 (1966), Nr. 2, S. 267–279
- [43] XILINX® INC. (Hrsg.): *Platform Studio and EDK Details*. [http://www.xilinx.com/ise/embedded/edk\\_pstudio.htm](http://www.xilinx.com/ise/embedded/edk_pstudio.htm): Xilinx® Inc., 2008. – Download am 14.07.2008
- [44] XILINX® INC. (Hrsg.): *ISE® WebPACK™*. [http://www.xilinx.com/ise/logic\\_design\\_prod/web-pack.htm](http://www.xilinx.com/ise/logic_design_prod/web-pack.htm): Xilinx® Inc., 2008. – Download am 14.07.2008
- [45] XILINX® INC. (Hrsg.): *DDS v5.0*. [http://www.xilinx.com/support/documentation/ip\\_documentation/dds.pdf](http://www.xilinx.com/support/documentation/ip_documentation/dds.pdf): Xilinx® Inc., April 2005. – Download am 10.10.2008
- [46] XILINX® INC. (Hrsg.): *Multiplier v.10.1*. <http://www.xilinx.com/products/ipcenter/multiplier.htm>: Xilinx® Inc., April 2008. – Download am 10.10.2008
- [47] SIKORA, Axel ; LEIPZIG, Fachbuchverlag (Hrsg.): *Technische Grundlagen der Rechnerkommunikation*. ISBN 3-446-22455-6 : Carl Hanser Verlag München Wien, 2003
- [48] ATMEL® (Hrsg.): *AVR Studio®4*. [http://www.atmel.com/dyn/Products/tools\\_card.asp?tool\\_id=2725](http://www.atmel.com/dyn/Products/tools_card.asp?tool_id=2725): Atmel®, 2008. – Download am 16.06.2008
- [49] WINAVR™ (Hrsg.): *WINAVR™*. <http://winavr.sourceforge.net/>: WINAVR™, 2008. – Open Source Project; Download am 16.06.2008
- [50] DATATAKER (Hrsg.): *DT800 Data Logger*. [http://www.datataker.com/Library/Product\\_Data\\_Sheets\\_TS](http://www.datataker.com/Library/Product_Data_Sheets_TS) DataTaker, 2009. – Datenblatt; Download am 05.02.2009
- [51] HUNTER, Robert J.: *Foundations of colloid science*. Oxford University Press, Oxford, 1986. – ISBN 0–19–855188–6
- [52] JONES, G. ; BOLLINGER, G. M. ; PARKER, H. C. ; WIEN, M. ; JOOS, G. ; BLUMENTRITT, M. ; DEBYE, P. ; HÜCKEL, E. ; KRAUS, C. A. ; MORGAN, J. L. R. ; LAMMERT, O. M. ; STAMM, A. J. ; HASSELBLATT, M.: Leitfähigkeitsstudien. In: *Fresenius' Journal of Analytical Chemistry* 88 (1932), Nr. 9-10, S. 358–365

- [53] KORYTA, Jin ; DVORAK, Jin ; KAVAN, Ladislav: *Principles of electrochemistry*. John Wiley & Sons Ltd., England, 1993. – ISBN 0–471–93713–4
- [54] RAJENDRAN, A. ; NEELAMEGAM, P.: Measurement of conductivity of liquids using AT89C55WD microcontroller. In: *Measurement* 35 (2004), S. 58–65
- [55] PRADO, J. ; MARGO, C. ; KOUIDER, M. ; NADI, M.: Auto balancing bridge method for bioimpedance measurement at low frequency. In: *1st International Conference on Sensing Technology November 21-23 (2005)*
- [56] DELLAVALE, D. ; SONNAILLON, M.O. ; BONETTO, F.J.: FPGA based Multi-Harmonic Control System for Single Bubble Sonoluminescence. In: *Programmable Logic, 2008 4th Southern Conference on (2008)*, S. 269 – 272
- [57] SHIRTCLIFFE, T. G. L. ; HUPPERT, H. E. ; WORSTER, M. G.: Measurements of the solid fraction in the crystallisation of a binary melt. In: *J. Crystal Growth* 113 (1991), S. 566ff
- [58] XILINX®INC. (Hrsg.): *PLB IPIF (v2.02a)*. innerhalb des EDK: Xilinx®Inc., April 2005

## Anhang für Kapitel 4

---

### A.1 Schaltpläne und Layouts der realisierten Hardware

Seite 88: Schaltplan der Adapterplatine.

Seite 89: Layout der Adapterplatine von der Oberseite und der Unterseite.

Seite 90: Schaltplan der Spannungsversorgungsplatine des Basismoduls.

Seite 91: Schaltplan der Steuerungsplatine des Basismoduls.

Seite 92: Schaltplan der DAC/ADC Platine des Basismoduls.

Seite 93: Layout der Basismodul Platinen von der Oberseite.

Seite 94: Layout der Basismodul Platinen von der Unterseite.

Seite 95: Schaltplan für die Messsensoren der Platindrähte des Messmoduls.

Seite 95: Layout für die Messsensoren der Platindrähte des Messmoduls.

Seite 96: Schaltplan für die Adapterplatine der Platindrähte des Messmoduls.

Seite 97: Layout für die Adapterplatine der Platindrähte des Messmoduls.

Seite 97: Schaltplan für die Messsensoren der Thermistoren des Messmoduls.

Seite 98: Layout für die Messsensoren der Thermistoren des Messmoduls.

Seite 99: Schaltplan für die Adapterplatine der Thermistoren des Messmoduls.

Seite 100: Layout für die Adapterplatine der Thermistoren des Messmoduls.



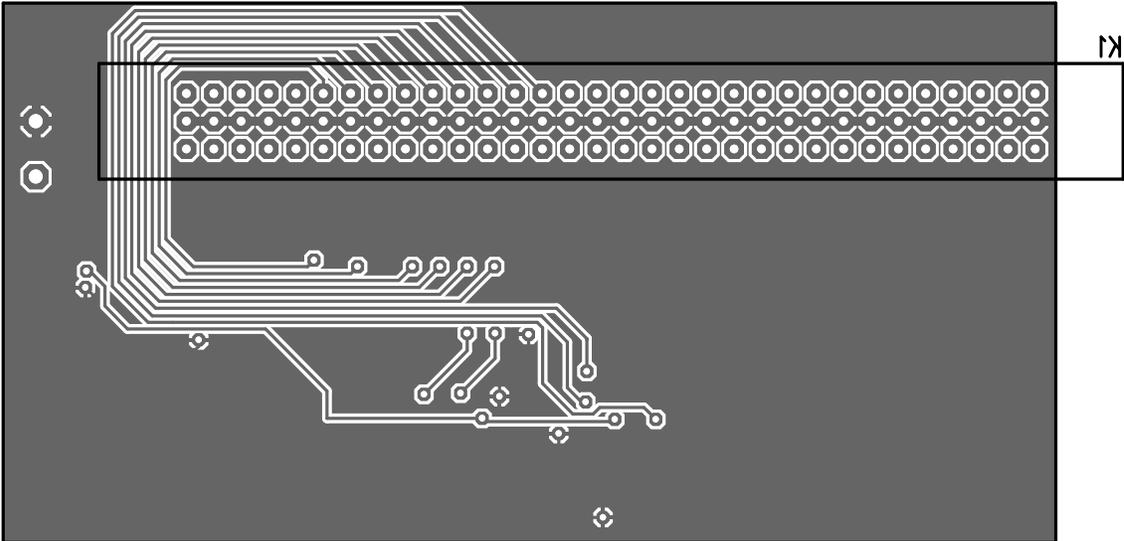
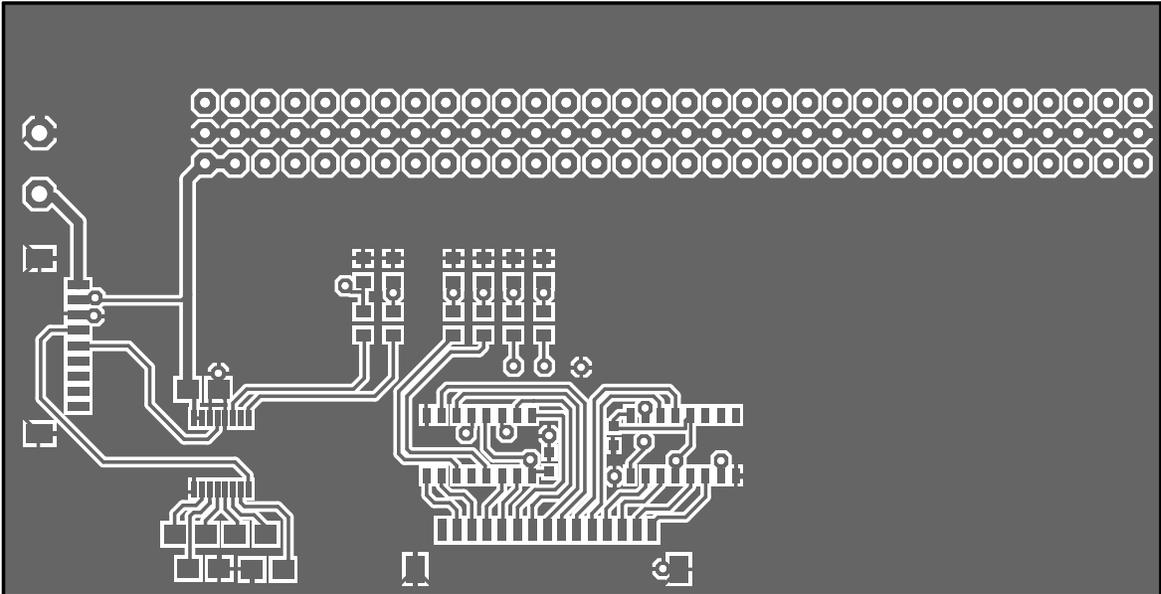


Abbildung A.2: Layout der Adapterplatine von der Oberseite (oben) und von der Unterseite (unten).

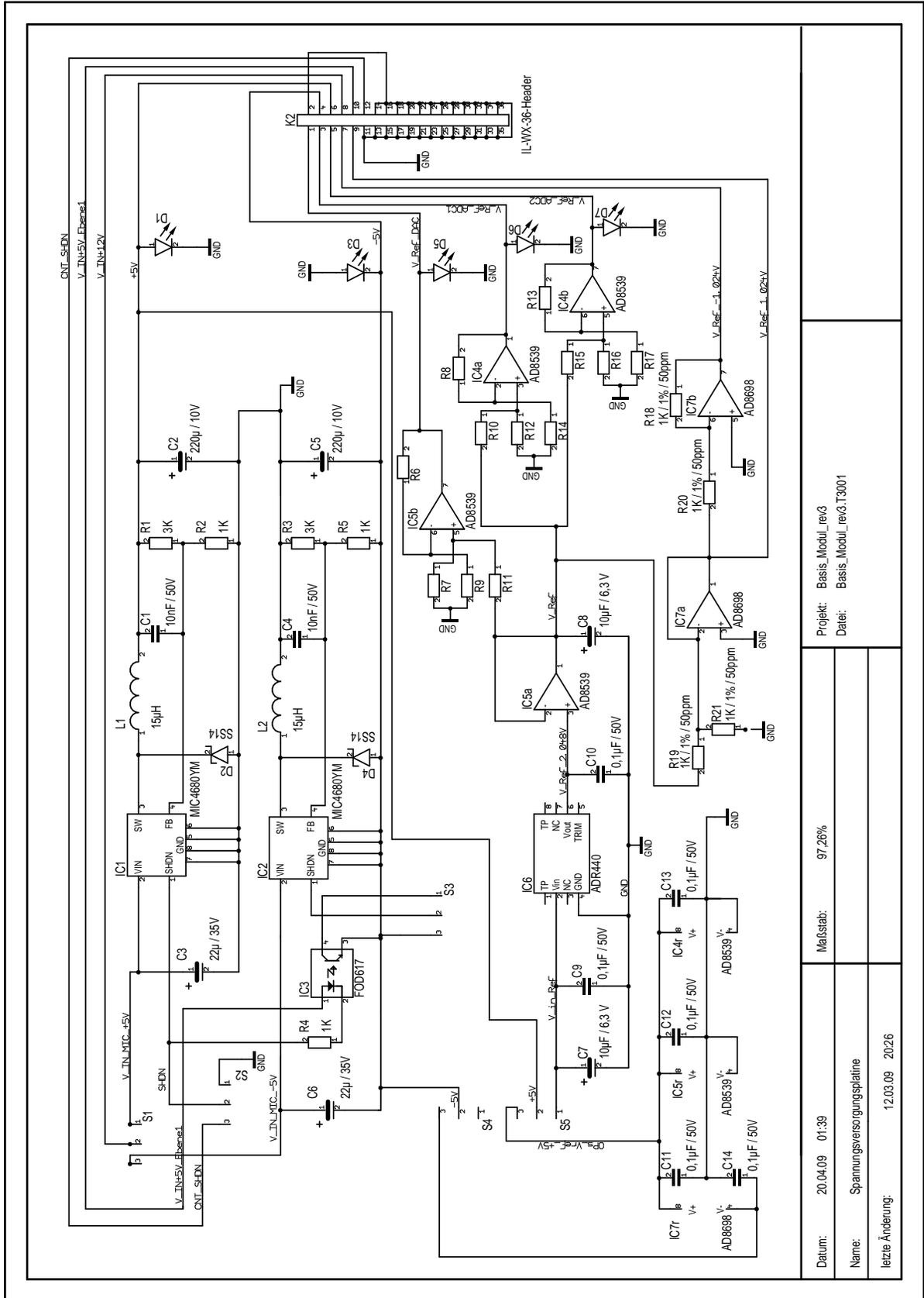


Abbildung A.3: Schaltplan der Spannungsversorgung des Basismoduls.

Datum:	20.04.09 01:39	Maßstab:	97,26%
Name:	Spannungsversorgungspaltine	Projekt:	Basis_Modul_rev3
letzte Änderung:	12.03.09 20:26	Datei:	Basis_Modul_rev3.T3001

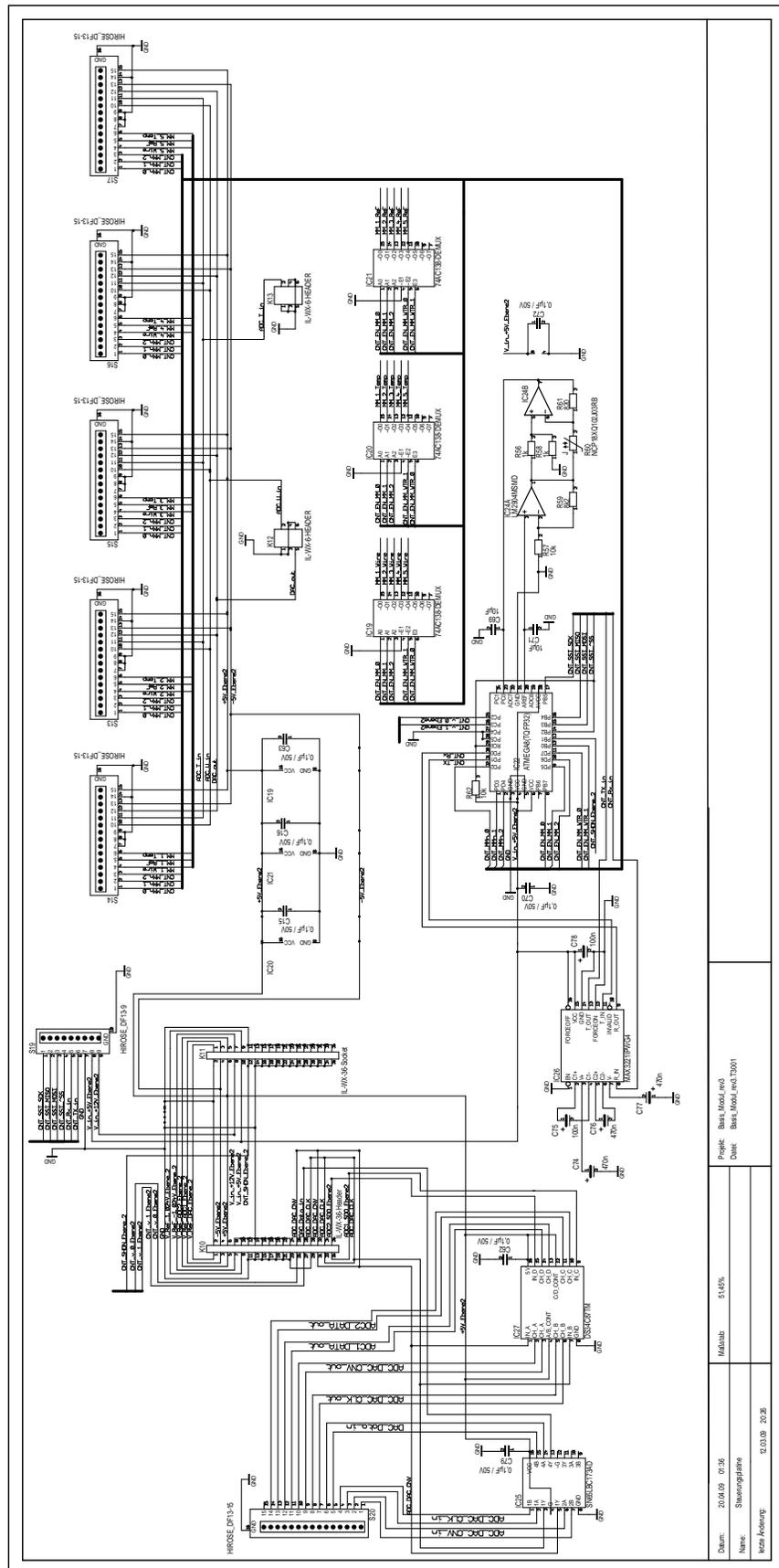


Abbildung A.4: Schaltplan der Steuerplatine des Basismoduls.

Projekt: Bas1_Modul_v04	Modulart: 5145%
Datum: 20.04.09 07:38	Name: Steuerungslatze
Dat: Bas1_Modul_v04_13001	Werk: K&H
	12.03.08 20:26





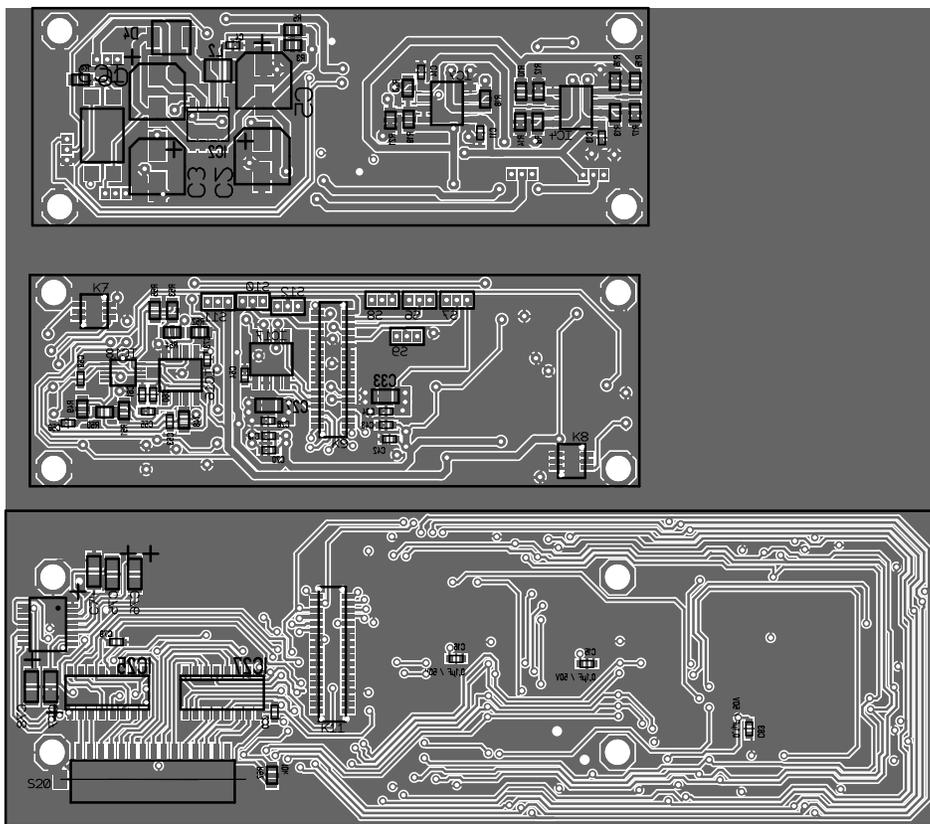


Abbildung A.7: Layout der Basismodul Platinen von der Unterseite (oben Spannungsversorgungsplatine, mitte DAC/ADC Platine und unten Steuerungsplatine)

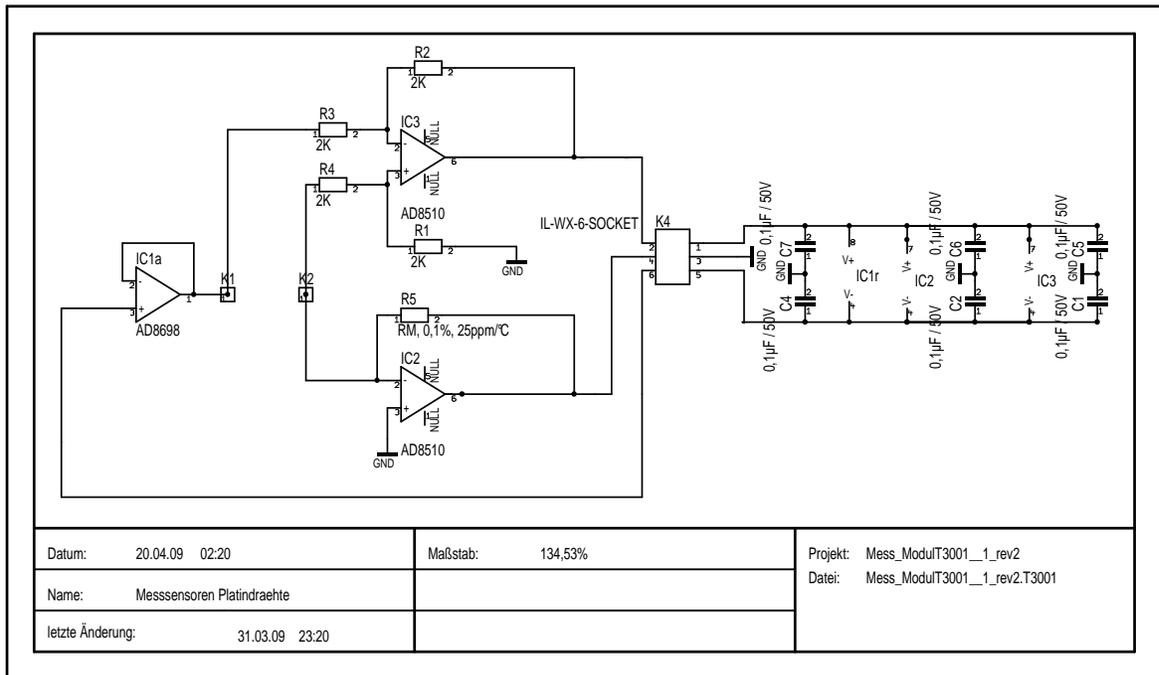


Abbildung A.8: Schaltplan der Platine für die Messsensoren der Platindrähte des Messmoduls.

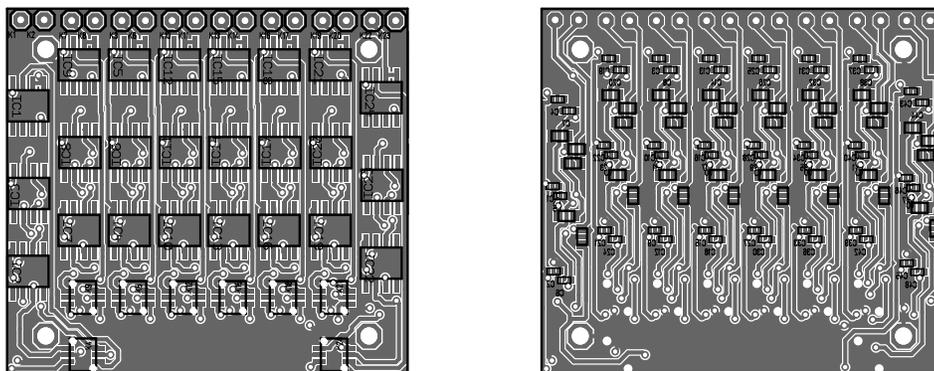


Abbildung A.9: Layout der Platine für die Messsensoren der Platindrähte des Messmoduls (rechts die Oberseite und links die Unterseite).

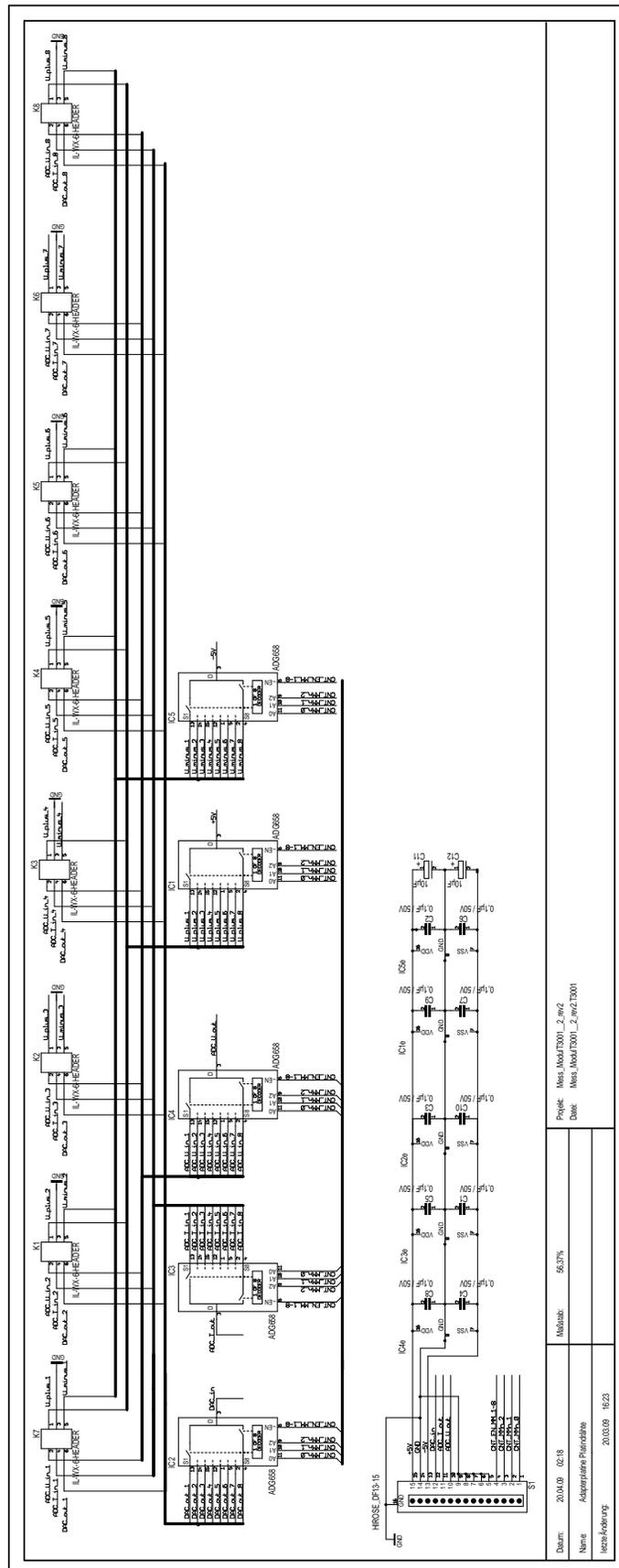


Abbildung A.10: Schaltplan der Platine für die Adapterplatine der Messsensoren der Platindrähte des Messmoduls.

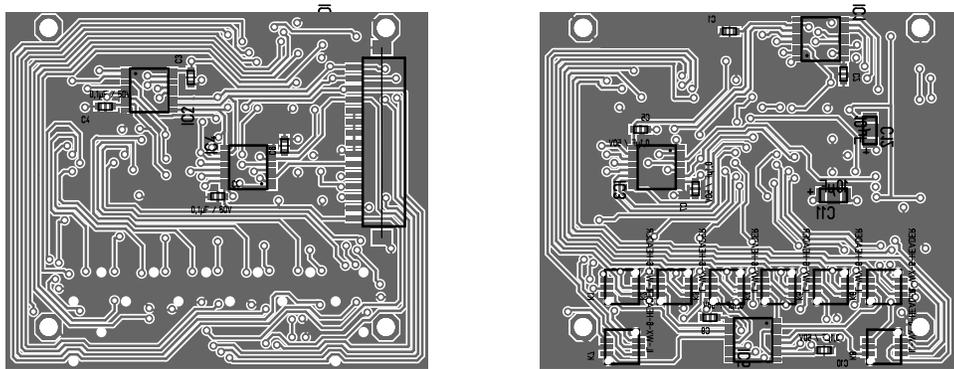


Abbildung A.11: Layout der Platine für die Adapterplatine der Platindrähte des Messmoduls (rechts die Oberseite und links die Unterseite).

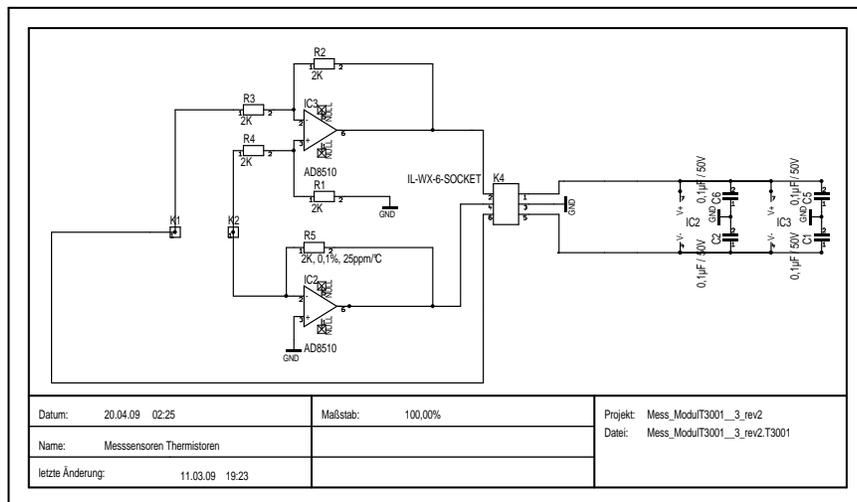


Abbildung A.12: Schaltplan der Platine für die Messsensoren der Thermistoren des Messmoduls.

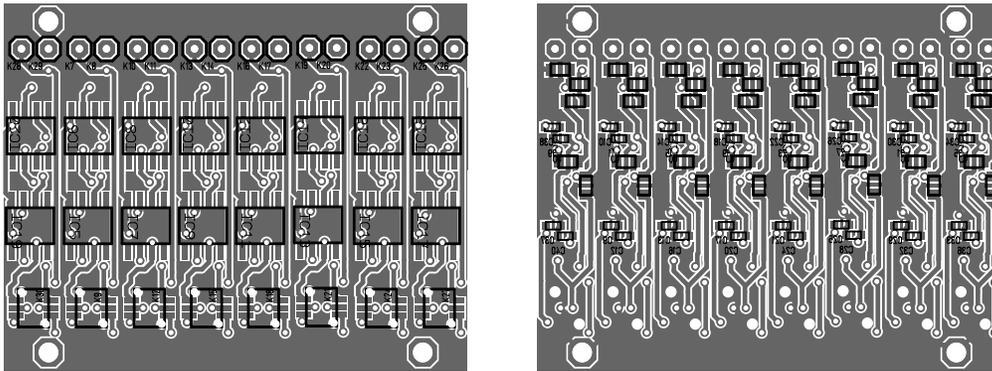


Abbildung A.13: Layout der Platine für die Messsensoren der Thermistoren des Messmoduls (rechts die Oberseite und links die Unterseite).

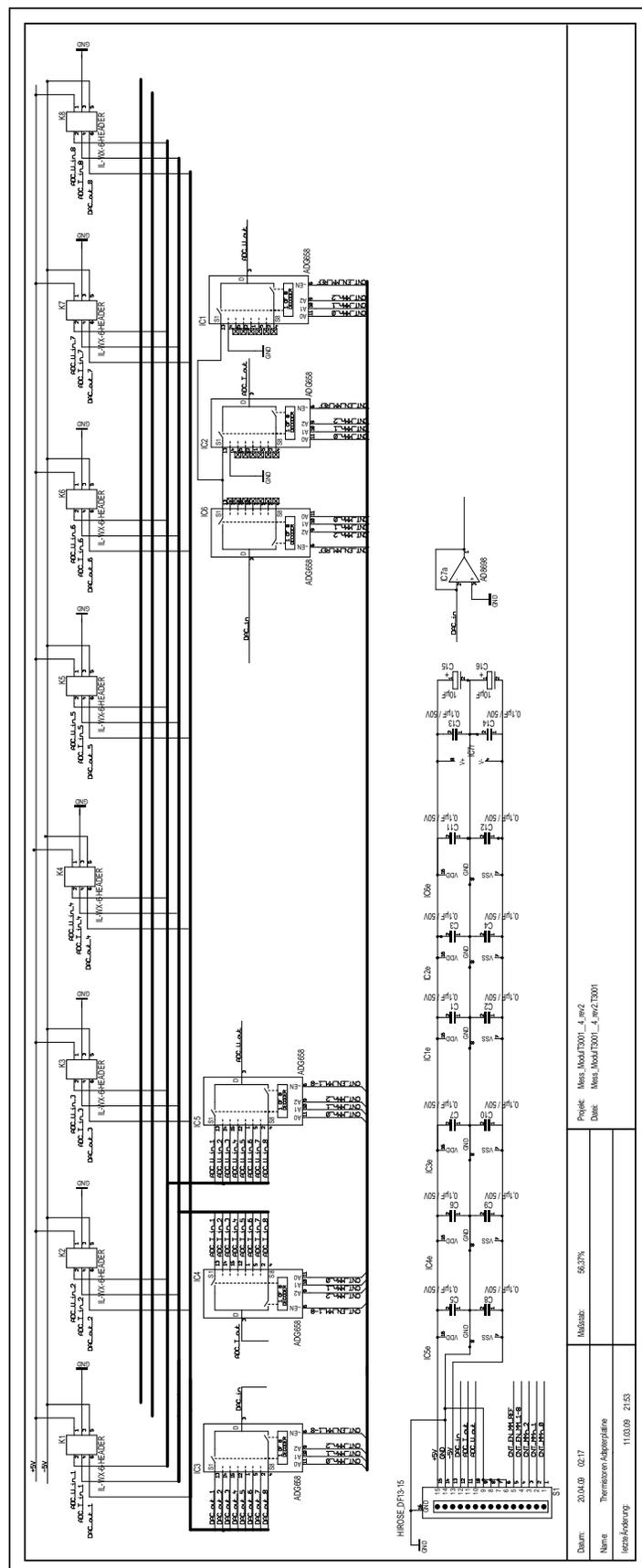


Abbildung A.14: Schaltplan der Platine für die Adapterplatte der Thermistoren der Platindröhre des Messmoduls.

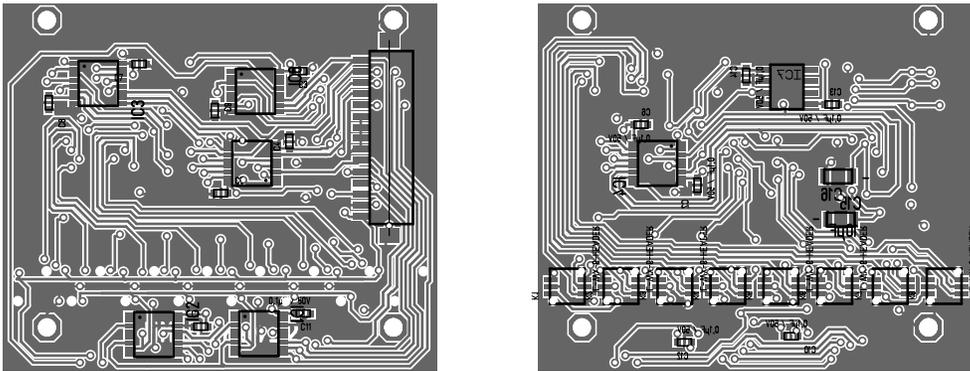


Abbildung A.15: Layout der Platine für die Adapterplatine der Thermistoren des Messmoduls (rechts die Oberseite und links die Unterseite).

## A.2 Dimensionierung des Interpolations-Tiefpasses

Auflistung der Bilder von der Erstellung des Interpolations-Tiefpasses mit Hilfe des *Analog Filter Wizard™* Tools.

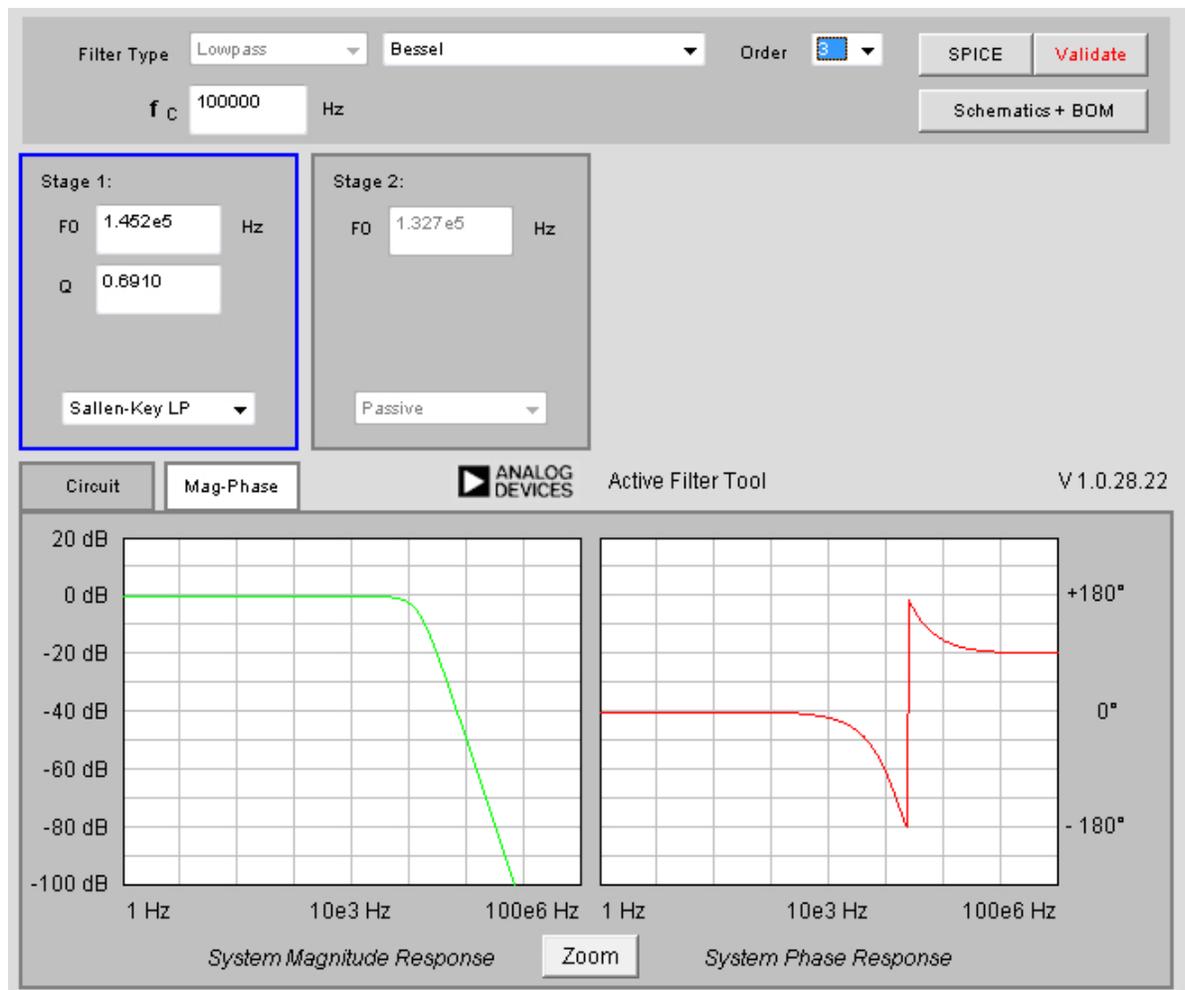
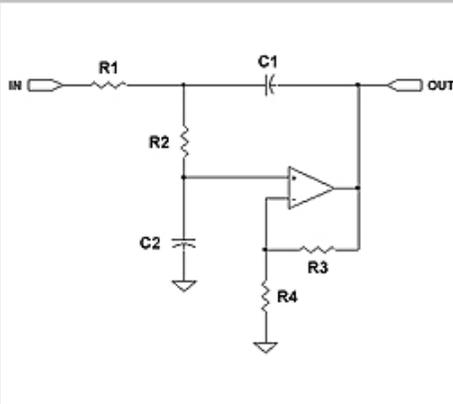


Abbildung A.16: Dimensionierung des Interpolations-Tiefpasses 1 [4].

Stage 1:  
 FO 1.452e5 Hz  
 Q 0.6910  
 Sallen-Key LP

Stage 2:  
 FO 1.327e5 Hz  
 Passive

Circuit Mag-Phase ANALOG DEVICES Active Filter Tool V 1.0.28.22



Gain 1.0  
 C1 1 nF  
 R3 22.0 K Ohms

Lock cap

R1 1.5 K Ohms R2 1.5 K Ohms C1 1.0 nF C2 0.5100 nF  
 R3 22.1 K Ohms R4 Infinity Ohms

Tolerance R 1.0% C 5.0%

Abbildung A.17: Dimensionierung des Interpolations-Tiefpasses 2 [4].

The screenshot shows the 'Active Filter Tool' interface. At the top, there are two stages of filter configuration:

- Stage 1:** FO: 1.452e5 Hz, Q: 0.6910, Filter Type: Sallen-Key LP.
- Stage 2:** FO: 1.327e5 Hz, Filter Type: Passive.

The 'Stage 2' section is highlighted with a blue border. Below the stages, there are tabs for 'Circuit' and 'Mag-Phase', and the 'ANALOG DEVICES' logo. The main area shows a circuit diagram of a passive RC low-pass filter with an input 'IN', a resistor 'R', a capacitor 'C' connected to ground, and an output 'OUT'. To the right of the diagram, the capacitor value is set to 1 nF. Below the diagram, there are input fields for R1 (1.21 K Ohms) and C1 (1.0 nF). At the bottom, there are tolerance settings for R (1.0%) and C (5.0%), and a 'Lock cap' checkbox which is checked.

Abbildung A.18: Dimensionierung des Interpolations-Tiefpasses 3 [4].

### A.3 Dimensionierung des Antialiasing-Filters

Auflistung der Bilder von der Erstellung des Antialiasing-Tiefpasses mit Hilfe des *Analog Filter Wizard™* Tools.

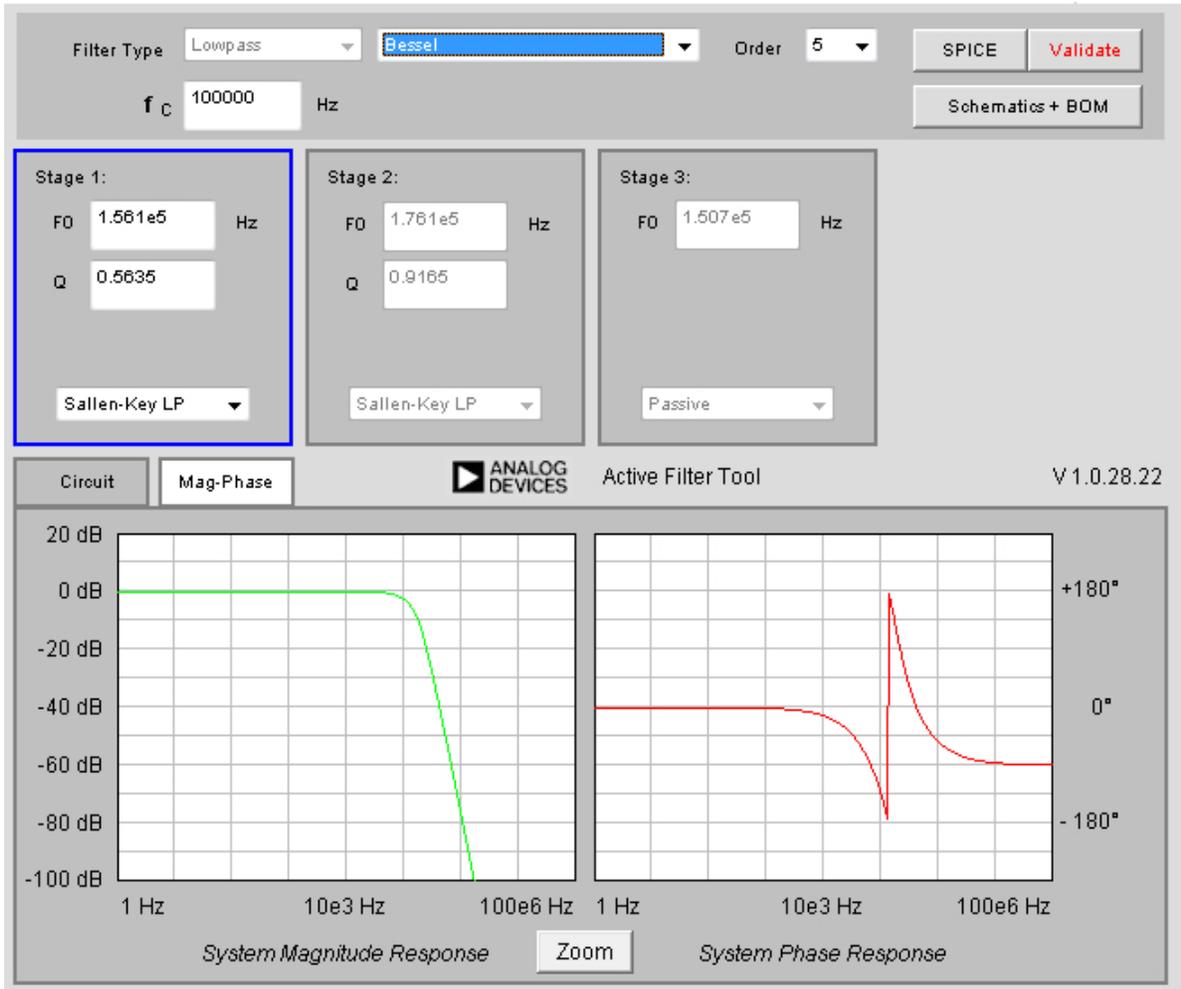


Abbildung A.19: Dimensionierung des Antialiasing-Tiefpasses 1 [4].

The screenshot displays the 'Active Filter Tool' interface for configuring a filter. It is divided into three stages:

- Stage 1:** F0 = 1.561e5 Hz, Q = 0.5635, Filter Type: Sallen-Key LP.
- Stage 2:** F0 = 1.761e5 Hz, Q = 0.9165, Filter Type: Sallen-Key LP.
- Stage 3:** F0 = 1.507e5 Hz, Filter Type: Passive.

The interface includes a 'Circuit' tab and a 'Mag-Phase' tab. The 'Circuit' tab shows a schematic diagram of a Sallen-Key LP filter with components R1, R2, R3, R4, C1, and C2. The 'Mag-Phase' tab shows the filter's gain and component values:

- Gain: 1.0
- C1: 1 nF
- R3: 22.0 K Ohms

At the bottom, a table lists the component values for the filter:

Component	Value	Unit
R1	1.150 K	Ohms
R2	1.150 K	Ohms
R3	22.1 K	Ohms
R4	Infinity	Ohms
C1	1.0	nF
C2	0.8200	nF

Tolerance settings: R 1.0%, C 5.0%. A 'Lock cap' checkbox is also present.

Abbildung A.20: Dimensionierung des Antialiasing-Tiefpasses 2 [4].

The screenshot displays the 'Active Filter Tool' interface for designing a 3-stage antialiasing low-pass filter. The tool is set to 'Mag-Phase' mode. The design parameters for the three stages are as follows:

Stage	Filter Type	Center Frequency (F0) [Hz]	Quality Factor (Q)
Stage 1	Sallen-Key LP	1.561e5	0.5635
Stage 2	Sallen-Key LP	1.761e5	0.9165
Stage 3	Passive	1.507e5	-

The circuit diagram shows an active filter stage with the following components and values:

- Gain: 1.0
- C1: 1 nF
- R3: 22.0 K Ohms
- R1: 1.650 K Ohms
- R2: 1.650 K Ohms
- R4: Infinity Ohms
- C2: 0.3000 nF

At the bottom, the tolerance settings are: R 1.0% and C 5.0%.

Abbildung A.21: Dimensionierung des Antialiasing-Tiefpasses 3 [4].

The screenshot displays the 'Active Filter Tool' interface for configuring a three-stage filter. The tool is set to 'Mag-Phase' mode. The configuration is as follows:

Stage	Filter Type	FO (Hz)	Q
Stage 1	Sallen-Key LP	1.561e5	0.5635
Stage 2	Sallen-Key LP	1.761e5	0.9165
Stage 3	Passive	1.507e5	-

The circuit diagram shows an input terminal 'IN' connected to a resistor 'R', which is then connected to an output terminal 'OUT'. A capacitor 'C' is connected from the node between 'R' and 'OUT' to ground. The component values are:

- R1: 1.05 K Ohms
- C1: 1.0 nF

Tolerance settings are: R 1.0%, C 5.0%. A 'Lock cap' checkbox is checked.

Abbildung A.22: Dimensionierung des Antialiasing-Tiefpasses 4 [4].

## A.4 Custom-IP-Core

Tabelle A.1: Zuordnung der we-Signale zu den Eingangswerten des harp3d\_data\_select-Signals.

harp3d_data_select-Wert	we-Signal	Funktion
1	measure_time_we	Messzeiteinstellung
2	freq_sinus_we	Frequenzeinstellung der Sinussignalerzeugung
3	mult_sinus_we	Amplitudeneinstellung der Sinussignalerzeugung
4	freq_sawtouth_we	Frequenzeinstellung der Dreiecksignalerzeugung
5	mult_sawtouth_we	Amplitudeneinstellung der Dreiecksignalerzeugung
6	dc_we	Amplitudeneinstellung der Gleichspannungssignalerzeugung
7	freq_square_we	Frequenzeinstellung der Rechtecksignalerzeugung
8	mult_square_we	Amplitudeneinstellung der Rechtecksignalerzeugung

Tabelle A.2: Verwendete Hardwareressourcen innerhalb des FPGA für den Custom-IP-Core.

Resource Type	Used	Available	Percent
Slices	1062	5472	19
Slice Flip Flops	1286	10944	11
4 input LUTs	1408	10944	12
IOs	282	NA	NA
bonded IOBs	0	320	0
FIFO16/RAMB16s	8	36	22
GCLKs	10	32	31
DSP48s	3	32	9

# Danksagung

Für die technische Unterstützung möchte ich mich bei Andreas Winderlich, Friedhelm Jansen und Björn Brügmann bedanken.

Für die tolle Teamarbeit bei der Entwicklung des gesamten Systems bedanke ich mich bei Iris Ehlert und Dirk Notz.

Bei meinen Eltern bedanke ich mich für die finanzielle Unterstützung.

Weiter möchte ich mich bei Marci @ Confoederatio Henschlumpfica und Laura bedanken, die mir den persönlichen Rückhalt während dieser Arbeit gegeben haben und mir darüber hinaus eine große Hilfe bei Schrift und Form der vorliegenden Arbeit waren.

Außerdem bedanke ich mich bei allen die hier nicht mit aufgeführt sind, es aber verdient hätten hier aufgeführt zu sein.



# Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen habe ich unter Angabe der Quellen kenntlich gemacht.

---

Ort, Datum

---

Unterschrift