

Diplomarbeit

DA595

Entwicklung einer Testumgebung für kontaktlose Chipkarten am Spea Tester

Bernhard Roitner

Institut für Elektronik

Technische Universität Graz

Vorstand: Univ.-Prof. Dipl.-Ing. Dr. techn. Wolfgang Pribyl



Begutachter: Ass.-Prof. Dipl.-Ing. Dr. techn. Peter Söser

Graz, im Oktober 2011



Infineon Technologies Austria AG
Development Center Graz
Betreuer: **Dipl.-Ing. Walter Kargl**

Abstract

During the last years fields of application for contactless microchips, so called radio frequency identification microchips, have increased continuously. These microchips have the advantage, that they can communicate without having electrical contact to a reader. Most of them do not even have a power supply like a battery. The main advantage of getting data from a microchip on a contactless way, is that there is no possibility of abrasion of contact pads or other parts. Further on there is less vulnerability to damage by vandalism, contrary to contact based chip cards. It is however necessary that the transponder is in close proximity to the reader antenna to establish a data connection.

The continuous rising of diversity of RFID-tags and their derivatives led to the need of optimisation in development time and particularly to a reduction of testing time. To reduce this time-consuming verification and characterisation, the idea was born to move tests on a wafer tester. Wafer testers are not built to make the kind of tests as described in ISO/IEC 10373-6[8] which is the reason why this diploma thesis was created. The main feature of wafer testers is, that they test dies directly on the wafer. Because of that nobody has to saw the dies and bond them in a package. These testers have many I/O lines, from 128 up to more than 1024, so one can test a lot of dies in parallel. Additionally the wafer prober can collect as many as 25 wafers, which is known as a lot, and can automatically supply them to the tester one at a time. Therefore no manpower is needed to change the dies. Further components can also be controlled like a chiller, where one can measure at different temperatures. At Infineon typically 64 RFID dies are tested in parallel. The main problem of this wafer tester is its low RFID functionality. It is not able to generate the analog high frequency signals with the impedance, which the dies see at the antenna pads in an radio environment. To build this missing component is a main part of this diploma thesis. Another part is to develop software to convert the required waveforms for the signals.

Kurzfassung

In den letzten Jahren stiegen Einsatzgebiete und Anwendungen für kontaktlose Chips, so genannte **R**adio **F**requency **I**Dentification Microchips, stetig an. Die Vorteile des berührungslosen Auslesens von Daten aus einem Chip sind im Wesentlichen, dass es zu keinen Kontaktproblemen und Verschleiß von mechanischen Teilen kommt. Der Chip mit seiner Antenne muss sich nur in der direkten Umgebung der Leseantenne befinden, um eine Datenkommunikation aufbauen und wenn nötig auch seine Versorgungsenergie aus dem HF-Feld des Lesegerätes beziehen zu können. Die immer größer werdende Vielfalt von RFID-Chips und deren Derivate führte zu einer Optimierung der Entwicklungszeiten, wovon ein Großteil Testzeiten ausmachen. Diese zeitintensive Verifikation und Charakterisierung erfordert umfangreiche Tests. Um diese Zeiten zu verkürzen, wurden Überlegungen angestellt Tests zu optimieren und zu automatisieren. Eine Idee war, gewisse Tests schon in der Frühphase auf einem Wafertester durchzuführen, da damit viele Chips gleichzeitig und automatisiert getestet werden können. Es müssen dafür keine CDIPs aufgebaut werden, die dann wiederum nur händisch gewechselt werden können. Auf so einem Wafertester können abhängig von Wafergröße und Chipgröße einige tausende Chips pro Wafer mit einem automatischen Waferprober 25 Wafer, das entspricht einem Los, in einem Zyklus automatisch getestet werden. Auf so einem Chiptester stehen abhängig vom Typ 128, 256, 512, oder auch mehr Signalleitungen zur Verfügung, die abhängig von der Anzahl der verwendeten Signalleitungen pro Chip eine hohe Parallelisierung der Messungen auf mehreren Chips ermöglicht. Typischerweise werden bei Infineon 64 RFID-Chips mit integriertem Mikrocontroller parallel getestet. Das große Problem dieser Chiptester mit Waferprober sind ihre geringe RFID Tauglichkeit, da sie entwickelt wurden um digital kontaktbasiert mit dem Chip zu kommunizieren, somit können sie nicht beliebige Kurvenformen ausgeben, die einem RFID-Chip im HF-Feld entsprechen. Da im Speziellen für die Charakterisierung und Verifikation des Analogteils an den Antennenanschlüssen des RFID-Chip Signale und Impedanzen erzeugt werden müssen, die dem entsprechen, wie es der Chip bei seiner Anwendung mit Antenne im HF-Feld vorfindet, gab es Überlegungen, ob nicht mit relativ wenig Schaltungsaufwand solche Signale nachgebildet werden können. Diese Diplomarbeit beschreibt eine mögliche Lösung dieses Problems.

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Ort

Datum

Unterschrift

Danksagung

Diese Diplomarbeit wurde am Institut für Elektronik an der Technischen Universität Graz in Kooperation mit der Firma Infineon Technologies Austria im Design Center Graz durchgeführt.

Mein besonderer Dank für die geduldige Betreuung am Institut für Elektronik gilt Herrn Ass.-Prof. Dipl.-Ing. Dr.techn. Peter Söser.

Außerdem möchte ich mich bei Herrn Ass.-Prof. Dipl.-Ing. Dr.techn. Gunter Winkler für die technische Unterstützung sehr bedanken.

Weiters möchte ich mich bei meinem Betreuer Dipl.-Ing. Walter Kargl bei Infineon bedanken, von dem die grundsätzliche Idee dieser Diplomarbeit ausging. Natürlich gehören auch die vielen Kollegen bei Infineon erwähnt, die mir mit Rat und Tat beiseite standen.

Bedanken möchte ich mich außerdem bei meinen vielen Freunden, die mir bei jeder Flaute mit neuem Rückenwind halfen weiterzumachen, um diese Arbeit zu beenden.

Besondere Anerkennung gebührt meinen Eltern, die mir ermöglichten dieses Studium zu absolvieren und nicht nur finanziellen Rückhalt gaben, sondern auch in vielen anderen Belangen unterstützten. Nicht zu vergessen meine Schwester, die beim Endspurt eine wichtige Unterstützung war.

Graz, im Oktober 2011

Bernhard Roitner

Inhaltsverzeichnis

1	Einleitung	14
1.1	Motivation	14
1.2	Zielsetzung	14
1.3	Gliederung	15
2	Allgemeines	17
2.1	Automatische Identifikationssysteme	17
2.1.1	Barcode-Systeme	18
2.1.2	Optical Character Recognition	18
2.1.3	Biometrische Verfahren	18
2.1.4	Chipkarten	20
2.1.4.1	Speicherkarten	20
2.1.4.2	Mikroprozessorkarten	21
2.1.5	RFID-Systeme	22
2.1.5.1	Kontaktlose Chipkarten	23
3	Theorie	25
3.1	Physikalische Grundlagen	25
3.1.1	Durchflutungsgesetz (Ampèresches Gesetz)	25
3.1.2	Induktionsgesetz (Faradaysches Gesetz)	27
3.2	RFID-Kopplung	27
4	Messverfahren	35
4.1	Testmethoden nach Norm	35
4.1.1	ISO-Turm	35
4.1.2	Calibration Coil	37
4.1.3	Reference PICC	37
4.2	Messung mit einem AWG-Setup	38
4.3	Messung mit dem CL-Emulator	40

5	Hardware	42
5.1	Einleitung	42
5.2	Evaluation Board	43
5.2.1	Digital-Analog-Umsetzer	44
5.2.1.1	AD9772 Evaluation Board von Analog Devices	44
5.2.1.2	AD9772 Evaluation Board Ausgangsverstärker	46
5.2.1.3	Rekonstruktionsfilter Teil 1	47
5.2.2	Eingangsverstärker	50
5.2.3	Rekonstruktionsfilter Teil 2	51
5.2.4	Stromquelle	54
5.2.5	Widerstandsnetzwerk	57
5.2.6	Testchip	59
5.2.7	Spannungsversorgungen	60
5.2.7.1	Spannungsversorgung des DA-Evaluation Boards	60
5.2.7.2	Spannungsversorgung Eingangsverstärker, Offsetabgleich, Filter und Stromquelle	62
5.2.7.3	Spannungsversorgung für den Dual Interface Chip	63
5.2.8	Inbetriebnahme	63
5.2.8.1	Verbindung zwischen DA- und CL Evaluation Board	64
5.2.8.2	Verbindung zwischen CL-Evaluation Board und Spea Load Board	65
5.2.8.3	Eingänge und Messpunkte	66
5.2.9	Messergebnisse	66
5.3	Weitere Schritte	68
6	Spea Chiptester	70
6.1	Das Comptestsystem	72
6.2	Load Board L306E	73
7	Software	75
7.1	Tabor zu Spea Konvertierungssoftware	75
7.1.1	Verwendung der Software mit einer <i>.ini</i> -Datei	77
7.1.2	Verwendung der Software mit einer <i>.bat</i> -Datei	78
7.1.3	Aufbau der Software	79
7.1.4	CL-template.p40	82
7.2	Spea Software	82

8	Zusammenfassung und Ausblick	85
A	Abkürzungsverzeichnis	87
B	Glossar	89
C	Literaturverzeichnis	90
C.1	Web links	91
D	Widerstandswertetabelle	92
E	Schaltplan	97
F	Layout	98
G	Product Brief: SLE 66CLxxxPE[6]	100
H	Quelltext	102
H.1	Konvertierungssoftware Tabor2Spea.exe	102
H.2	Spea .p40 Template	112
H.3	CL-Emulation.ini Datei	115
I	Fotos	117
J	Formeln	120

Abbildungsverzeichnis

2.1	Zusammenfassende Übersicht der wichtigsten Auto-ID-Verfahren[5]	17
2.2	European Article Number (EAN-13-Barcode)	18
2.3	Typische Architektur einer Speicherkarte mit Sicherheitslogik[5]	20
2.4	Typische Architektur einer Mikroprozessorkarte[5]	21
2.5	Aufbau eines RFID-Systems[5]	22
2.6	Zeiten eines Type-A Signals[7]	24
3.1	ESB magnetisch gekoppelte Leiterschleifen[5]	28
3.2	ESB RFID Transponder mit Spannungsquelle	28
3.3	Äquivalenz von realer Spannungs- und Stromquelle	29
3.4	ESB RFID Transponder mit Stromquelle	29
3.5	Äquivalente Spulenersatzschaltbilder	30
3.6	ESB RFID Transponder mit Stromquelle	31
3.7	ESB RFID Transponder Emulation	33
4.1	ISO-Turm Testanordnung[8]	35
4.2	ISO-Turm Antennen Seitenansicht[8]	36
4.3	Calibration Coil[8]	37
4.4	Reference PICC[8]	38
4.5	AWG-Setup Blockdiagramm	38
4.6	Zeiten eines Type-A Signals[7]	39
4.7	ESB Contactless Emulator	40
4.8	Contactless Emulator	40
4.9	Vom <i>.wav</i> über <i>.p40</i> zum Ausgangssignal	41
5.1	Hardwareablauf	42
5.2	Blockdiagramm Evaluation Board	43
5.3	AD9772A Blockdiagramm[2]	44
5.4	Schaltplan des AD9772 Evaluation Boards[2]	45

5.5	Ausgangsstufen vom AD9772 Evaluation Board[2]	47
5.6	Harmonic Images in the AD9772 D/A Converter	48
5.7	Eingangsverstärker mit einstellbarem Offset	50
5.8	Bodediagramm des Rekonstruktionsfilters	52
5.9	1. Stufe des Rekonstruktionsfilters	53
5.10	2. Stufe des Rekonstruktionsfilters	53
5.11	3. Stufe des Rekonstruktionsfilters	54
5.12	R_S vs. C_L des AD8056[1]	54
5.13	Butterworth Filter 5. Ordnung	55
5.14	Stromquelle	56
5.15	Stromquellen-Parallelschaltung	56
5.16	Widerstandsnetzwerk	58
5.17	Dual Interface Chip	60
5.18	Versorgungen	61
5.19	Einstellbare Versorgung für den Dual Interface Chip	63
5.20	Blockdiagramm Mehrkanalausführung	69
6.1	Spea - C320MX ¹	70
6.2	Comptestsystem[11]	72
6.3	LoadBoard L306[11]	74
7.1	Software Ablauf	75
7.2	Hilfe Ausgabe von <i>Tabor2Spea.exe</i>	76
7.3	<i>Tabor2Spea.exe</i> mit einer <i>.ini</i> -Datei aufgerufen	78
7.4	<i>Tabor2Spea.bat</i> Stapelverarbeitungsprogramm	79
7.5	<i>Tabor2Spea.exe</i> mit einer <i>.bat</i> -Datei aufgerufen	79
7.6	Flussdiagramm der Konvertierungssoftware <i>Tabor2Spea.exe</i>	80
7.7	Spea <i>Vector Explorer</i>	83
7.8	Spea <i>VectorView IDE</i>	83
7.9	Spea <i>Run Control Panel</i>	84
7.10	Messung am Spea Tester	84
E.1	Schaltplan	97
F.1	Layout Vorderseite	98
F.2	Layout Rückseite	99

Abbildungsverzeichnis

I.1	CL-Emulator	117
I.2	CL-Emulator	118
I.3	Calibration Coil	118
I.4	ISO-Turm	119
I.5	Reference PICC	119

Tabellenverzeichnis

2.1	Normen kontaktloser Chipkarten	23
2.2	Vergleich einiger RFID Standards	23
5.1	Konfiguration des AD9772 Evaluation Boards	46
5.2	Konfiguration des CL Evaluation Boards	64
5.3	Verbindung des CL Evaluation Board mit dem DA Evaluation Board	65
5.4	Verbindung des CL Evaluation Board mit dem Spea Load Board	65
5.5	Beschreibung der Messpunkte	66
5.6	Messergebnisse Versorgung	67
5.7	Messdaten der U/I-Wandler	67
D.1	Widerstandstabelle Seite 1/4	93
D.2	Widerstandstabelle Seite 2/4	94
D.3	Widerstandstabelle Seite 3/4	95
D.4	Widerstandstabelle Seite 4/4	96
J.1	Elektrische und magnetische Felder	120

1 Einleitung

In dieser Diplomarbeit wird eine Möglichkeit zur automatisierten Charakterisierung und Verifikation von RFID-Mikrochips behandelt.

1.1 Motivation

Der stetige Anstieg von Einsatzgebieten und Anwendungen der RFID Technik führte zur Entwicklung einer großen Palette unterschiedlicher RFID-Mikrochips. Durch die steigende Vielfalt und immer kürzer werdenden Entwicklungszeiten stieg auch der Bedarf an optimierteren Testverfahren. Eine gewisse Gruppe von Tests lässt sich aufgrund ihres Aufbaus nicht auf dem Wafer durchführen. Die Idee dieser Diplomarbeit war, diese Tests elektronisch nachzubilden, um schon im frühen Stadium der Wafertests Aussagen über die Funktionalität und Qualität der Chips treffen zu können. Der Vorteil dieser Wafertests ist natürlich auch die Möglichkeit der Parallelisierung, die enorm Zeit und somit auch Kosten spart. Hierfür wurde eine Mixed Signal Wafer Testmaschine der Firma Spea bei Infineon im Design Center Graz zur Verfügung gestellt. Die Aufgabe besteht nun darin eine Hardware zu entwickeln, die sich zwischen Testmaschine und Wafer befindet und die Chips mit den Signalen und Impedanzen versorgt, wie es der Chip in seiner natürlichen Anwendungsumgebung vorfindet.

1.2 Zielsetzung

Die Aufgabe der Diplomarbeit ist es den Spea-Tester um die Funktion einer Kontaktlos-Emulation zu erweitern, um Mikrochips mit kontaktloser Schnittstelle auf dem Wafer verifizieren und charakterisieren zu können. Hierzu muss die Übertragungstrecke zwischen Lesegerät und dem Chip nachgebildet werden.

- Im ersten Schritt wird das Konzept überprüft und in einem theoretischen Kontext mit mathematischen Methoden nachgewiesen, dass es aus theoretischer Sicht den bestehenden Messmethoden entspricht. Dabei wird von bekannten Grundgleichungen, wie den Maxwell'schen Gleichungen ausgegangen und auf das gewünschte Ergebnis hingerechnet. Hierbei werden natürlich einige Vereinfachungen durchgeführt um ein relativ kompaktes Ergebnis zu erreichen.
- In weiterer Folge wird eine Hardware entwickelt, die dem theoretischen Ergebnis entspricht. Dabei muss die Hardware entwickelt, dimensioniert und überprüft werden.
- Mit dieser Prototypenhardware werden Tests durchgeführt und ein Konzept für ein Mehrkanalmessverfahren entwickelt.
- Weiters muss die Hardware an den Wafertester angepasst werden um mit dem RFID-Chip Tests durchführen zu können. Dabei wird ein Prüfprogramm von Infineon an die gegebenen Bedürfnisse angepasst.
- Zu guter Letzt wird eine Konvertierungssoftware entwickelt, die Messsignale bereitstellt, die der Wafertester direkt verarbeiten kann.

1.3 Gliederung

- **Kapitel 1** beschreibt einleitend Motivation und Zielsetzung dieser Arbeit.
- **Kapitel 2** behandelt allgemeine Dinge und die Geschichte von RFID.
- **Kapitel 3** führt über mathematische Grundlagen zu dem theoretischen Konzept dieser Diplomarbeit.
- **Kapitel 4** beschäftigt sich mit den standardisierten Messungen, dem typischen Messaufbau bei Infineon und in Folge mit dem Messaufbau, der aus dieser Diplomarbeit hervor geht.
- **Kapitel 5** beschreibt die Hardware und ihre Dimensionierung.
- **Kapitel 6** gibt einen Überblick über den Wafertester und dessen Peripherie.
- **Kapitel 7** beschreibt den Software Aufbau und den Ablauf der Programme.
- **Anhang A bis C.1** beinhalten diverse Verzeichnisse wie Abkürzungsverzeichnis, Glossar, Literaturverzeichnis und Web links.

- **Anhang D** beinhaltet die Tabelle zur Konfiguration des Widerstandnetzwerkes.
- **Anhang E** und **F** zeigt Schaltplan und Layout der Hardware.
- **Anhang H** enthält den Quelltext der Konvertierungssoftware Tabor2Spea.exe, sowie einer Spea .p40 Template Datei und einer CL-Emulation.ini Datei.
- **Anhang I** zeigt einige Fotos der Hardwareaufbauten.
- **Anhang J** beinhaltet eine Zusammenstellung der wichtigsten Formeln für elektrische und magnetische Felder.

2 Allgemeines

RFID-Chips gehören zu der großen Gruppe von Automatischen Identifikationssystemen.

2.1 Automatische Identifikationssysteme

Schon seit dem Zweiten Weltkrieg werden Automatische Identifikationssysteme eingesetzt. Dort diente ein Sekundärradar zur Freund-Feind-Erkennung. Mittlerweile werden Auto-ID-Systeme im großen Stil für Bereitstellung von Informationen zu Personen, Tieren, Gütern und Waren verwendet. In der Abbildung 2.1 wird die Einteilung der Systeme nach der Art der Datenerfassung dargestellt.

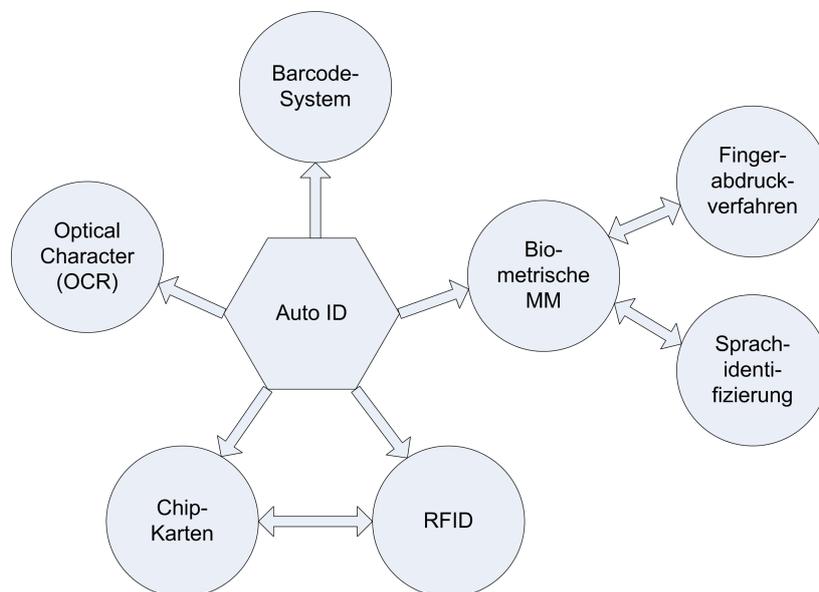


Abbildung 2.1: Zusammenfassende Übersicht der wichtigsten Auto-ID-Verfahren[5]

2.1.1 Barcode-Systeme

Barcode ist das in Europa am meisten verbreitete System zur Produktidentifikation. Es gibt beinahe kein Produkt auf dem Markt, das nicht durch einen Strichcode registriert wird. Der Barcode ist ein Binärcode aus einem Feld von parallel angeordneten breiten und schmalen Strichen mit definierten Lücken. In Abbildung 2.2 sieht man einen Strichcode der European Article Number, der nach der ISO/IEC 15420 genormt ist. Eine Weiterentwicklung ist der 2D-Code, bei dem die Fläche auf beiden Achsen als Informationsträger verwendet wird. Bei dem 3D-Code werden Farbton, Farbsättigung oder Farbhelligkeit als dritte Dimension verwendet. Im 4D-Code kommt noch die Zeit als Dimension dazu, es werden 3D-Codes hintereinander als Animation abgespielt.



Abbildung 2.2: European Article Number (EAN-13-Barcode)

2.1.2 Optical Character Recognition

In den 60er Jahren wurde damit begonnen Schrifttypen zu entwickeln, die sowohl von Menschen als auch von Maschinen gelesen werden können. Die Vorteile sind höhere Informationsdichte gegenüber dem Barcode und der Möglichkeit einer einfachen Kontrolle durch Menschen. OCR Verfahren werden hauptsächlich für Spezialanwendungen verwendet, da Lesegeräte wesentlich teurer sind als Barcode Lesegeräte.

2.1.3 Biometrische Verfahren

Die Biometrie beschäftigt sich mit Messungen an Lebewesen. Beim Menschen werden diese Messdaten zur Identifikation von Personen, zum Beispiel für Zutrittskontrollen in Hochsicherheitsbereichen, oder zur automatischen Kontrolle der Zugehörigkeit eines Reisepasses verwendet. Vor der Einführung von elektronischen Methoden wurde eine Personenidentifikation durch ein Kontrollorgan durchgeführt, das zum Beispiel das Gesicht mit dem Passfoto verglich.

Als biometrische Charakteristika werden unter anderem verwendet: [Wikipedia¹]

- Körpergröße (Anthropometrie)
- Iris (Regenbogenhaut)
- Retina (Augenhintergrund)
- Fingerabdruck (Fingerlinienbild)
- Gesichtsgeometrie
- Handgefäßstruktur / Venenstruktur
- Handgeometrie Handflächenscanner
- Handlinienstruktur
- Nagelbettmuster
- Zahnabdruck
- Stimme (nicht zu verwechseln mit Spracherkennung)
- Unterschrift (statisch, dynamisch, auch Handschrift)
- Tippverhalten auf Tastaturen (engl. keystroke dynamics)
- Lippenbewegung, meist im Zusammenhang mit Stimmerkennung (Klangfarbe)
- Gangstil (engl. automatic gait recognition)
- Körpergeruch
- DNA (mobiler DNA-Test, genetischer Fingerabdruck)

In der Praxis verwendete biometrische Erkennungsverfahren sind Sprachidentifikation, Finger- und Handabdruckverfahren, seltener auch Augen-Netzhaut-Identifizierung. Für die Tätererkennung wird auch der genetische Fingerabdruck immer interessanter.

¹<http://de.wikipedia.org/wiki/Biometrie>

2.1.4 Chipkarten

Als Chipkarte wird meistens eine Kunststoffkarte bezeichnet, die einen elektronischen Datenspeicher beinhaltet, der gegebenenfalls mit zusätzlicher Rechenleistung in Form eines Mikroprozessors ausgestattet ist. Solche Chipkarten werden als Bankomat-, Kredit- und Telefonkarten, Karte für Zutrittskontrollen, als SIM-Karte und für viele andere Anwendungen eingesetzt. Chipkarten wurden ursprünglich als kontaktbehaftete Karten ausgeführt, die in der ISO/IEC 7816 standardisiert und in dem *Handbuch der Chipkarten*[16] detailliert beschrieben sind. Über die Kontaktflächen wird der Chip mit Energie und Takt versorgt und über den I/O Kontakt eine bidirektionale serielle Kommunikation abgewickelt. Für deterministische Anfangszustände steht noch ein Resetkontakt zur Verfügung. Aktuell geht der Trend hin zu kontaktlosen Chipkarten, da Lesegeräte einfacher vor Sabotage und Vandalismus geschützt, gebaut werden können und es nicht zu Ausfällen durch Kontaktprobleme kommt. Es gibt auch Karten wo kontaktlose und kontaktbehaftete Schnittstellen, meistens für unterschiedliche Anwendungen, zur Verfügung stehen.

2.1.4.1 Speicherkarten

Speicherkarten arbeiten mit einer sequentiellen Logik zum Beispiel einer State Machine, mit welcher auf die Daten im Speicher zugegriffen wird. Sie sind wenig flexibel in ihrer Anwendung und in ihren Sicherheitsalgorithmen meistens einfach gelöst, was zur Reduktion der Chipfläche und somit zu einem günstigen Preis führt.

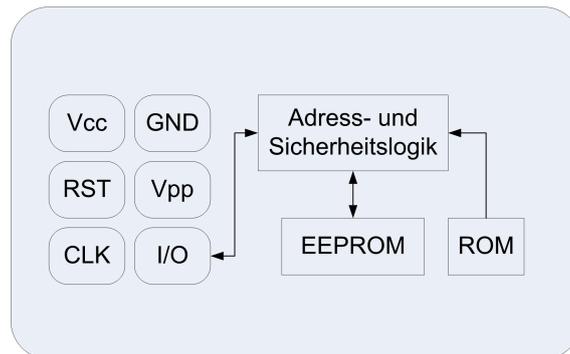


Abbildung 2.3: Typische Architektur einer Speicherkarte mit Sicherheitslogik[5]

2.1.4.2 Mikroprozessorkarten

Mikroprozessorkarten (Controller Cards) arbeiten mit integriertem Mikroprozessor, der mit einem sequenzierten Speicher verbunden ist

- ROM wird bei der Chipherstellung als Maske in Hardware aufgebaut und enthält das Betriebssystem, bei sehr großen Stückzahlen auf Kundenwunsch auch Applikationen. Inhalte können nicht mehr ohne Anfertigung neuer Masken, die sehr teuer sind, verändert werden.
- EEPROM enthält Applikationsdaten und kann im Betrieb verändert werden. Der Zugriff ist nur über das Betriebssystem möglich.
- RAM ist der temporäre Arbeitsspeicher, dessen Inhalt nach dem Abschalten der Versorgungsspannung verloren geht.

Mikroprozessorkarten sind sehr flexibel und können mehrere Applikationen beinhalten. Es lassen sich auch sehr aufwendige Sicherheitsalgorithmen realisieren, wie RSA- oder ECC-Verschlüsselung.

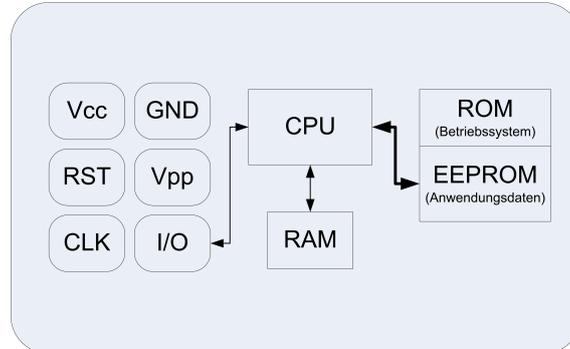


Abbildung 2.4: Typische Architektur einer Mikroprozessorkarte[5]

2.1.5 RFID-Systeme

RFID steht als Abkürzung für Radio Frequency Identification und würde man als Identifizierung mit Hilfe von elektromagnetischen Wellen ins Deutsche übersetzen. In den letzten Jahrzehnten sind eine große Vielzahl von verschiedenen RFID-Systemen entwickelt worden, auf die nicht genauer eingegangen werden kann. Manche dieser Systeme arbeiten mit elektrischen bzw. magnetischen Feldern oder elektromagnetische Wellen. RFID-Systeme gibt in den unterschiedlichsten Frequenzbereichen, von LF, wie zum Beispiel das System mit 135 kHz, bis zu Mikrowellensystemen im GHz Bereich. Außerdem wird noch unterschieden, ob es ein aktiver oder ein passiver Transponder ist. Aktive Transponder führen eine Versorgung mit sich und können dadurch höhere Reichweiten erreichen. Passive beziehen ihre Energie aus dem Feld des Lesegeräts. Wir beschränken uns hier auf ein sehr verbreitetes System auf 13,56 MHz im HF-Bereich. Dieses System arbeitet mit magnetischer Kopplung, wobei der Transponder seine Energie und den Takt aus dem HF-Feld bezieht. Diese RFID-Systeme sind im Kern den kontaktbasierten Chipkarten sehr ähnlich. Die großen Unterschiede bestehen in der Übertragung der Daten und der Energieversorgung. Es gibt auch Chipkarten bei denen kontaktbehaftete und kontaktlose Schnittstellen vorhanden sind, die so genannten Dualinterface Chipkarten. In Abbildung 2.5 ist der Aufbau eines RFID-Systems mit kontaktlosen Chipkarten zu sehen, sie bestehen aus einem Lesegerät und dem Transponder.

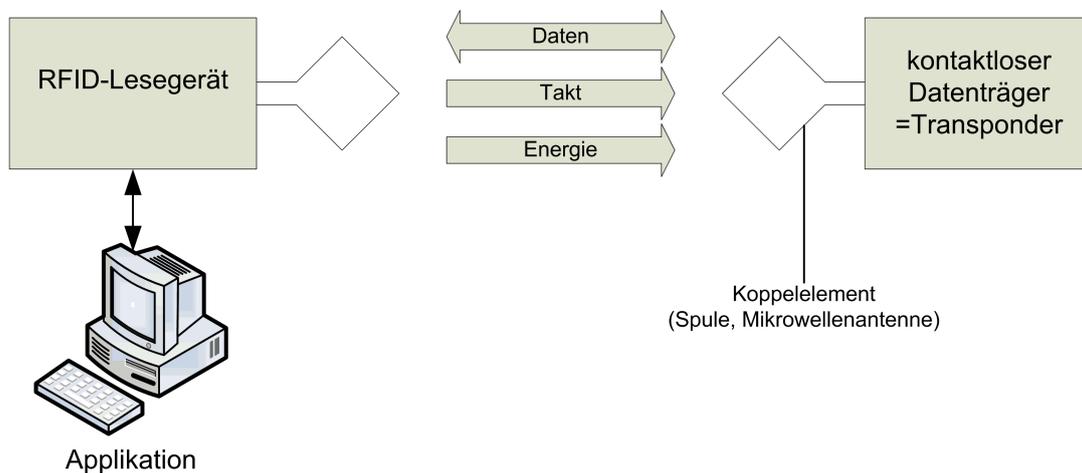


Abbildung 2.5: Aufbau eines RFID-Systems[5]

2.1.5.1 Kontaktlose Chipkarten

Die kontaktlosen Chipkarten werden abhängig von ihrer Reichweite in verschiedenen Normen definiert. Die Tabelle 2.1 gibt einen Überblick dieser Normen. Der Close-coupling

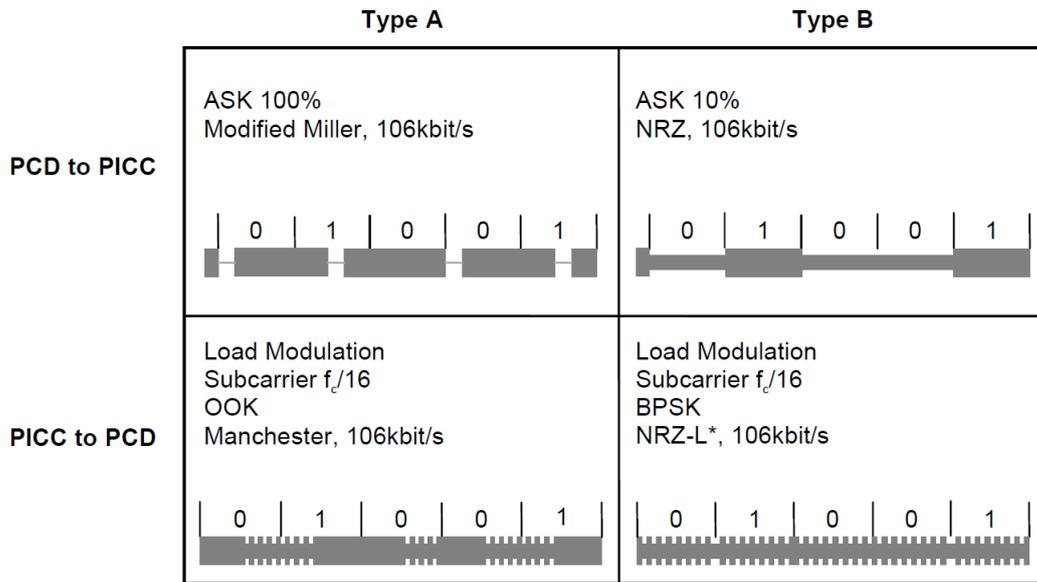
Norm	Kartentyp	Richtwert für die Reichweite
ISO/IEC 10536	Close coupling	0 ... 1 cm
ISO/IEC 14443	Proximity coupling	0 ... 10 cm
FeliCa	Proximity coupling	0 ... 15 cm
ISO/IEC 15693	Vicinity coupling	0 ... 1 m

Tabelle 2.1: Normen kontaktloser Chipkarten

Standard ist der älteste und schon obsolet, da er kaum Vorteile gegenüber kontaktbehafteten Chipkarten bringt. Chipkarten aus dem Bereich Proximity coupling sind in der ganzen Welt sehr verbreitet und werden als Testobjekte in dieser Diplomarbeit verwendet. Vicinity coupling Chipkarten haben eine hohe Reichweite, können aber nur mit geringen Datenraten kommunizieren. Außerdem fehlt ihnen die Energie um Controller mit höherer Rechenleistung zu versorgen. Beim Proximity coupling System haben sich drei Standards für die Kommunikation durchgesetzt. Type A und Type B sind in der ISO/IEC 14443 definiert. FeliCa ist ein von Sony entwickelter Standard, wobei man hier eher von Richtlinien für Hersteller sprechen kann, als von einem Standard. In der Abbildung 2.6 sind die aus der ISO/IEC 14443-2[7] vorgegebenen Modulationsarten bei 106 kbit/s zu sehen. In der Tabelle 2.2 werden einige RFID Standards für 13,56 MHz miteinander verglichen, dabei sind Feldstärke, Reichweite, Modulationsart, Codierung und Datenrate von Bedeutung.

Type	ISO/IEC 14443		FeliCa	ISO/IEC 15693	
	Type A	Type B		Vicinity	
Feldstärke	1,5 ... 7,5 A/m	1,5 ... 7,5 A/m	0,15 ... 12 A/m	0,15 ... 5 A/m	
Abstand	<10 cm	<10 cm	<15 cm	<1 m	
Datenflussrichtung: Reader ⇒ Transponder					
Modulation	ASK 100%(106) <60%(212-848)	ASK 10%	ASK 10%	ASK 10% oder 100%	
Codierung	Modified Miller	NRZ	Manchester	256 PPM, 4 PPM	
Datenrate	106, 212, 424, 848 kbit/s		212, 424 kbit/s	1,65 und 26,48 kbit/s	
Datenflussrichtung: Transponder ⇒ Reader					
Hilfsträger	847,5 kHz		nicht verwendet	423,75 kHz	424, 484 kHz
Codierung	Manchester(106), BPSK(212-848)	NRZ-L & BPSK	Manchester	Manchester	FSK
Datenrate	106, 212, 424, 848 kbit/s		212, 424 kbit/s	6,62 kbit/s	6,67 kbit/s
				26,48 kbit/s	26,69 kbit/s

Tabelle 2.2: Vergleich einiger RFID Standards



* Inversion of data is also possible

Abbildung 2.6: Zeiten eines Type-A Signals[7]

Mittlerweile etablieren sich auch neue Standards, die so genannten VHBR (Very High Bit Rate) Standards, die Datenraten von 1,695 Mbit/s, 3,39 Mbit/s und 6,78 Mbit/s ermöglichen.

3 Theorie

3.1 Physikalische Grundlagen

Die mathematische Grundlage für elektrische und magnetische Felder, die Abstrahlung von elektromagnetischen Wellen und somit auch die Energie- und Datenübertragung über diese, bilden die Maxwell'schen Gleichungen. Eine Gegenüberstellung von wichtigen Gleichungen und Einheiten, die elektrische und magnetische Felder beschreiben, befindet sich im Anhang J. Von den vier Maxwell'schen Gleichungen sind in dem von uns verwendeten RFID-System im Speziellen zwei Gleichungen wichtig, das Durchflutungsgesetz und das Induktionsgesetz, da dieses System auf der Übertragung von Energie und Daten über magnetische Kopplung basiert. Dabei benötigen wir das Durchflutungsgesetz um die Erzeugung des magnetischen Feldes in der Antenne des Lesegerätes und das Induktionsgesetz um die induzierte Spannung in der Antenne des Transponder zu beschreiben.

3.1.1 Durchflutungsgesetz (Ampèresches Gesetz)

Jedes zeitlich veränderliche elektrische Feld oder ein elektrischer Strom erzeugt ein magnetisches Wirbelfeld. Hierbei wird unterschieden zwischen der Leitungsstromdichte \vec{j}_l , die den Stromfluss in einem Leiter beschreibt und der Verschiebungsstromdichte, die eine zeitliche Änderung der elektrischen Flussdichte \vec{D} bedeutet. In der Formel 3.1 wird das Durchflutungsgesetz in seiner allgemeinen Form in differentieller Schreibweise links und in integraler Schreibweise rechts dargestellt.

$$\vec{\nabla} \times \vec{H} = \vec{j}_l + \frac{\partial \vec{D}}{\partial t} \quad \Leftrightarrow \quad \oint_{\partial A} \vec{H} \cdot d\vec{s} = \iint_A (\vec{j}_l \cdot d\vec{A}) + \frac{d}{dt} \left(\iint_A \vec{D} \cdot d\vec{A} \right) \quad (3.1)$$

Das Ampèresche Gesetz von André Marie Ampère, kannte die Verschiebungsstromdichte noch nicht und beschränkt sich somit auf das durch einen stromdurchflossenen Leiter erzeugte Magnetfeld.

Das Umlaufintegral der magnetischen Feldstärke längs einer geschlossenen Kurve ist gleich der Summe der Stromstärken der eingeschlossenen Ströme.[5]

$$\sum I = \oint \vec{H} \cdot \vec{ds} \quad (3.2)$$

Bei magnetisch gekoppelten RFID-Systemen werden häufig runde Antennen mit ein oder mehreren Leiterschleifen für Lesegeräte verwendet, so genannte kurze Zylinderspulen. Die Formel 3.3 [5] beschreibt den Feldstärkenverlauf einer solchen Antenne entlang der Spulenachse x , wobei N die Anzahl der Windungen, R der Kreisradius und I der Strom sind.

$$H = \frac{I \cdot N \cdot R^2}{2\sqrt{(R^2 + x^2)^3}} \quad (3.3)$$

Diese Gleichung gilt nur unter der Bedingung, dass die Spule sehr viel kürzer als ihr Radius ist und der Abstand x nur im Nahfeld betrachtet wird. Nahfeld ist gegeben, wenn $x < \lambda/2\pi$ ist. Da das verwendete RFID-System eine sehr große Wellenlänge von $\lambda = 22,1 \text{ m}$ hat, können wir bei den Betrachtungen in dieser Diplomarbeit immer von einem Nahfeld ausgehen.

$$\lambda = \frac{c}{f} = \frac{299792458 \frac{\text{m}}{\text{s}}}{13,56 \text{ MHz}} = 22,1 \text{ m} \quad (3.4)$$

Weitere wichtige Größen bei magnetischen Feldern sind der magnetische Fluss und die magnetische Flussdichte. Als magnetischen Fluss Φ bezeichnet man die Gesamtzahl der magnetischen Feldlinien, die den Spuleninnenraum durchsetzen. Bei N Windungen spricht man von dem verketteten Fluss Ψ . Wenn jede Leiterschleife mit gleichem magnetischem Flussanteil Φ zum gesamten Fluss Ψ beiträgt, ist der verkettete Fluss N mal so groß als der Fluss einer Leiterschleife. Die magnetische Flussdichte B ist der Fluss bezogen auf eine Fläche A .

$$B = \mu \cdot H \quad \Phi = B \cdot A \quad \Psi = \Phi \cdot N \quad (3.5)$$

Eine weitere Größe, die angibt wie viel verketteter Fluss mit einer Spule und einer bestimmten Stromstärke erzeugt werden kann, ist die Induktivität L . Diese ist nur von der Geometrie der Spule und der Permeabilität des Spulenkörpers abhängig.

$$L = \frac{\Psi}{I} = \frac{N \cdot \Phi}{I} = \frac{N \cdot \mu \cdot H \cdot A}{I} \quad (3.6)$$

Befindet sich in der Nähe einer stromdurchflossenen Leiterschleife eine 2. Leiterschleife so tritt ein Teil dieses Flusses, der so genannte Koppelfluss Ψ_{21} durch diese durch. Dieser Koppelfluss bezogen auf den Strom der 1. Leiterschleife ergibt die Gegeninduktivität M_{21} . Diese Gegeninduktivität ist so wie die Eigeninduktivität nur von geometrischen und permeablen Eigenschaften abhängig. Genau wie bei der Gegeninduktivität M_{21} kann ein

Strom in der 2. Leiterschleife einen Koppelfluss Ψ_{12} erzeugen, der zu einer Gegeninduktivität M_{12} führt. Diese beiden Gegeninduktivitäten sind laut dem Umkehrsatz gleich groß.

$$M_{21} = \frac{\Psi_{21}}{I_1} \quad M = M_{12} = M_{21} \quad M_{12} = \frac{\Psi_{12}}{I_2} \quad (3.7)$$

Eine qualitative Aussage, wie gut zwei Leiterschleifen miteinander verkoppelt sind, gibt der Kopplungsfaktor k , der bei völliger Kopplung zu 1 und bei keiner Kopplung zu 0 wird.

$$k = \frac{M}{\sqrt{L_1 \cdot L_2}} \quad (3.8)$$

3.1.2 Induktionsgesetz (Faradaysches Gesetz)

Jedes zeitlich veränderliche magnetische Feld erzeugt ein elektrisches Wirbelfeld. In der Formel 3.9 wird das Induktionsgesetz, auch nach seinem Entdecker Michael Faraday als Faradaysches Gesetz bezeichnet, in seiner allgemeinen Form in differentieller Schreibweise links und in integraler Schreibweise rechts dargestellt.

$$\vec{\nabla} \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \quad \Leftrightarrow \quad \oint_{\partial A} \vec{E} \cdot \vec{ds} = -\frac{d}{dt} \left(\iint_A \vec{B} \cdot \vec{dA} \right) \quad (3.9)$$

Daraus ergibt sich für eine Leiterschleife in einem homogenen Magnetfeld, dessen Flächenvektor in die gleiche Richtung wie die magnetische Flussdichte zeigt, die folgende induzierte Spannung u_{ind} .

$$u_{ind} = -\frac{d}{dt}(A \cdot B) = -\frac{d}{dt}(A \cdot \mu \cdot H) \quad (3.10)$$

Da die Fläche A und die Permeabilität μ keine zeitabhängigen Größen sind, entsteht bei einer zeitlich veränderlichen magnetischen Feldstärke die Formel 3.11. Für die Feldstärke H wird $\hat{H} \cdot \sin(\omega t)$ eingesetzt, wobei \hat{H} der Scheitelwert ist.

$$u_{ind} = -A \cdot \mu \cdot \frac{d(\hat{H} \cdot \sin(\omega t))}{dt} = -A \cdot \mu \cdot \omega \cdot \hat{H} \cdot \cos(\omega t) \quad (3.11)$$

$$\hat{u}_{ind} = A \cdot \mu \cdot 2 \cdot \pi \cdot f \cdot \hat{H} \quad (3.12)$$

3.2 RFID-Kopplung

Bei RFID-Systemen im LF und HF Bereich kann bei der Energie- und Datenübertragung von einer Transformatorkopplung gesprochen werden, wobei von einer schlechten Kopplung

und somit einer großen Streuinduktivität ausgegangen wird. In der Abbildung 3.1 ist das Ersatzschaltbild einer RFID-Koppelstrecke mit Chip dargestellt. Die Spule L_R ist die Sendeantenne des Lesegeräts und bildet mit der Spule L , die Antenne des RFID-Tags, einen lose gekoppelten Transformator.

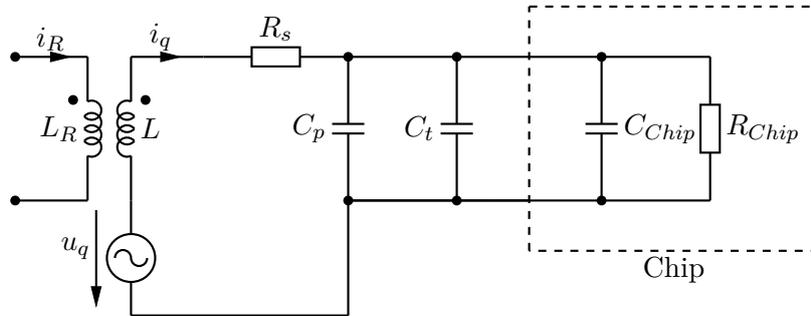


Abbildung 3.1: ESB magnetisch gekoppelte Leiterschleifen[5]

Die Spannung u_q stellt in diesem Ersatzschaltbild die induzierte Spannung dar. Die Transponderantenne mit der Induktivität L bildet mit der Windungskapazität C_p , der Trimmkapazität C_t und der Chipkapazität C_{Chip} einen Parallelschwingkreis. Dieser Schwingkreis führt zu einer Spannungsüberhöhung bei Resonanzfrequenz, deren Effekt sich positiv bei der Energieübertragung auswirkt. Der Widerstand R_s ist der elektrische Widerstand der Transponderantenne und ist neben dem Chipwiderstand R_{Chip} hauptverantwortlich für die Transpondergüte. Bei der Gesamtgüte im Betrieb ist natürlich auch der Chipwiderstand R_{Chip} zu berücksichtigen, der sich durch die Shuntwirkung, Lastmodulation und Laständerung im Digitalteil ständig ändert.

In den folgenden Schritten wird diese Ersatzschaltung auf die wichtigsten Komponenten, die für eine einfache Nachbildung nötig sind, reduziert. Dieser Vorgang erfolgt schrittweise bis bei bestimmten Bedingungen eine Quelle mit Widerstand über bleibt.

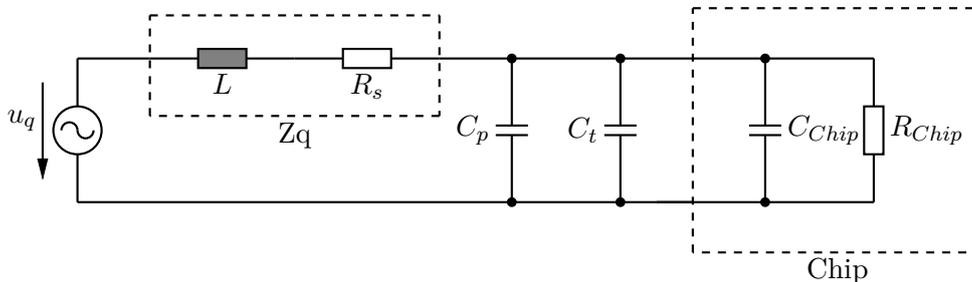


Abbildung 3.2: ESB RFID Transponder mit Spannungsquelle

Im Ersatzschaltbild 3.2 ist der RFID-Tag ohne Sendeantenne mit der induzierten Span-

nung als u_q dargestellt. Durch die Äquivalenz von realer Spannungs- und Stromquelle kann die Spannungsquelle durch eine Stromquelle ersetzt werden, dabei wird Z_q als Innenwiderstand gesehen. In der Gleichung 3.13 wird die Klemmenspannung für eine reale Spannungs- und Stromquelle berechnet. Es ist ersichtlich, dass die Quellen mit gleichem Innenwiderstand ersetzbar sind.

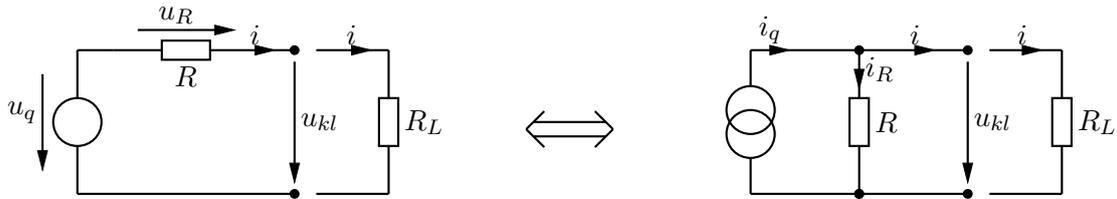


Abbildung 3.3: Äquivalenz von realer Spannungs- und Stromquelle

$$\begin{aligned}
 u_{kl} &= u_q \frac{R_L}{R + R_L} = i(R + R_L) \frac{R_L}{R + R_L} = i \cdot R_L \\
 u_{kl} &= i_q \frac{R \cdot R_L}{R + R_L} = (i + i_R) \frac{R \cdot R_L}{R + R_L} = \frac{i \cdot R \cdot R_L + i_R \cdot R \cdot R_L}{R + R_L} \\
 &= \frac{i \cdot R \cdot R_L + u_{kl} \cdot R_L}{R + R_L} = \frac{i \cdot R \cdot R_L + i \cdot R_L \cdot R_L}{R + R_L} = \frac{i \cdot R_L (R + R_L)}{R + R_L} = i \cdot R_L
 \end{aligned} \quad (3.13)$$

In der Abbildung 3.4 wurde die Spannungsquelle durch eine Stromquelle ersetzt.

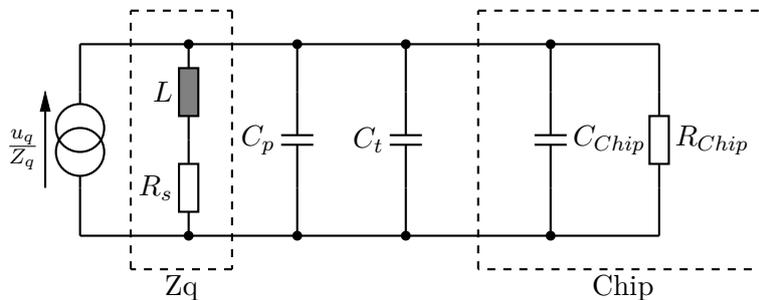


Abbildung 3.4: ESB RFID Transponder mit Stromquelle

Die Spule L bildet mit dem ohmschen Anteil des Spulendrahtes R_s und der Wicklungskapazität C_p das Ersatzschaltbild einer realen Spule. Da mit dieser Anordnung nicht immer leicht zu rechnen ist, kann der Spulenwiderstand R_s in einen äquivalenten Parallelwiderstand R_p übergeführt werden, was in der Abbildung 3.5 dargestellt ist.

Damit diese Äquivalenz gewährleistet ist, müssen zwei Bedingungen erfüllt werden. Die Beträge der Impedanzen und der Phasenwinkel zwischen Blind- und Wirkwiderstand müssen

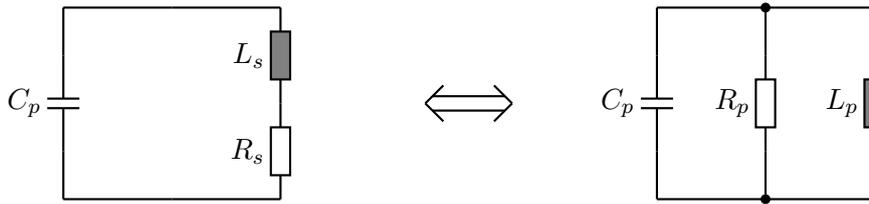


Abbildung 3.5: Äquivalente Spulenersatzschaltbilder

gleich bleiben.

$$|Z_s| = |Z_p| = \frac{1}{|Y_p|} \quad \varphi_s = \varphi_p \Rightarrow \tan\varphi_s = \tan\varphi_p$$

$$\sqrt{R_s^2 + X_s^2} = \frac{1}{\sqrt{G_p^2 + B_p^2}} \quad \frac{X_s}{R_s} = \frac{B_p}{G_p}$$

$$R_s^2 + X_s^2 = \frac{1}{\frac{1}{R_p^2} + \frac{1}{X_p^2}} \quad \frac{X_s}{R_s} = \frac{R_p}{X_p}$$

$$R_s^2 + X_s^2 = \frac{1}{\frac{1}{R_p^2} + \frac{X_s^2}{R_p^2 \cdot R_s^2}} = \frac{R_p^2 \cdot R_s^2}{R_s^2 + X_s^2} \Rightarrow$$

$$R_p^2 = \frac{(R_s^2 + X_s^2)^2}{R_s^2} \Rightarrow R_p = \frac{R_s^2 + X_s^2}{R_s} \quad (3.14)$$

$$R_s^2 + X_s^2 = \frac{1}{\frac{R_s^2}{X_p^2 \cdot X_s^2} + \frac{1}{X_p^2}} = \frac{X_p^2 \cdot X_s^2}{R_s^2 + X_s^2} \Rightarrow$$

$$X_p^2 = \frac{(R_s^2 + X_s^2)^2}{X_s^2} \Rightarrow X_p = \frac{R_s^2 + X_s^2}{X_s} \quad (3.15)$$

Mit den Formeln 3.14 und 3.15 können der äquivalente parallele Spulenwirkwiderstand und der äquivalente Spulenblindwiderstand errechnet werden. Mit den typischen Werten bei RFID-Transpondern kann davon ausgegangen werden, dass R_s kleiner als X_s ist, wodurch sich folgende Vereinfachungen ergeben.

$$R_s \ll X_s \quad \Rightarrow \quad R_p = \frac{X_s^2}{R_s} \quad X_p = X_s \quad (3.16)$$

In der Abbildung 3.6 wurde der Serienwiderstand R_s durch einen Parallelwiderstand R_p ersetzt. Weiters werden L , R_p und C_p zu einem Parallelschwingkreis mit der Impedanz Z_i zusammengefasst. Für diese Parallelschaltung empfiehlt es sich eher mit Leitwerten

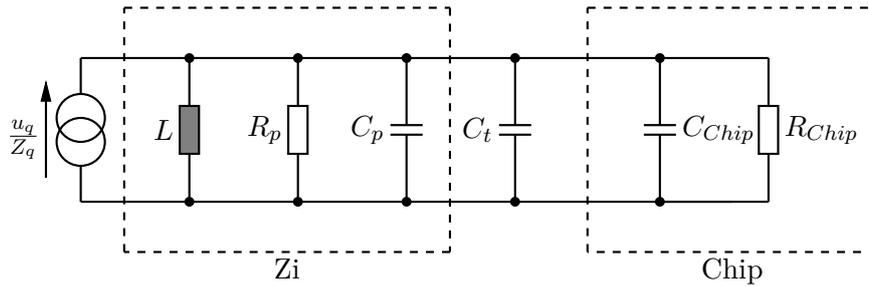


Abbildung 3.6: ESB RFID Transponder mit Stromquelle

zu rechnen, da sie so einfach addiert werden können. Der Scheinleitwert für den Parallelschwingkreis wird in der Formel 3.17 berechnet. Für den Wirkleitwert G_p werden im Folgeschritt die uns bekannten Bauteilwerte aus der Formel 3.16 eingesetzt.

$$\begin{aligned} \underline{Y}_i &= G_p + j\omega C_p + \frac{1}{j\omega L} = \frac{1}{R_p} + j\left(\omega C_p - \frac{1}{\omega L}\right) \\ &= \frac{R_s}{X_s^2} + j\left(\omega C_p - \frac{1}{\omega L}\right) = \frac{R_s}{(\omega L)^2} + j\left(\omega C_p - \frac{1}{\omega L}\right) \end{aligned} \quad (3.17)$$

Hier ist ein kleiner Abstecher zur Schwingkreistheorie[14] nötig. In elektrischen Schaltungen gibt es zwei Typen von Schwingkreisen, den Serienschwingkreis und den Parallelschwingkreis. Sie haben die Grundeigenschaft, dass zwei Energiespeicher abwechselnd ihre gespeicherte Energie austauschen. Die Spule speichert die Energie im magnetischen Feld und versucht dabei den Stromfluss aufrecht zu erhalten. Der Kondensator speichert die Energie im elektrischen Feld und versucht die Spannung aufrecht zu halten, dabei kommt es zum periodischen Energieaustausch. Im Resonanzfall werden die Blindwiderstände des Serienschwingkreis und die Blindleitwerte des Parallelschwingkreis zu null, somit wird ein Serienschwingkreis niederohmig und ein Parallelschwingkreis hochohmig. Weiters kommt es bei Schwingkreisen zu Resonanzüberhöhungen. Der Serienschwingkreis bildet eine Spannungsüberhöhung, der Parallelschwingkreis eine Stromüberhöhung. Dabei erreichen Spannungen und Ströme im Kreis höhere Werte als die der angelegten Quelle. Reale Schwingkreise sind durch Verluste wie ohmsche Verluste, dielektrische Verluste oder Abstrahlung gedämpft. Der Gütefaktor Q ist ein Maß für die Spannungs- und Stromüberhöhung im Schwingkreis bei Resonanzfrequenz.

Da es sich beim Ersatzschaltbild 3.6 um einen Parallelschwingkreis handelt, führen wir den Kennleitwert Y_0 ein, der dem Leitwert bei der Kennkreisfrequenz entspricht. Die Kennkreisfrequenz ω_0 entspricht der Frequenz, bei der die Suszeptanz des Parallelschwingkreises null wird. Diese Resonanzfrequenz kann mit der Thomsonschen Schwingungsgleichung

berechnet werden. Da Schwingkreise mit erzwungener Schwingung eine von außen eingeprägte periodische Erregergröße besitzen, deren Frequenz eine andere ist als die Eigenresonanzfrequenz, ist es von Vorteil die Frequenz zu normieren. Dies geschieht mit der normierten Frequenz Ω . Eine zusätzliche Erleichterung der formalen Darstellung ermöglicht die Verstimmung ν . All diese Vereinfachungen sind in der Formel 3.18 definiert.

$$\begin{aligned}
 Y_0 = \omega_0 C_p = \frac{1}{\omega_0 L} &\Rightarrow C_p = \frac{Y_0}{\omega_0} \quad L = \frac{1}{\omega_0 \cdot Y_0} & (3.18) \\
 \omega_0 = \frac{1}{\sqrt{L \cdot C}} &\Rightarrow Y_0 = \sqrt{\frac{C}{L}} \\
 \Omega = \frac{\omega}{\omega_0} &\Rightarrow \nu = \frac{\omega}{\omega_0} - \frac{\omega_0}{\omega} = \Omega - \frac{1}{\Omega}
 \end{aligned}$$

Bei den RFID Systemen auf 13,56 MHz ist es üblich, dass die Transponder Resonanzfrequenz höher liegt als die des Lesegeräts. Der Grund für die Verstimmung nach oben liegt in der Natur, dass mehrere Transponder im RFID-Feld eine Gesamtresonanzfrequenz haben, die unter der jedes einzelnen liegt.

In den Gleichungen 3.19 werden nun Kennleitwert, relative Frequenz und Verstimmung eingesetzt und ergeben somit eine recht übersichtliche Formel für die Admittanz.

$$\begin{aligned}
 \underline{Y}_i = R_s Y_0^2 \left(\frac{\omega_0}{\omega} \right)^2 + j Y_0 \left(\frac{\omega}{\omega_0} - \frac{\omega_0}{\omega} \right) &= & (3.19) \\
 R_s Y_0^2 \frac{1}{\Omega^2} + j Y_0 \left(\Omega - \frac{1}{\Omega} \right) &= \\
 R_s Y_0^2 \frac{1}{\Omega^2} + j Y_0 \nu &
 \end{aligned}$$

Eine übliche Größe, die bei Schwingkreisen angegeben wird und in RFID-Systemen häufig verwendet wird, ist die Güte. Eine Güte, die hier nützlich sein kann, ist die Spulengüte Q_L , da sie den größten Anteil an Verlusten bildet. Sie ist definiert als Verhältnis von der in der Spule maximal gespeicherten Gesamtenergie bezogen auf die in Wärme umgesetzte Energie. Daraus ergibt sich für die Güte eine recht einfache Formel 3.20, bei der der Blindwiderstand durch den Wirkwiderstand dividiert wird.

$$Q_L = \frac{X_s}{R_s} = \frac{\omega L}{R_s} \quad (3.20)$$

In die Gleichungen von vorher eingesetzt ergibt dies die Gleichung 3.21 für die Admittanz.

$$\underline{Y}_i = \frac{1}{\omega L Q_L} + j Y_0 \nu \quad (3.21)$$

Jetzt wird noch der Kennleitwert ersetzt, da er aus der ursprünglichen Schaltung nicht zur Verfügung steht.

$$\underline{Y}_i = \frac{1}{\omega L Q_L} + j \frac{\nu}{\omega_0 L} \quad (3.22)$$

Die Impedanz Z_i ergibt sich aus dem Kehrwert der Admittanz.

$$\underline{Z}_i = \frac{1}{\underline{Y}_i} = \frac{1}{\frac{1}{\omega L Q_L} + j \frac{\nu}{\omega_0 L}} \quad (3.23)$$

Der letzte Schritt in der Vereinfachung ist die Impedanz Z_i durch einen ohmschen Widerstand R_i zu ersetzen, was natürlich nur bei einer Frequenz gültig ist.

$$R_i = |Z_i| = \frac{1}{\sqrt{\left(\frac{1}{\omega L Q_L}\right)^2 + \left(\frac{\nu}{\omega_0 L}\right)^2}} \quad (3.24)$$

Für diesen Schritt wird vorausgesetzt, was eigentlich schon beim Einführen von Kennleitwert und Verstimmung geschehen ist, dass alle Kapazitäten wie Antennenkapazität, Trimmkapazität und Chipkapazität zu C zusammengefasst werden. Übrig bleibt ein Ersatzschaltbild mit einer idealen Stromquelle und zwei parallel geschalteten Widerständen, oder ersatzweise eine ideale Spannungsquelle mit einem Spannungsteiler. Das Ersatzschaltbild ist in Abbildung 3.7 zu sehen.

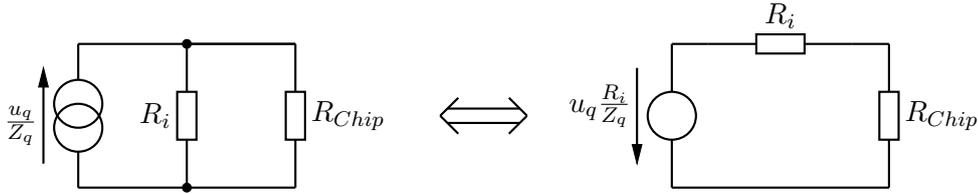


Abbildung 3.7: ESB RFID Transponder Emulation

Für diese einfachen Anordnungen kann für den Chip der Strom über die Stromteilerformel 3.25, oder die Spannung über die Spannungsteilerformel 3.26 berechnet werden.

$$i_{Chip} = \frac{u_q}{Z_q} \cdot \frac{G_{Chip}}{G_{Chip} + |Y_i|} \quad (3.25)$$

$$u_{Chip} = u_q \frac{R_i}{Z_q} \cdot \frac{R_{Chip}}{R_{Chip} + R_i} \quad (3.26)$$

Eine Größe, die noch nicht berechnet wurde, ist der Strom i_q der Stromquelle, oder die Spannung der Spannungsquelle. Berechnet werden diese durch das Induktionsgesetz 3.12 aus Kapitel 3.1.2

$$u_q = A \cdot \mu \cdot 2 \cdot \pi \cdot f \cdot H = A \cdot \mu \cdot \omega \cdot H \quad (3.27)$$

Teile dieser Gleichung können zu einem Faktor, den so genannten Spulenfaktor k_s , zusammengefasst werden.

$$u_q = k_s \cdot H \quad (3.28)$$

Die Fläche A ist die Summe der Flächen aller Windungen. Gibt es zum Beispiel N Windungen und würde jede Windung die gleiche Fläche beinhalten, könnte das für eine rechteckige Antenne mit der Länge l und der Breite b eine Gesamtfläche von $A = N \cdot l \cdot b$ ergeben. Zur Berechnung der Kreisfrequenz ω wird die Arbeitsfrequenz des Lesegeräts eingesetzt, das für $\omega = 2 \cdot \pi \cdot 13,56 \text{ MHz} = 85,2 \cdot 10^6 \text{ s}^{-1}$ ergibt.

$$k_s = A \cdot \mu \cdot \omega \quad (3.29)$$

Dies kann jetzt noch in die Gleichung für den Quellstrom einfließen und ergibt die Gleichung 3.30

$$i_q = \frac{u_q}{Z_q} = \frac{k_s \cdot H}{Z_q} \quad (3.30)$$

Für die Anwendung im Kapitel 5 ist eher der Betrag des Quellstroms von Interesse, der voll ausformuliert in der Formel 3.31 zu sehen ist.

$$|i_q| = \frac{k_s \cdot H}{\sqrt{R_s^2 + (\omega L)^2}} \quad (3.31)$$

4 Messverfahren

4.1 Testmethoden nach Norm

In der ISO/IEC 10373-6[8] mit dem Titel *Identification cards - Test methods - Part 6: Proximity cards* sind Testmethoden zum Messen der Reader - Transponder Verbindung standardisiert. Diese Testmethoden unterscheiden sich kaum von denen, wie sie für Vicinity cards in der Norm ISO/IEC 10373-7[9] verwendet werden. Somit lassen sich diese Messungen relativ leicht auch auf Vicinity Karten erweitern.

4.1.1 ISO-Turm

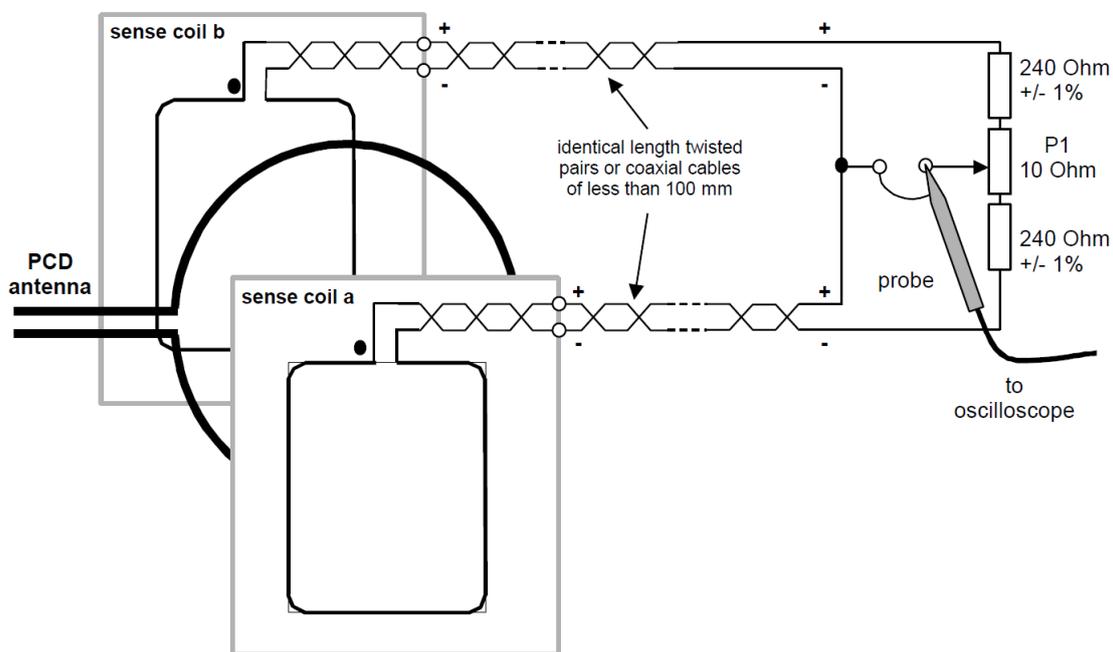


Abbildung 4.1: ISO-Turm Testanordnung[8]

Die Bezeichnung ISO-Turm oder auch ISO-Messturm wird umgangssprachlich für eine Messanordnung verwendet, wie sie in ISO/IEC 10373-6[8] Kapitel 6 beschrieben wird. Ein Foto I.4 von dem Aufbau gibt es im Anhang I. Mit der Abbildung 4.1 ist die Testanordnung schematisch dargestellt. In der Mitte befindet sich die PCD Antenne, an die entweder ein Lesegerät oder ein AWG (Arbitrary Waveform Generator) mit Leistungsverstärker über ein Anpassnetzwerk angeschlossen wird. Parallel dazu befinden sich zwei Sense Coils im gleichen Abstand zur PCD (Proximity Card Device) Antenne, da so das gleiche Träger-signal, aber jeweils gegenphasig abgegriffen werden kann. Sie sind mit einem verdrehten Aderpaar über ein abstimmbares Widerstandsnetzwerk verbunden. Mit dem Potentiometer P1 können so die 2 gegenphasigen Signale abgeglichen werden, damit der Träger um mehr als 40dB an dem Probe Anschluss unterdrückt wird. Die Antwort der PICC (Proximity Integrated Chip Card) wird somit sehr gut am Oszilloskop dargestellt, da sich das DUT (Device Under Test) nur an einer Sense Coil befindet und sich die Antwort der PICC nicht auslöscht. Eine Seitenansicht des ISO-Turms stellt die Abbildung 4.2 zur Verfügung, wo auch zu sehen ist, dass sich das DUT und die Calibration Coil auf jeweils einer gegenüberliegenden Sense Coil befinden. Alle Antennen befinden sich auf der selben Achse.

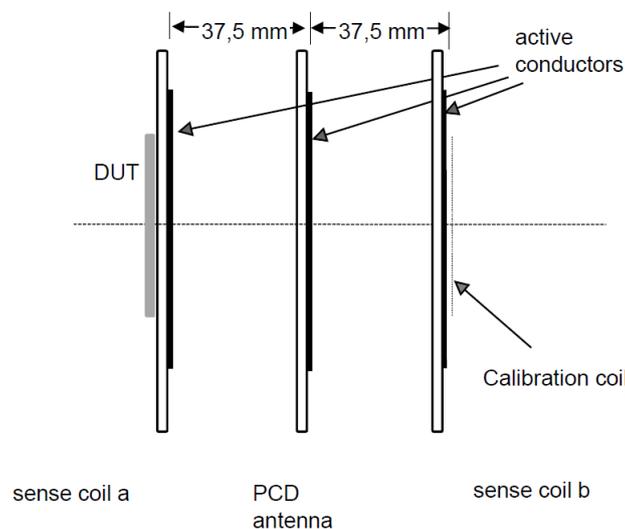


Abbildung 4.2: ISO-Turm Antennen Seitenansicht[8]

4.1.2 Calibration Coil

Die Calibration Coil ist eine genormte Antenne, mit der die Feldstärke beim ISO-Turm im gleichen Abstand zur PCD Antenne wie das DUT gemessen wird. Somit kann anhand des Messergebnisses der Calibration Coil die gewünschte Feldstärke eingestellt werden. Auf dem Foto I.3, mit dem in der Abbildung 4.3 dazugehörigen Aufbau, ist so eine ID1 Calibration Coil zu sehen. Aus der Spannung, die an der Calibration Coil gemessen wird,

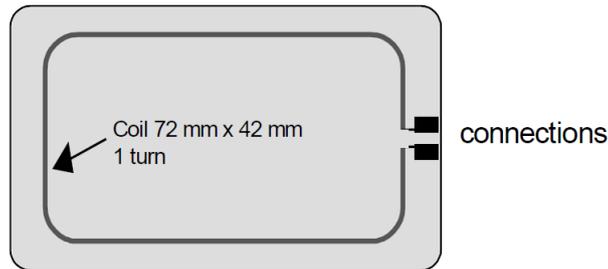


Abbildung 4.3: Calibration Coil[8]

kann die Feldstärke über das Induktionsgesetz 3.12 berechnet werden. Es wird für die Fläche A die Fläche der Windung, für die magnetische Permeabilität μ die magnetische Feldkonstante μ_0 und für die Frequenz $13,56 MHz$ eingesetzt. Das Ergebnis ist in der Formel 4.1 zu sehen, wobei hier gerne die Spannung von Spitze zu Spitze genommen wird, da diese leichter am Oszilloskop abzulesen ist.

$$\begin{aligned}
 H_{rms} &= \frac{U_{rms}}{A \cdot \mu \cdot 2 \cdot \pi \cdot f} = \frac{U_{rms}}{0,072 \cdot 0,042 \cdot 4 \cdot \pi \cdot 10^{-7} \cdot 2 \cdot \pi \cdot 13,56 \cdot 10^6} & (4.1) \\
 &= \frac{U_{rms}}{0,32} = \frac{U_{pp}}{0,9}
 \end{aligned}$$

4.1.3 Reference PICC

Die Reference PICC bildet eine RFID Karte im ID-1 Format nach. Sie kann entweder mit einer Schaltung bestückt sein, die einen Transponder nachbildet um die Empfindlichkeit des Lesegeräts zu prüfen, oder einfach einen Sockel besitzen mit dem ein Transponderchip als DUT aufgenommen wird. In der Abbildung 4.4 ist der Aufbau und in der Abbildung I.5 ein Foto solch einer Reference PICC zu sehen. Normalerweise hat die Antenne vier Windungen und die gleichen Außenabmessungen wie eine Calibration Coil

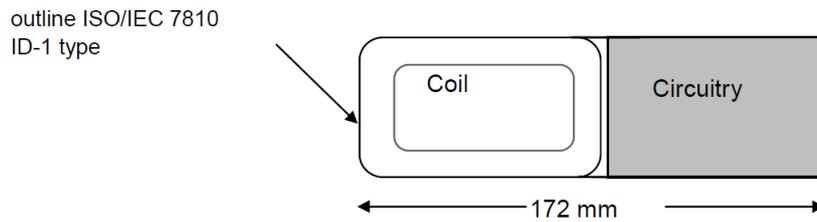


Abbildung 4.4: Reference PICC[8]

4.2 Messung mit einem AWG-Setup

Die Bezeichnung AWG-Setup ist bei Infineon eine übliche Bezeichnung, da für den Messaufbau als Signalgenerator ein AWG verwendet wird. Dabei steht AWG für Arbitrary Waveform Generator, mit dem beliebige Signalformen generiert werden können. Dieses AWG-Setup besteht aus einem PC, AWG, Messverstärker, ISO-Turm, Oszilloskop und einem Spektrumanalysator. Im Blockschaltbild 4.5 ist der Aufbau dargestellt. Mit dem PC wer-

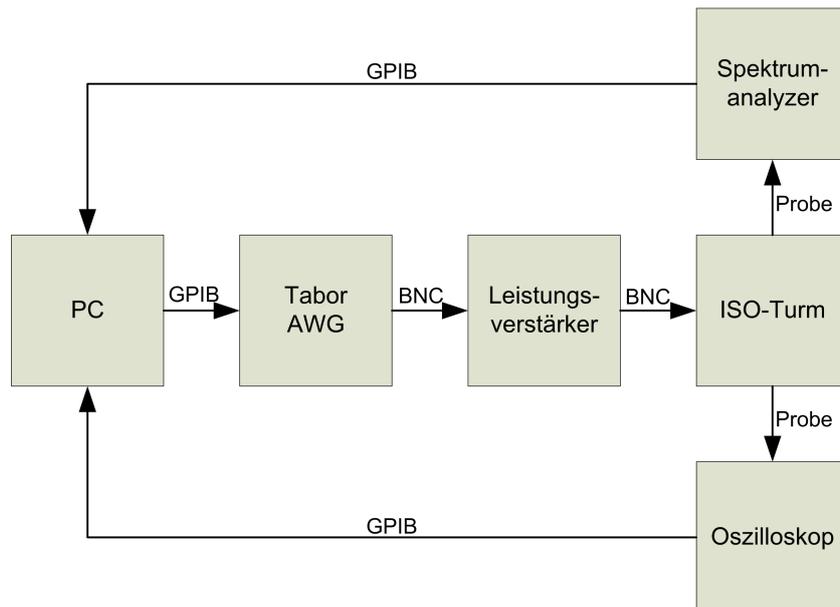


Abbildung 4.5: AWG-Setup Blockdiagramm

den *.wav* Dateien generiert, die über das Tabor AWG ausgegeben und dem Messverstärker verstärkt werden. Dieses Signal wird dann in die PCD Antenne eingespeist. An der Calibration Coil wird mit einem Oszilloskop die Feldstärke in dem definierten Abstand gemessen, in dem sich auch die Reference PICC mit dem DUT befindet. Nun kann die Leistung an dem Verstärker so eingestellt werden, dass die gewünschte maximale Feldstärke erreicht

wird. Mit dem Spektrumanalysator werden die Seitenbänder, die durch die Rückmodulation vom DUT (RFID-Chip) entstehen, gemessen. Der PC bekommt über eine GPIB Schnittstelle die Daten vom Oszilloskop und stellt den Pegel am AWG so ein, dass die gewünschte Feldstärke an der Reference PICC anliegt. Dann wird ein moduliertes Pattern über das AWG ausgegeben, das einen bestimmten Befehl beinhaltet, den der Chip, wenn der Befehl richtig dekodiert wird, mit einer dementsprechenden Rückantwort beantwortet. Diese lastmodulierte Rückantwort wird mit dem Spektrumanalysator als Seitenbänder erkannt und über GPIB zum PC übertragen. Somit kann bei verschiedenen Feldstärken ermittelt werden, ob das DUT den Befehl noch versteht und noch ausreichend Energie zur Verfügung hat um eine Rückantwort zu generieren. Um den Einfluss auf die Kommunikation, der durch hochgütige PCB Antennen entsteht, einerseits zu reduzieren, aber trotzdem zu berücksichtigen, wird hier eine PCB-Antenne mit geringer Güte verwendet und die Testpattern werden so generiert, dass Einschwing-, Abkling- und Überschwingverhalten mit definierten Zeiten und Werten eingespeist werden. Einen Zeitverlauf einer solchen Hüllkurve, wie sie in der ISO/IEC 14443-2[7] definiert ist, ist für 100% ASK in der Abbildung 4.6 zu sehen. Anhand dieser Beschreibung wird schnell klar, dass dieses Messverfahren sehr zeitaufwendig ist und etliche teure Messgeräte benötigt werden. Außerdem

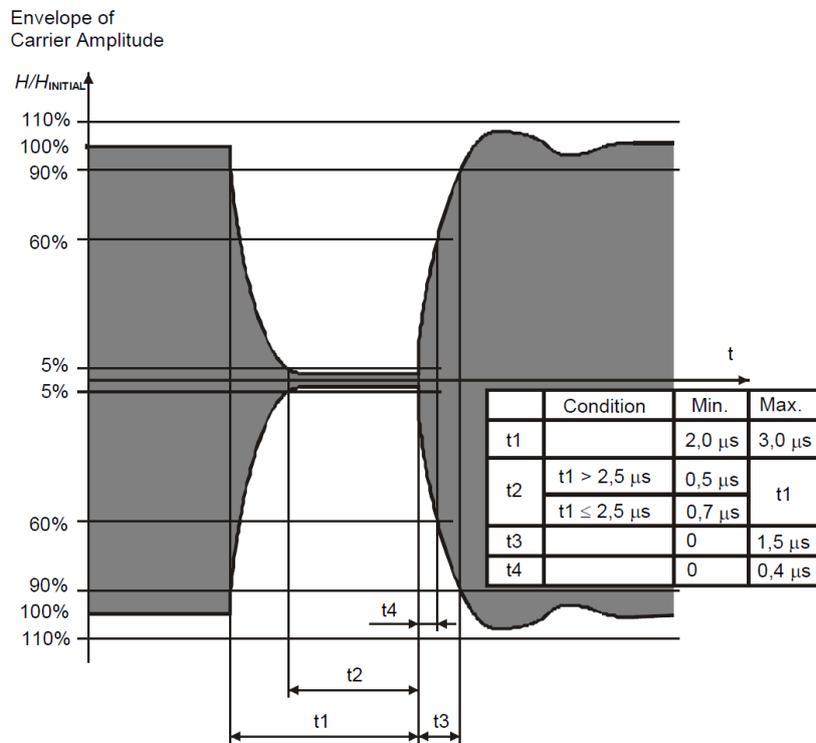


Abbildung 4.6: Zeiten eines Type-A Signals[7]

muss per Hand nach jeder Messung das DUT gewechselt werden, was nur ein Messen während regulärer Arbeitszeiten ermöglicht. Dies führte zu Überlegungen diese Messung zwar nicht zu ersetzen, aber eine Möglichkeit zu schaffen die Messung im Vorfeld auf einem Chiptester mit mehreren Chips gleichzeitig und voll automatisiert durchzuführen, um eine Abschätzung der Qualität der Chips zu ermöglichen. Diese Überlegungen führten in weiterer Folge zu dieser Diplomarbeit.

4.3 Messung mit dem CL-Emulator

Die Bezeichnung CL-Emulator steht für Contactless Emulator und bezeichnet das Konzept dieser Diplomarbeit die Übertragungsstrecke nachzubilden. Wie schon in dem Theoriekapitel 3.2 beschrieben, wurde das elektrische Netzwerk auf eine Stromquelle, einen Widerstand und dem DUT reduziert. In der Abbildung 4.7 ist noch einmal das Ergebnis aus dem Theorieteil zu sehen. Aus diesem Konzept wurde das Konzept in Abbildung

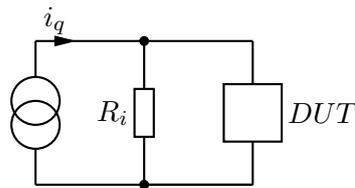


Abbildung 4.7: ESB Contactless Emulator

4.8 entwickelt, wo die Stromquelle durch einen U/I (Spannungs/Strom) Konverter ersetzt und dieser von einem D/A (Digital/Analog) Umsetzer gespeist wird. Gesteuert wird das

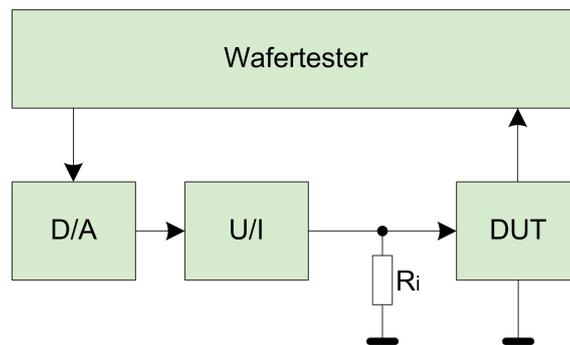


Abbildung 4.8: Contactless Emulator

Ganze von einem Wafertester, der im Kapitel 6 Spea Chiptester erläutert wird. Die Hardware, die aus D/A Umsetzer und U/I Konverter sowie dem einstellbaren Widerstand R_i

besteht, wird im Kapitel 5 Hardware beschrieben. Das DUT befindet sich dabei noch auf dem Wafer und wird von einem Prober gewechselt und kontaktiert. Die Software, die die Daten bereitstellt und dem Wafertester zur Verfügung stellt, wird im Kapitel 7 Software erklärt.

Der grobe Ablauf dieser Prüfanordnung: Mit dem Programm Matlab werden Hüllkurvensignale generiert und im *.wav* Format abgespeichert. Das Tool *Tabor2Spea.exe* konvertiert diese Dateien in ein für den Spea Wafertester verständliches Format. Auf diesem werden sie nach dem Kompilieren als Digitalwert dem Digital Analog Umsetzer zugeführt, welcher es in ein analoges Spannungssignal wandelt. Der Spannungs-Strom-Konverter erzeugt aus dem Spannungssignal ein äquivalentes Stromsignal. Dieses Stromsignal generiert, abhängig vom Innenwiderstand des RFID-Chips und R_i , eine Spannung am Eingang des DUT, die den RFID-Chip versorgt und von ihm demoduliert wird. Dieses demodulierte Signal wird vom DUT an einem I/O Pin ausgegeben und zum Auswerten an den Chiptester weitergeleitet. In Abbildung 4.9 ist dieser Ablauf, beginnend mit der *.wav* Kurvenform links oben, dem *.p40* Pattern unten bis zum braunen Analog- und blauen I/O-Signal rechts oben, zu sehen.

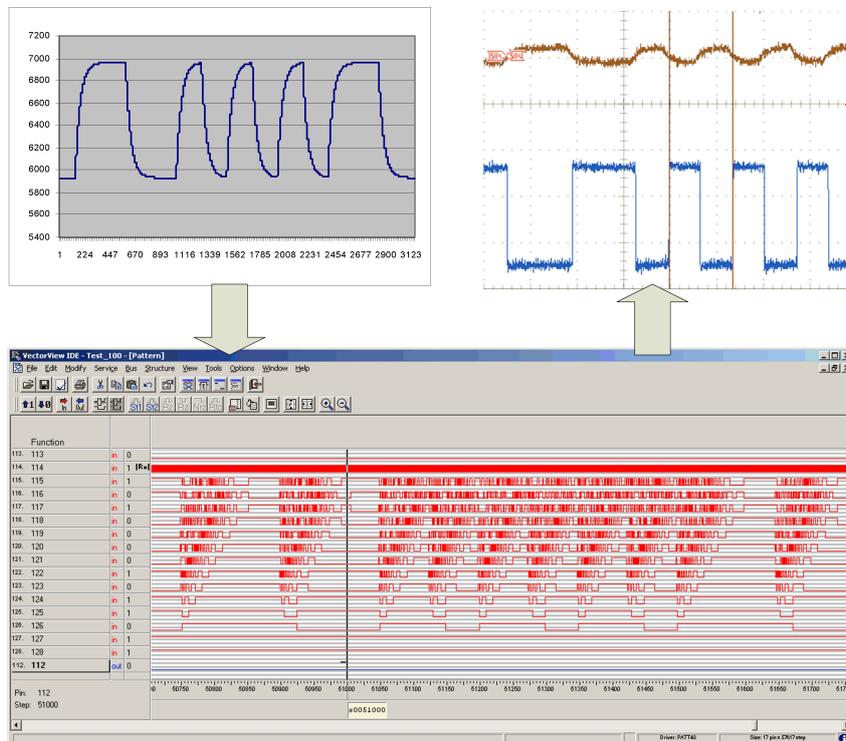


Abbildung 4.9: Vom *.wav* über *.p40* zum Ausgangssignal

5 Hardware

5.1 Einleitung

Die Aufgabe der Hardware ist es die digitalen Signale des Spea-Testers in einen analogen Stromverlauf zu konvertieren und dem „Device Under Test“ zuzuführen. Die durch das DUT demodulierten Signale werden an einem I/O Pin ausgegeben und können vom Tester auf Richtigkeit überprüft werden.

Das System auf dem alles aufsetzt ist der Spea Mixed Signal Semiconductor Tester, der im Kapitel 6 genauer beschrieben wird. Das Load Board (siehe Kapitel 6.2) setzt auf den Spea-Tester auf und führt die ganzen Schnittstellen nach außen und stellt somit die Digitalleitungen für den DA-Umsetzer und Rückleitungen zur Verifikation zur Verfügung. Über den DA-Umsetzer werden mittels Takt und 14 Bit Parallelbus vom Spea-Tester aus die einzelnen Samples der ursprünglichen Tabor *.wav* Datei hinausgetaktet. Dieses Analogsignal wird im nächsten Schritt verstärkt und tiefpassgefiltert dem Spannungs-Stromkonverter zugeführt. Dieser Stromverlauf wird dem Antenneneingang des DUT zugeführt, wobei ein parallel geschaltetes Widerstandsnetzwerk als Gütenachbildung dient. Die vom Demodulator im DUT erkannte Hüllkurve wird über den I/O Pin des DUT als Digitalsignal ausgegeben und dem Tester zur Analyse rückgeführt. In der Abbildung 5.1 ist der Ablauf als Blockdiagramm dargestellt.

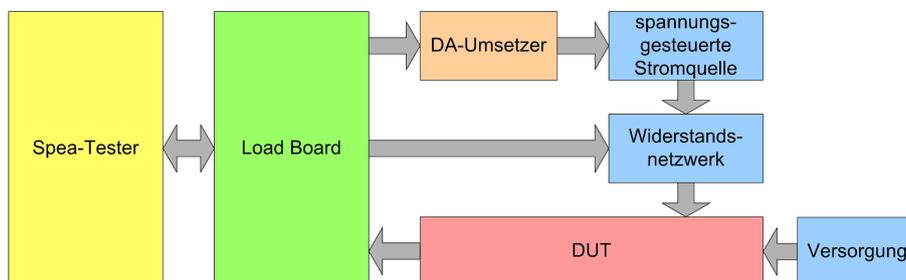


Abbildung 5.1: Hardwareablauf

5.2 Evaluation Board

Der Versuchsaufbau besteht aus mehreren Teilen, deren Verbindung im Blockdiagramm Evaluation Board (Abbildung 5.2) zu sehen ist. Auf dem Spea-Tester befindet sich das Load Board, mit dem sehr viele Verbindungen zum Tester zur Verfügung gestellt werden. Das DA Evaluation Board wird vom Load Board mit 14 Datenleitungen einer Clock- und einer Resetleitung versorgt, mit denen parallel ein 14 Bit Datenwort digital analog gewandelt wird. Das DA Evaluation Board wird dabei vom CL Evaluation Board mit 3 mal 3,3 V für den DA-Umsetzer und +5 V, -5 V für den Ausgangsverstärker versorgt. Das erzeugte Analogsignal wird dem Eingangverstärker des CL Evaluation Boards zugeführt. Dieser Verstärker ist neben der Verstärkung auch für den Offsetabgleich zuständig. Im nächsten Schritt wird das Signal tiefpassgefiltert um das Signal vollständig zu rekonstruieren und die höheren Frequenzen, die bei Nichteinhaltung des Nyquist-Shannon-Theorems entstehen, zu entfernen. Eine der wichtigsten Stufen ist der Spannungs/Strom-Umsetzer in der Form einer schnellen spannungsgesteuerten Stromquelle, der die Signale für den Antenneneingang des DUT zur Verfügung stellt. Im DUT werden die zur Verfügung gestellten Hüllkurven im Demodulator in digitale Signale umgewandelt, die über das Load Board dem Tester zugeführt und analysiert werden.

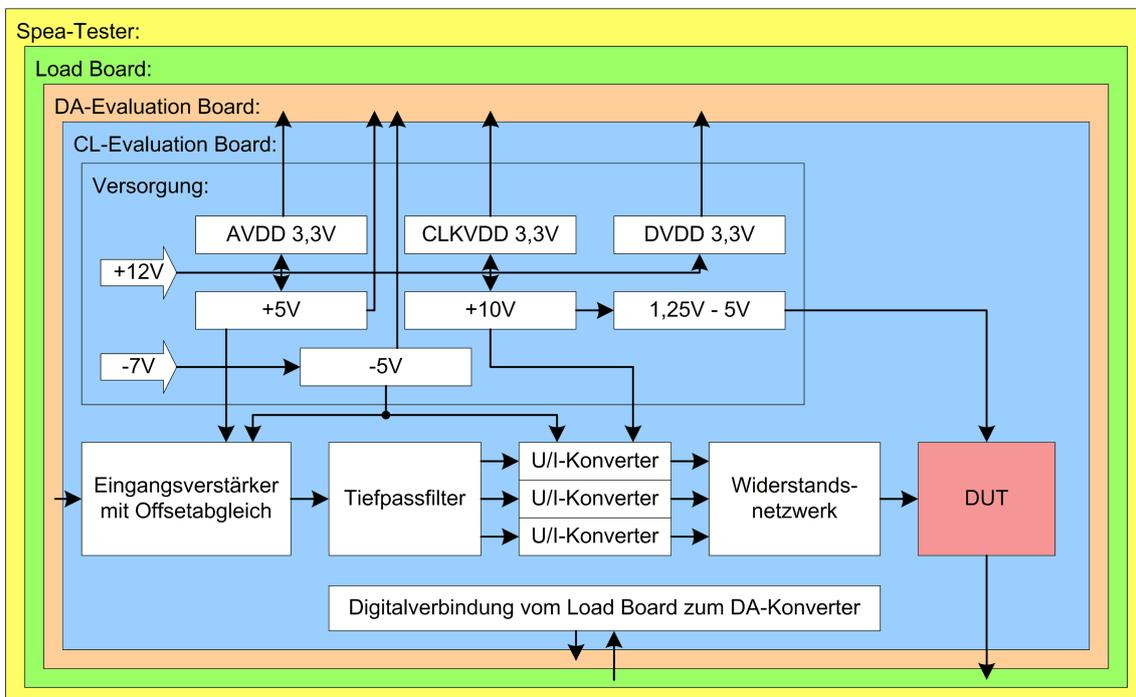


Abbildung 5.2: Blockdiagramm Evaluation Board

5.2.1 Digital-Analog-Umsetzer

Beim Digital-Analog-Umsetzer wurde für einen 14-Bit, 160 MSPS DAC von Analog Devices entschieden. 14-Bit wurde gewählt, da die Tabor *.wav* Dateien auch 14-Bit verwenden. Der AD9772A ist mit einem 2x FIR Interpolationsfilter für Überabtastung ausgestattet, was die Komplexität des Rekonstruktionsfilters auf ein Drittel reduziert. Eine integrierte PLL erzeugt aus dem Takt die für das digitale Filter und den DAC nötigen höheren Taktfrequenzen. Typische Applikationen laut Datenblatt sind Basisband- und Zwischenfrequenzrekonstruktion. Für die Basisbandapplikationen kann man das digitale Filter auf Tiefpass schalten, wobei Frequenzen im Bereich von $0,606 * f_{DATA}$ bis $1,394 * f_{DATA}$ mit mehr als 73 dB unterdrückt werden. Für Zwischenfrequenzapplikationen kann das Filter auch auf Hochpass geschaltet werden, wobei hier das Basisband unterdrückt wird. Zusätzlich für ZF-Applikationen verfügt der DA-Umsetzer über Zero-Stuffing, das im Wesentlichen das Sinc-Verhalten auf die doppelte Bandbreite ausdehnt und somit dieses Verhalten im Arbeitsbereich reduziert. Im Blockschaltbild 5.3 ist der strukturelle Aufbau des DA-Umsetzers dargestellt.

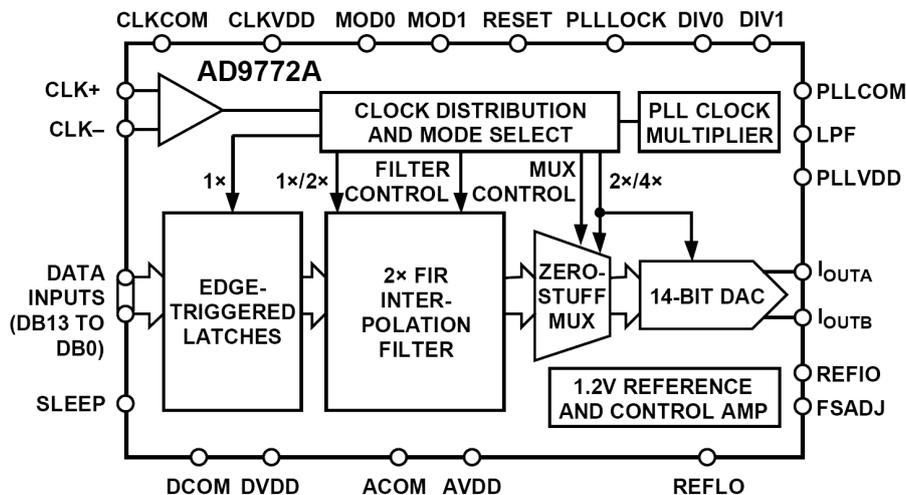


Abbildung 5.3: AD9772A Blockdiagramm[2]

5.2.1.1 AD9772 Evaluation Board von Analog Devices

Bei der Entwicklung des CL-Evaluation Boards wurde für den Einsatz des DA-Evaluation Boards von Analog Devices entschieden, da somit Signale mit ausreichendem Signal-Störabstand und geringen Verzerrungen gewährleistet sind. Da am Evaluation Board kein

Rekonstruktionsfilter vorhanden ist, musste dieser in der Folgeschaltung im Signalweg eingebaut werden um das Nyquist-Shannon-Theorem einzuhalten. Im Bild 5.4 sieht man den Schaltplan des Hauptteils des AD9772 Evaluation Boards.

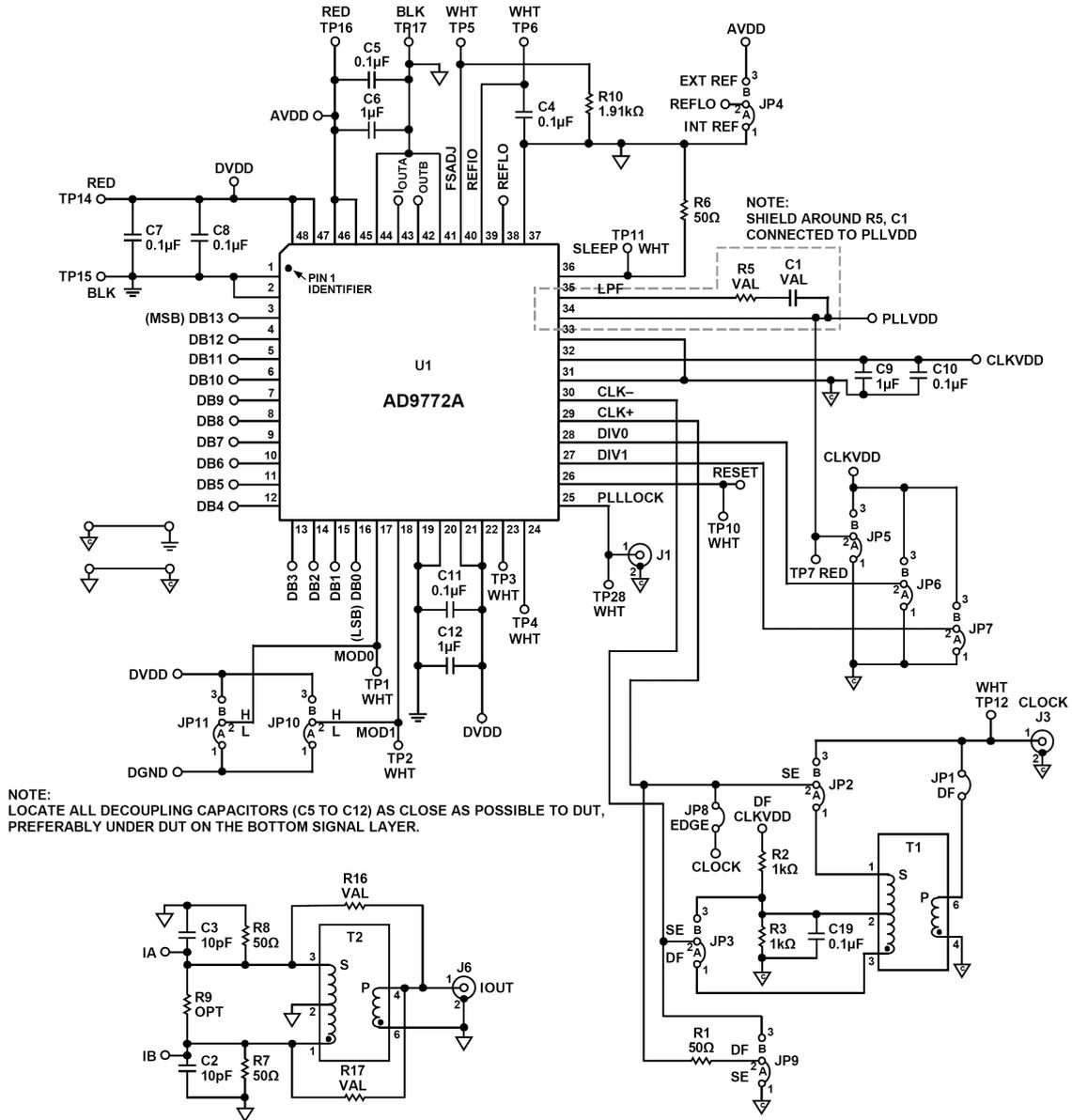


Abbildung 5.4: Schaltplan des AD9772 Evaluation Boards[2]

Folgende Modifikationen müssen am Evaluation Board durchgeführt werden um das Board in den richtigen Betriebsmodus zu versetzen.

Bauteil	Stellung	Beschreibung
T2	entfernt	T2 muss entfernt werden, da der AMPOUT verwendet wird
JP1	offen	Der Clock wird nicht über den Übertrager geführt
JP2	SE	Der Clock wird nicht über den Übertrager geführt
JP3	SE	Der Clock wird nicht über den Übertrager geführt
JP4	IN	Es wird die interne Referenzspannung verwendet
JP5	offen	PLLVDD hat eine eigene Versorgung
JP6	H	DIV0: 2 faches Oversampling wird verwendet
JP7	L	DIV1: 2 faches Oversampling wird verwendet
JP8	gesetzt	Clock wird über die 40-polige IDC-Steckverbindung geführt
JP9	SE	Clock wird auf $50\ \Omega$ terminiert
JP10	L	MOD1: Zero-Stuffing ist ausgeschaltet
JP11	L	MOD0: Digitalfilter wird als Tiefpass verwendet
JP12	gesetzt	AMPOUT wird verwendet
JP13	gesetzt	AMPOUT wird verwendet

Tabelle 5.1: Konfiguration des AD9772 Evaluation Boards

5.2.1.2 AD9772 Evaluation Board Ausgangsverstärker

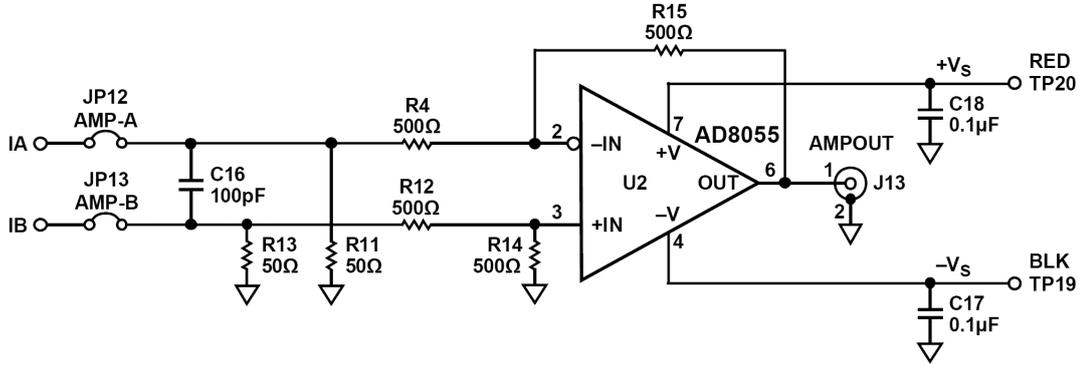
In der Abbildung 5.5 sind zwei von den drei möglichen Ausgängen des DA-Evaluation Boards abgebildet, wobei die Schaltung mit Ausgangsübertrager 5.5b für diese Anwendung nicht geeignet ist, da wir immer einen Gleichspannungsanteil auf den Signalen haben. Die Ausgangsschaltung 5.5a ist ein Subtrahierverstärker, der aus dem symmetrischen Stromausgang des DA-Umsetzers einen unsymmetrischen Ausgang mit der Ausgangsspannung U_a (Gleichung 5.1) erzeugt.

$$U_a = I_B \cdot (R_7 || R_{13}) - I_A \cdot (R_8 || R_{11}) = I_B \cdot 25\ \Omega - I_A \cdot 25\ \Omega \quad (5.1)$$

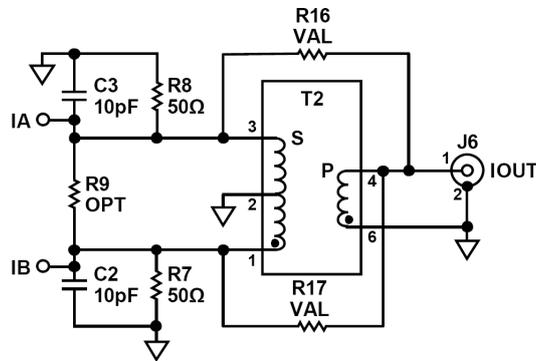
Die Ausgangsströme ergeben sich aus den Gleichungen 5.2 und 5.3

$$I_{OUTA} = (DAC\ CODE / 16384) \cdot I_{OUTFS} \quad (5.2)$$

$$I_{OUTB} = (16383 - DAC\ CODE) / 16384 \cdot I_{OUTFS} \quad (5.3)$$



(a) Ausgangsstufe mit Verstärker



(b) Ausgangsstufe mit Übertrager

Abbildung 5.5: Ausgangsstufen vom AD9772 Evaluation Board[2]

Der maximale Ausgangsstrom I_{OUTFS} errechnet sich aus dem Referenzstrom, der sich wiederum aus der Referenzspannung ergibt.

$$I_{REF} = \frac{U_{REF}}{R_{SET}} = \frac{1,2 V}{1910\Omega} = 628 \mu A \quad (5.4)$$

$$I_{OUTFS} = 32 \cdot I_{REF} = 32 \cdot 628 \mu A = 20,1 mA \quad (5.5)$$

Aus der Gleichung 5.1 und 5.5 lässt sich nun die maximale Ausgangsspannung U_{aFS} mit der Formel 5.6 errechnen.

$$U_{aFS} = \pm 20,1 mA \cdot 25 \Omega = \pm 0,5025 V \quad (5.6)$$

5.2.1.3 Rekonstruktionsfilter Teil 1

Bei der Umwandlung von zeitdiskreten Signalen in kontinuierliche muss der Bereich zwischen den einzelnen Abtastwerten interpoliert werden. Diese Interpolation wird mit einem

Rekonstruktionsfilter mit einer ausreichenden Tiefpasswirkung erreicht. Wird auf einen Rekonstruktionsfilter verzichtet, tritt das so genannte Aliasing auf, bei dem Frequenzen entstehen, die nicht im Digitalsignal vorhanden sind. Diese Frequenzen ergeben sich durch das immer wiederkehrende Spiegeln des Basisbandes um das Vielfache der Abtastfrequenz, wobei das Spektrum durch das Konstanthalten des Wertes über eine Abtastperiode zu höheren Frequenzen mit dem Betrag der Sinc-Funktion eingehüllt und somit gedämpft wird. Mit dem *Interactive Design Tool* (Abbildung 5.6) von Analog Devices kann man das Verhalten von Sinc-Funktion, FIR-Filter und Reonstruktionsfilter bei verschiedenen Ausgangsfrequenzen und deren Verzerrung in der Form der 2. oder 3. harmonischen Oberwelle im Spektrum betrachten.

Interactive Design Tools: Digital-to-Analog Converters:

Harmonic Images in the AD9772 D/A Converter

An applet for estimating harmonic images in the AD9772A TxDAC+®. ¹

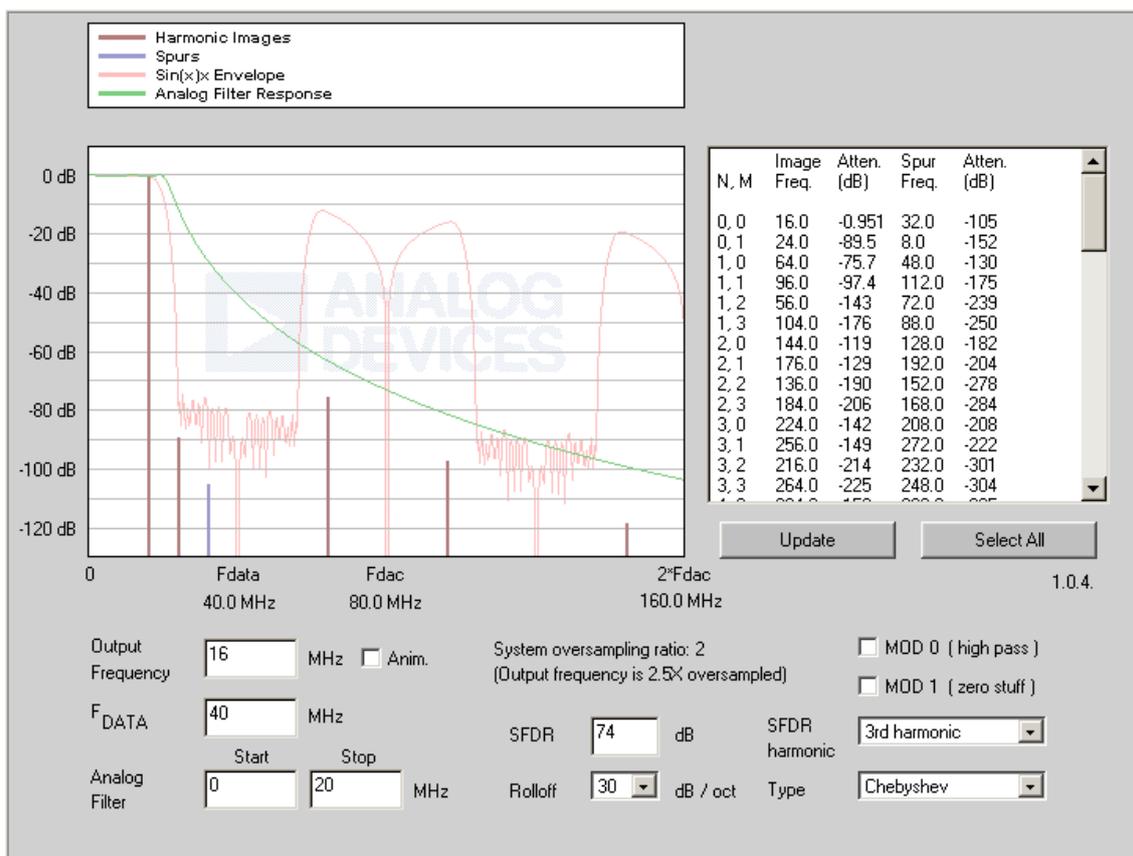


Abbildung 5.6: Harmonic Images in the AD9772 D/A Converter

¹<http://designtools.analog.com/dt/dac/ad9772image.html>

In dieser Simulation fällt auf, dass bei Frequenzen im Bereich von 16 MHz bis 20 MHz das integrierte FIR-Filter noch nicht greift. Es muss bei der Erzeugung der *.wav* Dateien darauf geachtet werden, dass keine Signale in diesem Frequenzbereich generiert werden, oder diese noch im Digitalen herausgefiltert werden. Weiters sieht man den Effekt der Sinc-Funktion in Kombination mit dem FIR-Filter wo das nächste Maximum bei $62,4\text{ MHz}$ auftritt. Die dort erreichte Dämpfung von $-12,4\text{ dB}$ wirkt sich positiv auf die Konstruktion des Rekonstruktionsfilters aus. Für die Dimensionierung des Rekonstruktionsfilters kann man davon ausgehen, dass der FIR-Filter eine Unterdrückung von maximal 73 dB schafft, und erst bei der Frequenz von 64 MHz ein Dämpfungsminimum von $-12,9\text{ dB}$ erreicht. Der Bereich von 60 MHz bis 64 MHz ist nicht zu berücksichtigen, da es derselbe Bereich wie die vorhergehenden 16 MHz bis 20 MHz ist und in diesem Bereich sowieso keine Signale vorhanden sein dürfen. Bei den 64 MHz soll der Rekonstruktionsfilter vollständig greifen und die restliche Dämpfung auf -73 dB übernehmen. Der erste Teil dieser Filterkette befindet sich in der Ausgangsstufe am Evaluation Board (Abbildung 5.5a). Dabei bildet der Kondensator $C16$ im Speziellen mit den vier Widerständen $R11||R8$ und $R13||R7$ einen Tiefpass, wobei $R4 + R15$ und $R12 + R14$ für die Berechnung der Grenzfrequenz vernachlässigt werden können, da sie um den Faktor 40 größer sind als die zu ihnen parallel liegenden Widerstände. Die Ausgangswiderstände des DA-Umsetzers können auch vernachlässigt werden, da sie gleich um einen Faktor 8000 größer sind. Die Kapazitäten $C2$, $C3$ und Ausgangskapazitäten wurden auch nicht berücksichtigt, da sie wesentlich kleiner als $C16$ sind und nur einen geringfügigen Einfluss auf die Grenzfrequenz nehmen. Dieser Filter ergibt einen RC-Tiefpass 1. Ordnung, deren Grenzfrequenz mit der Formel 5.7 berechnet wird. Da ein Filter 1. Ordnung nur 6 dB/Oktave schafft, erreicht man bei der Frequenz 64 MHz nach Formel 5.8 gerade einmal eine Dämpfung von ca. -7 dB . Die restliche Dämpfung von $73\text{ dB} - 12,9\text{ dB} - 7\text{ dB} = 53,1\text{ dB}$ muss mit dem Rekonstruktionsfilter nach dem Eingangsverstärker erreicht werden.

$$f_g = \frac{1}{2 \cdot \pi \cdot C \cdot R} = \frac{1}{2 \cdot \pi \cdot C \cdot \left(\frac{R11 \cdot R8}{R11 + R8} + \frac{R13 \cdot R7}{R13 + R7} \right)} = \frac{1}{2 \cdot \pi \cdot 100\text{ pF} \cdot 50\ \Omega} = 31,83\text{ MHz} \quad (5.7)$$

$$\begin{aligned} a_{16\text{ MHz}} &= 20 * \log \frac{1}{\sqrt{1 + (2 \cdot \pi \cdot f \cdot C \cdot R)^2}} \\ &= 20 * \log \frac{1}{\sqrt{1 + (2 \cdot \pi \cdot 16\text{ MHz} \cdot 100\text{ pF} \cdot 50\ \Omega)^2}} = -0,17\text{ dB} \end{aligned} \quad (5.8)$$

$$a_{64\text{ MHz}} = -7\text{ dB}$$

Bei der Rechnung 5.8 kann man erkennen, dass bei der höchsten Frequenz im Basisband dieser Filter nur eine Dämpfung von $-0,17\text{ dB}$ erzeugt. In der Abbildung 5.6 kann

man bei dieser Frequenz schon eine Abschwächung von $-0,951\text{ dB}$ erkennen, die durch das Sinc-Verhalten der DA-Umsetzung und dem FIR-Filter entsteht. Als Ziel wurde gesetzt nicht mehr als -3 dB Dämpfung bei der maximalen Signalfrequenz von 16 MHz zu erreichen, hierfür werden alle Dämpfungen von Sinc, 1. und 2. Rekonstruktionsfilter zusammengezählt. Der Grund für diese Begrenzung ist, dass sich leichte Dämpfungen im Arbeitsbereich durch Filter und Sinc-Funktion nicht vermeiden lassen, aber eine mögliche Maximalfrequenz der Anwendung von $13,56\text{ MHz}$ ohne größeren Einfluss erreicht werden soll.

5.2.2 Eingangsverstärker

Im Eingangsbereich sitzt ein Invertierverstärker (Abbildung 5.7) mit dem Verstärkungsfaktor $A = -2$. Es wurde ein Invertierverstärker gewählt, da der Subtrahierverstärker am

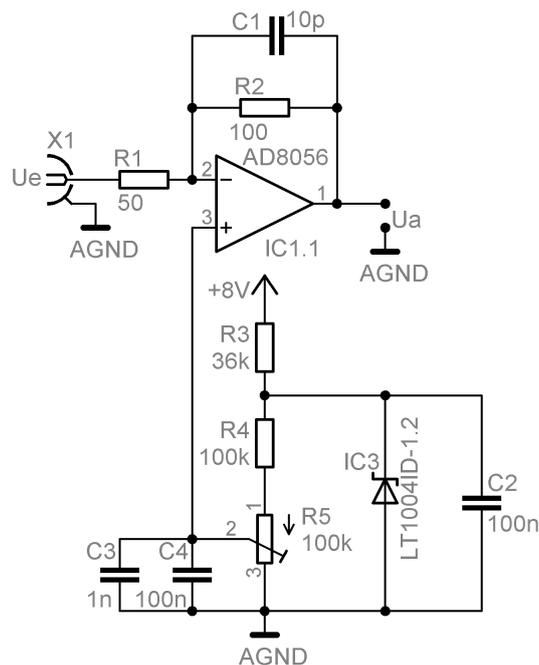


Abbildung 5.7: Eingangsverstärker mit einstellbarem Offset

Ausgang des DA Evaluation Boards das Signal invertiert ausgibt. Die Ausgangsspannung des Invertierverstärkers ergibt sich aus der Formel 5.9.

$$U_a = -U_e \cdot \frac{R_2}{R_1} + U_+ \cdot \left(1 + \frac{R_2}{R_1}\right) \quad (5.9)$$

Diese Formel ist natürlich nur eine Näherungsformel, da der Operationsverstärker als ideal betrachtet wurde. In dieser Schaltung liegt am negativen Eingang des Operationsverstärkers der $R1$ mit einem gewählten Wert von 50Ω , der somit die Eingangsimpedanz bildet. Am positiven Eingang liegt eine einstellbare Gleichspannung an, die mit der Spannungsreferenz IC LT1004-1.2 und einem Spannungsteiler erzeugt wird. Mit dieser einstellbaren Spannungsquelle lässt sich ein Gleichspannungsoffset einstellen. Der IC5 LT1004-1.2 ist eine Spannungsreferenz auf der Basis einer Bandgapreferenzdiode mit einer Spannung von $1,235V$ und einer garantierten Genauigkeit von $\pm 4mV$. Am positiven Eingang des Operationsverstärkers sollte eine Spannung von ca. $0,3V$ anliegen, damit aus den $\pm 0,5V$, die vom DA Evaluation Board kommen, ein Spannungsbereich von $0 - 2V$ entsteht. Mit dem Spannungsteiler, der durch $R4$ und dem Spindeltrimmpotentiometer $R5$ entsteht, lässt sich die Spannung zwischen $0V$ und $0,62V$ einstellen, somit kann auch der Offset von den einzelnen Verstärkerstufen mitkorrigiert werden. Der Kondensator $C1$ dient optional als Dämpfung um eine eventuelle Schwingneigung des Operationsverstärkers zu reduzieren. Die Kondensatoren $C2$, $C3$, $C4$ helfen die einstellbare Spannung konstant zu halten und dienen zur Reduktion von höherfrequenten Störungen.

5.2.3 Rekonstruktionsfilter Teil 2

Wie schon beim Rekonstruktionsfilter Teil 1 errechnet, soll mit dem Rekonstruktionsfilter Teil 2 eine Dämpfung von $53,1dB$ bei $64MHz$ erreicht werden. Nach dem Buch *Electronic Filter Design Handbook*[3], das doch schon älter ist, wird empfohlen bei diesem Frequenzbereich einen passiven LC Filter zu verwenden. Mittlerweile sind Operationsverstärker in diesem Frequenzbereich sehr günstig erhältlich und es spricht nichts gegen einen aktiven Ansatz. Bei einem passiven Ansatz müssten die Spulen auf Grund ihrer Toleranzen mit einer RLC-Brücke ausgemessen und selektiert werden. Zur Dimensionierung des Filters wurde die *Active Filter Design Toolbox* von Matlab verwendet. Der erste Ansatz war eine Chebychev II Charakteristik 4ter Ordnung in einer Multiple Feedback Anordnung. Dieser hat aber bei der Dimensionierung zu unrealistische Bauteilwerten geführt. Die Ackerberg-Mossberg Anordnung hat den großen Nachteil, dass dreimal so viele Operationsverstärker benötigt werden. Der Ansatz mit dem geringsten Bauteilaufwand und mit Bauteilwerten in einem realisierbaren Bereich lieferte ein Filter 5. Ordnung, in Butterworth Charakteristik, mit einer Sallen-Key Anordnung, bei einer Grenzfrequenz von $18,84MHz$. Dieser Filter hat keine Welligkeiten im Durchlass- sowie im Sperrbereich und einen sanften Übergang von Durchlass- in den Sperrbereich, der durch eine Dämpfung von $-0,774dB$

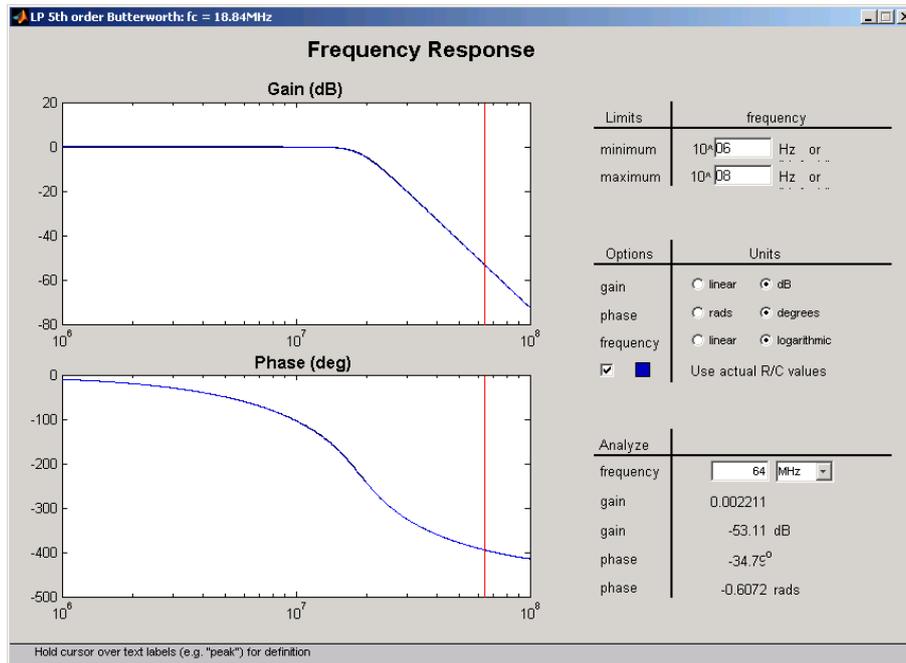


Abbildung 5.8: Bodediagramm des Rekonstruktionsfilters

bei 16 MHz erkaufte wird. Die geforderte Dämpfung von $-53,1 \text{ dB}$ bei 64 MHz wird erreicht. Im Bodediagramm (Abbildung 5.8) wird der Amplituden- und Phasenverlauf des Filters dargestellt. Das Filter setzt sich aus drei Stufen zusammen, wobei die ersten zwei Stufen in einer Sallen-Key Anordnung verwendet werden und die letzte Stufe aus einem RC-Tiefpass mit Impedanzwandler am Ausgang besteht. Dieser Impedanzwandler könnte eingespart werden, aber da dieses Board einen experimentellen Charakter hat, wo einzelne U/I-Konverterstufen zu- und weggeschaltet werden und das Messen am Messpunkt MP2 zu Kapazitätsänderungen und somit zum Verstimmen der letzten Filterstufe führt, ist eine Entkopplung von Vorteil. Die verwendeten Operationsverstärker sind von Analog Devices mit der Bezeichnung AD8056. Dieser Typ beinhaltet 2 Operationsverstärker in einem Gehäuse. Der erste wurde gleich für den Eingangsverstärker verwendet. Bei diesem Typ wird im Datenblatt auf die kapazitive Belastung hingewiesen, wobei die Kennlinie 5.12 den Mindestwert eines seriell vorgeschalteten ohmschen Widerstandes für einen bestimmten Kapazitätswert zeigt. Da bei der Dimensionierung keine Kapazitätswerte unter 20 pF vorkamen, musste darauf geachtet werden, dass ein serieller Widerstand von mindestens 25Ω vorhanden ist. In den Abbildungen 5.9 bis 5.11 befinden sich die Berechnungen der *Active Filter Design Toolbox*. Die fertige Schaltung ist in der Abbildung 5.13 zu sehen.

$x(t) \rightarrow$ Stage 1 \rightarrow Stage 2 \rightarrow Stage 3 $\rightarrow y(t)$

stage to edit:
 passband polarity:
 resistor tolerance:
 capacitor tolerance:

selected circuit:
Stage 1
 Stage details
 Print stage

recommended: Sallen-Key

select		calculated	
$C_a =$ <input type="text" value="270"/> pF	$R_a =$ <input type="text" value="0"/> ohms	$C_1 =$ 180 pF	$R_1 =$ 39 ohm
$C_b =$ <input type="text" value="0"/> pF	$R_b =$ <input type="text" value="0"/> ohms	$C_2 =$	$R_2 =$ Not present
		$C_3 =$	$R_3 =$
		$C_4 =$	$R_4 =$

Abbildung 5.9: 1. Stufe des Rekonstruktionsfilters

$x(t) \rightarrow$ Stage 1 \rightarrow Stage 2 \rightarrow Stage 3 $\rightarrow y(t)$

stage to edit:
 passband polarity:
 resistor tolerance:
 capacitor tolerance:

selected circuit:
Stage 2
 Stage details
 Print stage

recommended: Sallen-Key

select		calculated	
$C_a =$ <input type="text" value="270"/> pF	$R_a =$ <input type="text" value="0"/> ohms	$C_1 =$ 27 pF	$R_1 =$ 100 ohm
$C_b =$ <input type="text" value="0"/> pF	$R_b =$ <input type="text" value="0"/> ohms	$C_2 =$	$R_2 =$ Not present
		$C_3 =$	$R_3 =$
		$C_4 =$	$R_4 =$

Abbildung 5.10: 2. Stufe des Rekonstruktionsfilters

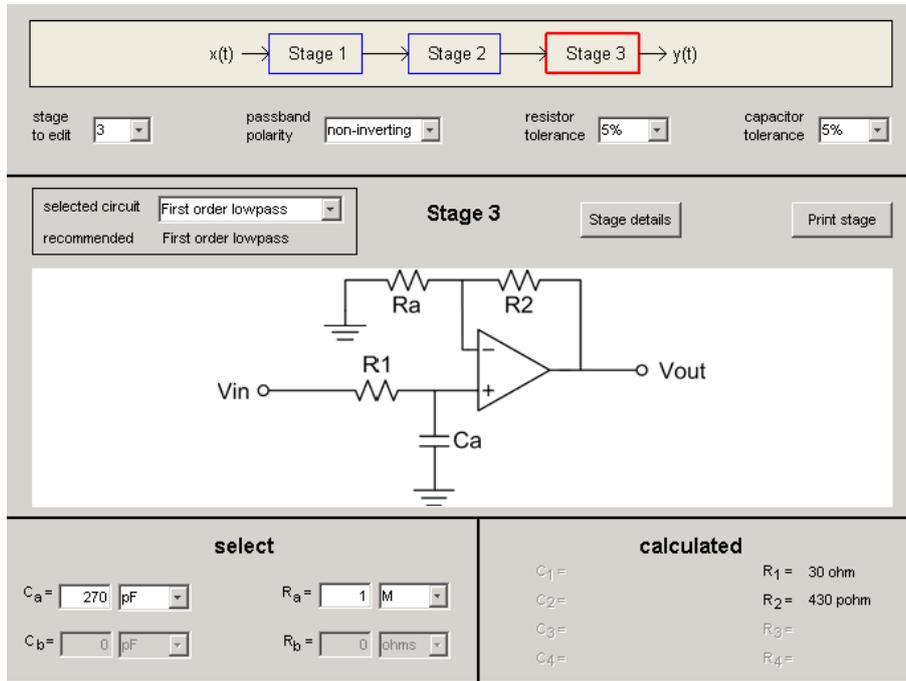


Abbildung 5.11: 3. Stufe des Rekonstruktionsfilters

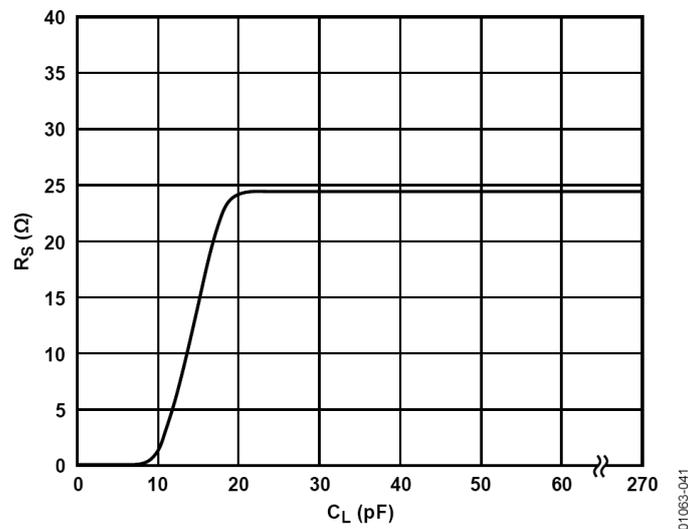


Abbildung 5.12: R_S vs. C_L des AD8056[1]

5.2.4 Stromquelle

Der Kernteil dieser Anlogschaltung ist die Stromquelle. Es ist ein schneller Spannungs-Strom-Konverter, der die Signale vom Digital-Analog-Umsetzer in einen Strom umwandelt. Die Schwierigkeit bei U/I-Konvertern ist das Erreichen von hohen Grenzfrequenzen, da sich

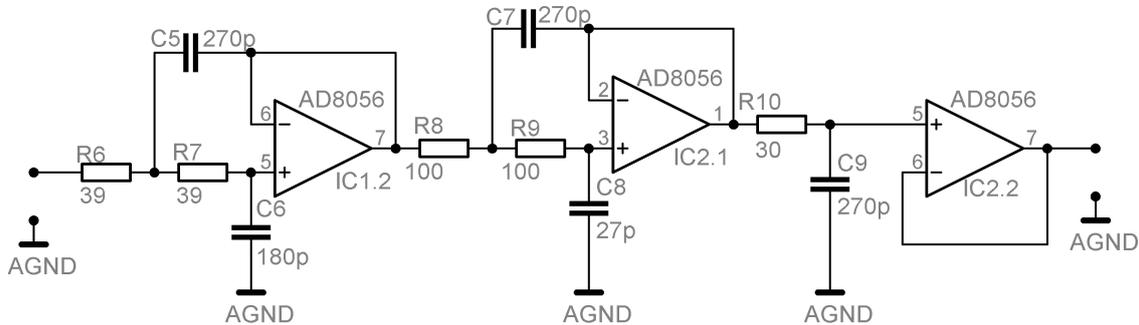


Abbildung 5.13: Butterworth Filter 5. Ordnung

parasitäre Kapazitäten bedingt durch den hohen Ausgangswiderstand stark negativ auf die Bandbreite auswirken. Im Vorfeld wurden mehrere Schaltungen aus dem Buch *Halbleiter-Schaltungstechnik*[13] in diskreter Bauform oder mit Operationsverstärker ausprobiert, wobei das Erreichen hoher Grenzfrequenzen und das Reduzieren von Temperaturdrift sich als schwierig herausstellte. Es wurde schlussendlich durch den Artikel *Sehr schnelle spannungsgesteuerte Stromquelle*[4] aus der Elektor Zeitschrift „7-8/98“ angeregt, eine einfache Schaltung mit nur einem integrierten Bauteil und einem passiven Bauteil verwendet. Das Prinzip dieser Schaltung ist einfach, der Differenzverstärker AD8130 hat als Eingang zwei Spannungsdifferenzverstärker, deren Ausgangsströme in einem Knotenpunkt addiert und über eine Treiberstufe am Ausgang ausgegeben werden. Bedingt durch die starke Gegenkopplung versucht der AD8130 in dieser Schaltung den Ausgang so auszuregeln, dass sich im Knotenpunkt die beiden Ströme aufheben. Somit ergibt sich, dass der Spannungsabfall U_r am Widerstand R gleich groß wie die Eingangsspannung U_e sein muss. Das ergibt die Formel 5.10.

$$U_e = U_r = R \cdot I_a \Rightarrow I_a = \frac{U_e}{R} \quad (5.10)$$

Der große Vorteil dieser Schaltung ist, dass man mit nur zwei Bauteilen auskommt. Der Nachteil ist der geringe Strom der Ausgangstreiber, der typischerweise beim AD8130 bei 35 mA liegt, aber Werte von 100 mA erreichen soll. Deswegen werden drei Stromquellen parallel geschaltet, was kein Problem ist, da hochohmige Spannungseingänge und hochohmige Stromausgänge ohne weiteres zusammengeschlossen werden können. Der gewählte Widerstand R beträgt 50Ω , daraus ergibt sich bei einer maximalen Eingangsspannung von 2 V ein maximaler Ausgangsstrom von 40 mA pro Stromquelle und bei Betrieb aller Stromquellen ein maximaler Ausgangsstrom von 120 mA . Die 40 mA sind ein bisschen über der Spezifikation von Analog Devices, aber durch die Angabe des Kurzschlussstroms von

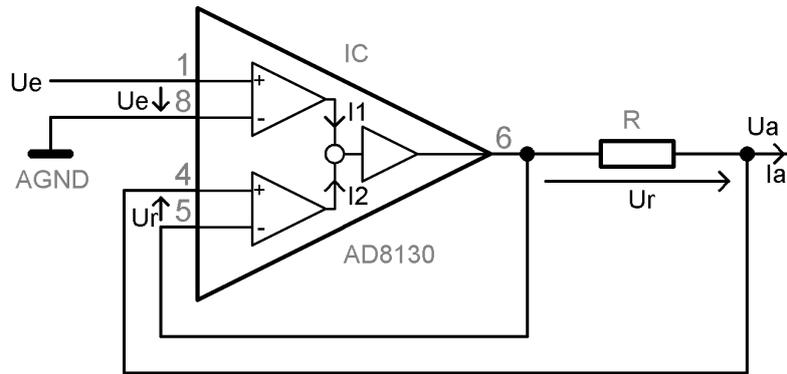


Abbildung 5.14: Stromquelle

55mA und durch Messungen wurde ermittelt, dass es bei diesem Wert noch zu keinen Einschränkungen kommt. In der Schaltung 5.15 sieht man die Parallelschaltung der einzelnen Stromquellen wie sie am Evaluation Board über Jumper hinzu und wieder weggeschaltet werden können.

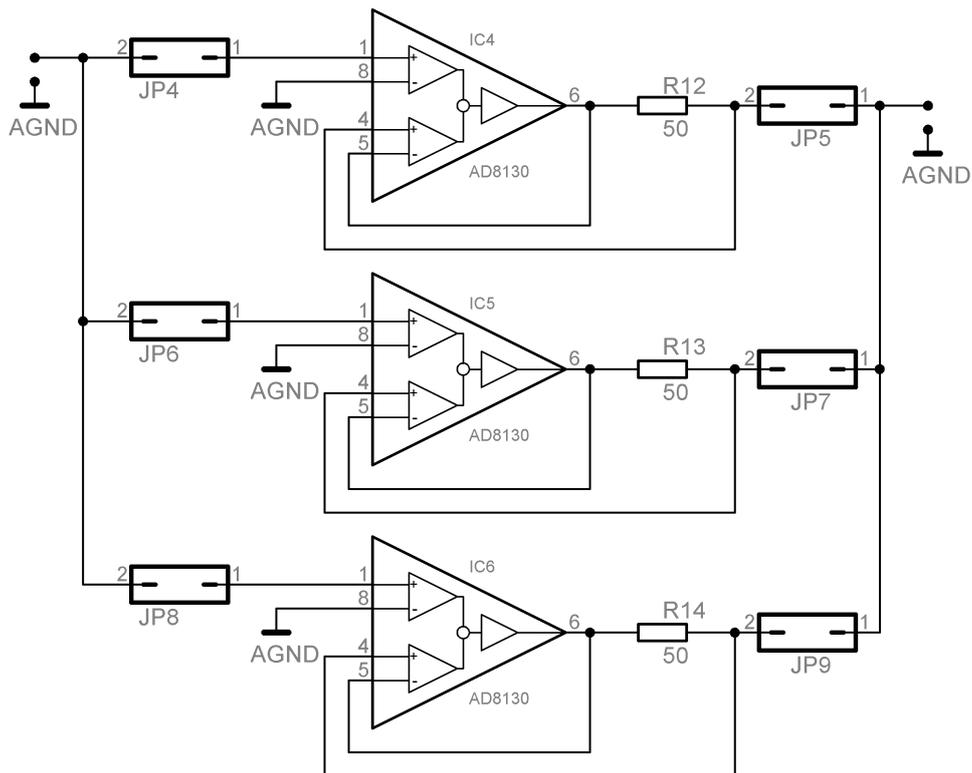


Abbildung 5.15: Stromquellen-Parallelschaltung

5.2.5 Widerstandsnetzwerk

Wie schon aus dem theoretischen Kapitel bekannt ist, wird die Güte des Systems auf einen Widerstandswert reduziert. Damit verschiedenste Gütewerte eingestellt werden können muss ein einstellbarer Widerstand verwendet werden. Digital einstellbare Widerstände auf Chipbasis sind nicht geeignet, da sie für diesen Frequenzbereich und diese Aufgabenstellung zu große Kapazitätswerte besitzen. Potentiometer mit einem Einstellmotor sind aufwendig und führen zu schnellen Alterungserscheinungen, wie zum Beispiel Kontaktprobleme des Schleiferkontaktes, außerdem ist das Einstellen eines genauen Widerstandswertes ohne Messung schwierig zu lösen. Das gleiche Problem entsteht wenn ein FET als einstellbarer Widerstand verwendet wird. Als sinnvollste Lösung schien ein Widerstandsnetzwerk, dass in einer automatisierten Ausführung Relais zum Umschalten benützt. Dadurch können im Testprogramm automatisch verschiedene Widerstandswerte eingestellt werden. Mit dem Widerstandsnetzwerk in Abbildung 5.16a lassen sich Werte von $50\ \Omega$ bis $51200\ \Omega$ in $50\ \Omega$ Schritten einstellen. Bei dieser Evaluation Ausführung wurde auf Relais verzichtet und durch DIP-Schalter ersetzt. Da nicht alle Widerstandswerte erhältlich sind, wurden manche Werte durch Parallel- und Serienschaltung realisiert, was in Abbildung 5.16b zu sehen ist. Später sollen dann Relais von Pickering mit der Typenbezeichnung 108-1-A-5/1D eingesetzt werden. Der große Vorteil dieser Relais sind ihre geringen parasitären Kapazitäten, die laut dem Dokument *Contact-Less Emulation System* [10] $0,2\ pF$ zwischen den zwei Schaltkontaktanschlüssen und $0,37\ pF$ zwischen den zwei Schaltkontaktanschlüssen und der Spule haben. Diese parasitären Kapazitäten können in drei wirkende Kapazitäten aufgeteilt werden. Dabei ist C_s die Kapazität zwischen den Schalterkontakten und beträgt $0,2\ pF$. C_c ist die Kapazität von einem Schalterkontakt zur Spule und beträgt $0,17\ pF$. Aus dem Dokument [10] geht auch hervor, dass die beiden Schalterkontakte fast identische Kapazitäten zur Spule haben, deswegen genügt es nur einen C_c Wert zu verwenden. In der Schaltung 5.16c sind die parasitären Kapazitäten eingezeichnet. Es wird dabei ausgegangen, dass die Relaispulen an einem Kontakt mit Masse verbunden sind. Die Spulen selber sind in der Schaltung nicht eingezeichnet. Mit der Einstellung, bei der alle Schalter offen sind, ergibt das laut Formel 5.11 eine Gesamtkapazität von $0,31\ pF$.

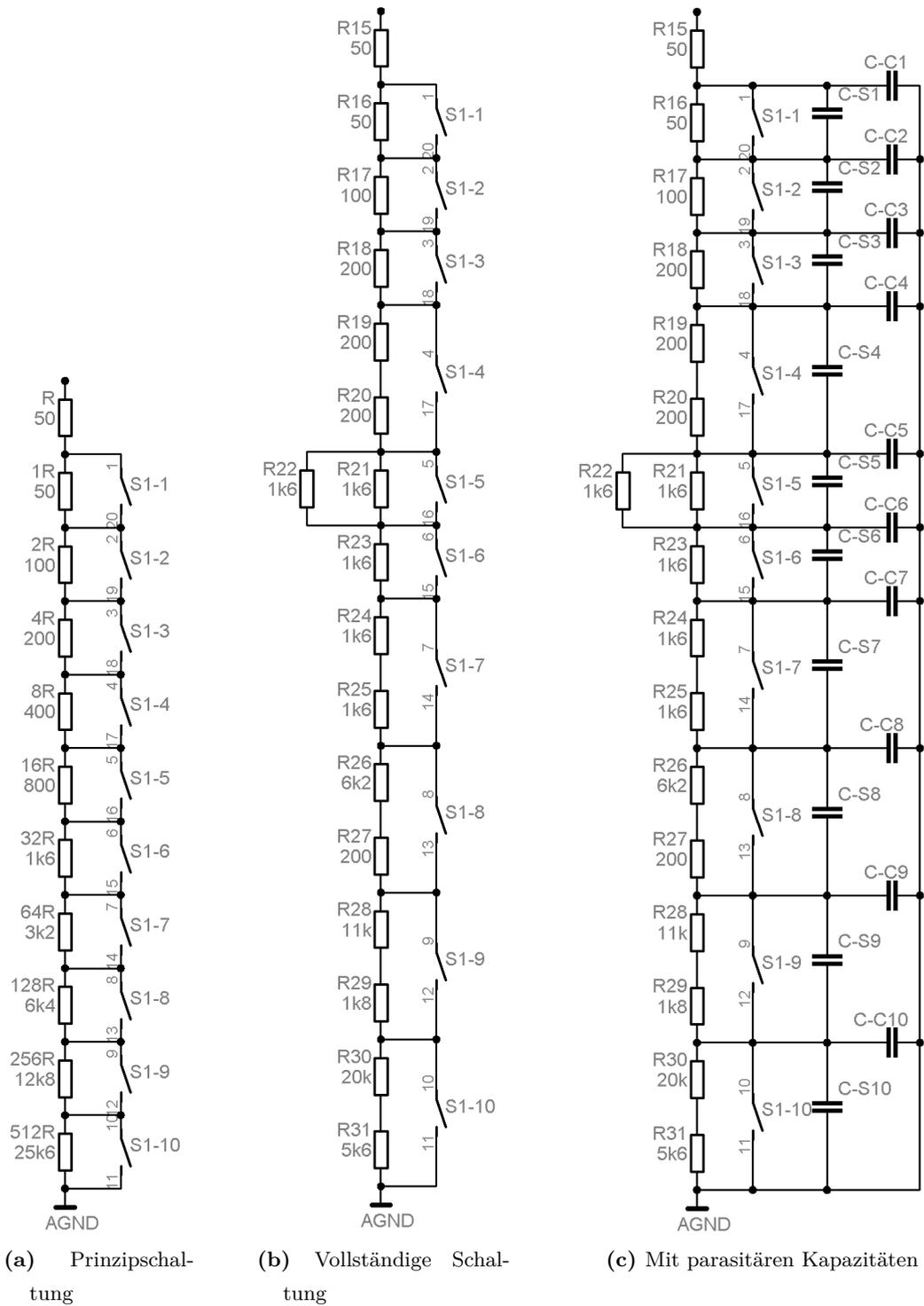


Abbildung 5.16: Widerstandsnetzwerk

auf der kontaktbasierten Schnittstelle über den I/O -Pin ausgeben kann. Damit das auch funktioniert, müssen die I/O -Pads über den V_{CC} -Pin mit einer Spannung versorgt werden. Diese Schaltung zur Erzeugung von Spannungen zwischen 1,25 V und 5 V wird im Kapitel 5.2.7.3 beschrieben. Das CL Evaluation Board stellt, wie in der Schaltung 5.17 erkennbar ist, einige nützliche Jumper und Messpunkte zur Verfügung. Am $MP2$ kann die Spannung von L_a -Pin gegen Masse gemessen werden, wobei mit $JP11$ und einer Active Differential Probe die direkte Aufnahme der Signale zwischen L_a und L_b möglich ist. Am $MP3$ wird das I/O Ausgangssignal gemessen. $JP12$ und $JP13$ dienen dazu L_b und V_{SS} mit der Masse zu verbinden. Über $MP4$ und $MP5$ können Signale auf den RES und CLK Eingang gelegt oder mit Masse verbunden werden. Mit $JP14$ lässt sich die Versorgung der kontaktbasierten Schnittstelle zu- und wegschalten.

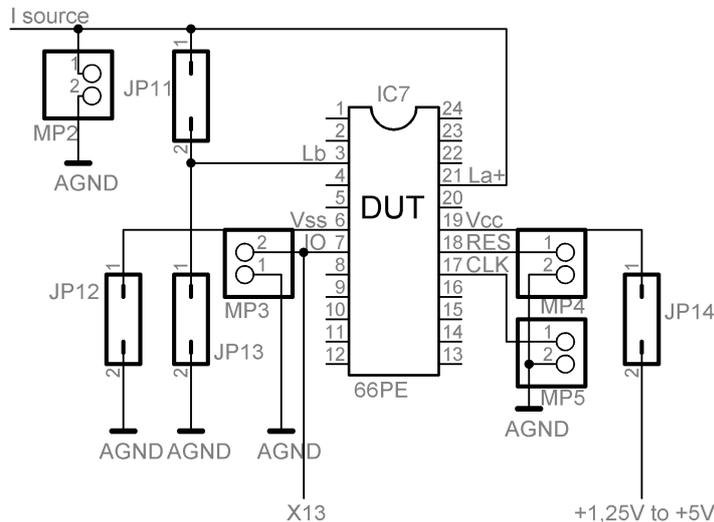


Abbildung 5.17: Dual Interface Chip

5.2.7 Spannungsversorgungen

5.2.7.1 Spannungsversorgung des DA-Evaluation Boards

Für das AD9772 Evaluation Board werden dreimal 3,3 V Spannungen für Analogteil, Takt und Digitalteil, die jeweils separat versorgt werden, benötigt. Die Massen sind sternförmig direkt unter dem AD9772 Chip verbunden. Zur Versorgung wurden drei Linearregler mit einer Festspannung von 3,3 V verwendet. Der Linearregler LM1117IMP-3.3 benötigt

als externe Bauteile nur einen Eingangs- und Ausgangskondensator, die aus dem Material Tantal bestehen sollen. Als Eingangskondensator werden $10\ \mu F$ empfohlen, wenn der Abstand zum Netzteilfilter sehr groß ist. Am Ausgang werden mindestens $10\ \mu F$ benötigt aber typischerweise auch höhere Werte verwendet. Der ESR des Kondensators soll zwischen $0,3\ \Omega$ und $22\ \Omega$ liegen. Die maximal zulässige Eingangsspannung beträgt $20\ V$. In der Schaltung 5.18a kann man sehen, dass jede Versorgung über Jumper separat zuschaltbar ist. Außerdem sind die Masseverbindungen über Jumper trennbar um eventuelle Masseschleifen aufzutrennen zu können.

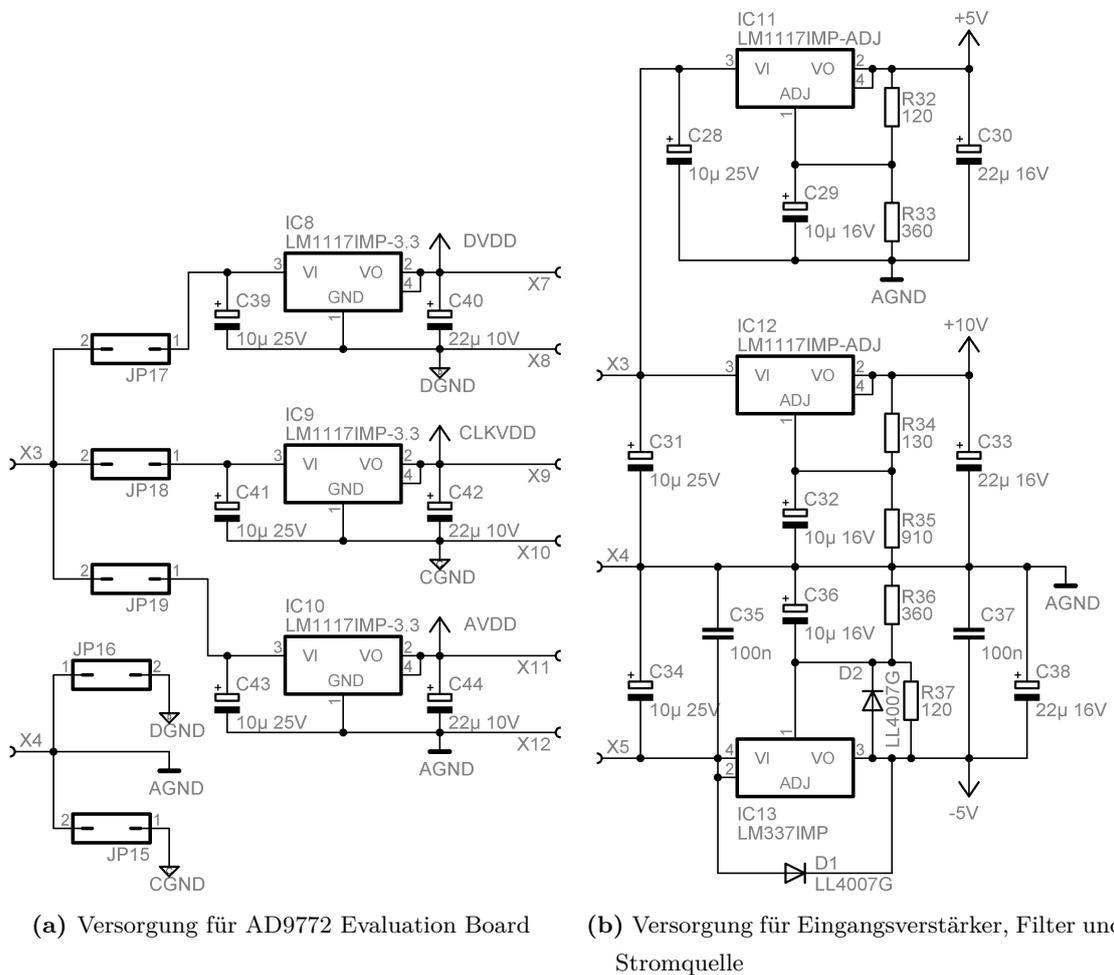


Abbildung 5.18: Versorgungen

5.2.7.2 Spannungsversorgung Eingangsverstärker, Offsetabgleich, Filter und Stromquelle

Der AD8055 auf dem AD9772 Evaluation Board und der Eingangsverstärker AD8056 benötigen eine Versorgung von minimal $\pm 4 V$, typisch $\pm 5 V$ und maximal $\pm 6 V$, daher wurde eine Versorgungsspannung von $\pm 5 V$ gewählt. Diese Spannungen werden ebenfalls für die Schaltung zum Offsetabgleich verwendet. Der für die Stromquelle verwendete AD8130 benötigt eine Versorgung von minimal $\pm 2,25 V$ und maximal $\pm 12,6 V$. Die Stromquelle muss mit $+10 V$ und $-2 V$ versorgt werden, da eine maximale Ausgangsspannung von $8 V$ erreicht werden soll und $2 V$ Reserve nötig sind. Für die negative Versorgung werden die vorhandenen $-5 V$ verwendet. Zur Erzeugung dieser drei verschiedenen Spannungen wurden einstellbare Linearregler verwendet, deren Ausgangsspannung über zwei Widerstände einstellbar sind. In der Schaltung 5.18b kann man sehen, dass für die Eingangskondensatoren wieder $10 \mu F / 25 V$ Tantalkondensatoren verwendet wurden. Für die Ausgangskondensatoren werden $22 \mu F$ empfohlen. Am Adjust-Eingang befindet sich ebenfalls ein $10 \mu F$ Tantalkondensator, der das Schwingungsverhalten vom Regelkreis des Linearreglers reduzieren soll. Beim negativen Linearregler LM337 wurden für die Kondensatoren die gleichen Werte gewählt. Zusätzlich gibt es am Ein- und Ausgang $100 nF$ Kondensatoren, die die Störungen und die Schwingungsneigung des Reglers reduzieren sollen. Zur Sicherheit sind dort noch zwei Dioden nötig, die verhindern sollen, dass beim Abschalten der Versorgung die gespeicherten Ladungen in den Kondensatoren den LM337 zerstören. Die Formel zur Dimensionierung des Spannungsteilers am Adjustpin kann man aus dem Datenblatt entnehmen und wird in der Formel 5.14 dargestellt. Daraus ergibt sich die Formel 5.15 zur Dimensionierung von R_{33} bei der Wahl von $R_{32} = 120 \Omega$. R_{32} soll laut Datenblatt zwischen 100Ω und 200Ω liegen. Beim Rückrechnen ergibt sich eine Spannung von $5,02 V$.

$$U_{OUT} = U_{REF} \left(1 + \frac{R_{33}}{R_{32}} \right) + I_{ADJ} R_{33} \quad (5.14)$$

$$R_{33} = \frac{U_{OUT} - U_{REF}}{\frac{U_{REF}}{R_{32}} + I_{ADJ}} = \frac{8 V - 1,25 V}{\frac{1,25 V}{120 \Omega} + 60 \mu A} = 357,94 \Omega \approx 360 \Omega \quad (5.15)$$

Die Berechnung von R_{34} und R_{35} erfolgt genau wie vorher und ist in der Formel 5.16 zu sehen. Es wurde für $R_{35} = 130 \Omega$ gewählt. Die Ausgangsspannung beträgt $10,05 V$.

$$R_{35} = \frac{U_{OUT} - U_{REF}}{\frac{U_{REF}}{R_{34}} + I_{ADJ}} = \frac{10 V - 1,25 V}{\frac{1,25 V}{130 \Omega} + 60 \mu A} = 904,36 \Omega \approx 910 \Omega \quad (5.16)$$

Bei dem negativen Linearregler LM337 ist die Formel ähnlich, es sind nur die Vorzeichen negativ und der Strom am Adjustpin erhöht sich auf $65 \mu A$. Für den Widerstand R_{36}

werden $120\ \Omega$ empfohlen, woraus sich eine Ausgangsspannung von $-5,02\ V$ rückrechnen lässt.

$$-U_{OUT} = -U_{REF} \left(1 + \frac{R_{36}}{R_{37}} \right) - I_{ADJ} R_{36} \quad (5.17)$$

$$R_{36} = \frac{-U_{OUT} + U_{REF}}{-\frac{U_{REF}}{R_{37}} - I_{ADJ}} = \frac{-2\ V + 1,25\ V}{-\frac{1,25\ V}{120\ \Omega} - 65\ \mu A} = 357,77\ \Omega \approx 360\ \Omega \quad (5.18)$$

5.2.7.3 Spannungsversorgung für den Dual Interface Chip

Für den Dual Interface Chip wird eine einstellbare Versorgung von $1,25\ V$ bis $5\ V$ benötigt. Die Berechnung für R_{38} und R_{39} erfolgt genau wie bei R_{32} und R_{33} und ist in der Formel 5.19 zu sehen. Es wurde für $R_{38} = 160\ \Omega$ gewählt. Da R_{39} ein Potentiometer ist, beträgt die maximale Ausgangsspannung $5,19\ V$ und die minimale Ausgangsspannung $1,25\ V$, Werte dazwischen sind kontinuierlich einstellbar.

$$R_{39max} = \frac{U_{OUT} - U_{REF}}{\frac{U_{REF}}{R_{38}} + I_{ADJ}} = \frac{5\ V - 1,25\ V}{\frac{1,25\ V}{160\ \Omega} + 60\ \mu A} = 476,34\ \Omega \approx 500\ \Omega \quad (5.19)$$

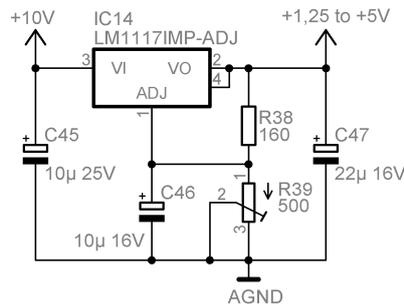


Abbildung 5.19: Einstellbare Versorgung für den Dual Interface Chip

5.2.8 Inbetriebnahme

Bei der Inbetriebnahme ist zu beachten, dass alle Jumper richtig konfiguriert werden. Die Jumperstellungen für das DA Evaluation Board wurden schon im Kapitel 5.2.1.1 beschrieben. Die Jumperstellungen für das CL Evaluation Board sind in der Tabelle 5.2 beschrieben. Der vollständige Schaltplan zum CL Evaluation Board befindet sich im Anhang E gefolgt vom Layout im Anhang F. Die hier beschriebenen Jumperstellungen sind ein Richtwert um das Evaluation Board in einer Standardkonfiguration zu betreiben.

Jumper	Stellung	Beschreibung
JP1	1-2	Leitet das Signal vom Eingangsverstärker weiter zum Filter
JP2	2-3	Das Signal vom Filterausgang wird verwendet
JP3	offen	Es wird kein externes Signal angeschlossen
JP4	1-2	Es wird der 1. U/I Wandler verwendet
JP5	1-2	Es wird der 1. U/I Wandler verwendet
JP6	offen	Der 2. U/I Wandler ist deaktiviert
JP7	offen	Der 2. U/I Wandler ist deaktiviert
JP8	offen	Der 2. U/I Wandler ist deaktiviert
JP9	offen	Der 2. U/I Wandler ist deaktiviert
JP10	1-2	Widerstandsnetzwerk wird zugeschaltet
JP11	offen	Dieser Jumper dient primär als Messanschluss
JP12	1-2	V_{SS} wird mit $AGND$ verbunden
JP13	1-2	L_b wird mit $AGND$ verbunden
JP14	1-2	V_{CC} wird mit der Versorgung verbunden
JP15	offen	Die Massen $CGND$ und $AGND$ werden hier nicht verbunden
JP16	offen	Die Massen $DGND$ und $AGND$ werden hier nicht verbunden
JP17	1-2	Der Linearregler für $DVDD$ wird versorgt
JP18	1-2	Der Linearregler für $CLKVDD$ wird versorgt
JP19	1-2	Der Linearregler für $AVDD$ wird versorgt

Tabelle 5.2: Konfiguration des CL Evaluation Boards

5.2.8.1 Verbindung zwischen DA- und CL Evaluation Board

Vor Inbetriebnahme muss das CL Evaluation Board auf das DA Evaluation Board gesteckt werden. Die dafür nötigen Verbindungen sind in der Tabelle 5.3 aufgelistet. Es werden 4mm Laborsteckerstifte zwischen den Versorgungsanschlüssen der Platinen gesteckt. Diese Verbindungen geben schon eine gute mechanische Stabilität, zusätzlich können die Platinen auch noch mit Distanzhülsen an den Ecken zusammengeschraubt werden. Zur Versorgung des Ausgangsverstärkers am DA Evaluation Board werden Prüflleitungen mit 0,64mm Buchse verwendet, die zu den 5V und -5V Pins des CL Evaluation Board führen. Das Analogsignal wird über ein SMA Kabel verbunden. Für den Digitalteil wird ein 40-poliges Flachbandkabel benötigt.

CL-Board	DA-Board	Verbindungsart	Beschreibung
X13	J9	4 mm Laborstecker	<i>AVDD</i> Versorgung
X14	J10	4 mm Laborstecker	<i>AGND</i> Versorgung
X11	J11	4 mm Laborstecker	<i>CLKVDD</i> Versorgung
X12	J12	4 mm Laborstecker	<i>CLKGND</i> Versorgung
X9	J7	4 mm Laborstecker	<i>DVDD</i> Versorgung
X10	J8	4 mm Laborstecker	<i>DGND</i> Versorgung
X1	J13	SMA-Kabel	Analog Signal
X6-1	TP20	Prüfleitung mit 0,64 mm Buchse	+8V Versorgung
X6-3	TP19	Prüfleitung mit 0,64 mm Buchse	-2V Versorgung
X13	P1	Flachbandkabel 40 polig	Digitalverbindung

Tabelle 5.3: Verbindung des CL Evaluation Board mit dem DA Evaluation Board

5.2.8.2 Verbindung zwischen CL-Evaluation Board und Spea Load Board

Für die Verbindung zum Load Board L306E von Spea werden Prüfleitungen mit 0,64 mm Buchsen verwendet. Die Pinbelegung für diese Verbindungen sind aus der Tabelle 5.4 zu entnehmen.

CL-Board	Load Board	Verbindungsart	Beschreibung
X14-1	CH112	Prüfleitung mit 0,64 mm Buchse	DUT I/O
X14-2	GND	Prüfleitung mit 0,64 mm Buchse	GND
X14-3	CH113	Prüfleitung mit 0,64 mm Buchse	DAC Reset Input
X14-4	CH114	Prüfleitung mit 0,64 mm Buchse	DAC Clock Input
X14-5			GND
X14-6			GND
X14-7	CH115	Prüfleitung mit 0,64 mm Buchse	DAC Data Input DB0
X14-8	CH116	Prüfleitung mit 0,64 mm Buchse	DAC Data Input DB1
...
X14-20	CH128	Prüfleitung mit 0,64 mm Buchse	DAC Data Input DB13

Tabelle 5.4: Verbindung des CL Evaluation Board mit dem Spea Load Board

5.2.8.3 Eingänge und Messpunkte

Versorgt wird das CL-Evaluation Board am Stecker X3 mit $12V$, X4 mit GND und X5 mit $-7V$. Das analoge Eingangssignal wird an X1 angeschlossen. Um den Eingangverstärker mit Offsetabgleich zu überbrücken kann über X2 ein Analogsignal direkt an den U/I-Wandler gelegt werden. Weitere Eingänge und Messpunkte sind in der Tabelle 5.5 beschrieben.

Messpunkt	Messsignalbeschreibung
MP1	Ausgangssignal des Filters bzw. Eingangssignal des U/I-Wandlers
MP2	Ausgangssignal des U/I-Wandlers bzw. L_a -Pin
MP3	I/O -Ausgang
MP4	RES -Pin kann als Eingang verwendet werden
MP5	CLK -Pin kann als Eingang verwendet werden
MP6	Versorgung $+5V$
MP7	Versorgung $-5V$

Tabelle 5.5: Beschreibung der Messpunkte

5.2.9 Messergebnisse

Vor der endgültigen Inbetriebnahme wurden noch ein paar Messungen durchgeführt, um eventuelle Fehler in den Berechnungen oder im Aufbau zu finden. Im ersten Schritt wurden alle Versorgungsspannungen gemessen. Die Platine wurde dabei von der Dual Output DC Power Supply E3646A von Agilent mit $+12V$ und $-7V$ versorgt. Zur Messung wurde ein Fluke 187 Multimeter verwendet. In der Tabelle 5.6 sind alle Messergebnisse aufgelistet.

Weiters sind noch ein paar Kenndaten der U/I-Wandler interessant. Gemessen wurden diese mit dem Digital Phosphor Oscilloscope DPO7104 1 GHz der Firma Tektronix. Wobei als Tastkopf am Ausgang der Stromquelle eine 1 GHz Tektronix Differential Probe mit der Bezeichnung TDP1000 verwendet wurde. Als Signalgenerator diente der 1,2 GS/s Waveform Generator WW1281A von Tabor. Zur Versorgung wurde eine Dual Output DC Power Supply E3646A von Agilent verwendet. Die Messergebnisse sind in der Tabelle 5.7 aufgelistet und wurden mit einem 50Ω Lastwiderstand gemessen. Beim Ermitteln der Verstärkung, sowie der $-3dB$ Grenzfrequenz, wurde ein sinusförmiges Eingangssignal von $1V_{pp}$ mit einem Offset von $0,5V$ verwendet. Das Signal wurde bei dem SMA-Anschluss X2 eingespeist

Messpunkt	berechnet	gemessen	Messsignalbeschreibung
AVDD	3,3 V	3,30 V	AVDD Versorgung
CLKVDD	3,3 V	3,29 V	CLKVDD Versorgung
DGND	3,3 V	3,29 V	DVDD Versorgung
MP6	5,02 V	5,04 V	Versorgung +5 V
MP7	-5,02 V	-5,03 V	Versorgung -5 V
IC12-4	10,05 V	10,03 V	Versorgung 10 V
JP14	1,25 V	1,25 V	minimale Versorgung DUT
JP14	5,19 V	5,00 V	maximale Versorgung DUT
IC1-3	0 V	0,07 mV	minimale Offsetspannung
IC1-3	0,62 V	640,6 mV	maximale Offsetspannung

Tabelle 5.6: Messergebnisse Versorgung

Anzahl der U/I-Wandler:	1	2	3
Verstärkungsfaktor bei 1 MHz:	1,00	2,00	3,00
-3 dB Grenzfrequenz:	56 MHz	46,5 MHz	39 MHz
Verstärkung und Frequenz bei 45° Phasenverschiebung			
Verstärkungsfaktor:	1,16	2,23	3,23
Frequenz:	13,1 MHz	11,6 MHz	10,3 MHz
Maximale Überhöhung der Verstärkung			
Verstärkungsfaktor:	1,57	2,80	3,77
Frequenz:	38 MHz	30 MHz	23 MHz
Phasenverschiebung:	180°	150°	125°
Rechteck Anstiegszeit und Abfallzeit bei 1 MHz			
Laststrom:	20 mA	40 mA	60 mA
Anstiegszeit:	1,55 ns	3,12 ns	4,39 ns
Abfallzeit:	1,65 ns	2,15 ns	4,03 ns
Laststrom:	40 mA	80 mA	120 mA
Anstiegszeit:	2,03 ns	3,54 ns	5,09 ns
Abfallzeit:	1,75 ns	2,33 ns	4,40 ns

Tabelle 5.7: Messdaten der U/I-Wandler

und bei dem Messpunkt *MP2* mit der Tektronix Differential Probe abgegriffen. Weiters wurden Verstärkung und Frequenz bei einer Phasenverschiebung von 45° zwischen Ein-

und Ausgang ermittelt. Bedingt durch das Regelverhalten des U/I-Wandlers gibt es eine Überhöhung in der Verstärkung, von der die Verstärkung sowie Frequenz und Phasenlage ermittelt wurde. Außerdem sind noch Anstiegszeiten und Abfallzeiten unter Verwendung von Rechtecksignalen von Interesse. Die Anstiegszeiten werden von 10% bis 90% und die Abfallzeiten von 90% bis 10% der Signalamplitude gemessen, wobei Überschwinger nicht berücksichtigt werden. Hierfür wurde ein Rechtecksignal mit einer Spannung von 0 bis 1 V als Eingangssignal verwendet, um Ausgangsströme von 20 mA bis 60 mA zu erreichen. Für die Ausgangsströme von 40 mA bis 120 mA wurde diese Spannung auf 0 bis 2 V erhöht. Beim Eingangssignal, vom Waveform Generator kommend, wurde eine Anstiegszeit von 0,73 ns und eine Abfallzeit von 0,71 ns, bei einer Amplitude von 1 V, ermittelt.

Weiters sind noch die Grenzfrequenzen von Eingangsverstärker und Filter von Interesse. Hierfür wurde ein sinusförmiges Signal mit einer Spannung von 1 V_{pp} an dem SMA-Buchse X1 angeschlossen. Für die Messung des Eingangsverstärkers wurde an der SMA-Buchse X2 abgegriffen und für den Filter die Differential Probe am Messpunkt MP1 verwendet. Die gemessene -3 dB Grenzfrequenz des Eingangsverstärkers beträgt 21 MHz und des Eingangsverstärkers plus Filter 17,1 MHz. Zu guter Letzt wurde noch versucht die Filterdämpfung zu ermitteln, was daran scheiterte, dass schon bei 50 MHz das Signal im Rauschen unterging.

5.3 Weitere Schritte

Der nächste Schritt ist die Erweiterung der Schaltung auf Mehrkanalbetrieb, wodurch sich die Verifikationsdauer oder Charakterisierungsdauer um den Faktor $1/n$ reduziert, dabei ist n die Anzahl der gleichzeitig gemessenen Chips. Hierfür müssen nur die U/I-Wandler und das Widerstandsnetzwerk n mal ausgeführt werden und die Versorgungen für die U/I-Wandler und die DUTs müssen für die n -fache Stromaufnahme dimensioniert werden. Sollte die Anzahl der gemessenen Chips sehr groß werden, muss natürlich der Treiber der letzten Filterstufe für die große Menge der U/I-Wandler ausgelegt sein und bei langen Leitungen der Wellenwiderstand der Leitung beachtet und die Leitung terminiert werden. Hierfür würde sich das System, wie es bei 10Base2 Ethernet verwendet wird, eignen. Bei diesem System ist nur das erste und das letzte Glied der Kette auf 50 Ω Wellenwiderstand des Kabels terminiert, alle anderen Teilnehmer, die am Netz lauschen, sind hochohmig und haben somit keinen Einfluss auf die Leitung, daher wird verhindert, dass sich Reflexionen bilden können.

Um verschiedene Widerstandswerte automatisiert verwenden zu können, macht es Sinn das Widerstandsnetzwerk über Relais umzuschalten. Geht man von einem Tester wie den C320MX aus, stehen 128 Digitalkanäle zu Verfügung, wovon 15 Kanäle für den DA-Umsetzer benötigt werden. Die restlichen 113 Kanäle können für die Verifikation oder Charakterisierung von bis zu 113 Chips parallel verwendet werden. Im Blockdiagramm 5.20 der Mehrkanalausführung sind Bereiche, die einfach ausgeführt werden und Teile, die n-fach ausgeführt werden müssen, ersichtlich. Bei dieser stark steigenden Anzahl der Bau-

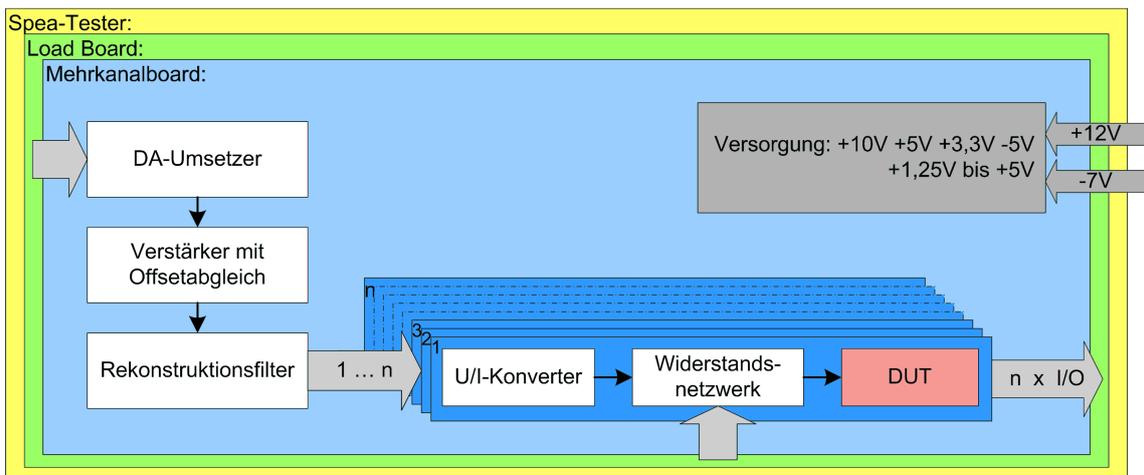


Abbildung 5.20: Blockdiagramm Mehrkanalausführung

teile ist es sinnvoll sich über Vereinfachungen der Schaltung Gedanken zu machen. Das größte Einsparungspotential ist hier beim Widerstandsnetzwerk gegeben. Es wäre zu überlegen, ob es nicht reicht auf wenige Widerstandswerte zu reduzieren, da sowieso niemals alle 1024 möglichen Werte verwendet werden.

6 Spea Chiptester

Beim verwendeten Halbleiter Wafer Tester für digitale und mixed signal ICs handelt es sich um das Modell „Comptest 320MX“ der Firma Spea. Die Grundidee solcher Testmaschinen ist, dass der Chip noch am Wafer getestet werden kann, ohne den Aufwand den Chip zu sägen und in ein Gehäuse einbauen zu müssen. Ein weiterer großer Vorteil ist, dass solche Testmaschinen mit sehr vielen Testkanälen ausgestattet sind und somit sehr viele Chips parallel geprüft werden können. Außerdem wechselt der Wafer Prober selbständig die Wafer, was zusätzlich Standzeiten reduziert und Personal einspart.



Abbildung 6.1: Spea - C320MX¹

Die Grundfunktionalität des Testers besteht darin, digitale Signale auszugeben und empfangene Signale zu vergleichen. Dazu steht ein 128 Kanal Bitmustergenerator zur Verfügung. Außerdem kann ein Tester auch andere interne und externe Geräte steuern. Solche typischen Geräte sind Versorgungen, Prober, Klimagerät und Messgeräte.

Die Ausstattungsmerkmale des Spea Comptest C320MX ¹

- Channels: 128
- Pattern Frequency: 40 MHz
- Clock Frequency: 50 MHz
- Pattern generation
- RAM: Up to 2 Mstep x PIN
- Programmable Logic per PIN: Yes
- High Precision PMU per PIN: 128 ch
- Digitizer: Up to 32 ch (4 ch x 8)
- Arbitrary Waveform Generator: Up to 32 ch
- Counter: 8 Units
- DC Source: 6 units (up to 40 V)
- RF Source: 1 (2 GHz)

Zusätzlich zu diesen Ausstattungsmerkmalen gibt es auch noch 64 OutC Leitungen, die zum Ansteuern von Relais dienen. Die vorhandene User Power Supply kann leider nicht verwendet werden, da sie nur Ströme bis 0,3 A liefert. Der integrierte AWG ist leider für unsere Bedürfnisse auch nicht ausreichend, da er auf 1 MHz limitiert ist. Sollten die 128 Testerkanäle nicht ausreichen kann über externe Hardware noch einmal um 128 Kanäle erweitert werden.

Ein gravierendes Problem dieses Chiptesters ist seine 40 MHz Patternfrequenz, da bei zukünftigen Datenraten von bis zu 27,12 Mbit das Abtasttheorem nicht eingehalten werden kann.

¹C320MX - Mixed Signal Semiconductor Tester

6.1 Das Comptestsystem

Wie in der Abbildung 6.2 zu sehen ist setzt sich das System aus mehreren Komponenten zusammen. Direkt auf dem Tester wird das Load Board montiert. Dabei führt der Tester

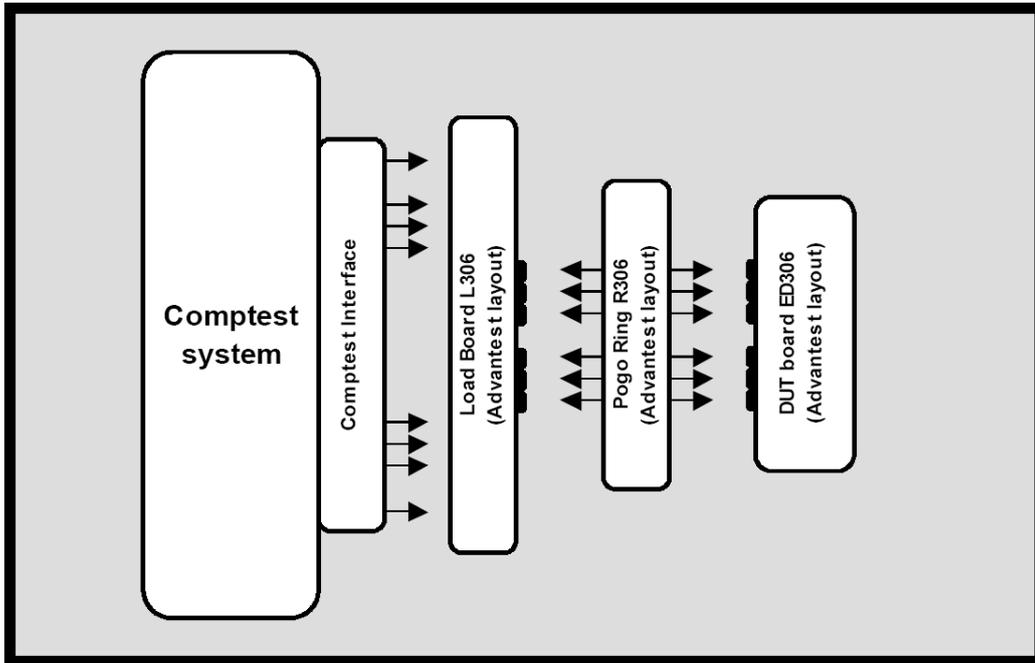


Abbildung 6.2: Comptestsystem[11]

alle Leitungen des Comptest Interface über gefederte Kontakte nach außen. Auf dem Load Board können noch Schaltungen dazwischen geschaltet werden, die Manipulationen an den Signalen vornehmen bevor sie weiter über den Bogo Ring und die Nadelkarte an den Chip gelegt werden. Die Schaltung aus Kapitel 5 setzt auch auf dem Load Board auf, aber es werden zum Testen noch kein Bogo Ring und keine Nadelkarte verwendet, da hierfür erst ein eigenes Load Board entwickelt werden müsste. Das passiert erst dann wenn eine Schaltung so ausgereift ist, dass sicher nichts mehr an ihr geändert werden muss. Der Bogo Ring ist nur ein Ring mit gefederten Kontakten auf beiden Seiten und verbindet das Load Board mit der Nadelkarte. Die Nadelkarte kann auch kleine Schaltungen beinhalten und hat auf der einen Seite die Kontakte für den Bogo Ring und auf der anderen Seite die feinen Nadeln zum Kontaktieren der Bonding Pads am Chip. Da jeder Chip eine andere Anordnung seiner Pads hat, muss diese Nadelkarte für jeden Chip neu entwickelt werden. Die Kontaktherstellung zum Wafer funktioniert nicht dadurch, dass die Nadelkarte auf den Wafer aufsetzt, sondern der Wafer wird vom Prober bewegt. Der Prober ist eine Maschine

zum Wechseln und Positionieren des Wafers. Dabei kann ein ganzes Los (25 Wafer) in einer Box deponiert werden. Die Wafer werden dann automatisch gewechselt und in x, y und z Richtung zur Nadelkarte positioniert. Ist ein Wafer einmal eingerichtet, findet die Repositionierung automatisch statt. Außerdem kann ein Prober auch ein Klimagerät beinhalten, somit können die Tests auch bei verschiedenen Temperaturen durchgeführt werden. Wie in Abbildung 6.1 zu sehen ist, ist der Teil des Testers, auf dem Load Board, Bogo Ring und Nadelkarte aufsetzen, in zwei Achsen schwenkbar und höhenverstellbar, denn dieser Teil wird auf den Prober aufgesetzt.

6.2 Load Board L306E

Das Load Board stellt die Verbindung zwischen Tester und Nadeladapter, der auch umgangssprachlich Nadelkarte bezeichnet wird, dar. In der Abbildung 6.3 sieht man den Bestückungsplan des in dieser Arbeit verwendeten Load Boards.

Spezifikation des L306E:

- 4 lagige Platine
- Kanalimpedanz beträgt 50Ω
- 3 Reihen Servicebus
- 256 mögliche 2 pin Komponenten im SMD Gehäuse 1206
- 64 mögliche Relays
- 8 Bit Jumper für Load Board Kodeerkennung
- 128 System channel pads
- 128 Extended channel pads
- 496 Messgeräte und Service I/O pads
- 8 External clock pads
- System Kompatibilität: Comptest C300MX, Comptest C320MX, Comptest C340MX.

Kontakte auf Pogo Ring Seite:

- 128 System channels pads
- 128 Extended channels pads

- 32 Available pads
- Pogo ring compatibility R306

Weitere Informationen zu dem Loadboard können aus dem Dokument „L306 UserGuide.pdf“[11] entnommen werden, welches sich bei den „Comptest MX series system manuals“ befindet.

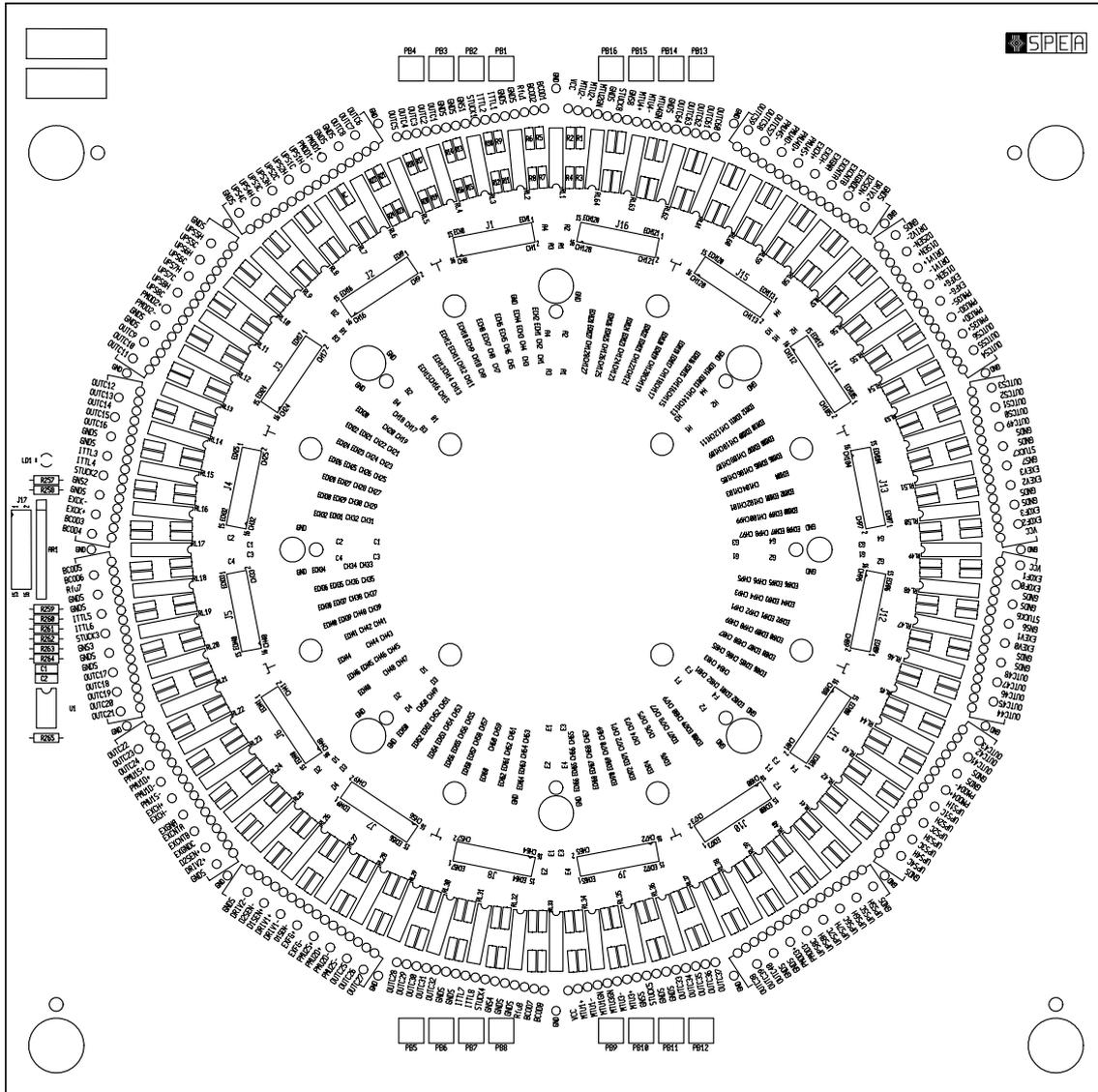


Abbildung 6.3: LoadBoard L306[11]

7 Software

Die Software besteht aus mehreren Teilen. Das Analogsignal, welches der Digital-Analog Umsetzer ausgeben soll, wird zuerst mit Matlab berechnet und in eine Tabor *.wav* Datei ausgegeben. Diese Datei kann verwendet werden, um Analogsignale direkt auf einem Tabor Arbitrary Waveform Generator auszugeben. Um diese auf einem Spea-Tester verwenden zu können, wird sie mit dem Programm *Tabor2Spea.exe* in die Spea PAGEL-C *.p40* Datei konvertiert. Mit dem *Vector Explorer* von Spea kann die generierte *.p40* Datei kompiliert werden. Dabei entsteht ein für den Tester verwendbares Testpattern. Solche Testpattern werden mit dem Programm *VectorView IDE* grafisch in einem Zeitdiagramm dargestellt. Aus der Spea Oberfläche *Comptest MX* kann mit dem Button „Test management“ das Prüfprogramm gestartet werden, mit dem der Tester initialisiert, die Pattern hochgeladen und der Prüfablauf gesteuert wird.

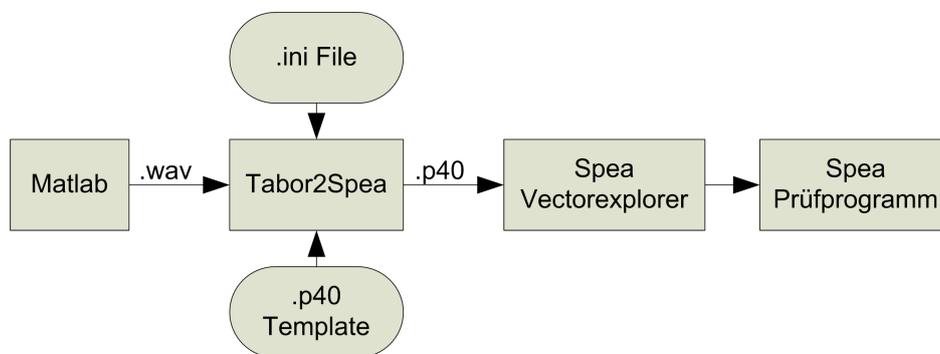


Abbildung 7.1: Software Ablauf

7.1 Tabor zu Spea Konvertierungssoftware

Dieses Programm konvertiert Tabor *.wav* Dateien in Spea *.p40* Dateien. Diese Tabor *.wav* Dateien sind Binärdateien, in denen die Wellenformen ähnlich wie Audio *.wav* Dateien

als Folge von zeit- und wertdiskreten Amplitudenwerten (Samples) gespeichert sind. Diese Samples haben eine Auflösung von 14 Bit, werden aber trotzdem in 16 Bit Wörtern gespeichert, wobei die obersten 2 Bits unbenutzt bleiben. Das oberste nächste Bit wird dabei als Vorzeichen verwendet und muss gegebenenfalls entfernt werden, da hier nur Hüllkurven verwendet werden. Die Abtastrate wird wie beim Tabor AWG auch beim Spea Tester direkt am Gerät eingestellt, wobei schon beim Generieren der *.wav* Dateien das Nyquist-Shannonsche Abtasttheorem von $f_{sample} > 2 \cdot f_{max}$ zu beachten ist. Eine üblich verwendete Samplefrequenz für den Tabor AWG ergibt sich aus $f_{sample} = 13,56 \text{ MHz} \cdot 7 = 94,92 \text{ MHz}$. Der Spea Tester benötigt spezielle Pattern Dateien, in denen die Signale für den D/A-Umsetzer aus Folge von High- und Low-Level definiert werden. Aufgerufen wird das Programm in einem „Command Line Window“ (DOS prompt) unter Windows. Eine kleine Hilfe zur Verwendung des Programms wird ausgegeben, wenn die Anzahl der Übergabeparameter eine Falsche ist, wie in der Abbildung 7.2 zu sehen ist.

```

C:\WINDOWS\system32\cmd.exe
E:\Diplomarbeit\Roy Diploma Thesis\sources>tabor2spea
Converts Tabor ".wav"-File to Spea ".p40"-File
ERROR: Wrong number of arguments

Usage: tabor2spea INI-FILE
       or: tabor2spea SOURCE-FILE DESTINATION-FILE TEMPLATE-FILE TABOR-FREQUENCY
              SPEA-FREQUENCY AMPLITUDE CURRENT DUMMY

SOURCE-FILE      Tabor wave-file
DESTINATION-FILE Spea pattern-file
TEMPLATE-FILE    Spea template pattern-file
TABOR-FREQUENCY  Tabor wave-file sample rate [MHz]
SPEA-FREQUENCY   Maximum spea tester frequency [MHz]
AMPLITUDE        Amplitude factor of wave-file: max_level/full_scale e.g.:0.85
CURRENT          Current factor: ouput_current/max_current e.g.:0.1
DUMMY            [yes/no] for generating an empty dummy pattern

for example:
  tabor2spea CL-emu.ini
or:
  tabor2spea pattern.wav pattern.p40 CL_template.p40 94.92 31.64 0.85 0.1 no
E:\Diplomarbeit\Roy Diploma Thesis\sources>

```

Abbildung 7.2: Hilfe Ausgabe von *Tabor2Spea.exe*

Aus entwicklungsgeschichtlichen Gründen gibt es zwei Möglichkeiten zu einer *.p40* Datei zu gelangen. Die eine ist über eine *.ini*-Datei und die andere durch die Angabe von acht Parametern. Die zweite Möglichkeit wurde erst später eingeführt und ist die bessere Variante, da sie sehr gut für Stapelverarbeitungsprogramme, wie *.bat* Dateien, zu verwenden ist.

Das Programm benötigt folgende Parameter:

SOURCE-FILE	Tabor <i>.wav</i> Datei
DESTINATION-FILE	Spea <i>.p40</i> Datei
TEMPLATE-FILE	Spea <i>.p40</i> Template Datei 7.1.4 mit den Voreinstellungen
TABOR-FREQUENCY	Sample Frequenz der Tabor <i>.wav</i> Datei in <i>MHz</i>
SPEA-FREQUENCY	Sample Frequenz der Spea <i>.p40</i> Datei in <i>MHz</i>
AMPLITUDE	Amplitudenfaktor zum Kalibrieren des Maximal Levels ≤ 1
CURRENT	Stromfaktor gibt den Strom abhängig von Maximal Level an ≤ 1
DUMMY	Ermöglicht eine <i>.p40</i> Datei ohne Signal zu erzeugen

Die grundsätzliche Funktion des Programms ist es, eine Template *.p40* Datei als Grundgerüst zu verwenden, einzelne Werte zu ersetzen und die für Spea konvertierte Tabor *.wav* einzufügen und wieder als *.p40* Datei abzuspeichern. Die von dem Programm benötigten Sample-Frequenzen dienen dazu, um eine eventuelle Unterabtastung durchzuführen, da standardmäßig 94,92 *MHz* für den Tabor AWG verwendet werden und der Spea Chiptester nur maximal 40 *MHz* ausgeben kann. Da hier keine komplizierten Filter verwendet, sondern nur einzelne Samples ausgelassen werden, muss beim Erstellen der *.wav* Dateien in Matlab besonders auf das Nyquist-Shannonsche Abtasttheorem geachtet werden, oder die Dateien müssen gleich mit der richtigen Abtastrate erstellt werden.

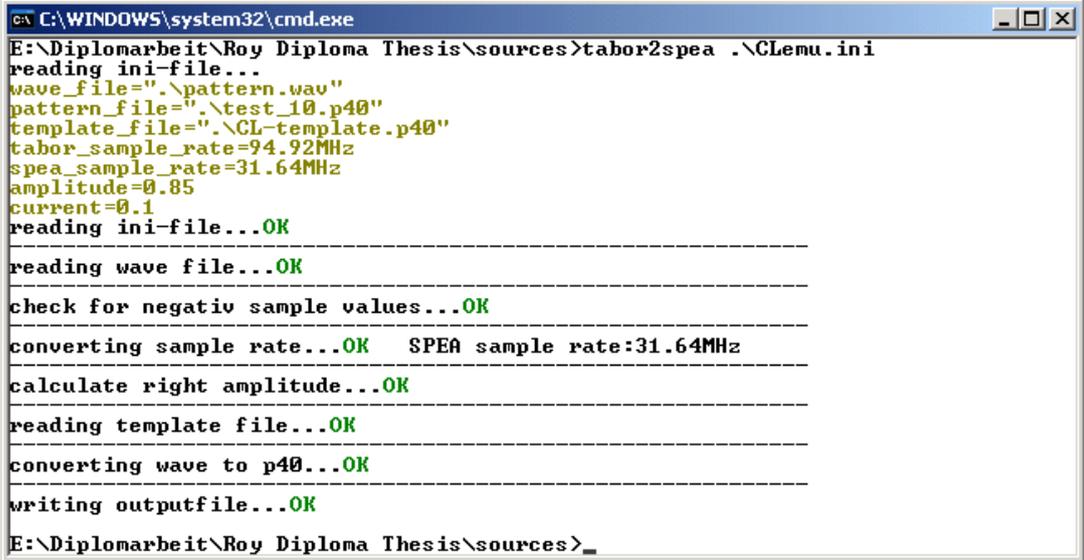
Mit dem Amplitudenfaktor und dem Stromfaktor werden die Signale auf den richtigen Amplitudenwert skaliert. Dabei werden die beiden Faktoren miteinander und dem Signal multipliziert. Die Wahl von zwei Faktoren hat den Vorteil, dass mit dem Amplitudenfaktor der Maximalwert z.B.: 10 *A/m* kalibriert werden kann und dass mit einem Stromfaktor von 0,1 ein Ausgabewert von 1 *A/m* erzeugt wird.

Die Dummy-Funktion ermöglicht das Erzeugen einer *.p40* Datei ohne Signal. Die restlichen Funktionen wie Chipochlauf und Einstellungen aus der Template Datei bleiben erhalten. Damit kann am Tester bei einem bereits gestarteten Chip sehr schnell zwischen verschiedenen Signalen umgeschaltet werden, da der Chipochlauf doch mehrere *ms* dauert.

7.1.1 Verwendung der Software mit einer *.ini*-Datei

Die anfängliche Idee eine *.ini* Datei zu verwenden beruht darauf, dass mit der Funktion `GetPrivateProfileString` sehr komfortabel Daten aus einer Textdatei ausgelesen werden können. Es hat sich in der Praxis herausgestellt, dass die nötigen Parameter nicht so viele sind, aber viele Dateien erstellt werden müssen, sodass es praktischer ist das Programm

einfach öfters aufzurufen und nicht diese Funktionalität in das Programm einzubauen. Die *CLemu.ini*-Datei hat natürlich noch andere Daten gespeichert, die von dem Spea-Prüfprogramm benötigt werden. Im Anhang H.3 ist so eine *.ini*-Datei zu sehen. Unter Sektion [General] werden allgemeine Informationen abgelegt, die primär für das Prüfprogramm von Bedeutung sind. Mit dem Schlüssel *wave* in der Sektion [General] wird ausgewählt, in welcher Sektion die Daten für das Programm *Tabor2Spea.exe* zu finden sind. Diese Sektion muss dann mit dem Schlüssel *type=tabor wave* beginnen. Alternativ kann auch ein Schlüssel *type=predefined shape* definiert sein, mit dessen Daten das Prüfprogramm direkt ein Testsignal erzeugen kann. Weiters gibt es die Schlüssel *wave_file*, *pattern_file*, *template_file*, *tabor_sample_rate*, *amplitude*, *current*, *dummy*. Der Schlüssel *spea_sample_rate* befindet sich in der Sektion *spea*. In der Abbildung 7.3 ist so eine typische Ausgabe des Programms bei Verwendung einer *.ini*-Datei zu sehen.



```

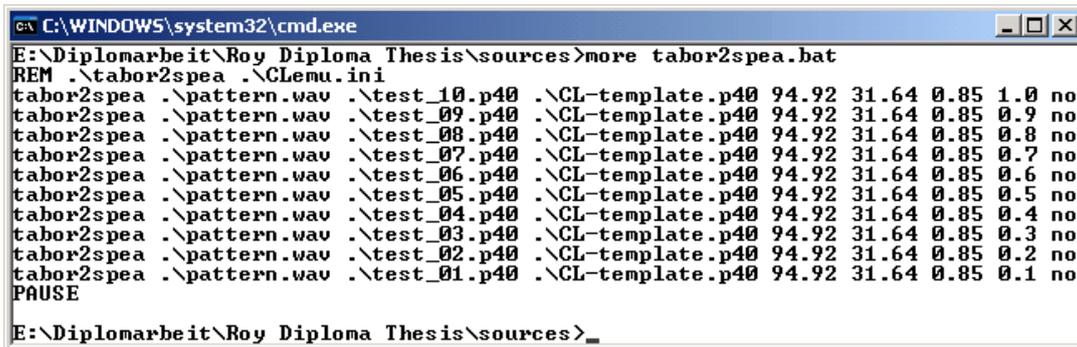
C:\WINDOWS\system32\cmd.exe
E:\Diplomarbeit\Roy Diploma Thesis\sources>tabor2spea .\CLemu.ini
reading ini-file...
wave_file=". \pattern.wav"
pattern_file=". \test_10.p40"
template_file=". \CL-template.p40"
tabor_sample_rate=94.92MHz
spea_sample_rate=31.64MHz
amplitude=0.85
current=0.1
reading ini-file...OK
-----
reading wave file...OK
-----
check for negativ sample values...OK
-----
converting sample rate...OK   SPEA sample rate:31.64MHz
-----
calculate right amplitude...OK
-----
reading template file...OK
-----
converting wave to p40...OK
-----
writing outputfile...OK
E:\Diplomarbeit\Roy Diploma Thesis\sources>_

```

Abbildung 7.3: *Tabor2Spea.exe* mit einer *.ini*-Datei aufgerufen

7.1.2 Verwendung der Software mit einer *.bat*-Datei

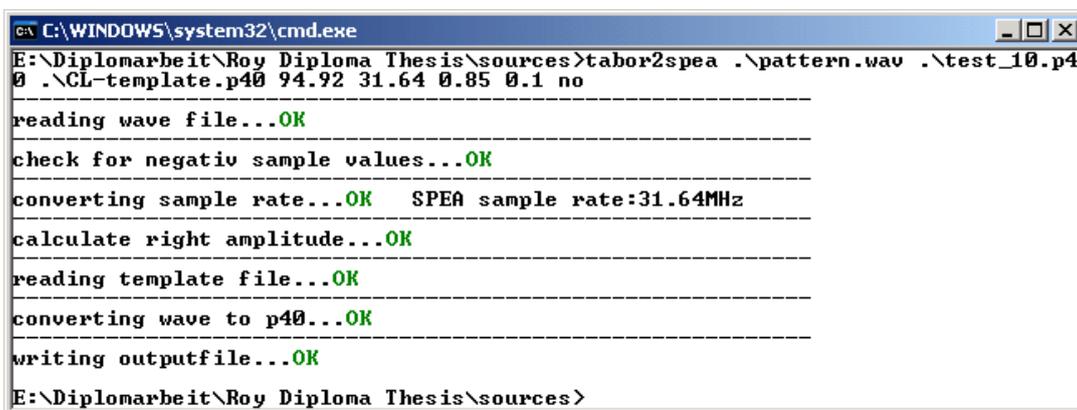
Nachdem immer mehrere Pattern Dateien mit verschiedenen Feldstärken generiert werden müssen, fiel die Entscheidung ein Stapelverarbeitungsprogramm zu verwenden und die Parameter gleich direkt zu übergeben. So eine typische *.bat* Datei ist in der Abbildung 7.4 zu sehen, mit welcher sehr viele Dateien mit einem Aufruf konvertiert werden können. Bei dem Aufruf solch einer *.bat*-Datei sieht die Ausgabe wie in Abbildung 7.5 aus.



```

C:\WINDOWS\system32\cmd.exe
E:\Diplomarbeit\Roy Diploma Thesis\sources>more tabor2spea.bat
REM .\tabor2spea .\CLemu.ini
tabor2spea .\pattern.wav .\test_10.p40 .\CL-template.p40 94.92 31.64 0.85 1.0 no
tabor2spea .\pattern.wav .\test_09.p40 .\CL-template.p40 94.92 31.64 0.85 0.9 no
tabor2spea .\pattern.wav .\test_08.p40 .\CL-template.p40 94.92 31.64 0.85 0.8 no
tabor2spea .\pattern.wav .\test_07.p40 .\CL-template.p40 94.92 31.64 0.85 0.7 no
tabor2spea .\pattern.wav .\test_06.p40 .\CL-template.p40 94.92 31.64 0.85 0.6 no
tabor2spea .\pattern.wav .\test_05.p40 .\CL-template.p40 94.92 31.64 0.85 0.5 no
tabor2spea .\pattern.wav .\test_04.p40 .\CL-template.p40 94.92 31.64 0.85 0.4 no
tabor2spea .\pattern.wav .\test_03.p40 .\CL-template.p40 94.92 31.64 0.85 0.3 no
tabor2spea .\pattern.wav .\test_02.p40 .\CL-template.p40 94.92 31.64 0.85 0.2 no
tabor2spea .\pattern.wav .\test_01.p40 .\CL-template.p40 94.92 31.64 0.85 0.1 no
PAUSE
E:\Diplomarbeit\Roy Diploma Thesis\sources>_

```

Abbildung 7.4: *Tabor2Spea.bat* Stapelverarbeitungsprogramm


```

C:\WINDOWS\system32\cmd.exe
E:\Diplomarbeit\Roy Diploma Thesis\sources>tabor2spea .\pattern.wav .\test_10.p40
0 .\CL-template.p40 94.92 31.64 0.85 0.1 no
-----
reading wave file...OK
-----
check for negativ sample values...OK
-----
converting sample rate...OK   SPEA sample rate:31.64MHz
-----
calculate right amplitude...OK
-----
reading template file...OK
-----
converting wave to p40...OK
-----
writing outputfile...OK
E:\Diplomarbeit\Roy Diploma Thesis\sources>

```

Abbildung 7.5: *Tabor2Spea.exe* mit einer *.bat*-Datei aufgerufen

7.1.3 Aufbau der Software

Das Programm ist mit dem GNU C++ Compiler in der Cygwin Umgebung Kompiliert worden, deswegen muss in dem Ordner, aus dem *Tabor2Spea.exe* aufgerufen wird, auch die *cygwin1.dll* vorhanden sein. Der GNU C++ Compiler wurde gewählt da er zur Verfügung stand und für diese Aufgabe mehr als ausreichend war. Der Quelltext des Programms befindet sich im Anhang H.1. Das Programm beginnt typisch mit den diversen **#include**-Anweisungen, in welchen die Libraries eingebunden werden, gefolgt von den Konstanten-Definitionen. Diese Konstanten können in mehrere Gruppen geteilt werden.

<code>//Return values</code>	Fehler die von Funktionen zurückgegeben werden
<code>//Arguments</code>	Übergebene Kommandozeilenargumente
<code>//SPEA and Tabor settings</code>	Initialisierungswerte von Variablen
<code>//Ini file read buffer</code>	Lesebuffergröße zum Auslesen der .ini Datei
<code>//Data conversion & set endian</code>	Umschaltung zwischen big- und little endian
<code>//Colors</code>	Ausgabefarben der Kommandozeile

```
//Template find points
```

Strings, nach denen im Template File gesucht wird

Mit dem **struct** Parameter werden alle übergebenen Argumente gespeichert, wie die Namen der Ein- und Ausgabedateien, sowie Abtastraten und Verstärkungsfaktoren. Gefolgt von den Deklarationen der einzelnen Funktionen. Das Hauptprogramm ist so strukturiert, dass der Ablauf des Programms sehr gut erkennbar ist. Dieser ist in groben Zügen im Flussdiagramm 7.6 zu sehen. Am Anfang vom `main` wird die Anzahl der übergebenen Argumente geprüft, sind es neun, werden die Parameter direkt übernommen und im **struct** Parameter abgelegt. Sind die Argumente nur zwei, wird davon ausgegangen, dass eine `.ini` Datei übergeben und somit die Funktion `readIniFile` aufgerufen wurde, die die Daten aus der `.ini` Datei extrahiert und ebenfalls in das **struct** Parameter speichert. Ist die Anzahl der übergebenen Argumente keine von diesen gewesen, wird ein Text ausgegeben, der in Abbildung 7.2 zu sehen ist. Dieser Text dient als Hilfe für den Anwender, um fehlerhafte Eingaben richtig interpretieren zu können. Im nächsten Schritt wird mit der Funktion `readWaveFile` die `.wav` Datei ausgelesen und die Daten werden in einen `vector<uint16>` gespeichert. Diese Klasse hat die Eigenschaft, dass sie den Aufwand der dynamischen Speicherallokation übernimmt. Danach werden die Daten mit der Funktion `negativSamples` auf negative Werte geprüft und gegebenenfalls wird das Vorzeichen entfernt. Mit der Funktion `sampleRateConversion` wird das Signal auf die Wiedergabefrequenz des Spea Testers angepasst. In der aktuellen Version werden nur einzelne Samples entfernt, das geht aber nur wenn das Abtasttheorem bei der Erzeugung der `.wav` Datei für die resultierende Frequenz eingehalten wurde. In zukünftigen Versionen können hier auch Filter verwendet wer-

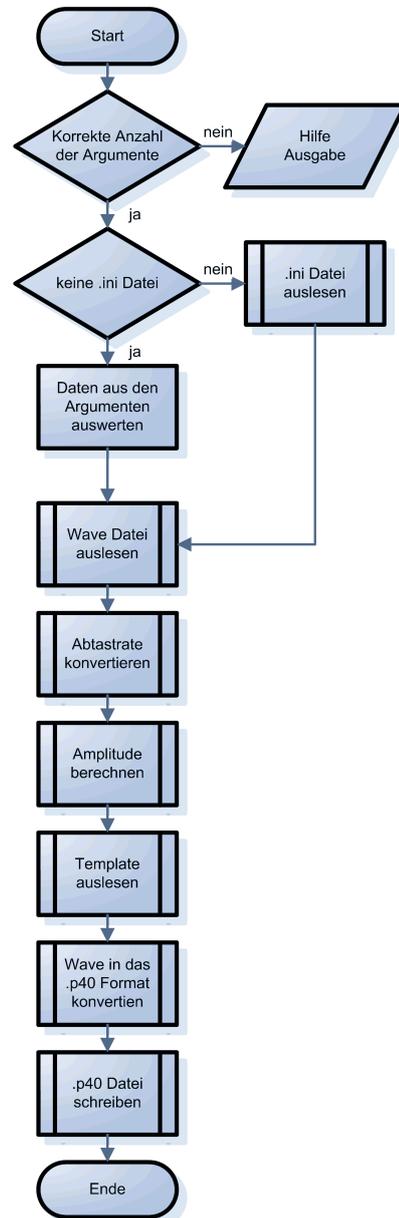


Abbildung 7.6: Flussdiagramm der Konvertierungssoftware *Tabor2Spea.exe*

den. Die beste Variante ist natürlich, wenn die Dateien im Matlab gleich mit der richtigen Frequenz ausgegeben werden. In der Funktion `multiplyAmplitude` werden die Amplitudenwerte auf die endgültigen Ausgabewerte umgerechnet. Dabei werden nur der Amplitudenwert des Sample mit dem Amplitudenfaktor und dem Stromfaktor multipliziert. Anschließend wird das Template eingelesen und der Funktion `convertWave2P40` übergeben. Hier werden einzelne Daten aus dem Template ausgelesen oder dort ersetzt. Markiert werden diese Stellen mit `///**** tabor2spea -> *****` wenn Daten eingefügt beziehungsweise ersetzt werden, oder mit `///**** -> tabor2spea *****` wenn Daten dort entnommen werden. Gesucht wird aber direkt nach den Variablendeklarationen oder Schlüsselwörtern. Mit der Variable `"long wave_begin="` wird bestimmt ab welchen Zeitpunkt im Pattern das Signal beginnen soll, vorher wird ein Chip-Hochlauf durchgeführt, damit der Dekoder richtig initialisiert ist und das Dekodersignal auch an dem I/O-Anschluss weitergeleitet wird. Mit `"long wave_end="` wird die Endposition des Signals im Pattern festgelegt. Die Variable `"long wave_length="` bestimmt die Länge des Signals. Mit `"double current_ratio="` und `"double amplitude_ratio="` werden die Amplitudenwerte festgelegt, die auch für den Hochlauf verwendet werden können. Das Schlüsselwort `"// Waveform:"` definiert den Beginn des Signals. Die einzelnen Samples werden parallel auf 14 Leitungen mit `H` und `L` definiert, wobei `H` für logisch 1 und `L` für logisch 0 steht. Dabei wird für jedes Sample eine eigene Zeile verwendet. In dem Kodeausschnitt 7.1 sind solche Samples, wie sie für den Tester aufbereitet werden, zu sehen.

Listing 7.1: Waveform

```

131  ///**** tabor2spea -> *****
132  ///Waveform
133  Pw(p1,50001..50001,"LHLLHLLHLL");
134  Pw(p1,50002..50002,"HLHLHLLHLL");
135  Pw(p1,50003..50003,"LHLLHLLHLL");
136  ...

```

Nun müssen die modifizierten Daten nur noch in die Ausgabedatei geschrieben werden. Es gibt dann noch ein paar Hilfsfunktionen, die den Programmablauf vereinfachen und übersichtlicher machen, wie zum Beispiel die Funktion `getInt64FromString`, die Zahlen aus einem String extrahiert, oder `replaceNumberInString` mit welcher Zahlen im String ersetzt werden. Mit der Funktion `numberToString` werden Zahlen im String eingefügt. Zum Setzen der Textfarbe im Ausgabefenster gibt es dann noch `setColor`.

7.1.4 CL-template.p40

Die Template *.p40* Datei (siehe Anhang H.2) ist eine Vorlage für eine PAGEL C Skriptdatei. Die Skriptsprache PAGEL C wird von Spea verwendet um Testpattern zu generieren und basiert auf der C Syntax. Weitere Informationen zu dieser Syntax befinden sich im Dokument **“PAGEL-C Test Pattern Generator Language Reference Guide“**[12] der Firma Spea.

Diese Template Datei besteht aus mehreren Bereichen. Zu Beginn werden im Settingsbereich einige Konstanten definiert, dann folgen die Variablen und im `main` befindet sich das Programm. Ein paar Einträge, die im Speziellen vom Programm *Tabor2Spea.exe* gemacht oder verändert werden, sind schon im Kapitel 7.1.3 beschrieben worden. Der Settingsbereich beginnt gleich mit den Clock settings, wo die genauen Timings des Testers eingestellt werden. Mit den Startpulse Settings kann ein Overshoot zu Beginn eingestellt werden. Die Wake up pulse settings erlauben, dass eine bestimmte Anzahl von Taktzyklen rausgetaktet werden bevor das eigentliche Signal kommt, damit der Controller im Chip einen Takt hat, den er braucht um die Startroutine durchzulaufen und das Programm zu starten. Mit Hilfe der Label settings kann eingestellt werden, wie oft Labels vorkommen, mit denen Patternteile direkt angesprungen werden können. `int * total_pin_list` und `int * p1` definieren die verwendeten Kanäle des Testers, wobei `p1` nur die Datenleitungen des DACs sind. Im `main` wird zuerst das Pattern initialisiert und der Clock eingestellt, gefolgt von der Startpulse- und Wakeup-Generierung. Nach dem vom *Tabor2Spea.exe* eingetragenen `“// Waveform:“` werden noch die Labels generiert.

7.2 Spea Software

Nachdem die *.p40* Testpattern Dateien erzeugt worden sind, werden sie mit dem Spea *Vector Explorer* geöffnet, von dem ein Screenshot in Abbildung 7.7 zu sehen ist.

Durch das Kompilieren der selektierten Daten im *Vector Explorer* werden die Patterndateien in ein für den Tester direkt verwendbares Format gespeichert. Eine graphische Darstellung dieser Pattern ist in dem *VectorView IDE* 7.8 möglich.

Das Prüfprogramm wird mit dem *Run Control Panel* 7.9 gestartet. Dieses Programm steuert direkt den Spea Tester und verwendet die *AtosC* Umgebung von Spea. Diese *AtosC* Software beinhaltet eine Vielzahl von *DLLs*, mit denen auf die Funktionen des Testers zugegriffen werden kann. Geschrieben und kompiliert werden die Programme in einer Borland

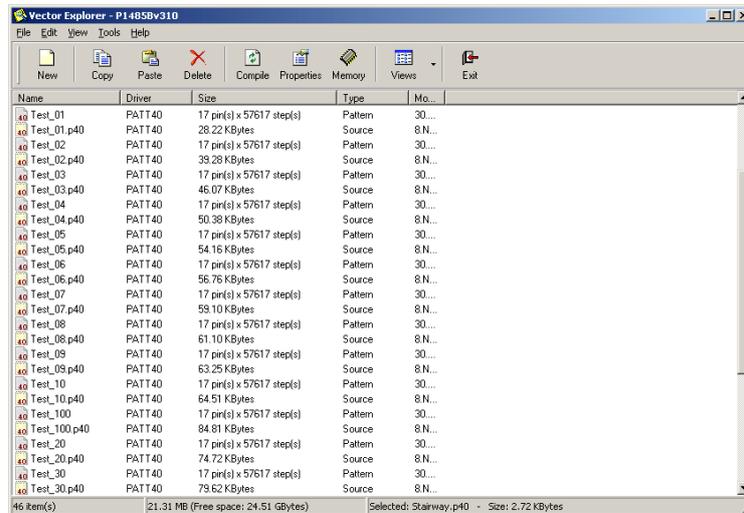


Abbildung 7.7: Spea Vector Explorer

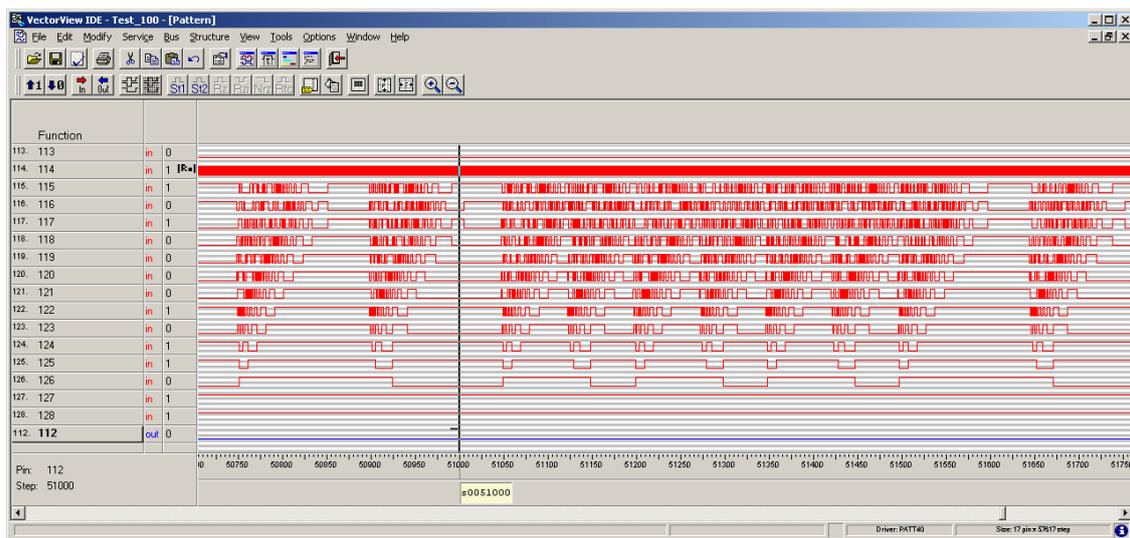


Abbildung 7.8: Spea VectorView IDE

C++ Umgebung. Von Infineon wurde diese *AtosC* Umgebung mit der Infineon internen *ModPFT*, der so genannten „modularen Prüftechnik“ erweitert, die noch weitere *DLLs* zur Verfügung stellt. Für die Anwendung in dieser Diplomarbeit wurde ein bestehendes Infineon *AtosC* Programm adaptiert.

In Abbildung 7.10 ist abschließend noch ein Screenshot von einem Tektronix Oszilloskop zu sehen, der bei einer Messung mit fertigem Aufbau am Spea Tester gemacht wurde. Dabei ist das braune Signal das Signal, das am La-Anschluss des Chips anliegt. Das blaue Signal wird vom I/O-Pin abgegriffen und stellt das vom Chip demodulierte Signal dar.

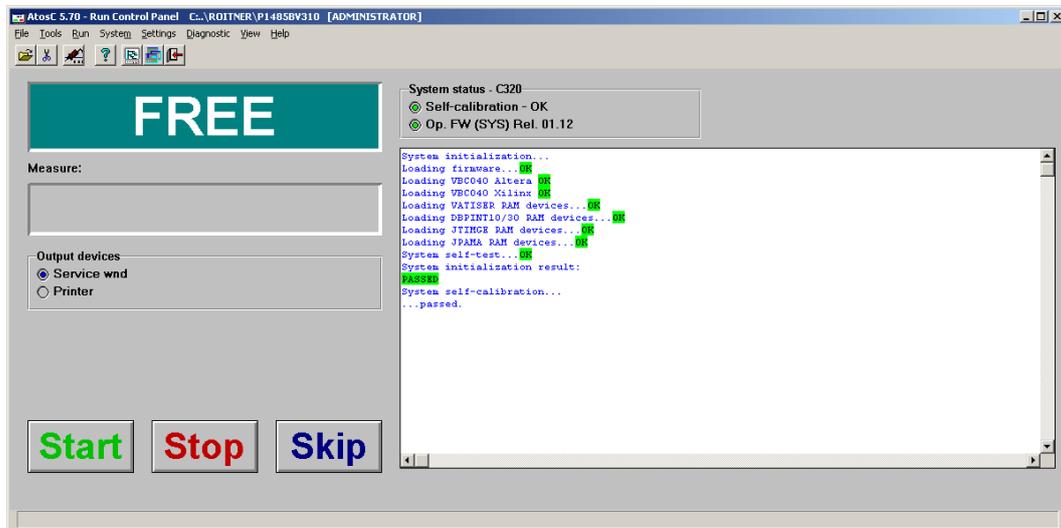


Abbildung 7.9: Spea Run Control Panel

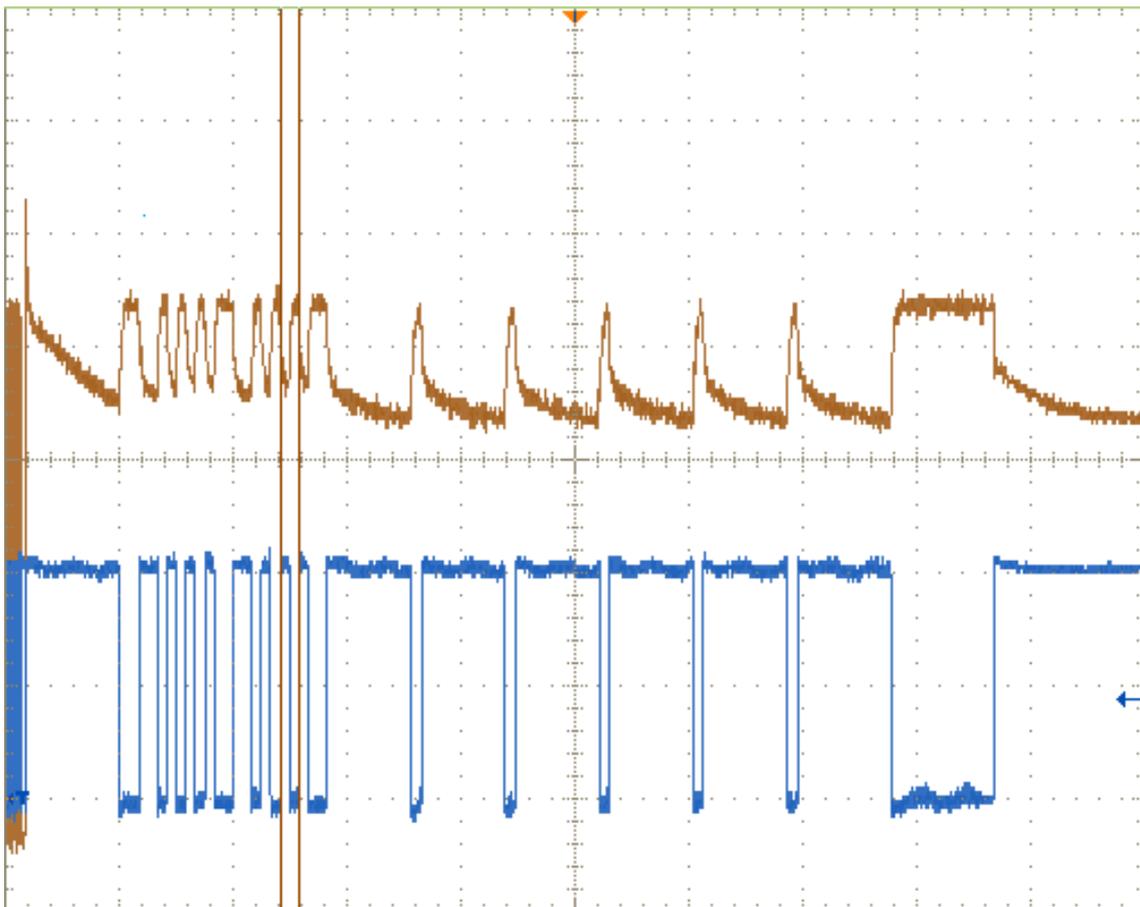


Abbildung 7.10: Messung am Spea Tester

8 Zusammenfassung und Ausblick

Die Aufgabe dieser Diplomarbeit war es eine Testumgebung zu schaffen, die es ermöglicht den Demodulator von RFID-Chips direkt am Wafer mit einem Wafertester verifizieren und qualifizieren zu können. Hierfür wurde eine elektronische Schaltung entwickelt, die dem Testchip Signale zur Verfügung stellt, die dem realen Umfeld entsprechen. Gesteuert wird der Messaufbau mit einem Spea Wafertester, für den ein Prüfprogramm adaptiert wurde. Zusätzlich wurde eine Software entwickelt, die die Signaldaten für den Spea Wafertester konvertiert und diesem zur Verfügung stellt.

Abschließend kann gesagt werden, dass das in dieser Diplomarbeit vorgestellte Verfahren sehr vielversprechend wirkt und es sich lohnt es in einen robusten, ausgereiften und mit Vergleichsmessungen untermauerten Zustand weiter zu entwickeln. Im Speziellen gehören Vergleichsmessungen mit dem AWG-Setup durchgeführt, damit von einer hohen Wahrscheinlichkeit ausgegangen werden kann, dass die Testergebnisse mit den Messungen eines AWG-Setups korrelieren. Um endgültig parallel am Wafer testen zu können, müssen die Überlegungen aus dem Kapitel 5.3 umgesetzt und verfeinert werden. Im Speziellen müssen die Anzahl der Bauteile reduziert werden. Durch geschickte Wahl der Widerstandswerte ist es möglich das große Widerstandsnetzwerk zu verkleinern, da bei weitem nicht alle möglichen einstellbaren Widerstandswerte benötigt werden. Bei den Stromquellen kann auf maximal 2 parallel geschaltete Stromquellen reduziert werden, da Messungen ergeben haben, dass der maximal zulässige Strom von 60 mA kein Problem darstellt. Auch beim Rekonstruktionsfilter sind Vereinfachungen möglich, da andere Glieder in der Signalkette ebenfalls Tiefpasswirkung mit sich bringen, die bei der Dimensionierung des Filters nicht eingeflossen sind. Eine weitere Idee um Bauteile einzusparen ist Messungen zu gruppieren. Eine Gruppe von Chips wird nur mit geringen Strömen betrieben, eine andere nur mit hohen Strömen, die nächste hat nur geringe Widerstandswerte und eine weitere nur hohe Widerstandswerte. Durch Umschalten mit Relais müssen somit nicht alle Messschaltungen für den vollständigen Messbereich ausgelegt werden, was zusätzlich eine erhebliche Einsparung von Bauteilen mit sich bringt.

Für die nächsten Generationen von RFID-Chips mit höheren Datenraten, so genannte **Very High Bit Rate** RFID-Chips, wo Datenraten bis zu 27,12 Mbit/s geplant sind, muss die Schaltung nur geringfügig modifiziert werden. Das Hauptproblem stellt eher der Spea-Tester dar, der mit seinen 40 Mbit/s an die Grenzen dieser Aufgabe stößt. Hier kann nur der Ersatz durch einen schnelleren Wafertester helfen. Wenn dieser noch zusätzlich einen schnelleren AWG mit sich bringen würde, hätte das zusätzlich großes Potential den Schaltungsaufwand wesentlich zu reduzieren. Bei der bestehenden Hardware muss für die VHBR Anwendung nur das Rekonstruktionsfilter neu dimensioniert werden, da die restlichen Komponenten dieser Schaltung für diesen Frequenzbereich bereits gerüstet sind.

A Abkürzungsverzeichnis

AM	Amplitude Modulation
ASK	Amplitude Shift Keying
AWG	Arbitrary Waveform Generator
BNC	Bayonet Neill Concelman
BPSK	Binary Phase Shift Keying
CDIP	Glass Sealed Ceramic Dual In-Line Package, wie DIP im Keramikgehäuse
CB	Contactbased
CL	Contactless
D/A	Digital/Analog
DAC	Digital to Analog Converter
DLL	Dynamic Link Library
dB	deziBel
DIP	Dual In-Line Package
DUT	Device Under Test
ECC	Elliptic Curve Cryptography
EEPROM	Electrically Erasable Programmable Read-Only Memory
ESB	Ersatzschaltbild
ESR	Equivalent Series Resistance
FET	Feldeffekttransistor
FIR	Finite Impulse Response (FIR-Filter)
GPIO	General Purpose Interface Bus
HF	Low Frequency (Frequenzband im Kurzwellenbereich)
IDC	Insulation Displacement Connector
I/O	Input/Output
LF	High Frequency (Frequenzband im Langwellenbereich)
NRZ-L	Non-Return to Zero (L for level)
OOK	On/Off Keying
PCD	Proximity Card Device (siehe ISO/IEC 14443[7])

A Abkürzungsverzeichnis

PC	Personal Computer
PCB	Printed Circuit Board
PICC	Proximity Integrated Chip Card (siehe ISO/IEC 14443[7])
PLL	Phase Locked Loop
PPM	Puls Positions Modulation
RAM	Random-Access Memory
RefPICC	Reference PICC
RF	Radio Frequency
RFID	Radio Frequency Identification
ROM	Read-Only Memory
RSA	asymmetrisches Kryptosystem von Ronald L. R ivest, Adi S hamir und Leonard A dleman am MIT entwickelt
U/I	Spannung/Strom
VHBR	Very High Bit Rate
WAV	Wave Dateiformat
ZF	Zwischenfrequenz
ZIF	Zero Insertion Force

B Glossar

Die in dieser Arbeit häufig verwendeten englischen Begriffe werden hier beschrieben.

Barcode	Strichcode
Bonding Pad	Anschlussfläche am Chip, an der er kontaktiert wird
Differential Probe	Tastkopf für Oszilloskope mit Differentialeingang
Overshoot	Überschwingen
Parser	Syntaxanalysator
Pattern	Muster
Probe	Tastkopf für Oszilloskope
Reader	Lesegerät
Routine	Computerprogramm
Tag	Anhängeschildchen, Etikett
Transponder	Ein Gerät, das empfangene Signale automatisch beantwortet oder weiterleitet
Template	Vorlage
Sample	Abtastwert
Screenshot	Bildschirmkopie
Settings	Einstellungen
Wafer Prober	Testgerät mit dem über eine Nadelkarte die Pads des Wafers kontaktiert werden

C Literaturverzeichnis

- [1] Analog Devices. *AD8056: Low Cost, 300 MHz, Voltage Feedback Amplifiers*, rev. j edition, 2006.
- [2] Analog Devices. *AD9772A: 14-Bit, 160 MSPS TxDAC+ with 2x Interpolation Filter*, rev. c edition, 2008.
- [3] Fred J. Taylor Arthur B. Williams. *Electronic Filter Design Handbook*. McGRAW-HILL, INC., 1221 Avenue of the Americas New York, NY 10020, 3. edition, 1995.
- [4] Elektor. *Sehr schnelle spannungsgesteuerte Stromquelle*, chapter 045, page 57. Number 984091 in Halbleiterheft. Elektor, 7-8 1998.
- [5] Klaus Finkenzeller. *RFID - Handbuch, Grundlagen und praktische Anwendungen induktiver Funkanlagen, Transponder und kontaktloser Chipkarten*. HANSER, 4. edition, 2006.
- [6] Infineon Technologies AG. *Product Brief: SLE 66CLxxxPE Contactless and Dual Interface Controller PE Family*, 2006.
- [7] ISO/IEC. Fdis 14443-2:2000(e), identification cards - contactless integrated circuit(s) cards - proximity cards - part 2: Radio frequency power and signal interface, 2000.
- [8] ISO/IEC. Fdis 10373-6:2001(e), identification cards - test methods - part 6: Proximity cards, 2001.
- [9] ISO/IEC. Fdis 10373-7:2001(e), identification cards - test methods - part 7: Vicinity cards, 2001.
- [10] Simon Pilz. Contact-less emulation system. Master's thesis, FH Joanneum, Kapfenberg, Austria, September 2006.
- [11] Spea, SPEA SpA - 16, Via Torino - 10088 Volpiano (TO) - Italy. *Load Board „L306“ - Advantest Layout Load Board for Comptest MX and accessories - User Guide*, 1.00 edition.

- [12] Spea, SPEA SpA - 16, Via Torino - 10088 Volpiano (TO) - Italy. *PAGEL-C - Test Pattern Generator Language - Reference Guide*, 1.20 edition.
- [13] Christoph Schenk Ulrich Tietze. *Halbleiter-Schaltungstechnik*. Springer-Verlag, 12. edition, 2002.
- [14] Franz Moeller / Hans Fricke / Heinrich Frohne / Paul Vaske. *Grundlagen der Elektrotechnik*. B. G. Teubner Stuttgart, 17. edition, 1986.
- [15] Harald Hartl / Edwin Krasser / Wolfgang Pribyl / Peter Söser / Gunter Winkler. *Elektronische Schaltungstechnik*. PEARSON Studium, 2008.
- [16] Wolfgang Effing Wolfgang Rankl. *Handbuch der Chipkarten, Aufbau - Funktionsweise - Einsatz von Smart Cards*. HANSER, 4. edition, 2002.

C.1 Web links

Analog Devices AD8130	http://www.analog.com/en/audiovideo-products/video-ampsbuffersfilters/ad8130/products/product.html
Biometrie	http://de.wikipedia.org/wiki/Biometrie
Contactless SLE 66 family	http://www.infineon.com/cms/en/product/channel.html?channel=ff80808112ab681d0112ab69357c0158
Infineon Technologies	http://www.infineon.com
SPEA - Automatic Test Equipment	http://www.spea.com
SPEA - C320MX	http://www.spea-ate.de/index.php?id=116&L=2&tx_ttnews[backPid]=66&tx_ttnews[tt_news]=35&cHash=7e68e873e3

D Widerstandswertetabelle

Die genauere Beschreibung des Widerstandnetzwerkes ist aus dem Kapitel 5.2.5 zu entnehmen. Auf den folgenden Seiten sind alle einstellbaren Widerstandswerte, abhängig von den Schalterstellungen dargestellt. Dabei sind je Spalte die Schalter S1-10 bis S1-6 und je Zeile die Schalter S1-5 bis S1-1 gleich. Der Wert '1' bedeutet Schalter geschlossen und '0' steht für Schalter offen. Die Widerstandswerte sind in Ω angegeben.

Als kleines Beispiel wollen wir den Wert $5550\ \Omega$ einstellen, der aus der Tabelle D.1 den binären Wert '1110010001' ergibt. Das würde ergeben, dass Schalter S1-10, S1-9, S1-8, S1-5 und S1-1 geschlossen sein müssen und die restlichen offen bleiben.

D Widerstandswertetabelle

	11111x	11110x	11101x	11100x	11011x	11010x	11001x	11000x
x11111	50	1650	3250	4850	6450	8050	9650	11250
x11110	100	1700	3300	4900	6500	8100	9700	11300
x11101	150	1750	3350	4950	6550	8150	9750	11350
x11100	200	1800	3400	5000	6600	8200	9800	11400
x11011	250	1850	3450	5050	6650	8250	9850	11450
x11010	300	1900	3500	5100	6700	8300	9900	11500
x11001	350	1950	3550	5150	6750	8350	9950	11550
x11000	400	2000	3600	5200	6800	8400	10000	11600
x10111	450	2050	3650	5250	6850	8450	10050	11650
x10110	500	2100	3700	5300	6900	8500	10100	11700
x10101	550	2150	3750	5350	6950	8550	10150	11750
x10100	600	2200	3800	5400	7000	8600	10200	11800
x10011	650	2250	3850	5450	7050	8650	10250	11850
x10010	700	2300	3900	5500	7100	8700	10300	11900
x10001	750	2350	3950	5550	7150	8750	10350	11950
x10000	800	2400	4000	5600	7200	8800	10400	12000
x01111	850	2450	4050	5650	7250	8850	10450	12050
x01110	900	2500	4100	5700	7300	8900	10500	12100
x01101	950	2550	4150	5750	7350	8950	10550	12150
x01100	1000	2600	4200	5800	7400	9000	10600	12200
x01011	1050	2650	4250	5850	7450	9050	10650	12250
x01010	1100	2700	4300	5900	7500	9100	10700	12300
x01001	1150	2750	4350	5950	7550	9150	10750	12350
x01000	1200	2800	4400	6000	7600	9200	10800	12400
x00111	1250	2850	4450	6050	7650	9250	10850	12450
x00110	1300	2900	4500	6100	7700	9300	10900	12500
x00101	1350	2950	4550	6150	7750	9350	10950	12550
x00100	1400	3000	4600	6200	7800	9400	11000	12600
x00011	1450	3050	4650	6250	7850	9450	11050	12650
x00010	1500	3100	4700	6300	7900	9500	11100	12700
x00001	1550	3150	4750	6350	7950	9550	11150	12750
x00000	1600	3200	4800	6400	8000	9600	11200	12800

Tabelle D.1: Widerstandstabelle Seite 1/4

D Widerstandswertetabelle

	10111x	10110x	10101x	10100x	10011x	10010x	10001x	10000x
x11111	12850	14450	16050	17650	19250	20850	22450	24050
x11110	12900	14500	16100	17700	19300	20900	22500	24100
x11101	12950	14550	16150	17750	19350	20950	22550	24150
x11100	13000	14600	16200	17800	19400	21000	22600	24200
x11011	13050	14650	16250	17850	19450	21050	22650	24250
x11010	13100	14700	16300	17900	19500	21100	22700	24300
x11001	13150	14750	16350	17950	19550	21150	22750	24350
x11000	13200	14800	16400	18000	19600	21200	22800	24400
x10111	13250	14850	16450	18050	19650	21250	22850	24450
x10110	13300	14900	16500	18100	19700	21300	22900	24500
x10101	13350	14950	16550	18150	19750	21350	22950	24550
x10100	13400	15000	16600	18200	19800	21400	23000	24600
x10011	13450	15050	16650	18250	19850	21450	23050	24650
x10010	13500	15100	16700	18300	19900	21500	23100	24700
x10001	13550	15150	16750	18350	19950	21550	23150	24750
x10000	13600	15200	16800	18400	20000	21600	23200	24800
x01111	13650	15250	16850	18450	20050	21650	23250	24850
x01110	13700	15300	16900	18500	20100	21700	23300	24900
x01101	13750	15350	16950	18550	20150	21750	23350	24950
x01100	13800	15400	17000	18600	20200	21800	23400	25000
x01011	13850	15450	17050	18650	20250	21850	23450	25050
x01010	13900	15500	17100	18700	20300	21900	23500	25100
x01001	13950	15550	17150	18750	20350	21950	23550	25150
x01000	14000	15600	17200	18800	20400	22000	23600	25200
x00111	14050	15650	17250	18850	20450	22050	23650	25250
x00110	14100	15700	17300	18900	20500	22100	23700	25300
x00101	14150	15750	17350	18950	20550	22150	23750	25350
x00100	14200	15800	17400	19000	20600	22200	23800	25400
x00011	14250	15850	17450	19050	20650	22250	23850	25450
x00010	14300	15900	17500	19100	20700	22300	23900	25500
x00001	14350	15950	17550	19150	20750	22350	23950	25550
x00000	14400	16000	17600	19200	20800	22400	24000	25600

Tabelle D.2: Widerstandstabelle Seite 2/4

D Widerstandswertetabelle

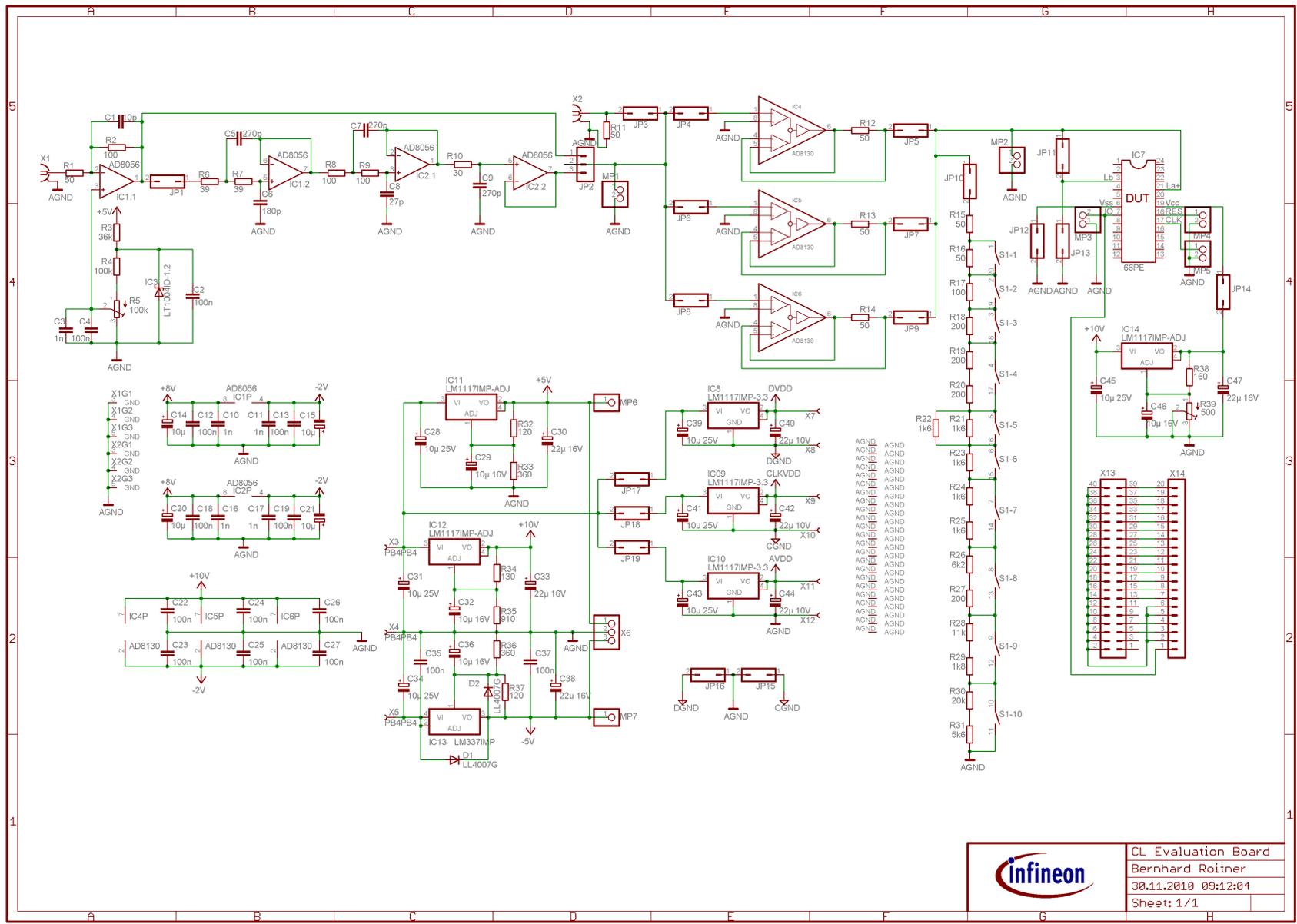
	01111x	01110x	01101x	01100x	01011x	01010x	01001x	01000x
x11111	25650	27250	28850	30450	32050	33650	35250	36850
x11110	25700	27300	28900	30500	32100	33700	35300	36900
x11101	25750	27350	28950	30550	32150	33750	35350	36950
x11100	25800	27400	29000	30600	32200	33800	35400	37000
x11011	25850	27450	29050	30650	32250	33850	35450	37050
x11010	25900	27500	29100	30700	32300	33900	35500	37100
x11001	25950	27550	29150	30750	32350	33950	35550	37150
x11000	26000	27600	29200	30800	32400	34000	35600	37200
x10111	26050	27650	29250	30850	32450	34050	35650	37250
x10110	26100	27700	29300	30900	32500	34100	35700	37300
x10101	26150	27750	29350	30950	32550	34150	35750	37350
x10100	26200	27800	29400	31000	32600	34200	35800	37400
x10011	26250	27850	29450	31050	32650	34250	35850	37450
x10010	26300	27900	29500	31100	32700	34300	35900	37500
x10001	26350	27950	29550	31150	32750	34350	35950	37550
x10000	26400	28000	29600	31200	32800	34400	36000	37600
x01111	26450	28050	29650	31250	32850	34450	36050	37650
x01110	26500	28100	29700	31300	32900	34500	36100	37700
x01101	26550	28150	29750	31350	32950	34550	36150	37750
x01100	26600	28200	29800	31400	33000	34600	36200	37800
x01011	26650	28250	29850	31450	33050	34650	36250	37850
x01010	26700	28300	29900	31500	33100	34700	36300	37900
x01001	26750	28350	29950	31550	33150	34750	36350	37950
x01000	26800	28400	30000	31600	33200	34800	36400	38000
x00111	26850	28450	30050	31650	33250	34850	36450	38050
x00110	26900	28500	30100	31700	33300	34900	36500	38100
x00101	26950	28550	30150	31750	33350	34950	36550	38150
x00100	27000	28600	30200	31800	33400	35000	36600	38200
x00011	27050	28650	30250	31850	33450	35050	36650	38250
x00010	27100	28700	30300	31900	33500	35100	36700	38300
x00001	27150	28750	30350	31950	33550	35150	36750	38350
x00000	27200	28800	30400	32000	33600	35200	36800	38400

Tabelle D.3: Widerstandstabelle Seite 3/4

D Widerstandswertetabelle

	00111x	00110x	00101x	00100x	00011x	00010x	00001x	00000x
x11111	38450	40050	41650	43250	44850	46450	48050	49650
x11110	38500	40100	41700	43300	44900	46500	48100	49700
x11101	38550	40150	41750	43350	44950	46550	48150	49750
x11100	38600	40200	41800	43400	45000	46600	48200	49800
x11011	38650	40250	41850	43450	45050	46650	48250	49850
x11010	38700	40300	41900	43500	45100	46700	48300	49900
x11001	38750	40350	41950	43550	45150	46750	48350	49950
x11000	38800	40400	42000	43600	45200	46800	48400	50000
x10111	38850	40450	42050	43650	45250	46850	48450	50050
x10110	38900	40500	42100	43700	45300	46900	48500	50100
x10101	38950	40550	42150	43750	45350	46950	48550	50150
x10100	39000	40600	42200	43800	45400	47000	48600	50200
x10011	39050	40650	42250	43850	45450	47050	48650	50250
x10010	39100	40700	42300	43900	45500	47100	48700	50300
x10001	39150	40750	42350	43950	45550	47150	48750	50350
x10000	39200	40800	42400	44000	45600	47200	48800	50400
x01111	39250	40850	42450	44050	45650	47250	48850	50450
x01110	39300	40900	42500	44100	45700	47300	48900	50500
x01101	39350	40950	42550	44150	45750	47350	48950	50550
x01100	39400	41000	42600	44200	45800	47400	49000	50600
x01011	39450	41050	42650	44250	45850	47450	49050	50650
x01010	39500	41100	42700	44300	45900	47500	49100	50700
x01001	39550	41150	42750	44350	45950	47550	49150	50750
x01000	39600	41200	42800	44400	46000	47600	49200	50800
x00111	39650	41250	42850	44450	46050	47650	49250	50850
x00110	39700	41300	42900	44500	46100	47700	49300	50900
x00101	39750	41350	42950	44550	46150	47750	49350	50950
x00100	39800	41400	43000	44600	46200	47800	49400	51000
x00011	39850	41450	43050	44650	46250	47850	49450	51050
x00010	39900	41500	43100	44700	46300	47900	49500	51100
x00001	39950	41550	43150	44750	46350	47950	49550	51150
x00000	40000	41600	43200	44800	46400	48000	49600	51200

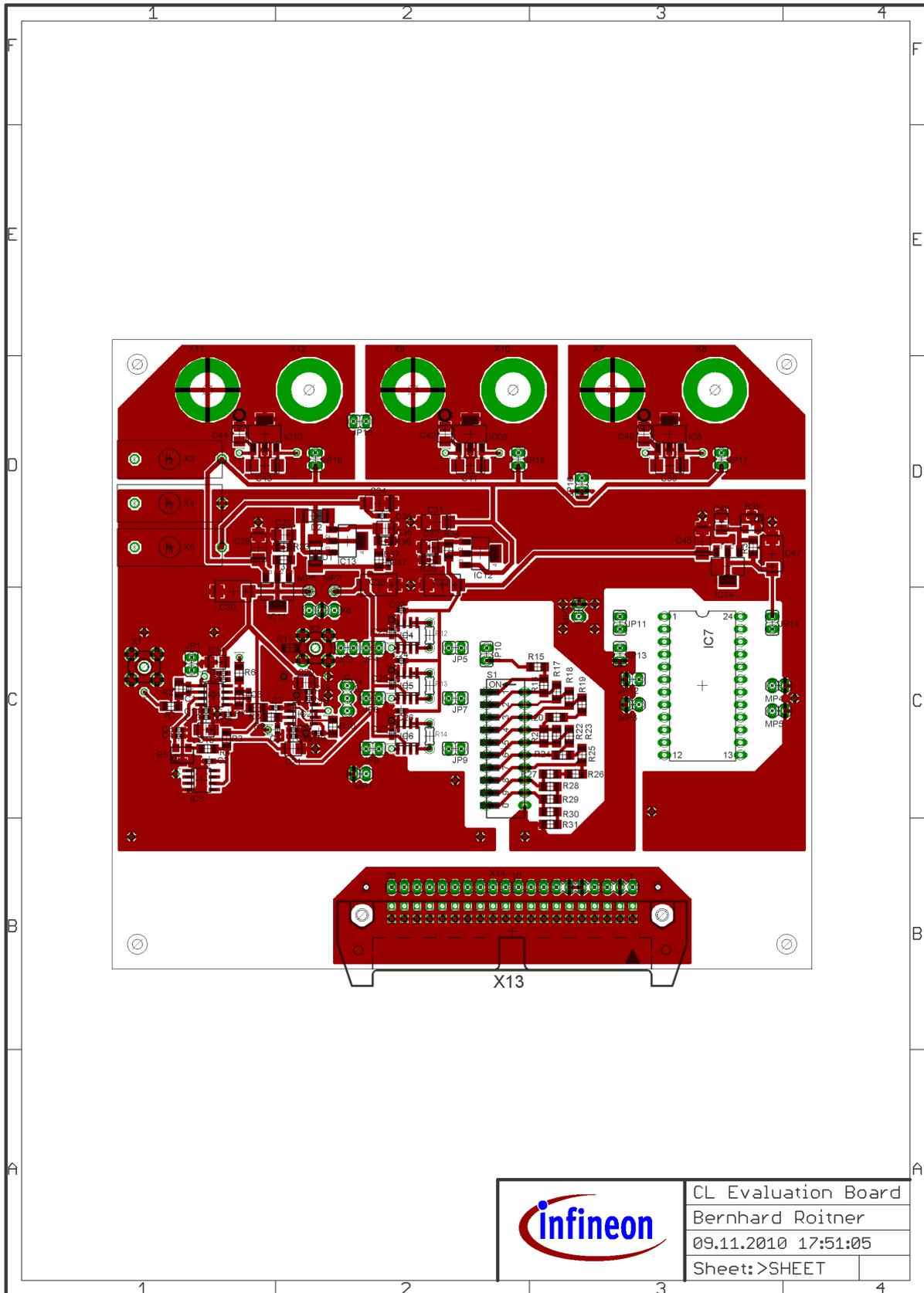
Tabelle D.4: Widerstandstabelle Seite 4/4



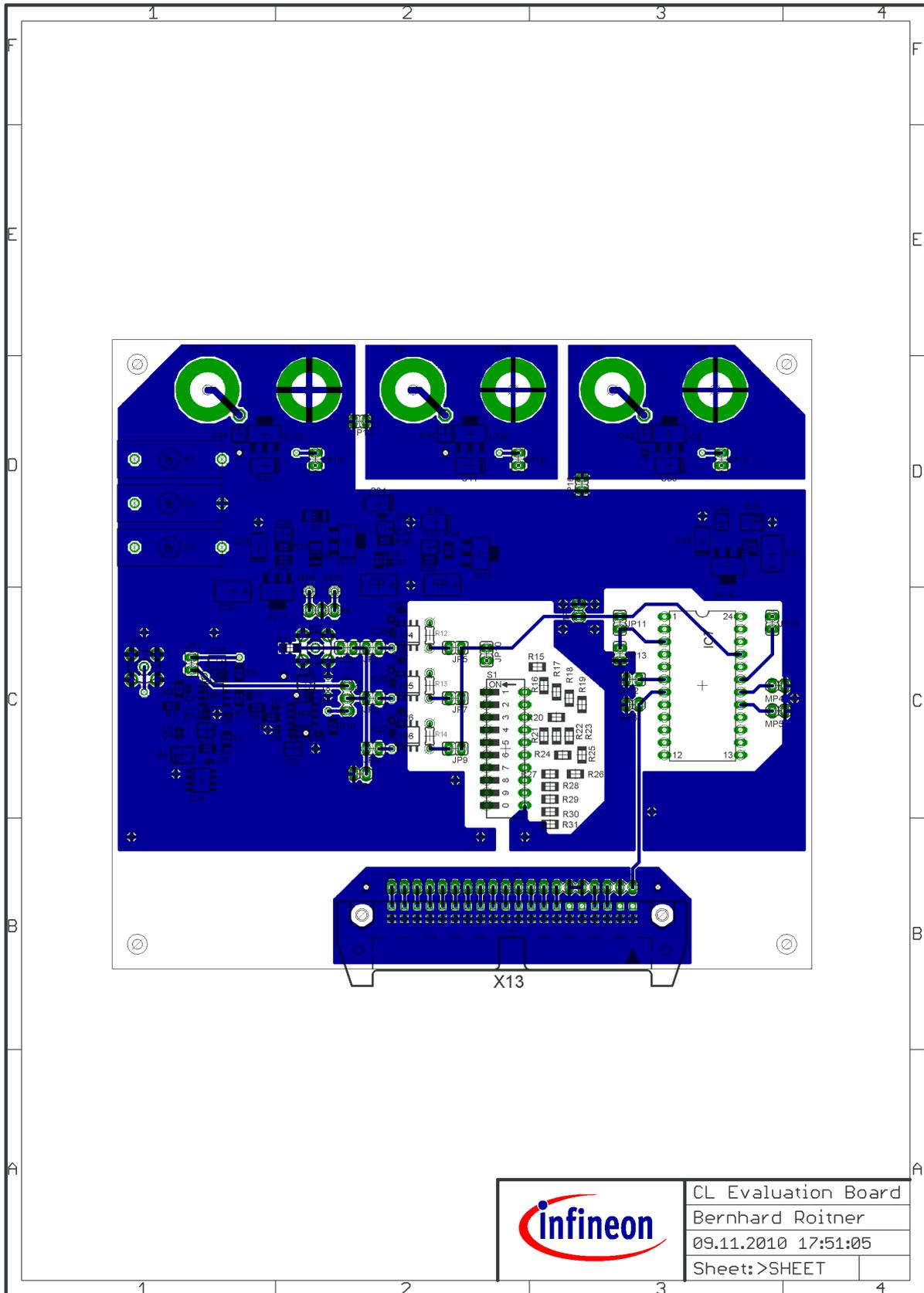
E Schaltung

	CL Evaluation Board
	Bernhard Roitner
	30.11.2010 09:12:04
	Sheet: 1/1

F Layout



F Layout



CL Evaluation Board	
Bernhard Roitner	
09.11.2010 17:51:05	
Sheet:	>SHEET

SLE 66CLxxxPE

Contactless and Dual Interface Controller PE Family

THE SLE 66CLxxxPE contactless and dual interface controller portfolio supports a complete range of contactless proximity interfaces for global interoperability. Designers can choose from the wide range of family members to serve individual applications and contactless infrastructure needs.

Applications

- Identification (e-Passports, National ID, Driver Licence, Health Care, Digital Signature, Access Control)
- Payment
 - Finance (Mastercard PayPass®, Visa Wave®, American Express® Expresspay)
 - E-purse
 - Transportation
- Multiapplication (Finance, Transportation, Access Control)

Features

- Contactless interface acc. to ISO 14443 B&A, transmission speeds RX (Reader to Card) and TX (Card to Reader) up to 848 kbit/s
- Contactless interface acc. to ISO 18092 passive mode (SONY FeliCa® communication interface), transmission speeds RX and TX 212 and 424 kbit/s
- Contact based interface acc. to ISO 7816/ETSI/EMV – Class A/B/C
- Widely compatible to Infineon's SLE 66P – Platform
- MIFARE® classic 1k emulation

Security Features

- High performance contactless cryptography ACE (RSA), DDES (DES, 3DES)
- Enhanced sensor concept (low and high voltage sensors, frequency sensors and filters, light sensor, glitch sensor, temperature sensor, life test function for sensors (UMSLC))
- Bus confusion
- Security reset detection
- Current control oscillator (ICO)

- Sparkling SFR encryption for DDES and ACE, CRC module and RNG
- 32 bytes security PROM, HW protected for batch-, wafer-, die-individual security data
- MED- memory encryption/decryption device for XRAM, ROM and EEPROM
- Security optimized layout and layout scrambling
- Fast IRAM erase
- Enhanced Error correction unit (ECU)
- CC EAL 5 + high planned
- Visa level 3 planned
- CAST planned

Delivery Forms

- Wafer
 - 8", sawn, unsawn, 150 µm thickness, NiAu bumps
 - 8", sawn, 55 µm thickness, NiAu bumps
- Dual interface module M8.4
- Contactless only module MCC8

Support

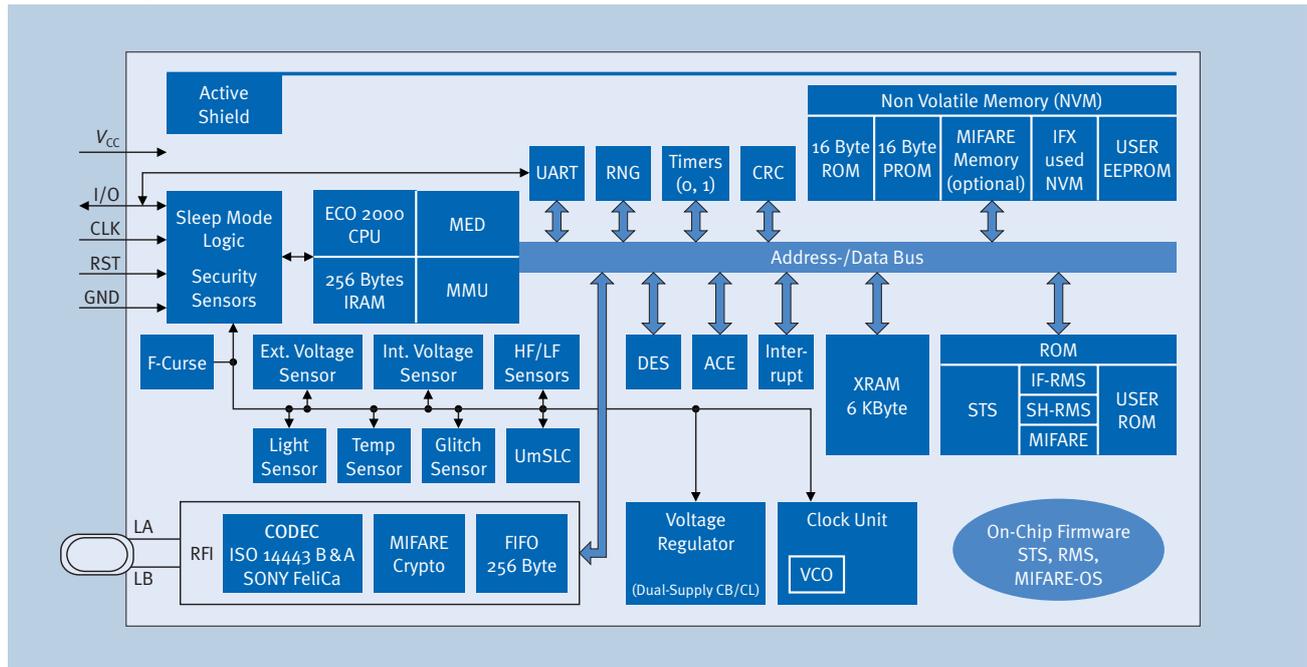
- Hardware Emulator (KSC, Hitex, Rom Monitor)
- Keil SDK
- Application Notes
- Evaluation Kit (Baltech)

PayPass® is a registered trademark of Mastercard
Visa® is a registered trademark of Visa USA Inc.
American Express® is a registered trademark of American Express Company
FeliCa® is a registered trademark of Sony Corp.
MIFARE® is a registered trademark of Philips Electronics N.V.

www.infineon.com/contactless

Chip Card & Security ICs

Block Diagram



PE Product Portfolio

Product	EEPROM	Interface	ROM	Cryptography
SLE 66CL41PE	4 KB	ISO 14443 B&A	92 KB	DDES
SLE 66CL80PE	8 KB	ISO 14443 B&A, ISO 7816	92 KB	DDES
SLE 66CL80PEM	8 KB + 1 KB MIFARE Data	ISO 14443 B&A, ISO 7816 MIFARE Classic	88 KB	DDES
SLE 66CL80PES	8 KB	ISO 14443 B&A, ISO 7816 Acc. ISO 18092 Passive Mode	92 KB	DDES
SLE 66CL81PE	8 KB	ISO 14443 B&A	92 KB	DDES
SLE 66CL81PEM	8 KB + 1 KB MIFARE Data	ISO 14443 B&A, MIFARE Classic	88 KB	DDES
SLE 66CL180PE	18 KB	ISO 14443 B&A, ISO 7816	92 KB	DDES
SLE 66CL180PEM	16 KB + 1 KB MIFARE Data	ISO 14443 B&A, ISO 7816 MIFARE Classic	88 KB	DDES
SLE 66CL180PES	18 KB	ISO 14443 B&A, ISO 7816 Acc. ISO 18092 Passive Mode	92 KB	DDES
SLE 66CLX360PE	36 KB	ISO 14443 B&A, ISO 7816	240 KB	DDES ACE
SLE 66CLX360PEM	36 KB + 1 KB MIFARE Data	ISO 14443 B&A, ISO 7816 MIFARE Classic	236 KB	DDES ACE
SLE 66CLX360PES	36 KB	ISO 14443 B&A, ISO 7816 Acc. ISO 18092 Passive Mode	240 KB	DDES ACE
SLE 66CLX800PE	80 KB	ISO 14443 B&A, ISO 7816	240 KB	DDES ACE
SLE 66CLX800PEM	78 KB + 1 KB MIFARE Data	ISO 14443 B&A, ISO 7816 MIFARE Classic	236 KB	DDES ACE
SLE 66CLX800PES	80 KB	ISO 14443 B&A, ISO 7816 Acc. ISO 18092 Passive Mode	240 KB	DDES ACE

For further information, please contact:

Infineon Technologies AG, e-mail: security.chipcard.ics@infineon.com, www.infineon.com/contactless

How to reach us:
<http://www.infineon.com>

Published by
Infineon Technologies AG
81726 München, Germany

© Infineon Technologies AG 2006.
All Rights Reserved.

Legal Disclaimer

The information given in this Product Brief shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffheitsgarantie"). With respect to any examples or hints given herein, any typical values stated herein and/or any information regarding the application of the device, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

Information

For further information on technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies Office (www.infineon.com).

Warnings

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies Office.

Infineon Technologies Components may only be used in life-support devices or systems with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system, or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body, or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

Ordering No. B116-H8834-X-7600
Printed in Germany
PS 05061. nb

H Quelltext

H.1 Konvertierungssoftware Tabor2Spea.exe

Quelltext zum Konvertierungsprogramm Tabor2Spea.exe

```
1 #include "user_types.h"
2 #include <iostream>
3 #include <iomanip>
4 #include <fstream>
5 #include <sstream>
6 #include <string>
7 #include <vector>
8 #include <windows.h>
9 #include <cstdlib>
10
11 using std :: cerr ;
12 using std :: endl ;
13 using namespace std;
14
15 typedef int ret; // Type of return value
16
17 // Return value
18 const ret OK=0;
19 const ret FAILED=1;
20 const ret END_OF_FILE=255;
21 const ret WRONG_NUMBER_OF_ARG=1;
22 const ret WRONG_INI_FILE=3;
23 const ret WRONG_SOURCE_FILE=2;
24 // Arguments
25 const int RIGHT_NUMBER_OF_ARG=9;
26 const int TABOR_ARG=1;
27 const int SPEA_ARG=2;
28 const int TEMPLATE_ARG=3;
29 const int TABOR_SAMPLE_RATE_ARG=4;
30 const int SPEA_SAMPLE_RATE_ARG=5;
31 const int AMPLITUDE_ARG=6;
32 const int CURRENT_ARG=7;
33 const int DUMMY_ARG=8;
34 // SPEA and Tabor settings
35 const uint8 NUMBER_OF_PINS=14;
```

```

36 const double TABOR_SAMPLE_RATE=94.92;
37 const double SPEA_SAMPLE_RATE=31.64;
38 const double TABOR_AMP=0.85;
39 const double CURRENT=0.1;
40 // Ini file read buffer
41 const size_t INI_BUFFER_SIZE = 255;    // Number of characters for output messages
42 // Data conversion & set endian
43 const int BIG_ENDIAN = 0;
44 const int LITTLE_ENDIAN = 1;
45 const int ENDIAN=LITTLE_ENDIAN;
46 // Colors
47 // colors @ black background
48 //const uint16 KHAKI=6;
49 //const uint16 GREEN=10;
50 //const uint16 RED=12;
51 //const uint16 WHITE=15;
52 // colors @ white background
53 const uint16 WHITE=240;
54 const uint16 GREEN=242;
55 const uint16 KHAKI=246;
56 const uint16 RED=252;
57 // Template find points
58 const char* WAVE_BEGIN="// Waveform:";
59 const char* VAR_WAVE_BEGIN="long wave_begin=";
60 const char* VAR_WAVE_END="long wave_end=";
61 const char* VAR_WAVE_LENGTH="long wave_length=";
62 const char* VAR_CURRENT_RATIO="double current_ratio=";
63 const char* VAR_AMPLITUDE_RATIO="double amplitude_ratio=";
64
65 struct Parameter
66 {
67     string ini_file;
68     string wave_file;
69     string pattern_file;
70     string template_file;
71     double tabor_sample_rate; // Tabor wave's sample frequency
72     double spea_sample_rate; // Spea wave's sample frequency
73     double amplitude;       // Tabor wave's nominal amplitude for calibration
74     double current;        // Tabor wave's actual current
75     bool dummy;
76 };
77
78 // Declaration of functions
79 ret readWaveFile (ifstream&,vector<uint16>&);
80 ret negativSamples (vector<uint16>&);
81 ret sampleRateConversion (vector<uint16>&,&b>double,double&);
82 ret multiplyAmplitude (vector<uint16>&,&b>double,double);
83 ret convertWave2P40 (vector<uint16>,stringstream&,stringstream&,Parameter);
84 ret readIniFile(Parameter&);
85 int64 getInt64FromString(string,string,string);

```

```

86 template <class ElementType> ret replaceNumberInString(string& string_to_insert, string begin,
87                                     string end, ElementType data);
88 template <class ElementType> string numberToString(const ElementType input);
89 void setColor(uint16);
90
91
92 int main (int argc, char* argv[])
93 {
94     Parameter settings;
95     if (argc==RIGHT_NUMBER_OF_ARG) // read data from input
96     {
97         settings.wave_file = argv[TABOR_ARG];
98         settings.pattern_file = argv[SPEA_ARG];
99         settings.template_file = argv[TEMPLATE_ARG];
100        settings.tabor_sample_rate = strtod(argv[TABOR_SAMPLE_RATE_ARG],NULL);
101        settings.spea_sample_rate = strtod(argv[SPEA_SAMPLE_RATE_ARG],NULL);
102        settings.amplitude = strtod(argv[AMPLITUDE_ARG],NULL);
103        settings.current = strtod(argv[CURRENT_ARG],NULL);
104        if(strcmp(argv[DUMMY_ARG],"yes")==0)
105        {
106            setColor(KHAKI); cerr << "Generating a dummy pattern" << endl;
107            settings.dummy=true;
108        }
109        else settings.dummy=false;
110    }
111    else if (argc==2) // read data from ini.file
112    {
113        settings.ini_file = argv[1];
114        setColor(WHITE); cerr << "reading ini-file...";
115        if (readIniFile(settings)==WRONG_INI_FILE)
116            { setColor(RED); cerr << "Error in reading ini-file" << endl; setColor(WHITE);
117              return WRONG_INI_FILE;}
118        else {setColor(GREEN);cerr << "OK" << endl;}
119    }
120    else if (argc!=RIGHT_NUMBER_OF_ARG)
121    {
122        setColor(WHITE);
123        cerr
124        << "\nConverts Tabor \".wav\"-File to Spea \".p40\"-File \n \n"
125        << "ERROR: Wrong number of arguments \n \n"
126        << "Usage: tabor2spea INI-FILE" << endl
127        << "   or: tabor2spea SOURCE-FILE DESTINATION-FILE TEMPLATE-FILE TABOR-FREQUENCY" << endl
128        << "           SPEA-FREQUENCY AMPLITUDE CURRENT DUMMY" << endl << endl
129        << "SOURCE-FILE      Tabor wave-file" << endl
130        << "DESTINATION-FILE  Spea pattern-file" << endl
131        << "TEMPLATE-FILE    Spea template pattern-file" << endl
132        << "TABOR-FREQUENCY   Tabor wave-file sample rate [MHz]" << endl
133        << "SPEA-FREQUENCY    Maximum spea tester frequency [MHz]" << endl
134        << "AMPLITUDE        Amplitude factor of wave-file: max_level/full_scale e.g.:0.85" << endl
135        << "CURRENT          Current factor: ouput_current/max_current e.g.:0.1" << endl

```

```

136 << "DUMMY          [yes/no] for generating an empty dummy pattern" << endl << endl
137 << "for example:" << endl
138 << "  tabor2spea CL-emu.ini" << endl
139 << "or:" << endl
140 << "  tabor2spea pattern.wav pattern.p40 CL_template.p40 94.92 31.64 0.85 0.1 no" << endl;
141   return WRONG_NUMBER_OF_ARG;
142 }
143
144 vector<uint16> samples;
145 setColor(WHITE); cerr
146 << "-----" << endl
147 << "reading wave file...";
148 ifstream in_file (settings.wave_file.c_str(),ifstream::binary);
149 if (!in_file) { setColor(RED); cerr
150   << "unable to open file" << endl; setColor(WHITE); return FAILED;}
151 if (readWaveFile (in_file,samples)==OK) {setColor(GREEN); cerr
152   << "OK" << endl;}
153 else {setColor(RED);cerr
154   << "\nERROR: odd number of bytes in tabor .wav file" << endl;}
155 in_file.close(); setColor(WHITE); cerr
156 << "-----" << endl
157 << "check for negativ sample values...";
158 if (!negativSamples (samples)){setColor(GREEN);cerr
159   << "OK" << endl;}
160 else {setColor(RED); cerr << endl
161   << "\nWarning: Negative value in tabor pattern found!\n"
162   << "Pattern is maybe modulated.\nUse unmodulated pattern for CL-Emulation!\n"
163   << "Negative samples are replaced by 0" << endl; } setColor(WHITE); cerr
164 << "-----" << endl
165 << "converting sample rate...";
166 if (sampleRateConversion (samples,settings.tabor_sample_rate,settings.spea_sample_rate)==OK)
167   {setColor(GREEN);cerr
168   << "OK"; setColor(WHITE); cerr
169   << "  SPEA sample rate:" << settings.spea_sample_rate << "MHZ" << endl;}
170 else {cerr
171   << "no need" << endl;} setColor(WHITE); cerr
172 << "-----" << endl
173 << "calculate right amplitude...";
174 if (multiplyAmplitude (samples,settings.amplitude,settings.current)==OK) {setColor(GREEN);cerr
175   << "OK" << endl;}
176 else {setColor(RED);cerr
177   << "FAILED" << endl;} setColor(WHITE); cerr
178 << "-----" << endl
179 << "reading template file...";
180 stringstream template_string;
181 ifstream template_file (settings.template_file.c_str(),ifstream::in);
182 if (template_file) {template_string << template_file.rdbuf(); template_file.close();
183   setColor(GREEN); cerr
184   << "OK" << endl;}
185 else {setColor(RED); cerr

```

```
186     << "unable to open file" << endl;} setColor(WHITE); cerr
187 << "-----" << endl
188 << "converting wave to p40...";
189 stringstream pattern_string;
190 if (convertWave2P40 (samples,template_string,pattern_string,settings)==OK) {setColor(GREEN);cerr
191     << "OK" << endl;}
192 else {setColor(RED);cerr
193     << "FAILED" << endl;} setColor(WHITE); cerr
194 << "-----" << endl
195 << "writing outputfile...";
196 ofstream outfile (settings.pattern_file.c_str(),ofstream::binary);
197 if (outfile) {outfile << pattern_string.str(); setColor(GREEN);cerr
198     << "OK" << endl;}
199 else {setColor(RED); cerr
200     << "unable to open file" << endl;}
201 outfile.close();
202 setColor(WHITE);
203 return OK;
204 }
205
206 ret readIniFile(Parameter &settings)
207 {
208     if (settings.ini_file.empty()) return WRONG_INI_FILE;
209     long buf_len=0;
210     char ini_buffer[INI_BUFFER_SIZE];
211     // start reading cl.ini: check if cl_emu.ini exist
212     buf_len=GetPrivateProfileString("CL-Emulation", NULL, ";;;", ini_buffer, INI_BUFFER_SIZE,
213         settings.ini_file.c_str());
214     // ";;;" is a comment and can't be returned by routine (" " is returned when there are no subkeys)
215     if(strcmp(ini_buffer,";;;")==0) {setColor(RED); cerr
216         << "file not found" << endl;
217         return WRONG_INI_FILE;}
218     cerr << endl;
219     // load selected wave's name (=section name):
220     string wave_section;
221     buf_len=GetPrivateProfileString("General", "wave", "", ini_buffer, INI_BUFFER_SIZE,
222         settings.ini_file.c_str());
223     wave_section=ini_buffer;
224     if(buf_len==0){ setColor(RED); cerr
225         << "Wave specification in [General] missing!" << endl
226         << "use \"wave=_sectionname_\" to select a wave section." << endl;
227         return WRONG_INI_FILE;}
228     // Test if section exists:
229     buf_len=GetPrivateProfileString(wave_section.c_str(), NULL, ";;;", ini_buffer, INI_BUFFER_SIZE,
230         settings.ini_file.c_str());
231     if(strcmp(ini_buffer,";;;")==0) {setColor(RED); cerr
232         << "Selected wave section doesn't exist!" << endl
233         << "A section named as specified by \"wave=\" has to exist." << endl;
234         return WRONG_INI_FILE; }
235     // load selected wave's type and according settings:
```

```

236 string wave_type;
237 buf_len=GetPrivateProfileString(wave_section.c_str(), "type", "", ini_buffer, INI_BUFFER_SIZE,
238                               settings.ini_file.c_str());
239 wave_type=ini_buffer;
240 if(buf_len==0) {setColor(RED); cerr
241 << "Wave type specification missing!" << endl
242 << "use \"type=\" to specify the wave section's type. (e.g. \"type=tabor wave\")" << endl;
243 return WRONG_INI_FILE;}
244 if(wave_type.compare("tabor wave")!=0) {setColor(RED); cerr
245 << "Unknown \"type=\" statement!" << endl
246 << "Use one of the following: \"tabor wave\", \"predefined shape\", \" << endl;
247 return WRONG_INI_FILE;}
248 // load wave_file
249 buf_len=GetPrivateProfileString(wave_section.c_str(), "wave_file", "", ini_buffer, INI_BUFFER_SIZE,
250                               settings.ini_file.c_str());
251 settings.wave_file = ini_buffer;
252 if(buf_len==0) {setColor(RED); cerr
253 << "Missing \"wave_file=\" in ini-file\"" << endl;
254 return WRONG_INI_FILE;}
255 setColor(KHAKI);cerr
256 << "wave_file=\"" << settings.wave_file << "\"" << endl;
257 // load pattern_file
258 buf_len=GetPrivateProfileString(wave_section.c_str(), "pattern_file", "", ini_buffer, INI_BUFFER_SIZE,
259                               settings.ini_file.c_str());
260 settings.pattern_file = ini_buffer;
261 if(buf_len==0) {setColor(RED); cerr
262 << "Missing \"pattern_file=\" in ini-file" << endl; settings.pattern_file = "pattern.p40";}
263 else settings.pattern_file = ini_buffer;
264 setColor(KHAKI);cerr
265 << "pattern_file=\"" << settings.pattern_file << "\"" << endl;
266 // load template_file
267 buf_len=GetPrivateProfileString(wave_section.c_str(), "template_file", "", ini_buffer,
268                               INI_BUFFER_SIZE, settings.ini_file.c_str());
269 settings.template_file = ini_buffer;
270 if(buf_len==0) {setColor(RED); cerr
271 << "Missing \"template_file=\" in ini-file" << endl; setColor(KHAKI); cerr
272 << "No template is used" << endl;}
273 else {setColor(KHAKI);cerr
274 << "template_file=\"" << settings.template_file << "\"" << endl;}
275 // load tabor_sample_rate
276 buf_len=GetPrivateProfileString(wave_section.c_str(), "tabor_sample_rate", "", ini_buffer,
277                               INI_BUFFER_SIZE, settings.ini_file.c_str());
278 if(buf_len==0) {setColor(RED); cerr
279 << "Missing \"tabor_sample_rate=\" in ini-file" << endl;
280 settings.tabor_sample_rate = TABOR_SAMPLE_RATE;}
281 else settings.tabor_sample_rate = strtod(ini_buffer, NULL);
282 setColor(KHAKI);cerr
283 << "tabor_sample_rate=" << settings.tabor_sample_rate << "MHz" << endl;
284 // load spea_sample_frequency
285 buf_len=GetPrivateProfileString("Spea", "spea_sample_rate", "", ini_buffer, INI_BUFFER_SIZE,

```

```

286         settings.ini_file.c_str());
287 if (buf_len==0) {setColor(RED); cerr
288     << "Missing \"spea_sample_rate=\" in ini-file" << endl; settings.spea_sample_rate = SPEA_SAMPLE_RATE
        ;}
289 else settings.spea_sample_rate = strtod(ini_buffer,NULL);
290 setColor(KHAKI);cerr
291     << "spea_sample_rate=" << settings.spea_sample_rate << "MHz" << endl;
292 // load amplitude
293 buf_len=GetPrivateProfileString(wave_section.c_str(), "amplitude", "", ini_buffer, INI_BUFFER_SIZE,
294     settings.ini_file.c_str());
295 if(buf_len==0) {setColor(RED); cerr
296     << "Missing \"amplitude=\" in ini-file" << endl; settings.amplitude = TABOR_AMP;}
297 else settings.amplitude = strtod(ini_buffer,NULL);
298 setColor(KHAKI);cerr
299     << "amplitude=" << settings.amplitude << endl;
300 // load current
301 buf_len=GetPrivateProfileString(wave_section.c_str(), "current", "", ini_buffer, INI_BUFFER_SIZE,
302     settings.ini_file.c_str());
303 if(buf_len==0) {setColor(RED); cerr
304     << "Missing \"current=\" in ini-file" << endl; settings.current = CURRENT;}
305 else settings.current = strtod(ini_buffer,NULL);
306 setColor(KHAKI);cerr
307     << "current=" << settings.current << endl;
308 // load dummy
309 buf_len=GetPrivateProfileString(wave_section.c_str(), "dummy", "", ini_buffer, INI_BUFFER_SIZE,
310     settings.ini_file.c_str());
311 if(strcmp(ini_buffer,"yes")==0) {setColor(KHAKI); cerr
312     << "Generating a dummy pattern" << endl; settings.dummy=true;}
313 else settings.dummy=false;
314 setColor(WHITE);cerr << "reading ini-file...";
315 return OK;
316 }
317
318 ret readWaveFile (ifstream &infile,vector<uint16> & samples)
319 {
320     ret state=OK;
321     uint8 last_byte=0;
322     if (infile.good()) last_byte = infile.get();
323     while (infile.good())
324     {
325         // putting bytes together
326         uint16 sample=0;
327         uint8 first_byte=0;
328         uint8 next_byte=0;
329         first_byte = last_byte;
330         next_byte = infile.get();
331         if (infile.good())
332             last_byte = infile.get();
333         else
334             state=FAILED;

```

```
335     if (state==OK)
336     {
337         if (ENDIAN==BIG_ENDIAN)
338         {
339             sample = first_byte;
340             sample = sample << 8;
341             sample |= next_byte;
342         }
343         else // little_endian
344         {
345             sample = next_byte;
346             sample = sample << 8;
347             sample |= first_byte;
348         }
349         samples.push_back(sample);
350     }
351 }
352 return state;
353 }
354
355 ret negativSamples (vector<uint16>& samples)
356 {
357     ret negative_samples=OK;
358     for (uint64 index=0; index<samples.size(); index++)
359     {
360         if (samples[index]&0xE000) // check if the upper 3 bits are set
361         {
362             negative_samples=FAILED;
363             samples[index]&=0x1FFF; // delete the upper 3 bits
364         }
365     }
366     return negative_samples;
367 }
368
369 // downsampling -> skip samples not needed
370 ret sampleRateConversion (vector<uint16>& samples, double tabor_sample_rate , double &spea_sample_rate)
371 {
372     uint32 sample_ratio=static_cast<int>(tabor_sample_rate/spea_sample_rate);
373     if (sample_ratio<(tabor_sample_rate/spea_sample_rate)) sample_ratio++;
374     if (sample_ratio==0 || sample_ratio==1) return FAILED;
375     if (samples.size()<2) return FAILED;
376     for (uint32 index=1; index<(samples.size()-1); index++)
377     {
378         for (uint64 cnt=0; cnt<(sample_ratio-1) && index<(samples.size()-1); cnt++)
379             samples.erase(samples.begin()+index);
380     }
381     spea_sample_rate=tabor_sample_rate/sample_ratio;
382     return OK;
383 }
384
```

```
385 ret multiplyAmplitude (vector<uint16>& samples,double amplitude,double current)
386 {
387     for (uint64 index=0; index<samples.size() ; index++)
388         samples[index]=static_cast<uint16>((static_cast<double>(samples[index])/amplitude)*current*2.0);
389     return OK;
390 }
391
392 ret convertWave2P40 (vector<uint16> in_buffer,stringstream& template_stringstream,
393                     stringstream& out_buffer,Parameter settings)
394 {
395     string template_string;
396     template_string=template_stringstream.str();
397     int64 wave_begin=getInt64FromString(template_string,VAR_WAVE_BEGIN,"");
398     int64 wave_end=wave_begin+in_buffer.size();
399     if (replaceNumberInString(template_string,VAR_WAVE_END,"",wave_end)==FAILED)
400         {setColor(RED); cerr << endl << VAR_WAVE_END << " not found in template_file "};
401     if (replaceNumberInString(template_string,VAR_WAVE_LENGTH,"",wave_end-wave_begin)==FAILED)
402         {setColor(RED); cerr << endl << VAR_WAVE_LENGTH << " not found in template_file "};
403     if (replaceNumberInString(template_string,VAR_AMPLITUDE_RATIO,"",settings.amplitude)==FAILED)
404         {setColor(RED); cerr << endl << VAR_AMPLITUDE_RATIO << " not found in template_file "};
405     if (replaceNumberInString(template_string,VAR_CURRENT_RATIO,"",settings.current)==FAILED)
406         {setColor(RED); cerr << endl << VAR_CURRENT_RATIO << " not found in template_file "};
407     int64 begin_position=template_string.find(WAVE_BEGIN);
408     begin_position=template_string.find("\n",begin_position);
409     if (begin_position!=string::npos)
410     {
411         out_buffer << template_string.substr(0,begin_position-1) << endl;
412     }
413     if (!settings.dummy)
414     {
415         for (uint64 index=0;index < in_buffer.size();index++)
416         {
417             uint64 loop_begin=index+1;
418             uint16 sample = 0;
419             sample = in_buffer[index];
420             while (index < in_buffer.size() && in_buffer[index+1]==sample)
421                 index++;
422             out_buffer << "Pw(p1," << loop_begin+wave_begin << ".." << index+wave_begin+1 << ",\n";
423             uint16 curl=1;
424             for (uint8 pin_number = 0; pin_number < NUMBER_OF_PINS; pin_number++)
425             {
426                 if (sample&curl)
427                     out_buffer << 'H';
428                 else
429                     out_buffer << 'L';
430                 curl = curl << 1;
431             }
432             out_buffer << "\n");\n";
433         }
434     }
```

```
435 if (begin_position!=string::npos)
436 {
437     out_buffer << template_string.substr(begin_position);
438 }
439 return OK;
440 }
441
442 int64 getInt64FromString(string string_to_parse,string begin,string end)
443 {
444     size_t begin_position;
445     size_t end_position;
446     begin_position=string_to_parse.find(begin);
447     if (begin_position!=string::npos)
448     {
449         begin_position+=begin.size();
450         end_position=string_to_parse.find(end,begin_position);
451         if (end_position!=string::npos)
452         {
453             string str;
454             str=string_to_parse.substr(begin_position,end_position-begin_position);
455             return strtol(str.c_str(),NULL,10);
456         }
457         else return 0;
458     }
459     else return 0;
460 }
461
462 // find position between begin-string and end-string delete the string between
463 // and insert the converted data
464 template <class ElementType> ret replaceNumberInString(
465     string& string_to_insert,string begin,string end,ElementType data)
466 {
467     size_t begin_position;
468     size_t end_position;
469     begin_position=string_to_insert.find(begin);
470     if (begin_position!=string::npos)
471     {
472         begin_position+=begin.size();
473         end_position=string_to_insert.find(end,begin_position);
474         if (end_position!=string::npos)
475         {
476             string str=numberToString(data);
477             string_to_insert.erase(begin_position,end_position-begin_position);
478             string_to_insert.insert(begin_position,str);
479             return OK;
480         }
481         else return FAILED;
482     }
483     else return FAILED;
484     return OK;
```

```
485 }
486
487 template <class ElementType> string numberToString(const ElementType input)
488 {
489     stringstream stream_buffer;
490     stream_buffer << input;
491     string string_buffer;
492     string_buffer = stream_buffer.str();
493     return string_buffer;
494 }
495
496 void setColor(unsigned short color)
497 {
498     HANDLE hcon = GetStdHandle(STD_OUTPUT_HANDLE);
499     SetConsoleTextAttribute(hcon,color);
500 }
```

Listing H.1: Tabor .wav auf Spea .p40 Konvertierungssoftware

H.2 Spea .p40 Template

```
1 // Version:    01.00
2 // Date:      10.03.2007
3 // Description: CL-Emulation test
4 // Last change: 13.05.2011
5 //
6 // Pin description:
7 // Pin 112 = I/O
8 // Pin 113 = DAC RESET
9 // Pin 114 = DAC CLOCK
10 // Pin 115 = DAC DBO
11 // Pin 116 = DAC DB1
12 // ...
13 // Pin 128 = DAC DB13
14
15 //***** Settings BEGIN *****
16
17 // Clock settings
18 double steptime = 0.031606; // test period 31.606n (31.64MHz)
19 double strobe = 0.020; // strobe edge 180ns
20 double edge1 = 0.017777; // CLK rising edge
21 double edge2 = 0.030804; // CLK falling edge
22
23 // Start pulse settings // Start up pulse is a stronger single pulse at the beginning
24 long start_pulse_length = 8; // Start pulse length in cycles
25 double start_pulse_overshoot = 10; // % Overshoot
26
27 // Wake up pulse settings // Wake up pulses are rectangle wave pulses at the beginning
```

```

28                                     // to give the CPU a clock for executing a startup software
29 int wakeup_cycles=2;                 // Wake up pulses half periodic time
30 double wakeup_level=0.25;           // The level of the wake up pulse. 1 = full scale
31
32 // Label settings
33 long label_step=1000;                // to jump directly to a point in the wave you can use labels
34                                     // label_step is the distance between 2 labels
35
36 //***** tabor2spea -> *****
37 double amplitude_ratio=0.85;
38 //***** tabor2spea -> *****
39 double current_ratio=0.1;
40
41 // Wave settings:
42 //***** -> tabor2spea *****
43 long wave_begin=50000;
44 //***** tabor2spea -> *****
45 long wave_end=0;
46 //***** tabor2spea -> *****
47 long wave_length=0;
48
49 long end;                            // absolut end position
50
51 // Pin settings:
52 // pin_list
53 int * total_pin_list = {113,114,115,116,117,118,119,120,121,122,123,124,125,126,127,128,112};
54 // dac_pin_list
55 int * pl = {115,116,117,118,119,120,121,122,123,124,125,126,127,128};
56
57 //***** Settings END *****
58
59 // variables declaration
60 long position=0;                     // label position
61 long number=0;                       // for intermediate result
62 long index=1;                        // position counter for start pulses and wake up
63 char Buffer[15];                     // data vector
64 int loop_cnt;                        // counter for bit operation
65 long bit_operation;                 // for bit manipulations
66
67 // End of pin assignment and variable declaration
68
69 main ()
70 {
71     end=wave_end;
72     wakeup_level=current_ratio;
73
74     // Initialize the whole pattern
75     STime (step_time);                 // Step Time: STime(double time)
76     // Parallel Write: // Pw(int *pin_list, long *step_list, char *sequence)
77     // sequence: H | L | I | O | C | X | U | D

```

```

78 //          High | Low | In | Out | ...
79 Pw(total_pin_list,1..wave_end,"LHLLLLLLLLLLLLLLLL");
80 Pw(total_pin_list,1..wave_end,"IIIIIIIIIIIIIIIO");
81 Pw(total_pin_list,1..wave_end,"XXXXXXXXXXXXXXXXXC");
82
83 // Set edges for the clock:
84 FormatEdgeSel(114,1,2); // FormatEdgeSel(int *pin_list, int rise_edge, int fall_edge)
85 ETime(edge1,1,1);      // This function allows to set the time for the specified edge
86 ETime(edge2,1,2);      // ETime(double time[us][, int set_up=1[, int edge]])
87 ETime(strobe,1,3);     //
88 PulsEn(114,0);        // PulsEn(int *pin_list, [int pt, int pts, int ph])
89                        // pt(pulse type) : 0=RZ Return to Zero
90                        //                1=RZI Return to Zero Inverted
91                        //                2=DNrz Do Not Return to Zero
92                        //                3=XOR  eXclusive OR
93
94 // Start puls
95 // calculating the start up level
96 // wakeup_level*FS*start_pulse_overshoot
97 //16383=2^14-1 at 14bit is the full scale value
98 bit_operation=wakeup_level*amplitude_ratio*16383.*((start_pulse_overshoot+100.)/100.);
99 // converting a number in to a string of 'H' and 'L'
100 for (loop_cnt=0; loop_cnt<14; loop_cnt=loop_cnt+1)
101 {
102     if (bit_operation & 0x01) // find out if LSB is set or not
103         Buffer[loop_cnt]='H';
104     else
105         Buffer[loop_cnt]='L';
106     bit_operation >>= 1;      // bit operation: shift right
107 }
108 Buffer[14]='\0';             // End of string
109 Pw(pl,1..start_pulse_length,Buffer);
110
111
112 // Chip wakeup
113 // calculating the wake up clock level
114 // wakeup_level*FS
115 //16383=2^14-1 at 14bit is the full scale value
116 bit_operation=wakeup_level*amplitude_ratio*16383;
117 // converting a number in to a string of 'H' and 'L'
118 for (loop_cnt=0; loop_cnt<14; loop_cnt=loop_cnt+1)
119 {
120     if (bit_operation & 0x01)
121         Buffer[loop_cnt]='H';
122     else
123         Buffer[loop_cnt]='L';
124     bit_operation >>= 1;
125 }
126 Buffer[14]='\0';
127 for (index=start_pulse_length+1; index+(wakeup_cycles*2)<wave_begin; index=index+(wakeup_cycles*2))

```

```

128 Pw(pl,index..(index+wakeup_cycles-1),Buffer);
129
130 //***** tabor2spea -> *****
131 // Waveform:
132
133 // add step labels:
134 // typical label name: 1 <= "s0000001", 2 <= "s0000002", ...
135 // label name is equal to step position
136 // labels are starting at wave_begin and coming every label_step and ending at wave_end
137 Buffer[0]='s';
138 Buffer[8]='\0';
139 for(position = wave_begin; position <= wave_end ; position=position+label_step)
140 {
141     number=position;
142     Buffer[1]='0'+(number/1000000); // number is converted to a string
143     number=number-(number/1000000)*1000000; // starting with 's' and stepposition as ASCII
144     Buffer[2]='0'+(number/100000); //
145     number=number-(number/100000)*100000; //
146     Buffer[3]='0'+(number/10000); //
147     number=number-(number/10000)*10000; //
148     Buffer[4]='0'+(number/1000); //
149     number=number-(number/1000)*1000; //
150     Buffer[5]='0'+(number/100); //
151     number=number-(number/100)*100; //
152     Buffer[6]='0'+(number/10); //
153     number=number-(number/10)*10; //
154     Buffer[7]='0'+number; //
155     Label(position, Buffer); // Label(long step, char * name)
156 }
157 EndP(end); // Pattern End
158 }

```

Listing H.2: CL-template.p40

H.3 CL-Emulation.ini Datei

```

1 [CL-Emulation]
2 ; This ini file is to setup the CL-Emulation
3 ;
4 ; The ini file can contain more wave sections
5 ; In the wave section you have to define the parameters of the wav file
6 ; You have to select the section in [General] with key: wave=tabor1
7
8 [General]
9 ;wave=TestWave
10 wave=tabor1
11 number_of_bits=14
12 PINS_DAC=128,127,126,125,124,123,122,121,120,119,118,117,116,115

```

```
13 channels_dac=128,127,126,125,124,123,122,121,120,119,118,117,116,115
14 channel_dac_clock=114
15 channel_dac_reset=113
16 channel_IO=112
17
18 [Spea]
19 spea_sample_rate=31.64 ;[MHz]
20 delay=1;
21
22 [tabor1]
23 type=tabor wave
24 wave_file=. \pattern.wav
25 pattern_file=. \test_10.p40
26 template_file=. \CL-template.p40
27 tabor_sample_rate=94.92 ;[MHz]
28 amplitude=0.85
29 current=0.1
30 dummy=no
31
32 [TestWave]
33 type=predefined shape
34 ; r# for rising slope steps, f# for falling slope steps
35 ; start with r0/f0 and go on for as many steps as needed (r1, r2 ,r3,...)
36 ; 1.0 equals nominal amplitude, use values up to 1.25 for overshoots!
37 r0=0
38 r1=0.1
39 r2=0.2
40 r3=0.3
41 r4=0.4
42 r5=0.5
43 r6=0.6
44 r7=0.7
45 r8=0.8
46 r9=0.9
47 r10=1
48 f0=1
49 f1=0.9
50 f2=0.8
51 f3=0.7
52 f4=0.6
53 f5=0.5
54 f6=0.4
55 f7=0.3
56 f8=0.2
57 f9=0.1
58 f10=0
```

Listing H.3: CLemu.ini

I Fotos

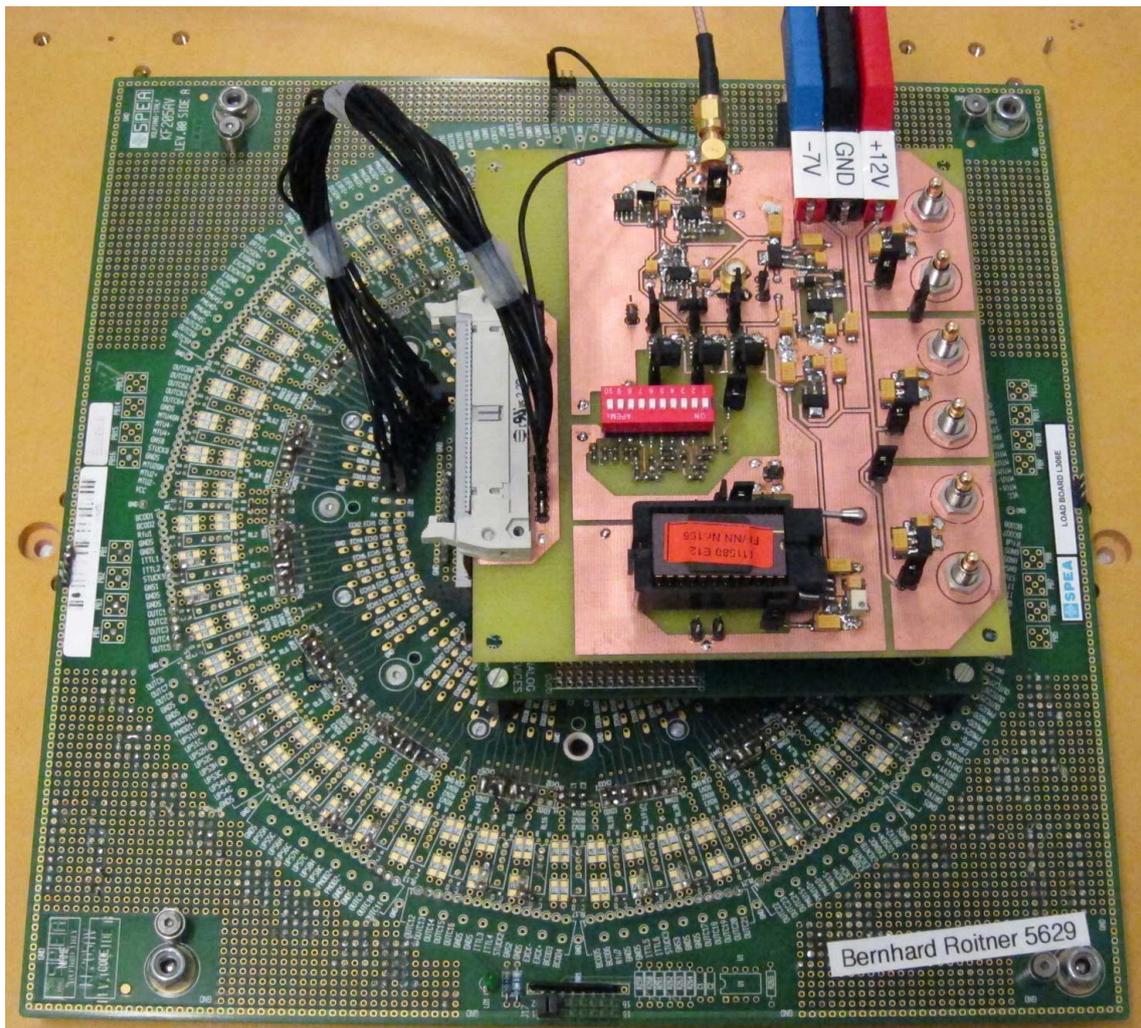


Abbildung I.1: CL-Emulator

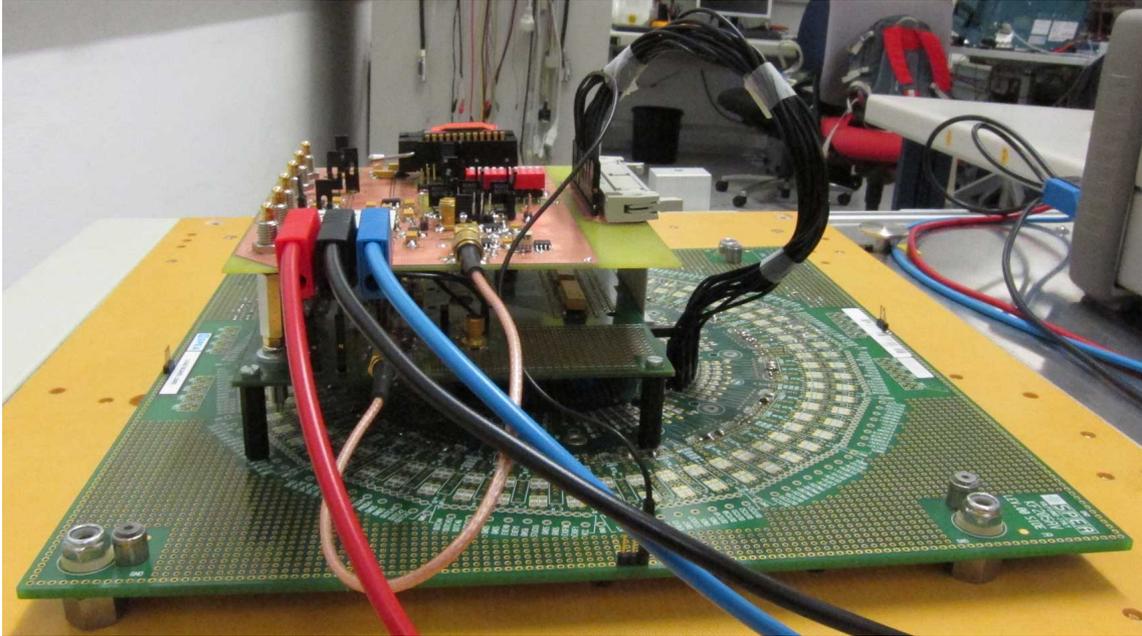


Abbildung I.2: CL-Emulator

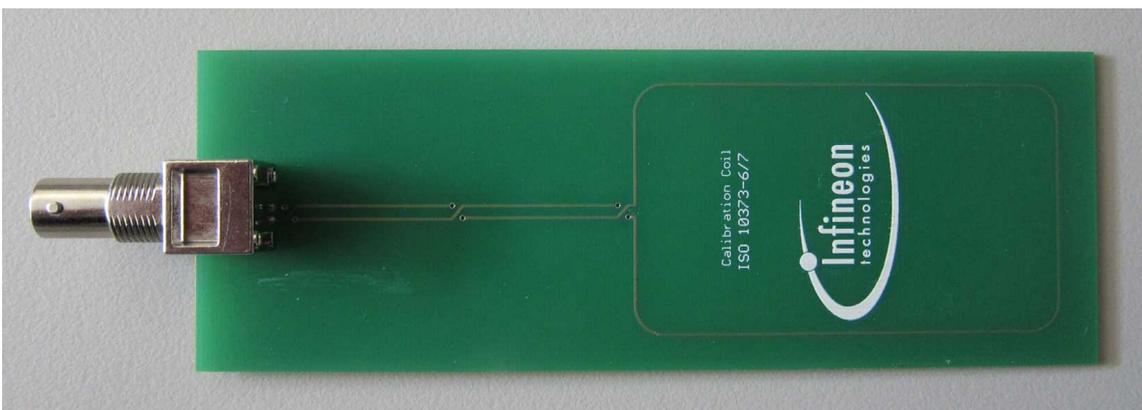


Abbildung I.3: Calibration Coil

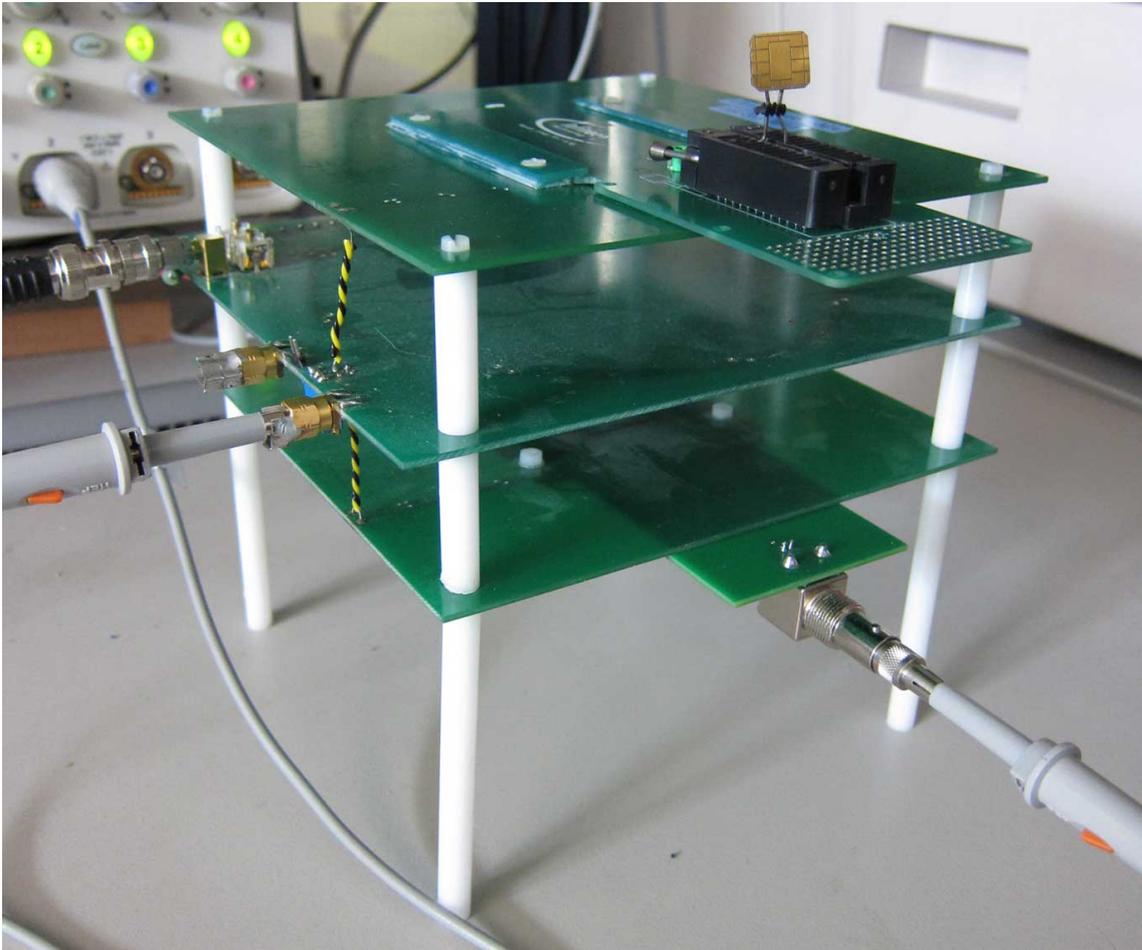


Abbildung I.4: ISO-Turm

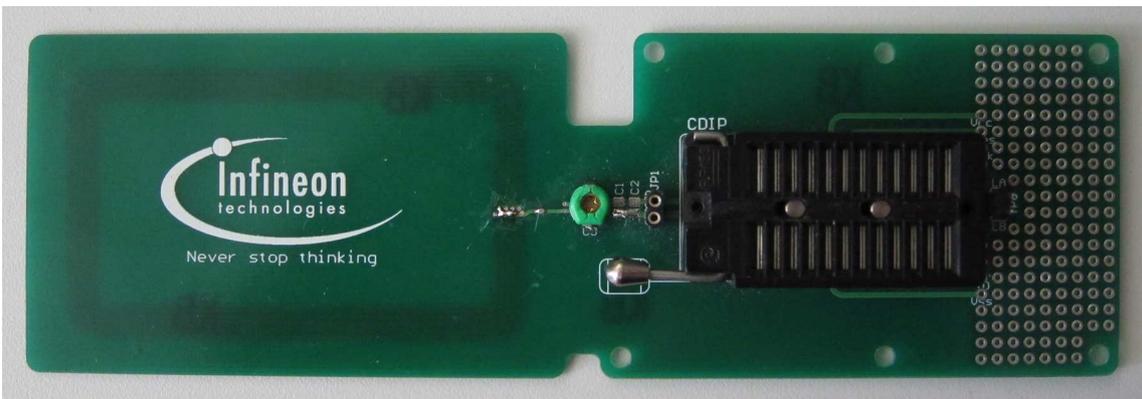


Abbildung I.5: Reference PICC

J Formeln

elektrisch	magnetisch
Induktionsgesetz $\vec{\nabla} \times \vec{E} = -\frac{\partial \vec{B}}{\partial t}$ $\oint_{\partial A} \vec{E} \cdot d\vec{s} = -\frac{d}{dt} \left(\iint_A \vec{B} \cdot d\vec{A} \right)$	Durchflutungsgesetz $\vec{\nabla} \times \vec{H} = \vec{j}_l + \frac{\partial \vec{D}}{\partial t}$ $\oint_{\partial A} \vec{H} \cdot d\vec{s} = \iint_A \left(\vec{j}_l \cdot d\vec{A} \right) + \frac{d}{dt} \left(\iint_A \vec{D} \cdot d\vec{A} \right)$
elektrische Feldstärke $[E] = \frac{V}{m}$	magnetische Feldstärke $[H] = \frac{A}{m}$
elektrische Flussdichte $[D] = \frac{A \cdot s}{m^2} = \frac{C}{m^2}$ $\vec{D} = \epsilon \cdot \vec{E}$	magnetische Flussdichte $[B] = \frac{V \cdot s}{m^2} = T$ $\vec{B} = \mu \cdot \vec{H}$
dielektrische Leitfähigkeit $[\epsilon] = \frac{A \cdot s}{V \cdot m}$	magnetische Leitfähigkeit $[\mu] = \frac{V \cdot s}{A \cdot m}$
elektrischer Fluss $[\Psi] = A \cdot s = C$ $\Psi = \int_A \vec{D} \cdot d\vec{A}$	magnetischer Fluss $[\Phi] = V \cdot s = Wb$ $\Phi = \int_A \vec{B} \cdot d\vec{A}$ Verketteter Fluss $\Psi = N \cdot \Phi$
Kapazität $[C] = \frac{A \cdot s}{V} = F$ $C = \frac{\Psi}{U}$	Induktivität $[L] = \frac{V \cdot s}{A} = H$ $L = \frac{\Psi}{I}$

Tabelle J.1: Elektrische und magnetische Felder