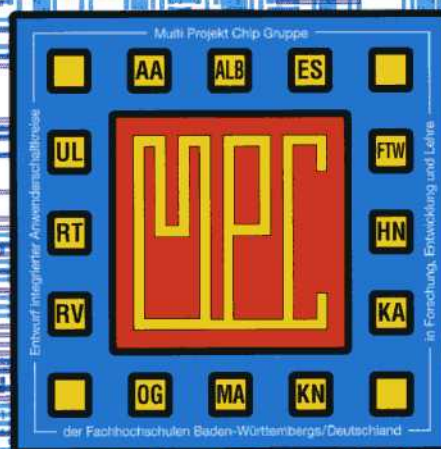


MULTIPROJEKT CHIP-GRUPPE

BADEN - WÜRTTEMBERG

MPC-Workshop Juni 2002

Reutlingen



MULTIPROJEKT CHIP-GRUPPE

BADEN - WÜRTTEMBERG

MPC-Workshop Juni 2002

Reutlingen

Cooperating Organization
Solid-State Circuits Society Chapter
IEEE Germany Section



Herausgeber: Fachhochschule Ulm

© 2002 Fachhochschule Ulm

Das Werk und seine Teile sind urheberrechtlich geschützt. Jede Verwertung in anderen als den gesetzlich zugelassenen Fällen bedarf deshalb der vorherigen schriftlichen Einwilligung des Herausgebers Prof. A. Führer, Fachhochschule Ulm, Prittwitzstraße 10, 89075 Ulm.

Adressen der

MULTIPROJEKT-CHIP-GRUPPE (MPC-Gruppe) **BADEN - WÜRTTEMBERG**

<http://www.mpc.belwue.de>

Fachhochschule Aalen

Prof. Dr. Kohlhammer, Postfach 1728, 73428 Aalen

Tel.: 07361/576-296, Fax: -324, Email: bernd.kohlhammer@fh-aalen.de

Fachhochschule Albstadt-Sigmaringen

Prof. Dr. Rieger, Johannesstr. 3, 72458 Albstadt-Ebingen

Tel.: 07431/579-124, Fax: -149, Email: rieger@fh-albsig.de

Fachhochschule Esslingen

Prof. Dr. Kampe, Flandernstr. 101, 73732 Esslingen

Tel.: 0711/397-4221, Fax: -4212, Email: gerald.kampe@fht-esslingen.de

Fachhochschule Furtwangen

Prof. Dr. Rülling, Postfach 28, 78113 Furtwangen

Tel.: 07723/920-503, Fax: -610, Email: ruelling@fh-furtwangen.de

Fachhochschule Heilbronn

Prof. Dr. Clauss, Max-Planck-Str. 39, 74081 Heilbronn

Tel.: 07131/504400, Fax: /252470, Email: clauss@fh-heilbronn.de

Fachhochschule Karlsruhe

Prof. Dr. Koblitz, Postfach 2440, 76012 Karlsruhe

Tel.: 0721/925-2238, Fax: -2259, Email: koblitz@fh-karlsruhe.de

Fachhochschule Konstanz

Prof. Dr. Voland, Brauneggerstraße 55, 78462 Konstanz

Tel.: 07531/206-644, Fax: -559, Email: voland@fh-konstanz.de

Fachhochschule Mannheim

Prof. Dr. Albert, Speyerer Str. 4, 68136 Mannheim

Tel.: 0621/2926-351, Fax: -454, Email: g.albert@fh-mannheim.de

Fachhochschule Offenburg

Prof. Dr. Jansen, Badstr. 24, 77652 Offenburg

Tel.: 0781/205-267, Fax: -242, Email: d.jansen@fh-offenburg.de

Fachhochschule Pforzheim

Prof. Dr. Kesel, Tiefenbronner Str. 65, 75175 Pforzheim

Tel.: 07321/28-6567, Fax: -6060, Email: kesel@fh-pforzheim.de

Fachhochschule Ravensburg-Weingarten

Prof. Dr. Ludescher, Postfach 1261, 88241 Weingarten

Tel.: 0751/501-9685, Fax: -9876, Email: ludescher@fbe.fh-weingarten.de

Fachhochschule Reutlingen

Prof. Dr. Kreutzer, Federnseestr. 4, 72764 Reutlingen

Tel.: 07121/341-108, Fax: -100, Email: hans.kreutzer@fh-reutlingen.de

Fachhochschule Ulm

Prof. Führer, Postfach 3860, 89028 Ulm

Tel.: 0731/50-28338, Fax: -28363, Email: fuehrer@fh-ulm.de

Inhaltsverzeichnis

Workshop-Vorträge

1. Methoden, Werkzeuge und Architekturen zum Entwurf und zur Realisierung von Signal- und Bildverarbeitungsalgorithmen für die Umsetzung in programmierbarer Hardware 5
R. Bartholomä, F. Kesel, T. Greiner, FH Pforzheim
2. ATM-/UTOPIA-Testsystem als FPGA IP-Core 11
D. Schirmer, FH Ulm
3. CIS - Chip im Schmuck 21
A. Ratz, D. Ernst, F. Kesel, FH Pforzheim
4. Induktive bidirektionale Schnittstelle ähnlich ISO/IEC 14443-A 25
F. Baier, D. Jansen, FH Offenburg
5. DQPSK Modulator for Inductive Data Transmission 31
N. Fawaz, D. Jansen, FH Offenburg
6. FPGAs entwerfen - ein Lernprogramm im Selbststudium 49
E. Futterer, IMS Chips Stuttgart
7. ESD Schutz in Deep-Sub-Micron Technologien 57
H. Goßner, Infineon Technologies München
8. Digitale Designmethodik für die System-on-Chip Integration in Deep-Sub-Micron Technologien 89
K. M. Just, Infineon Technologies München

Methoden, Werkzeuge und Architekturen zum Entwurf und zur Realisierung von Signal- und Bildverarbeitungsalgorithmen für die Umsetzung in programmierbarer Hardware

R. Bartholomä, Prof. F. Kesel, Prof. T. Greiner
Fachhochschule Pforzheim, Tiefenbronnerstraße 65, 75175 Pforzheim

1. Einleitung

Der steigende Bedarf von immer komplexer werdenden Hardwaresystemen erfordert intelligente Synthesewerkzeuge, die abstrakte und übersichtliche Hardwarebeschreibungen erlauben. Seit einiger Zeit existieren Synthesewerkzeuge, die aus algorithmischen Beschreibungen Hardware erzeugen können. In diesem Beitrag wird eine Methodik vorgestellt, die unter Verwendung solcher Synthesewerkzeuge die Realisierung von Signal- und Bildverarbeitungsalgorithmen in Hardware erleichtert.

2. Aufgabenstellung

Die Hauptaufgabe der Arbeit bestand im Entwurf einer Methodik, die eine Umsetzung von Signal- und Bildverarbeitungsalgorithmen in programmierbare Hardware ermöglicht. Als Grundlage hierfür sollten High Level Synthese Werkzeuge und das Werkzeug MATLAB zum Einsatz kommen. High Level Synthese Werkzeuge erlauben im Gegensatz zu anderen bekannten Entwurfsmethoden eine abstraktere Modellierung von Hardware. Bei MATLAB handelt es sich um ein leistungsfähiges Softwarewerkzeug, das häufig zur Lösung von mathematischen und technischen Problemstellungen verwendet wird. MATLAB besitzt zahlreiche Visualisierungsmöglichkeiten und erlaubt im Vergleich mit anderen Hochsprachen (wie beispielsweise C/C++) ebenfalls eine abstraktere Modellierung von Algorithmen. Weiterhin wurde die entstandene Entwurfsmethodik

anhand eines Bildanalyse-Systems evaluiert. Dieses System soll quantitative Aussagen über Texturmerkmale von Bildern treffen können. Ein derartiges System kann beispielsweise für die Analyse von Oberflächeneigenschaften in der Metallindustrie eingesetzt werden. Abbildung 1 zeigt zwei typische Metalloberflächen mit unterschiedlichen Oberflächenbeschaffenheiten.

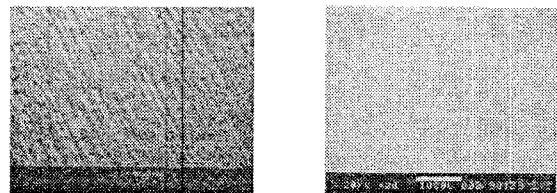


Abbildung 1 : Metallische Oberflächen mit unterschiedlichen Oberflächenbeschaffenheiten

Das Bildanalyse-System besteht im wesentlichen aus einer zweistufigen FIR-Filterbank mit 81 Kanälen. Da es sich bei Bildsignalen um zweidimensionale Signale handelt, müssen auch zweidimensionale Filter verwendet werden. Falls sich die zweidimensionale Impulsantwort eines Filters jedoch als Produkt zweier eindimensionaler Funktionen mit nur einer ortsabhängigen Variablen darstellen lässt ($h(n_1, n_2) = h(n_1) \cdot h(n_2)$), so kann ein separabler Filteransatz gewählt werden. In diesem Fall kann das zweidimensionale Filter durch zwei hintereinander geschaltete eindimensionale Filter realisiert werden. Dieser Ansatz verringert den Rechenaufwand erheblich. Die Struktur dieser Filterbank ist in Abbildung 2

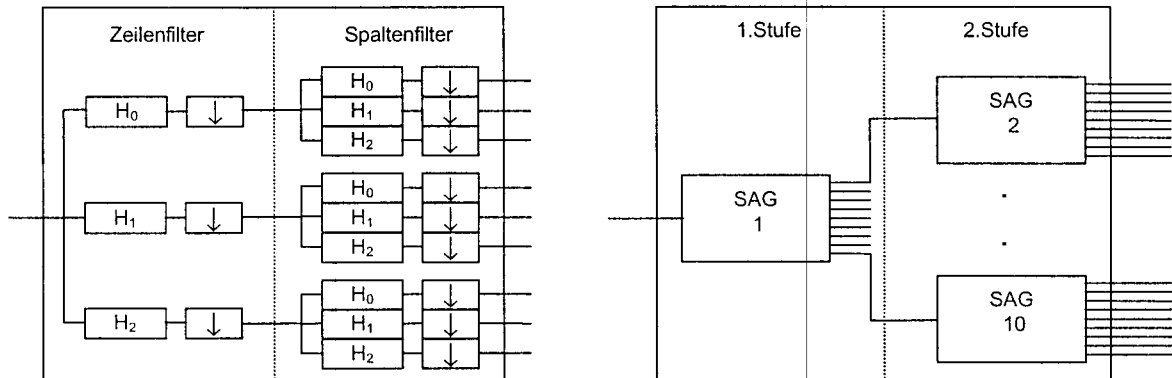


Abbildung 2 : Separable Analysegrundstruktur SAG (links) und Filterbank mit 81 Kanälen (rechts)

dargestellt. Die Filterbank besteht aus Tief-, Band- und Hochpässen, die das gesamte Spektrum in gleichmäßige Teile unterteilen. Jedem Filter wird ein Dezimierer nachgeschaltet, der die Bilddaten um den Faktor 3 dezimiert. Bei geeigneter Wahl der Filter kann aus den Energien der gefilterten Bilder eine Aussage über die Oberflächeneigenschaften des Quellbildes getroffen werden.

3. High Level Synthese Werkzeuge

Das Y Diagramm nach Gajski (Abbildung 3) zeigt die unterschiedlichen Abstraktionsebenen einer Hardware aus den Sichten Verhalten, Struktur und Geometrie.

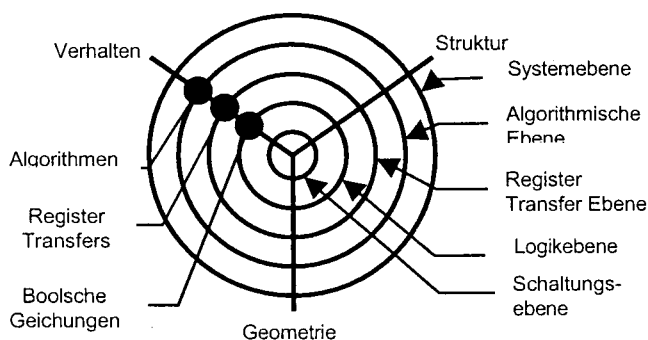


Abbildung 3 : Y-Diagramm nach Gajski

Während Hardware früher auf der Logikebene entworfen wurde, findet der derzeitige Hardwareentwurf in den meisten Fällen auf der Registertransferebene (RT-Ebene, engl. RTL -

Register Transfer Level) statt. Hierbei werden die Register und die Übertragungsfunktionen (Transfer) zwischen den Registern bitgenau und takttrichtig beschrieben. Diese RTL Beschreibungen werden dann automatisch mittels Synthesewerkzeug auf die Logikebene transferiert. Die Modellierung von Registern auf der RT-Ebene geschieht jedoch manuell. High Level Synthese Werkzeuge erlauben eine Modellierung von Hardware auf der algorithmischen Ebene. Auf dieser Ebene wird die gesamte zu synthetisierende Hardware in einer reinen Verhaltensbeschreibung ohne Nebenläufigkeiten modelliert. Die explizite Modellierung von Registern entfällt, was zu einer bitgenauen aber nicht taktgenauen Beschreibung der Hardware führt. Das Synthesewerkzeug generiert aus der Verhaltensbeschreibung und weiteren Randbedingungen (wie z.B. verfügbare Ressourcen, Flächen bzw.

Geschwindigkeitsoptimierungen) eine RTL-Beschreibung der Hardware. Die Aufteilung der Architektur in Transferfunktionen und Register wird vom Synthesewerkzeug in Abhängigkeit von den geforderten Randbedingungen vorgenommen, so dass durch Änderung der Randbedingungen eine komplett neue Hardwarestruktur erzeugt werden kann.

Die Vorteile einer abstrakteren Hardwarebeschreibung ergeben sich unmittelbar aus der Eigenschaft, dass eine derartige Beschreibung viel weniger Details bezüglich der tatsächlichen (physikalischen) Realisierung enthält. Da abstrakte Beschreibungen weniger Details enthalten, werden diese übersichtlicher und eignen sich somit besser für komplexe Systeme. Außerdem lassen sich in abstrakten Beschreibungen auch viel leichter Fehler

Methoden zum Entwurf von Signalverarbeitungsalgorithmen in programmierbarer Hardware

lokalisieren, was zu einer Verkürzung der Entwicklungszeiten führt.

Der High Level Syntheseprozess [1] ist in Abbildung 4 schematisch dargestellt. Ausgehend von der Verhaltensbeschreibung des Designs wird in mehreren Schritten eine Netzliste der Architektur auf RT-Ebene erzeugt.

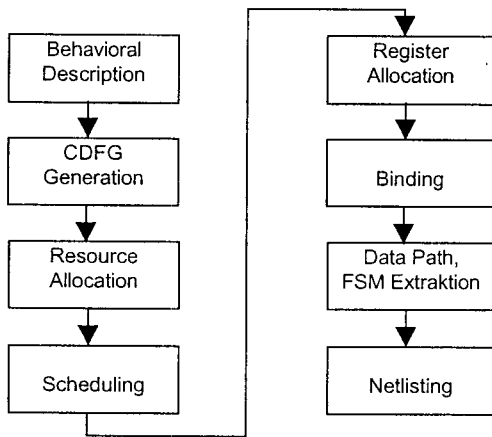


Abbildung 4 : High Level Synthese Prozess

Aus der Verhaltensbeschreibung wird zunächst ein Kontrolldatenflussgraph (CDFG) erzeugt, der die zeitlichen Abhängigkeiten der einzelnen Operationen modelliert. Die Knoten dieses Graphs modellieren die Operationen (Komparatoren, Multiplizierer, usw.) und die Kanten die zeitlichen Abhängigkeiten der Operatoren untereinander. Abbildung 5 zeigt ein Beispiel eines CDFGs. Bei der „Resource Allocation“ wird entschieden welche und wie viele Ressourcen im Design verwendet werden können. Dieser Schritt wird durch Randbedingungen wie maximale Anzahl der verfügbaren Ressourcen und gewünschte Zieltechnologie wesentlich beeinflusst. Das Synthese Werkzeug kann für die einzelnen Operationen im CDFG aus einer Menge von unterschiedlichen Komponententypen auswählen. Für eine Addition im CDFG kann beispielsweise ein Ripple-Addierer oder ein Carry-Save-Addierer verwendet werden. Der eigentliche Syntheseprozess findet beim „Scheduling“ statt. Dort werden die einzelnen Operationen aus dem CDFG an feste Taktzeitpunkte gebunden. Die zu generierende Architektur kann beim „Scheduling“ auf Zeit bzw. Fläche optimiert werden. Beim Schritt „Register Allocation“ werden in die Architektur Register eingefügt. Zuvor wird ermittelt welche Zwischenergebnisse von Operationen zu welchen Zeitpunkten präsent sein müssen („Lifetime

Analysis“). Wenn Zwischenergebnisse von unterschiedlichen Komponenten nicht zu einem gleichen Zeitpunkt gehalten werden müssen, kann für die Speicherung dieser Ergebnisse ein einziges Register verwendet werden. Beim „Binding“ wird jeder Operation eine allokierte Komponente zugeordnet. Schließlich werden in die Architektur Komponenten eingefügt, welche die Datenpfade steuern (Multiplexer usw.) und der Zustandsautomat für die Ablaufsteuerung des Designs wird extrahiert.

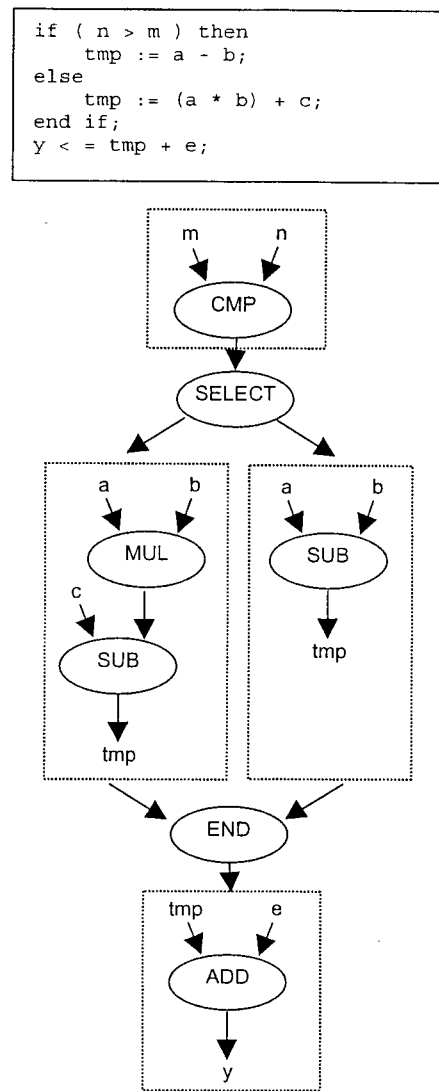


Abbildung 5 : Control Data Flow Graph (CDFG)

4. Entwurfsmethodik

Die in dieser Arbeit entstandene Entwurfsmethodik ermöglicht eine frühzeitige Simulation von Signal- bzw. Bildverarbeitungsalgorithmen für den Hardwareentwurf. Ein Problem beim Entwurf von Signalverarbeitungsalgorithmen für Hardware ist die begrenzte Möglichkeit der Zahlendarstellungsgenauigkeit. Es besteht immer ein Zusammenhang zwischen hoher Genauigkeit und schneller Rechengeschwindigkeit der Hardware. Wenn eine hohe Darstellungsgenauigkeit gewählt wird, dann müssen für die Implementierung der Hardware größere arithmetische Recheneinheiten verwendet werden. Diese besitzen eine größere Verzögerungszeit, was zu einer Verminderung der Rechengeschwindigkeit des gesamten Systems führt. Um die Rechengeschwindigkeit eines Systems optimieren zu können, ist es deshalb hilfreich das gesamte System zunächst hinsichtlich der minimalen Anforderungen von Genauigkeit aller Datenpfade und Rechenoperationen zu untersuchen.

Die entstandene Entwurfsmethodik ermöglicht daher Simulationen auf den folgenden vier unterschiedlichen Abstraktionsebenen:

- MATLAB Simulation (Gleitpunktarithmetik)
- MATLAB Simulation (Festpunktarithmetik)
- VHDL Simulation der Verhaltensbeschreibung
- VHDL Simulation der RTL Beschreibung

Das gesamte System wird zunächst in der MATLAB eigenen Sprache M modelliert und mit der in MATLAB üblichen 64 Bit Gleitpunktzahlendarstellung simuliert. Ausgehend von diesem Modell wird dann im zweiten Schritt eine MATLAB Simulation mit einer Festpunktarithmetik vorgenommen. Hierzu werden sämtliche Datenpfade und Rechenoperationen mit einer festen Datenwortbreite simuliert, so dass sich dieselben Quantisierungseffekte wie auf der Hardware ergeben. Mit diesen beiden Simulationen und den zahlreichen Darstellungsmöglichkeiten von MATLAB können die minimal erforderlichen Wortbreiten bereits im Vorfeld ermittelt werden. Im nächsten Schritt wird die VHDL-Verhaltensbeschreibung des Systems entworfen.

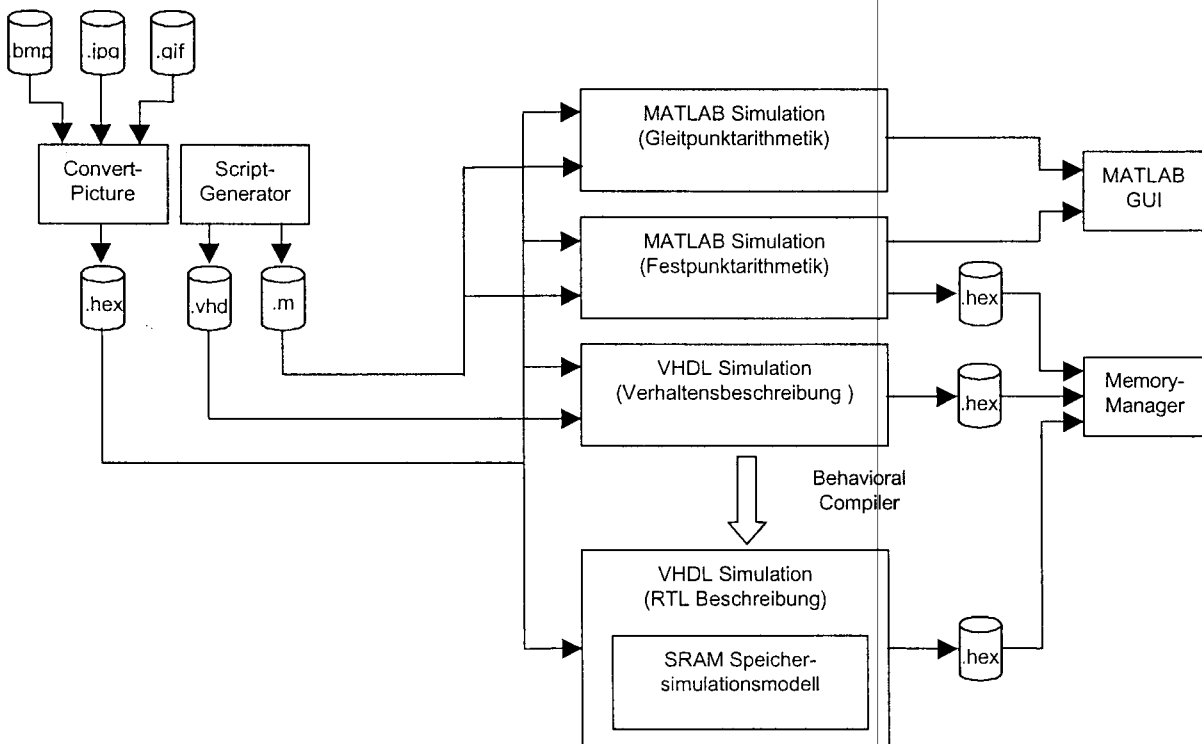


Abbildung 6 : Entwurfsablauf für das Bildanalyse-system aus Abschnitt 2



Für den Übergang von MATLAB nach VHDL ist im Rahmen dieser Arbeit eine gemeinsame Schnittstelle entstanden mit der Simulationsdaten und Testvektoren zwischen den beiden Sprachen ausgetauscht werden können. Für die Modellierung der Festpunktarithmetik in VHDL ist eine umfangreiche Bibliothek entstanden, die eine Simulation und Synthese mit dem High Level Synthese Werkzeug „Behavioral Compiler“ von Synopsys erlaubt. Nach erfolgreicher Simulation der VHDL Verhaltensbeschreibung wird mit dem High Level Synthese Werkzeug die RTL Architektur generiert und simuliert. Die MATLAB Festpunktarithmetik Simulation und beide VHDL Simulationen müssen bezüglich der Genauigkeit der Simulationsdaten exakt dasselbe Ergebnis liefern. Alle weiteren Syntheseschritte bis hin zur physikalischen Realisierung der Hardware unterscheiden sich nicht vom bisherigen Hardwareentwurf auf RT-Ebene. Abbildung 6 zeigt den Entwurfsablauf für das in Abschnitt 2 erwähnte Bildanalysesystem.

Das Werkzeug ConvertPicture erzeugt aus Bildern mit bekannten Formaten Simulationsdaten, die über die gemeinsame Schnittstelle in MATLAB und VHDL gelesen werden können. Das Werkzeug Script Generator erzeugt Simulationsskripte für MATLAB und VHDL mit globalen Parametern wie Filterkoeffizienten und Bildgrößen. Da die Simulationsergebnisse der MATLAB Festpunktarithmetiksimulation und beider VHDL Simulationen bezüglich der Genauigkeit exakt übereinstimmen müssen, werden diese über die gemeinsame Schnittstelle exportiert und mit dem Werkzeug MemoryManager auf Richtigkeit überprüft.

5. Zusammenfassung

In dieser Arbeit ist eine Methodik zum Entwurf von Signal - und Bildverarbeitungsalgorithmen für Hardware entstanden, welche auf der Verwendung von High Level Synthese Werkzeugen und MATLAB beruht. Für die Realisierung dieser Entwurfsmethodik wurden einige Werkzeuge entwickelt, die einen durchgängigen Entwurfsablauf ermöglichen. Es wurde eine gemeinsame Schnittstelle für den Austausch von Simulationsdaten zwischen den Sprachen MATLAB und VHDL geschaffen. Weiterhin entstand eine umfangreiche Bibliothek für die Simulation und Synthese von Festpunktarithmetikoperationen in VHDL. Mit der erarbeiteten Entwurfsmethodik

wurde ein Bildanalysesystem zur Erkennung von Texturmerkmalen entworfen und erfolgreich simuliert.

6. Literatur

- [1] John P. Elliott; Understanding Behavioral Synthesis; Kluwer Academic Publishers, 1999

ATM-/UTOPIA-Testsystem als FPGA IP-Core

Daniel Schirmer

Fachhochschule Ulm, Fachbereich Elektrotechnik und Informationstechnik, Prittwitzstr. 10,
89075 Ulm
schirmer@mail.fh-ulm.de

In der Hardwareabteilung von COM AC D AE der Firma Infineon Technologies AG werden Platinen für Halbleiterbausteine von Infineon für die drahtgebundene Kommunikation entwickelt. Diese Platinen dienen als Entwicklungsplattformen, Referenzdesigns und Designstudien. Auf diesen Boards ist häufig eine UTOPIA¹-Schnittstelle zu finden. Während der Entwicklung, Inbetriebnahme und Tests soll mit Hilfe eines FPGAs eine Möglichkeit geschaffen werden, die UTOPIA-Schnittstelle der Bausteine zu prüfen.

1 Einblick in ATM und UTOPIA

1.1 Asynchroner Transfer Modus

Der Asynchrone Transfer Modus ist ein verbindungsorientiertes Paketvermittlungsverfahren mit virtuellen Verbindungen in virtuellen Kanälen auf virtuellen Pfaden (siehe Bild 1).

Verbindungen laufen nach folgendem Schema ab: Verbindungsaufbau, Paketübertragung, Verbindungsabbau.

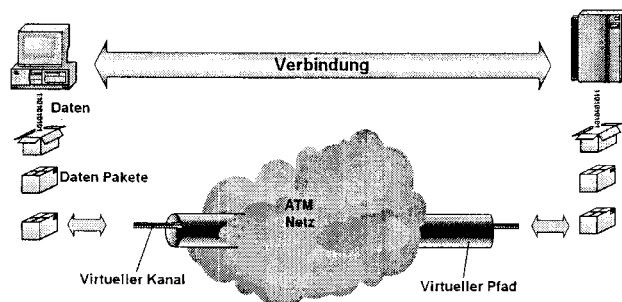


Bild 1 ATM Verbindung

ATM findet seinen Einsatz hauptsächlich in den Backbonenetzen von Netzbetreibern, aber auch in Local Area, Metropolitan Area und Wide Area Netzen ist der Einsatz denkbar. Bei dieser Übertragungstechnik bewegen sich die Anwendungen auf dem Gebiet der Multimedia und der breitbandigen Datenübertragung und sind weitestgehend vom Übertragungsmedium unabhängig.

Dabei stehen keine festen Bandbreiten zur Verfügung, sondern werden bei einem Verbindungsaufbau zwischen den Teilnehmern ausgehandelt.

Die zu übertragenden Daten werden in Pakete einer festen Größe umgewandelt. Diese Pakete sogenannte ATM Zellen, bestehen aus einem Zellheader und einem Teil der die Nutzdaten (Payload) enthält. In Bild 2 ist die Struktur einer solchen ATM-Zelle zu sehen. Die Größe der Zelle ist von der Datenbusbreite abhängig und beträgt zwischen 53 und 56 Bytes (siehe Tabelle 1). Der Zellheader enthält Informationen über die Nutzdaten, zur Adressierung, Priorisierung und Fehlerkorrektur.

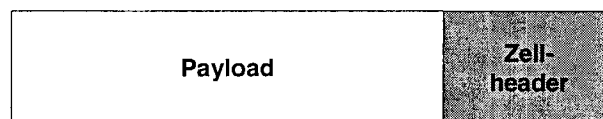
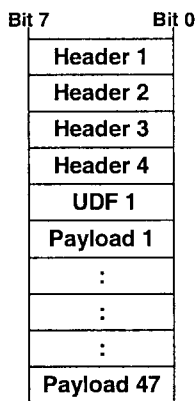


Bild 2 Struktur einer ATM Zelle

Über die Informationen aus dem Zellheader werden die ATM-Zellen durch das ATM-Netz geleitet. Dabei unterscheiden sich die Informationsfelder je nach dem welche Teilnehmer miteinander verbunden sind. Im allgemeinen sind Bereiche für die virtuellen Kanäle, Pfade, die Zugriffsteuerung und Nutzdateninformation vorhanden. Ergänzend sind in Abhängigkeit der Datenbusbreite (siehe Abschnitt 1.2) bis zu 4 benutzerdefinierte Bytes vorhanden.

Dabei dient das 5. Byte des Headers der Fehlerkorrektur im Header und wird als HEC¹-Byte bezeichnet. Der Aufbau einer ATM-Zelle für einen 8 und 16 Bit Datenbus ist in Bild 3 zu sehen, UDF 1² stellt dabei das HEC-Byte dar.

Zellaufbau 8 Bit:



Zellaufbau 16 Bit:

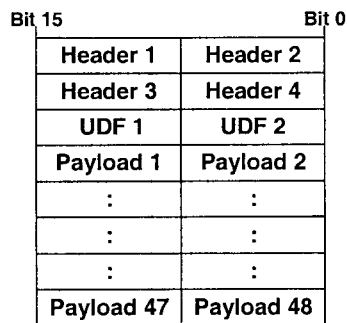


Bild 3 Zellenaufbau

In Anlehnung an das OSI Reference Modell [5] werden die Funktionen für den Asynchronen Transfer Modus in zwei ATM spezifischen Schichten definiert. In diesen Schichten findet die Anpassung der Nutzdaten an das Zellenformat und die Übertragung auf Zellebene statt [1].

1.2 UTOPIA-Interface

Zwischen den angesprochenen ATM-Schichten (ATM Layer) und der Bitübertragungsschicht, dem PHY Layer, befindet sich die UTOPIA-Schnittstelle. Sie bietet ein standardisiertes paralleles Interface, dass die modulare Anbindung bis zu 31 auch verschiedener Geräte der Bitübertragungsschicht an ein Gerät der ATM-Schicht ermöglicht. Aus der Tabelle 1 sind einige Rahmendaten der UTOPIA-Schnittstelle für einzelne Spezifikationslevel ersichtlich.

UTOPIA Spezifikation	Datenbusbreite (max.)	Übertragungsrate (max.)	Zellgröße (max.)
Level 1	8 Bit	155 Mb/s	53 Bytes
Level 2	16 Bit	622 Mb/s	54 Bytes
Level 3	32 Bit	3,2 Gb/s	56 Bytes
Level 4	32 Bit	10 Gb/s	56 Bytes

Tabelle 1: Datenrate UTOPIA

Die Schnittstelle selbst ist in eine Sende- (Tx) und Empfangsrichtung (Rx), aus Sicht des ATM Layer Device, aufgeteilt (siehe Bild 4). Die Signale des UTOPIA-Interfaces sind in Tabelle 2 erklärt.

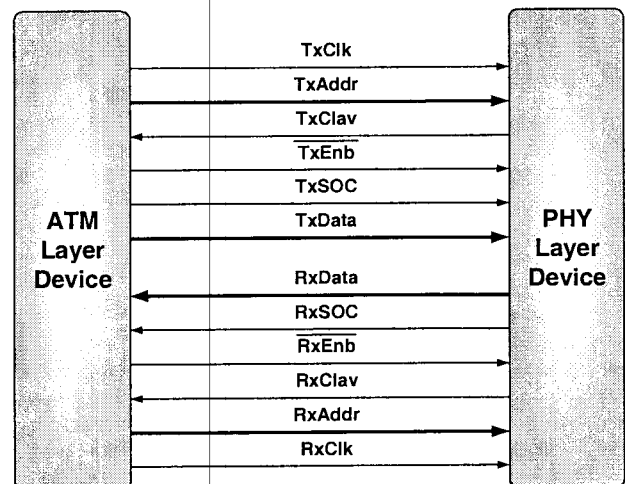


Bild 4 UTOPIA Interface

1. HEC Header Error Control
2. UDF User Defined (Benutzerdefiniertes Byte)

Signal	Erklärung
Clk	Taktsignal
Addr	5 Bit Adressbus
CLAV	Cell available
ENB	Enable
SOC	Start of Cell
DATA	Datenbus

Tabelle 2: UTOPIA Signale

Das ATM Layer Device stellt dabei immer den UTOPIA Master dar und die PHY Layer Devices verhalten sich als Slaves. Der Master besitzt die Kontrolle über die Schnittstelle und steuert die angeschlossenen PHYs. Dabei werden die PHYs nacheinander, über den Adressbus, abgefragt ob sie in der Lage sind ATM-Zellen zu senden bzw. zu empfangen. Dieser Vorgang wird als Standard Address Polling bezeichnet. Bild 5 zeigt ein Beispiel wie die Abfrage und Auswahl eines PHY Layer Devices durch den UTOPIA-Master auf dem Tx-Pfad aussehen kann. Im Detail läuft der Vorgang nach folgendem Schema ab. Über den UTOPIA-Adressbus werden die Slaves durch den Master angesprochen. Dazu legt der Master die Adresse eines angeschlossenen PHYs an den Adressbus (Takt 2,4,6,10). Die Slaves teilen dem Master über das Signal TxCLAV, während der folgenden Phase in der die Adresse $1F_h$ angelegt wird, mit, ob sie Zellen empfangen können (Takt 1,3,5,7,9,11). Da die Adresse $1F_h$ durch diesen Mechanismus für die Signalisierung verwendet werden können von den 32 möglichen UTOPIA-Adressen lediglich 31 durch PHY Layer Devices belegt werden. Die PHYs mit den Adressen 1 und 2 teilen dem Master in den Takten 3 und 5 mit, dass sie nicht in der Lage sind momentan Zellen zu empfangen, in dem sie das Signal TxCLAV während $1F_h$ auf Null legen. Mit Takt 8 stellt der Master fest, dass der Slave mit Adresse 3 bereit ist eine Zelle zu empfangen, da Slave 3 TxCLAV auf 1 zieht. In dem der Master erneut die Adresse 3 anlegt und gleichzeitig TxENB und TxSOC aktiviert wird der Slave 3 ausgewählt und die Zellübertragung gestartet. Mit Takt 10 setzt der Master das Standard Address Polling fort.

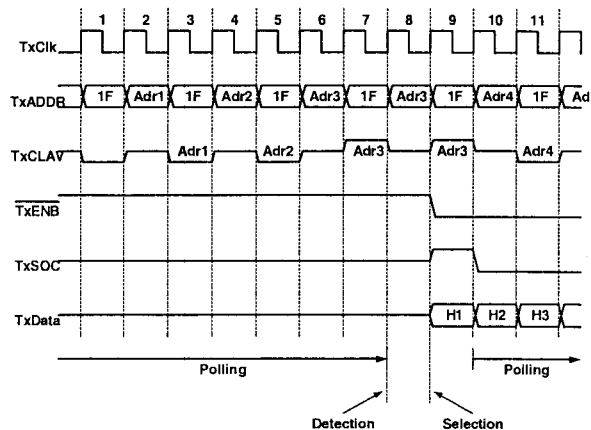


Bild 5 Standard Address Polling

Wenn nur ein PHY Layer Device an das ATM Layer Device angeschlossen ist erfolgt die Synchronisation über Direct Status Indication: Bei diesem Vorgang wird die Bezeichnung des Signals CLAV in FULL abgeändert. In Bild 6 ist der Ablauf einer Zellübertragung mit Direct Status Indication abgebildet. Nachdem der Slave dem Master mitteilt dass er in der Lage ist eine Zelle zu empfangen in dem er TxFULL auf 1 legt (Takt 0), startet der Master die Übertragung durch die Aktivierung von TxENB und TxSOC mit Takt 1. Im Takt 49 zeigt der Slave an, dass er nur noch 4 Bytes aufnehmen kann darauf hin stoppt der Master die Datenübertragung nach Takt 53. Bei diesem Modus sind der Master über Deaktivierung von TxENB und der Slave über TxFULL in der Lage die Übertragung während einer Zellübertragung zu unterbrechen.[2],[3],[4]

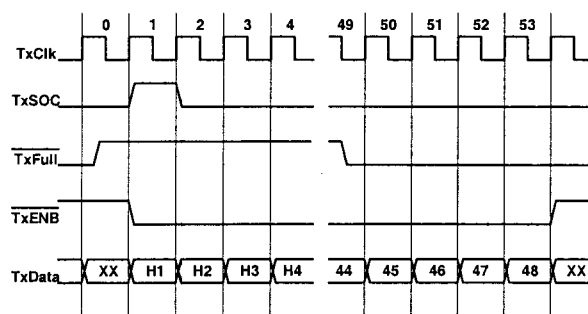


Bild 6 Direct Status Indication

2 Das Testsystem

Aufgabe des Testsystems ist es Funktionen für den Test und die Evaluierung von ATM basierten xDSL-Systemen bereit zu stellen. Zu diesem Zweck liegt das System an der UTOPIA-Schnittstelle zwischen der ATM und PHY Layer. Die Lage des Testsystems wird aus Bild 7 ersichtlich.

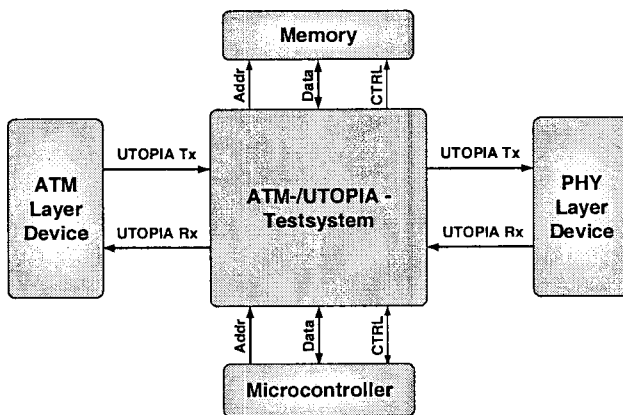


Bild 7 Testsystem

2.1 Aufbau

2.1.1 UTOPIA Slave Interface

Über diese Komponente wird die Verbindung zur ATM Layer hergestellt. Sie bildet dabei die Funktionen eines Gerät der PHY Layer mit bis zu 31 einzelnen PHYs nach. Die Anzahl der PHYs ist über generische Variablen wählbar. Diese Anzahl der PHY Layer Geräte kann dann während des Betriebes über eine Variable noch reduziert werden. Die Datenbusbreite des Tx- und Rx-Pfades ist ebenfalls über generische Variablen vor der Synthese zwischen 16 und 8 Bit konfigurierbar. Die Taktfrequenz des UTOPIA Slave Interfaces liegt bei maximal 50 MHz.

2.1.2 UTOPIA Master Interface

Durch diese Schnittstelle werden Funktionen eines UTOPIA-Masters realisiert. Über dieses Interface können maximal 31 PHY Layer Devices an das Testsystem angeschlossen werden. Dabei ist die Anzahl der PHYs und die Breite des Datenbusses über die gleichen Variablen wie beim Slave-Interface vor der Synthese auswählbar. Über eine weitere Variable ist die Anzahl der angeschlossenen PHYs während der Laufzeit reduzierbar.

Die mit der Schnittstelle verbundenen PHYs werden über, das in Abschnitt 1.2 besprochene, Standard Address Polling angesprochen und ausgewählt. Die maximale Taktfrequenz dieser UTOPIA-Schnittstelle liegt bei 50 MHz.

2.1.3 Microcontroller Interface

Diese Schnittstelle verfügt über einen 16 Bit breiten Adressbus und verfügt somit über einen Speicherbereich von 64 kB der vollständig für das Testsystem reserviert ist. In diesen Bereich bindet das Testsystem einen Teil des externen Speichers ein damit der Microcontroller direkt darin Daten ablegen und auslesen kann (Abschnitt 2.2.5). Die Daten werden byteweise auf dem 8 Bit Datenbus zwischen dem Testsystem und dem Microcontroller übertragen. Über eine Interruptleitung kann das System bestimmte Funktionen des Microcontroller auslösen. Diese Funktionen werden im Abschnitt 2.2.4 genauer erläutert. Bevor das System einen Interrupt auslöst wird die benötigte Funktion im Interrupt-Register über das Setzen des entsprechenden Bits angezeigt. Im Anschluß daran wird der Interrupt ausgelöst. Der Microcontroller liest das Interrupt-Register aus und löscht das Bit nachdem der Interrupt abgearbeitet wurde.

Bild 14 zeigt den internen Aufbau der Microcontroller Schnittstelle.

2.1.4 Memory Interface

Dem Testsystem steht ein externer Speicher zur Verfügung. Darin werden ATM-Zellen von der UTOPIA-Schnittstelle und vom Microcontroller abgelegt (siehe Abschnitt 2.2.5)

2.2 Testfunktionen

Die einzelnen Testfunktionen können während des Betriebs ausgewählt werden. Die Konfiguration dazu erfolgt durch Registerzugriffe über den Microcontroller. Verschiedene Statusinformationen können ebenfalls über den Microcontroller zur Laufzeit aus dem Testsystem ausgelesen werden.

2.2.1 Transparenter Mode

Bei dieser Testfunktion werden die ATM-Zellen ohne Veränderung durch das Testsystem weitergeleitet. Zellen werden von dem ATM Layer bzw. PHY Layer weiterhin an das System und nicht direkt zu Gegenstelle gesendet sondern es erfolgt zunächst eine Zwischenspeicherung der Daten bis die Zelle vollständig vom System empfangen ist. Danach werden die ATM-Zellen dann an die entsprechende Seite gesandt. Diese Vorgehensweise ist notwendig um die weiteren Testfunktionen zu realisieren. Auch können durch diese Entkopplung das Slave und das Master Interface mit unterschiedlichen Taktraten betrieben werden. Die maximale Taktfrequenz liegt für beide Schnittstellen bei 50 MHz.

Bild 8 zeigt die Struktur des transparenten Modes.

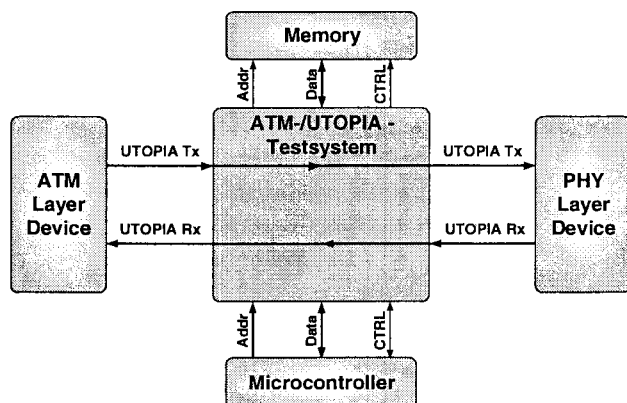


Bild 8 Transparenter Mode

2.2.2 Multiplex Mode

Diese Funktion tritt als Kombination zum transparenten Mode auf, und ermöglicht die Datenübertragung zwischen ATM Layer und PHY Layer Devices mit unterschiedlichen Datenbusbreiten. Es werden dabei die Datensignale der beiden UTOPIA Schnittstellen von 16 auf 8 und umgekehrt umgesetzt. Diese Konfiguration ist nicht während der Laufzeit durch den Microcontroller auswählbar sondern ergibt sich aus den generischen Vorgaben für die Datenbusbreite und wird automatisch während der Synthese im Testsystem eingebunden.

Die Umsetzung läuft nach folgendem Schema ab. Die Daten werden zunächst wie beim transparenten Mode vom System empfangen und in einem FIFO

zwischen gespeichert. Bei der Weitergabe der Zellen von unterschiedlichen Datenbusbreiten muss die Zelle allerdings durch das Testsystem verändert werden, da die Zellen unterschiedlich groß sind. Der Unterschied zwischen einer Zelle eines 16 Bit Datenbusses und der eines 8 Bit Datenbusses wird aus Bild 3 ersichtlich. Der Zellheader einer 16 Bit Zelle besitzt zwei benutzerdefinierbare (UDF) Bytes, in einer 8 Bit Zelle ist lediglich ein solches Feld vorhanden. Bei einem Austausch solcher Zellen wird nun das UDF 2 Feld aus dem Header der 16 Bit Zelle entfernt und bei einer 8 Bit Zelle ein solches Feld eingefügt.

Diese Veränderungen an den ATM-Zellen werden über folgenden Mechanismus realisiert. 16 Bit ATM-Zellen werden zunächst vollständig im System gespeichert, bei einer Weitergabe in Form einer 8 Bit Zelle wird die Speicherstelle die das nun überflüssige UDF 2 Byte enthält beim Auslesen übersprungen. Und somit das UDF 2 aus den Zellheader entfernt. Beim Einlesen einer 8 Bit Zelle wird die Speicherstelle für das UDF 2 Byte übersprungen und somit freigelassen. Bei der Ausgabe der 16 Bit Zelle erfolgt kein Adresssprung und es werden 54 Bytes ausgegeben was der Größe einer 16 Bit ATM-Zelle entspricht. Allerdings kann durch das Testsystem kein gültiges Datum für das UDF 2 eingeschoben werden, da die Konfiguration für dieses Byte dem System nicht bekannt ist. Aber dieses Byte ist bei einer Umsetzung von 8 auf 16 bzw. 16 auf 8 Bit ohnehin nicht zu verwenden da es bei 8 Bit Zellen nicht vorhanden ist.

Bild 9 zeigt ein Beispiel für die Umsetzung eines 16 Bit breiten Datenbus des Slave Interfaces auf den 8 Bit breiten Bus des Master Interfaces.

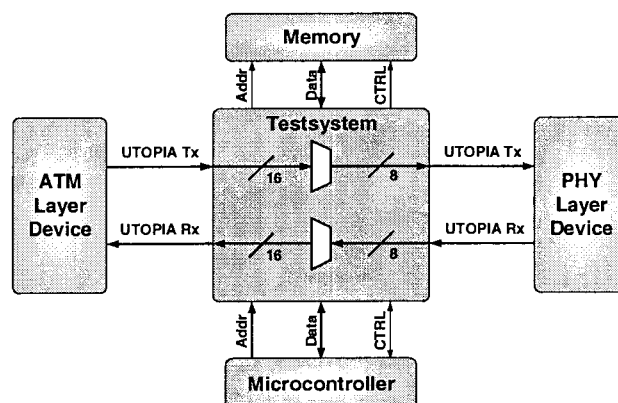


Bild 9 Multiplex Mode

2.2.3 Loop Back

Dieser Modus ist für einfache Tests an den UTOPIA-Interfaces der angeschlossenen ATM bzw. PHY Layer Devices vorgesehen. Dabei werden die Devices ohne die entsprechende Gegenstelle betrieben. Im Testsystem werden die Zellen ohne weitere Änderung direkt

zurückgeschickt. Und können beispielsweise miteinander verglichen werden. Wie in den zuvor beschriebenen Testfunktionen werden die Zellen vom Testsystem empfangen. Aber dann nicht weiter geschickt sondern an den Empfänger mit der gleichen UTOPIA-Adresse der selben Schnittstelle zurück gesandt.

Die Schleifen können während des laufenden Betriebs durch den Microcontroller für eine Schnittstelle geschlossen werden, die andere Schnittstelle wird dabei deaktiviert.

Bild 10 zeigt einer Schleife an der Slave Schnittstelle. In Bild 11 werden ATM-Zellen des PHY Layer Devices zurückgeschickt.

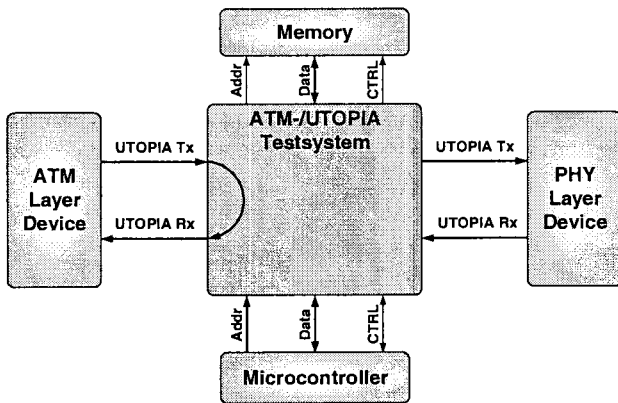


Bild 10 Schleife ATM Seite

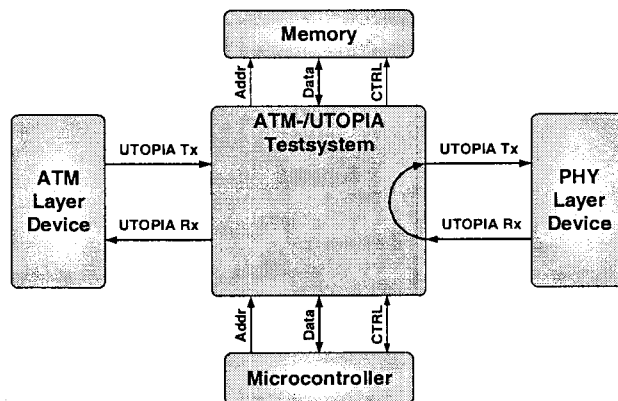


Bild 11 Schleife PHY Seite

2.2.4 Microcontroller Zugriff auf UTOPIA-Interface

Mit dieser Konfiguration wird ein Zugriff des Microcontrollers auf die UTOPIA-Schnittstelle ermöglicht. Es können somit ATM-Zellen vom Microcontroller über das Testsystem an ein ATM Layer oder PHY Layer Device, wie in Bild 12 und Bild 13 zu sehen, gesandt und von dort empfangen werden. Im Bild 12 kommuniziert der Microcontroller mit dem ATM Layer Device im Bild 13 mit einem PHY Layer Device.

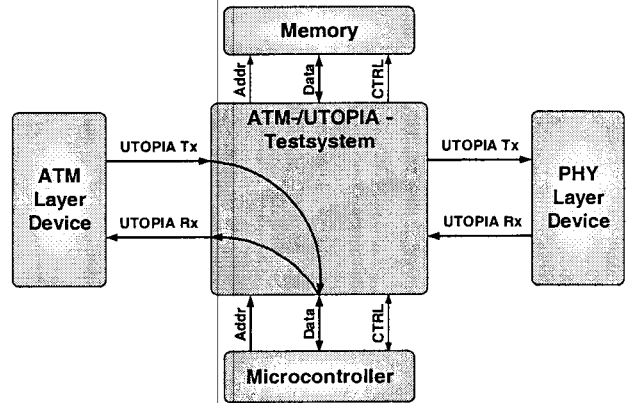


Bild 12 Microcontroller Zugriff auf UTOPIA Slave

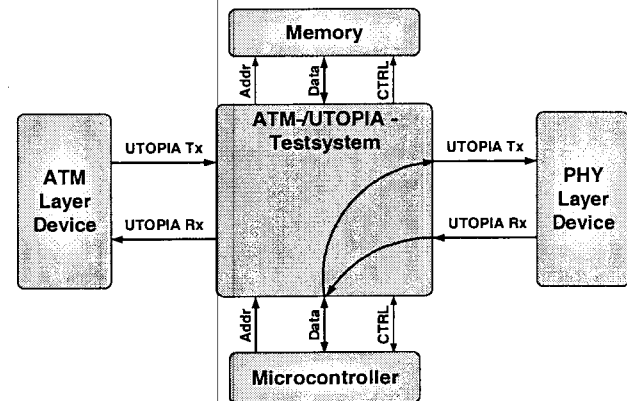


Bild 13 Microcontroller Zugriff auf UTOPIA Master

Für diese Testkonfiguration besitzt die Microcontroller Schnittstelle zwei FIFOs je einen für den Empfang und den Versand von ATM-Zellen (siehe Bild 14). Diese beiden FIFOs werden wie die Register für den Status und die Konfiguration des Testsystems angesprochen, wobei beim Sende FIFO nur ein Schreibzugriff und beim Empfangs FIFO nur ein Lesezugriff erlaubt ist. Zu übertragende Zellen werden byteweise auf die Adresse des Sende FIFOs geschrieben. Über die Interruptleitung wird die Übertragung der Daten gestoppt, wenn die Zelle vollständig in das FIFO geschrieben wurde. Nachdem die Zelle an das entsprechende Device gesandt wurde wird über einen weiteren Interrupt eine neue Zelle angefordert.

Von der UTOPIA Schnittstelle empfangene Zellen werden im Empfangs FIFO abgelegt. Wenn die gesamte Zelle im FIFO steht wird der Microcontroller über ein Interrupt aufgefordert die Zelle auszulesen. Hat der Microcontroller die Zelle vollständig ausgelesen wird diese Übertragung wiederum durch einen Interrupt gestoppt. Die UTOPIA Adressen der PHYs mit denen der Microcontroller kommuniziert werden in Registern hinterlegt und können während der Testphase geändert werden. Wird dadurch eine Zelle im Empfangs oder Sende FIFO ungültig werden die Daten

verworfen und neu angefordert.

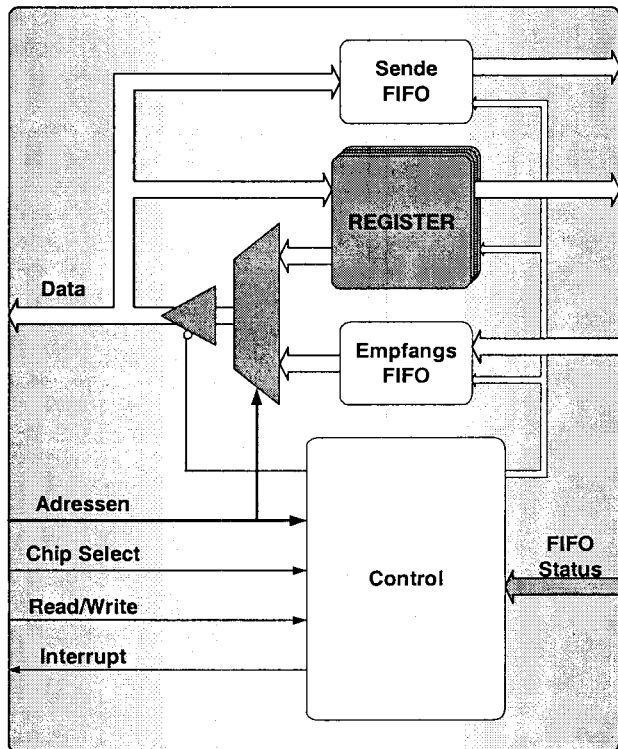


Bild 14 Microcontroller Schnittstelle

2.2.5 Speicherzugriff

Über den angeschlossenen Speicher ist das Testsystem in der Lage dort durch den Microcontroller abgelegte ATM-Zellen zu senden (siehe Bild 15 und Bild 16). Das Testsystem liest dazu aus dem Speicher bestimmte ATM-Zellen aus dessen Speicherbereich durch den Microcontroller vorgeben ist und sendet diese Zellen an die vom Microcontroller ausgewählten Devices. Im Gegenzug werden von dort empfangene Zellen in einem, über den Microcontroller ausgewählten, Speicherbereich, für eine weitere Verwendung, abgespeichert.

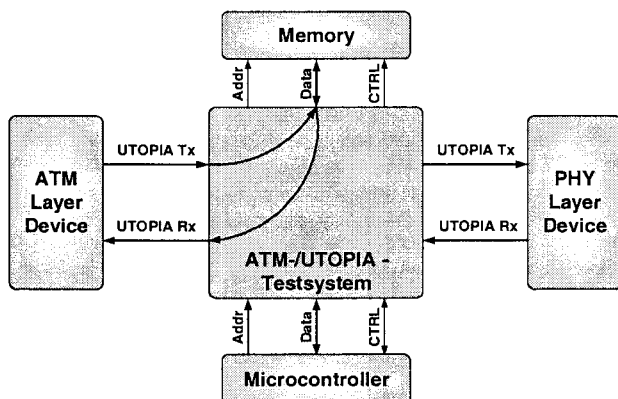


Bild 15 Speicherzugriff für ATM Device

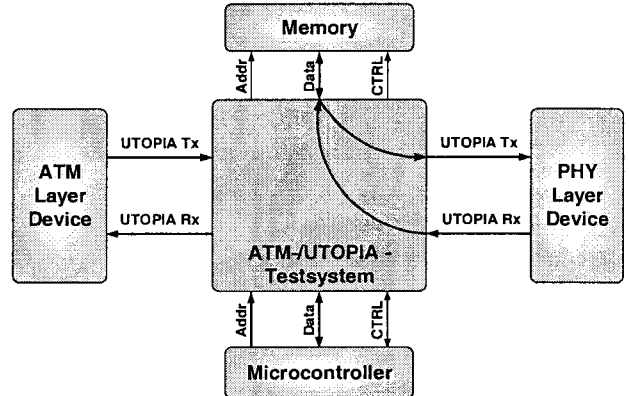


Bild 16 Speicherzugriff für PHY Device

2.2.6 HEC Byte Erzeugung

Für spätere Erweiterungen des Testsystems bei dem über eine serielle Schnittstelle Daten an eine der beiden UTOPIA Schnittstellen gesandt werden ist es notwendig aus dem seriellen Datenstrom den Beginn einer Zelle zu erfassen. Dafür lässt sich das in Abschnitt 1.1 behandelte HEC-Byte (Header Error Control) verwenden. Zunächst muss aus dem Datenstrom mit jedem Takt ein neues HEC-Byte wie in [6] spezifiziert, erzeugt werden. Dazu werden die ersten vier Bytes des Zellkopfes durch das Generator Polynom x^8+x^2+x+1 geteilt. Der verbleibende Rest stellt das HEC-Byte dar. Über einen sogenannten Hunting Mechanismus kann nun der Beginn einer Zelle aus einem Datenstrom gefiltert werden. Dieser Mechanismus ist in Bild 18 abgebildet und läuft nach folgendem Schema ab. Zunächst wird der Datenstrom durch ein Schieberegister (Bild 17), das das Generator Polynom nachbildet, und parallel durch ein neutrales Schieberegister geschoben. Mit jedem Takt werden nun die 8 Bit die sich in den beiden Schieberegistern befinden miteinander verglichen. Stimmen die beiden Registerinhalte überein wird davon ausgegangen, dass ein Zellenanfang gefunden wurde und es erfolgt der Übergang in den Presync-Status wird nun in Folge sieben mal immer nach 53 bzw. 54 Bytes ein Header gefunden geht man davon aus, dass der Zellenbeginn richtig erkannt wurde und befindet sich im Sync-Status. Wird nach Ablauf einer Zelle kein Header erkannt wird das Verfahren neu eingeleitet. Dieses Verfahren ist deshalb notwendig da auch im Nutzdatenteil einer ATM-Zelle eine Sequenz von Daten erscheinen kann, die auf einen Header schließen lässt. Durch die sieben in Folge gefundenen Header ist die statistische Wahrscheinlichkeit sehr groß, dass richtige Header erkannt wurden.

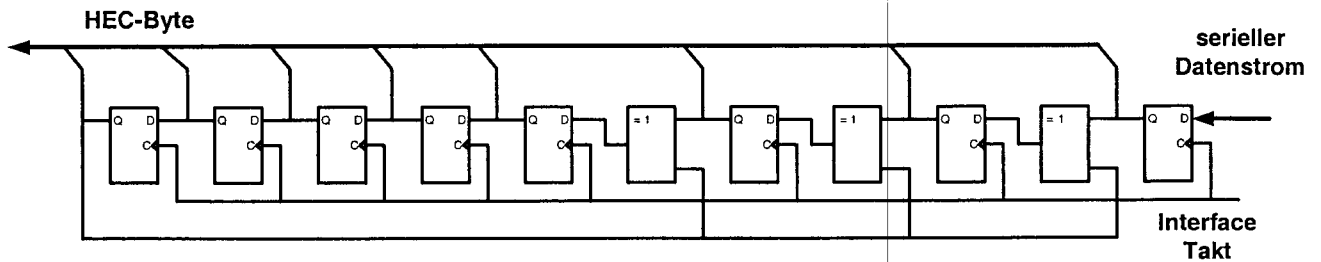


Bild 17 HEC Schieberegister

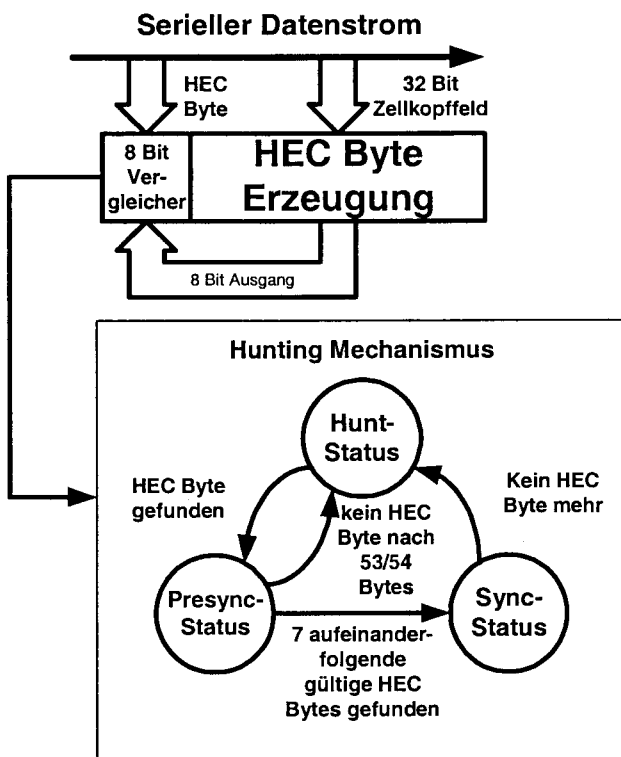


Bild 18 Hunting Mechanismus

2.2.7 Zellen Zähler

Dieser konfigurierbare Zähler zählt Zellen die einem bestimmten Kriterium entsprechen. Mögliche Kriterien sind:

- Alle Zellen mit einer bestimmten UTOPIA-Adresse.
- Zellen die einer bestimmten Richtung entsprechen.
- Alle Zellen die durch das Testsystem geleitet werden.

Der Zählerstand wird in einem Register abgelegt und kann durch den Microcontroller ausgelesen werden. Dieser Modus ist unabhängig von den übrigen Testfunktionen und wird ebenfalls über den Microcontroller konfiguriert.

3 Zusammenfassung

Um das Testsystem in verschiedenen Hardwareumgebungen einsetzen zu können wurde es so implementiert, dass es den unterschiedlichen Bedingungen auf den verschiedenen Platinen genau angepaßt werden kann. Über die generischen Variablen für die maximale Anzahl der anzuschließenden PHYs und der Datenbusbreite läßt sich starken Einfluß auf die benötigte Bausteingröße nehmen. Dadurch läßt sich das Testsystem, mit angepaßter Funktionalität, auch in kleinen FPGAs realisieren. Die verwendeten IP-Cores, ein Dual-Ported-Ram und ein FIFO stellen Standardelemente dar wie sie von den meisten FPGA Herstellern angeboten werden. Die Ansteuerung dieser eingebundenen Elemente wurde so vorgenommen, dass die verschiedenen Cores ohne oder mit nur wenig Änderungen ausgetauscht werden können. Die Komponenten des Systems insbesondere die Schnittstellen sind nicht speziell für das Testsystem entwickelt worden, sondern können in verschiedenen Designs mit entsprechend geforderter Funktionalität eingesetzt werden.

Mit dem Testsystem ist eine Entkopplung zwischen den ATM Layer Devices und den PHY Layer Devices möglich somit können Geräte unterschiedlicher Datenbusbreiten und Taktfrequenzen verbunden werden.

Durch die Konfigurierbarkeit während des Betriebs können verschiedene Test an einem Gesamtsystem vorgenommen werden ohne die Umgebungsbedingungen zu verändern. So werden die Ansprüche einer Testumgebung durch das Testsystem gering gehalten. Tests sind somit ohne der Möglichkeit einer FPGA-Programmierung mit einfachen Testmitteln möglich.

Quellen und weiterführende Literatur

- [1] Siegmund, Gerd: ATM – Die Technik. 3. Auflage. Hüthig Verlag Heidelberg
- [2] The ATM Forum: UTOPIA Specification Level 1. Version 2.01
- [3] The ATM Forum: UTOPIA Specification Level 2. Version 1.0
- [4] <http://www.atmforum.com/>
- [5] ISO–International Organization for Standardization
<http://www.iso.ch/>
- [6] ITU–International Telecommunication Union
Telecommunication Standardization:
Recommendation I.432 (02/99)
<http://www.itu.int/>

CIS – Chip im Schmuck

A. Ratz, D. Ernst, Prof. F. Kesel
 Fachhochschule Pforzheim, Tiefenbronnerstraße 65, 75175 Pforzheim

1. Einleitung

Im Rahmen des CIS Projekts wurde an der FH Pforzheim erstmals ein eigener ASIC entwickelt. Ziel war es, in Zusammenarbeit mit dem Schmucktechnologischen Institut (STI) und der Hochschule für Gestaltung (FHG) die Integration von Elektronik in Schmuckobjekte zu erproben. Pforzheim gilt als Uhren- und Schmuckstadt, deshalb liegt es nahe, sich gerade hier mit innovativem Schmuck zu beschäftigen. Als Anwendungsbeispiele lassen sich z.B. Zugangskontrollsysteme oder Identifikationssysteme nennen, wofür auch seitens der Industrie Interesse besteht.

2. Funktion des ASIC

Mit der Schaltung werden mehrfarbige LEDs pseudozufällig angesteuert. Die Ansteuerung unterschiedlicher Farben mit pulsweitenmodulierten Signalen beeinflusst die Leuchtdichte und ergibt somit einen charakteristischen Farbverlauf. Zur Erzeugung der Pseudozufallsfolgen dient ein linear rückgekoppeltes Schieberegister wie in Abbildung 1 dargestellt. Bei Verwendung von Minimalpolynomen als Rückkopplungspolynome entstehen Folgen der Länge $2^m - 1$.

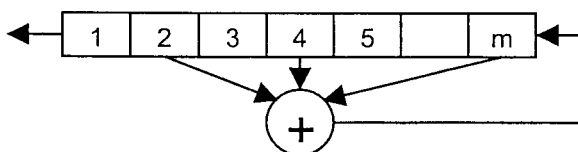


Abbildung 1: linear rückgekoppeltes Schieberegister

3. Aufbau des ASIC

Beim gewählten Fertigungsprozess handelt es sich um einen Alcatel Mietec 0.5µm Prozess mit zusätzlichen Analogoptionen für Kapazitäten und Widerstände. Kapazitäten werden zwischen zwei PolySi Schichten gebildet, für die Widerstände wird besonders hochohmiges PolySi verwendet. Abbildung 2 zeigt den ASIC einschließlich Pads. Die Abmessungen betragen ca. 1,8x1,9 mm.

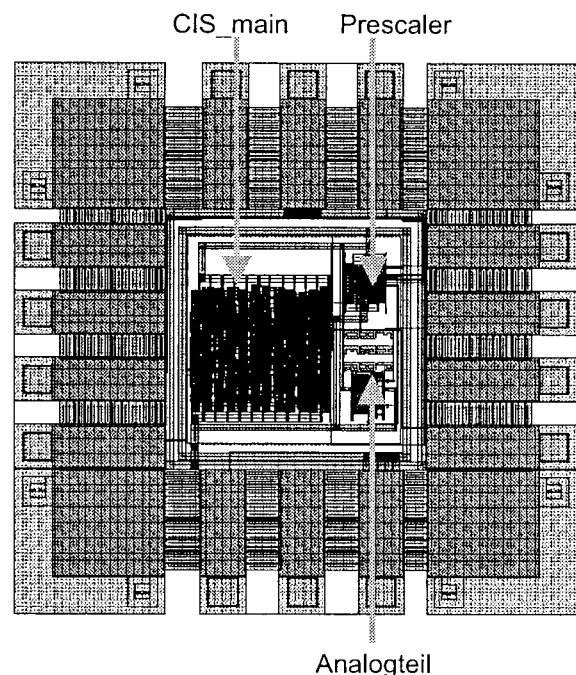


Abbildung 2: Aufbau des ASIC

4. Analogteil

Der Analogteil wurde mit gängigen IC Design Tools von Mentor Graphics entwickelt. Zur Eingabe des Schaltplans diente "design architect", die anschließende Simulation erfolgte mit "accusim". Das verwendete Layout – Tool ist "ic station". Bestandteile des Analogteils sind ein Oszillator (Abbildung 3) und eine Power – On Reset Schaltung (Abbildung 4).

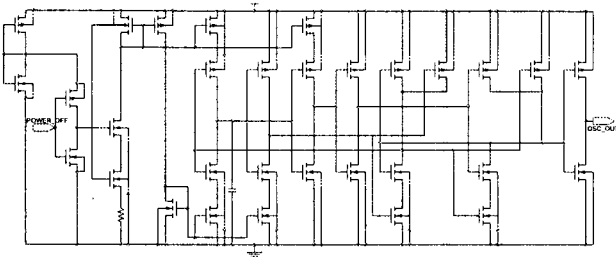


Abbildung 3: Schaltplan des Oszillators

Grundlage für den implementierten Oszillator ist ein PLL – Design der FH Offenburg [1], die Schaltung wurde dort als VCO eingesetzt. Das frequenzbestimmende Bauelement ist der Kondensator. Dieser wird über eine schaltbare Stromquelle abwechselnd ge- und wieder entladen. Über einen CMOS Vorspannungsteiler wird die Schaltung auf eine Frequenz von ca. 1,8 MHz eingestellt.

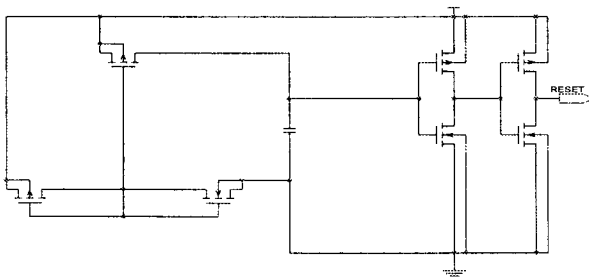


Abbildung 4: Schaltplan des Power - On Reset

Beim Power – On Reset handelt es sich um ein klassisches RC – Glied, wobei der Widerstand durch MOS Transistoren gebildet wird. Während der Aufladung wird ein Reset der digitalen Logik durchgeführt.

Abbildung 5 zeigt das Layout des Analogteils. Besonders auffällig ist die mäanderförmige Struktur des Widerstands links unten. Rechts daneben sind die beiden Kondensatoren, je einer für den Oszillator und die Reset – Schaltung.

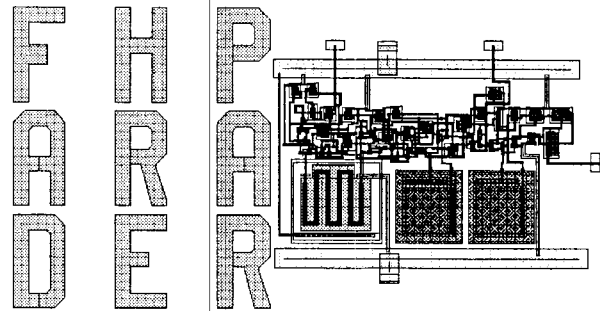


Abbildung 5: Layout des Analogteils

5. Digitalteil

Der Entwurf des Digitalteils erfolgte in VHDL, unter Verwendung der Mentor – Tools "Renoir" für die Codeeingabe und "ModelSim" zur Simulation. Die Synthese erfolgte mit Synopsys. Der Prescaler dient dazu, den Takt vom Oszillator in einem bestimmten Verhältnis zu teilen. Im Block "CIS_main" ist das linear rückgekoppelte Schieberegister implementiert. Außerdem befindet sich dort die Logik zur Erzeugung der PWM Signale und zur Ansteuerung der LEDs. Am Ende des Entwurfsablaufs wurde eine Simulation des gesamten ASIC, einschließlich Pads, durchgeführt. Hierzu war ebenfalls ModelSim im Einsatz. Für die Analogkomponenten entstanden dabei VHDL – Simulationsmodelle.

6. Inbetriebnahme

Alle Funktionen des IC konnten fehlerfrei in Betrieb genommen werden. Zwei Diagramme sollen dies verdeutlichen.

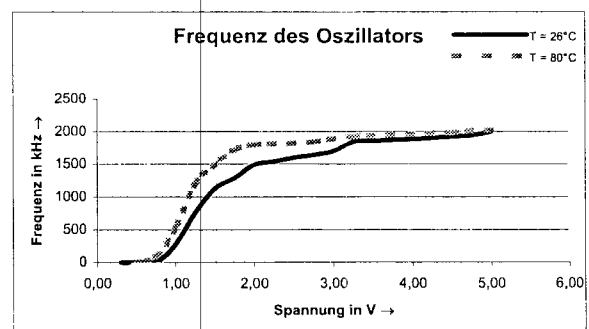


Abbildung 6: Frequenz des Oszillators

Abbildung 6 zeigt die Frequenz des Oszillators in Abhängigkeit von der angelegten Spannung und der Umgebungstemperatur. Man erkennt die Spannungscharakteristik der Schaltung. Besonders

erfreulich ist die Temperaturstabilität. Die durchgeführten Messungen ergaben ein besseres Temperaturverhalten als die Simulation erwarten ließ.

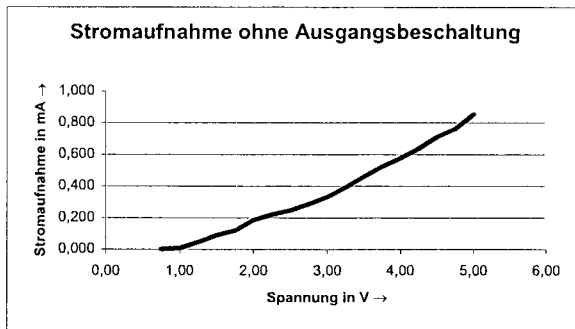


Abbildung 7: Stromaufnahme des ASIC

Abbildung 7 zeigt die Stromaufnahme des IC. Der Verlauf ist im wesentlichen linear, wie zu vermuten war. Im Hinblick auf ein Low – Power Design sind an der Schaltung noch Veränderungen denkbar. Damit könnte die Stromaufnahme der Schaltung weiter optimiert werden.

7. Zusammenfassung

Im Rahmen einer Studienarbeit wurde der beschriebene ASIC an der FH Pforzheim entwickelt. Dieser ASIC ist eine einfache Applikation, um mit den Tools und dem Entwurfsablauf vertraut zu werden. Für die Zukunft sind umfangreichere Projekte, auch in Zusammenarbeit mit Partnern aus der Industrie, geplant. Gemeinsam mit der Hochschule für Gestaltung und dem Schmucktechnologischen Institut sollen Schmuckobjekte erstellt werden, die praktisch einsetzbare Schaltungen beinhalten. Die ersten Schritte scheinen erfolgversprechend, denn der erste selbstgefertigte ASIC der FH funktioniert wie erwartet.

8. Literatur

- [1] Entwicklung eines PLL zur Generierung eines 11 MHz – Taktes aus einem 32 kHz – Takt. FH – Offenburg, ASIC – Design – Center, Oktober 1999
- [2] Chip im Schmuck – ein CMOS Demonstrator für die Integration in Schmuckobjekte. FH – Pforzheim, IC Design Labor, März 2002

Induktive bidirektionale Schnittstelle ähnlich ISO/IEC 14443-A

Frank Baier,
Dirk Jansen
Fachhochschule Offenburg, ASIC-Design-Center
Badstr. 24, 77652 Offenburg
☎ :0781/205-179, Fax: 0781/205-174

Im Folgenden wird die Entwicklung eines induktiven Moduls in VHDL zur bidirektionalen Datenübertragung nach dem Gütemodulationsverfahren für kontaktlose Chipkarten ähnlich ISO/IEC 14443-A beschrieben.

Ein System besteht aus zwei Komponenten, einem Erfassungs- oder Lesegerät und einer kontaktlosen Chipkarte (Transponder). Der Transponder ist vollständig von dem Lesegerät abhängig, sowohl der Systemtakt des Transponders als auch die Energie werden durch das magnetische Feld des Lesegerätes übertragen. Der Datenaustausch erfolgt im Halbduplexverfahren in beide Richtungen.

Das Lesegerät sendet die mit einem modifizierten Millercode codierten Daten durch eine Amplitudenmodulation des Trägers von 13.56 MHz.

Der Transponder codiert die Daten nach dem Manchestercode, diese werden dann mit Hilfe eines Lastmodulationsverfahrens mit Hilfst Träger gesendet wobei, der Träger 874 KHz beträgt.

1. Einleitung

Die ISO 14443-A ist eine Norm aus dem Bereich der *Radio Frequenz Identifikation RFID*. Sie spezifiziert die kontaktlose Kommunikation zwischen Lesegeräten und Chipkarten. Dieser Chipkartentyp erfreut sich wachsender Akzeptanz. Die Vorteile dieses kontaktlosen Systems liegen klar auf der Hand, so haben Umwelteinflüsse wie Schmutz und Nässe keinen Einfluss auf die Funktion. Auch ist die Handhabung dieser Karten sehr komfortabel, beispielsweise beim Einsatz in

bargeldlosen Kassensystemen verbleibt die Karte in der Brieftasche, diese wird dann einfach auf das Lesegerät gelegt. Eine Kommunikation wird aufgebaut, sobald eine Karte in den Einflussbereich eines Lesegeräts gelangt. Die maximale Entfernung zwischen Lesegerät und Transponder dieses Systems beträgt ca. 10 cm.

Erste Entwicklungen zu diesen Lesegeräten lagen bereits vor [6]. Es war nun die Aufgabe, ein komplettes Kommunikationsmodul mit Anbindung an den an der Fachhochschule Offenburg entwickelten Prozessorkern FHOP zu schaffen.

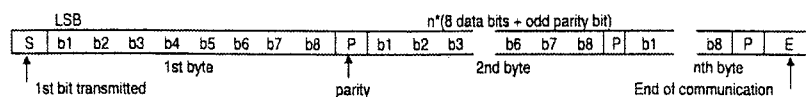
2. Kommunikation

2.1. Datenformat

Der gesamte Datenaustausch sowie die Antwort des Transponders erfolgt in einem bitorientierten *Frame* (Abbildung 2-1). Dieser beginnt mit einem *Startbit* (*S*). Es folgen die Datenbytes, die voneinander durch *Paritybits* (*P*) getrennt sind (ungerade).

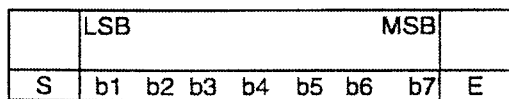
Das *Last Significant Bit LSB* folgt dabei zuerst. Der *Standardframe* endet mit einem *End of Frame* (*E*).

Um eine Verwechslung mit dem Standardframe auszuschließen, besteht das *Request* Kommando, welches die Kommunikation einleitet, aus lediglich 7 Bits daher der Name *Shortframe*. Diese Folge beginnt ebenfalls immer mit einem *Startbit* (*0*) und endet mit einem *End of Frame* (*0*). Sie wird von dem Lesegerät gesendet und startet die



Kommunikation mit dem Transponder.

Das Kommando hat grundsätzlich die folgende Sequenz: S 0110010 E. Abb.: 2-2 veranschaulicht den Aufbau eines *Shortframes*. Auch innerhalb dieses Rahmenformates wird zuerst das *Last Significant Bit LSB* gesendet.



↑
First bit transmitted

Abb. 2-2 Shortframe [3]

Werden Daten vom Lesegerät zum Transponder übertragen (*downlink*), so besteht das Startbit und der *End of Frame* aus einer 0.

Folgt die Datenübertragung in Richtung des Lesegerätes (*uplink*), so besteht der *Start of Frame* aus einer 1 und der *End of Frame* aus einer Modulationspause.

2.2. Initialisierung

Wird ein Transponder in den Einflussbereich des Lesegerätes gebracht, initialisiert sich dieser und begibt sich in den *IDLE*-Mode. Dieser Wartezustand ist notwendig, um sicherzustellen, dass der Transponder keine bereits laufende Kommunikation stört.

Eine Kommunikation wird immer von dem Lesegerät initiiert (*Master-Slave-Prinzip*). Befindet sich das Lesegerät nicht in einer Kommunikation mit einem Transponder, sendet dieses periodisch ($T=100\text{ms}$) ein sogenanntes *Request*-Signal aus.

Wird während des *IDLE* State ein gültiges *Request*-Kommando empfangen, so sendet der Transponder eine Standard Antwort (*ATQA*) an den Kartenleser zurück. Nach Beantwortung des *Request*-Kommandos befindet sich der Transponder im Bereit-Status.

Das Lesegerät hat nun erkannt, dass sich mindestens ein Transponder im Ansprechbereich befindet. Daraufhin startet das Lesegerät den Antikollisionsalgorithmus durch Senden des *Select* Befehls. Dies bewirkt, dass alle Transponder im *Ready*-Status Ihre Kennung senden. Die aufgrund des *Binary-Search-Algorithmus* selektierte Seriennummer des gewählten Transponders wird dann zurückgesandt. Erkennt der Transponder seine Nummer, antwortet dieser mit einer Bestätigung, setzt den Status *AUSGEWÄHLT* und ist nun für einen Datenaustausch mit dem

Kartenlesegerät bereit. Somit wird verhindert, dass es zu einer Datenkollision der beiden Transponder kommt. Prinzipiell nutzt man dabei die Eigenschaft der Manchester Codierung aus, die es ermöglicht, eine Kollision festzustellen.

2.3. Übertragungsprotokoll

Alle Informationen werden in einem Blockformat übertragen. Der I-Block ist der wichtigste Baustein. Er überträgt die relevanten Informationen im *Information Field INF*.

Abbildung 2-3 zeigt den Aufbau eines I-Blockes

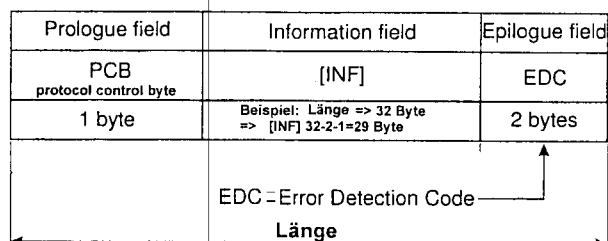


Abb. 2-3: I-Block [3]

Im *Prologue field* findet man das *Protocol Control Byte PCB*. Dies definiert den Typ des Blockes. Die Norm 14443 spezifiziert hierin drei unterschiedliche Blöcke:

- I-Block
- R-Block
- S-Block

Mit Hilfe des R-Blockes lässt sich eine Verkettung von mehreren Blöcken spezifizieren. Dies wird als *Chaining* bezeichnet. Das Prinzip beruht dabei auf einer Bestätigung jedes Blockes.

Durch den S-Block können kartenspezifische Parameter übertragen werden wie der *Card Identifier CID*. Dieser optionale Parameter wird im *Prologue Field* des Blockes spezifiziert.

Ein ebenfalls optionaler Parameter ist das *Node Address Field NAD*. Genauere Angaben hierzu findet man in ISO/IEC 7816-3.

Das Übertragungsprotokoll wurde auf die wichtigen Eigenschaften der bidirektionalen Datenübertragung reduziert. Die R- und S-Blöcke wurden nicht berücksichtigt, ebenso die optionalen Parameter *CID* und *NAD*.

Im Bit 1 des *PCB* wird die Blocknummer eines I-Blockes spezifiziert. Diese kann 0 oder 1 sein.

Jede Kommunikation wird vom Lesegerät initiiert. Dies bedeutet, dass das Lesegerät immer den ersten Block sendet.

Beispielszenario einer fehlerfreien Datenübertragung:

Im Folgenden entspricht ein I(x) für einen Informationsblock mit der Blocknummer x.

Lesegerät		Transponder
I(0)	=>	0
	<=	I(0)
I(1)	=>	I(1)
	<=	I(1)

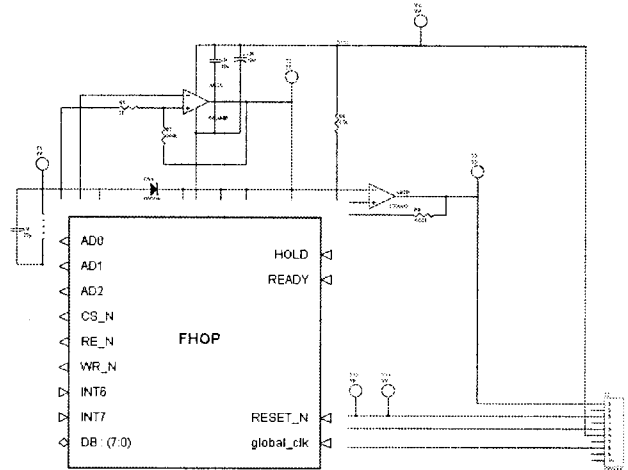
Das Lesegerät sendet einen I-Block mit der Blocknummer 0 an den Transponder.

Daraufhin sendet der Transponder einen I-Block mit der Blocknummer 0 an das Lesegerät. Infolgedessen wird das Lesegerät die Blocknummer des I-Blocks auf 1 erhöhen und den Block mit neuen Informationen an die Karte senden. Die Karte sendet den Block mit der Blocknummer 1 zurück an das Lesegerät, welches wiederum die Blocknummer auf 0 setzt und eventuell einen neuen I-Block sendet.

3. Realisierung

3.1. Der Prozessorkern FHOP

Um die Daten normgetreu weiterverarbeiten zu können, werden diese zunächst in einem RAM gespeichert. Der Protokollablauf wird in einem ROM



festgelegt. Hierzu wurde der FHOP Design Kit (MPC-Workshop, Januar 2002) modifiziert. Es wurden Adressleitungen zur Registeradressierung sowie Send- und Empfangsinterrupts geschaffen. RAM und ROM sind in der Größe parametrisierbar und somit an die Zellenzahl des FPGA gebunden.

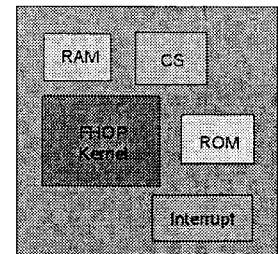


Abb. 3-1: Prozessorkern FHOP

3.2. Das Modul RF-Modem

Das *Radio Frequency RF-Modem* besteht aus einem analogen Frontend, das die analogen Signale zur digitalen Weiterverarbeitung analog aufbereitet. Abbildung 3-2 zeigt den Schaltplan des Frontends.

Abb. 3-2: Frontend

Weiterhin besteht es aus einem digitalen Logikblock, der vollständig in VHDL entworfen wurde [6]. Abbildung 3-3 zeigt die Entity des Moduls RF-Modem.

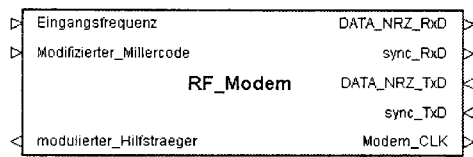


Abb. 3-3: Modul RF-Modem

Der rechte Teil der Entity des RF-Modems wird an die RF-Modem Schnittstelle angekoppelt. Diese Signale umfassen die Sende- und Empfangsdaten sowie die dazugehörigen Synchronisierungen. Außerdem wird der Schnittstelle ein Modemtakt von 105.9 kHz zugeführt. Dieser Takt wird aus dem modifizierten Millercode und der Signalträgerfrequenz von 13.56 MHz gewonnen.

3.3. Die Schnittstelle

Die vom Lesegerät gesendeten seriellen Daten, werden in Bytes gewandelt und in einem FIFO, der über eine Speichertiefe von 16 Bytes verfügt, abgelegt. Sind Daten im FIFO vorhanden, wird ein Interrupt ausgelöst. Hierzu liest der FHOP den FIFO in der Schnittstelle aus. Analoge Vorgehensweise gilt beim Senden der Bytes. Das *Chipselectsignal CS* dient der Aktivierung des Moduls. In Registern werden verschiedene protokollspezifische Parameter wie beispielsweise die Seriennummer der Karte abgelegt.

Abbildung 3-4 zeigt die Entity der Schnittstelle

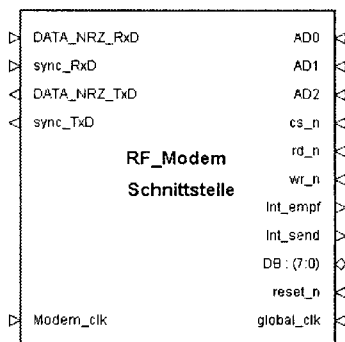


Abb. 3-4: Modul RF_Modem Schnittstelle

Die Hardware wurde erfolgreich simuliert. Synthetisiert wurden bereits der FIFO sowie der Block FHOP.

Die Norm ISO/IEC 14443-A spezifiziert leider nicht explizit jedes Bit. Daher konnten Hersteller wie Philips MIFARE zum Schutz ihrer Produkte verschiedene proprietäre Standards festlegen, die

nicht veröffentlicht werden. So nutzt beispielsweise MIFARE kryptologische Verfahren auf ihren Transpondern, um die gesendeten Informationen vor jeglichem Zugriff zu schützen.

Aufgrund dieser Tatsache können mit der Standardsoftware, die von Mifare bereits lange bevor die Norm ISO/IEC 14443-A veröffentlicht wurde, keine I-Blöcke gesendet werden, wie sie in der Norm spezifiziert sind. Daher ist zumindest eine Anpassung der Software unumgänglich für eine erfolgreiche Nutzung der Schnittstelle *Radio Frequency Identification RFID*.

4. Zusammenfassung

Es wurde ein Modul zur Kommunikation mit dem FHOP auf Basis der Norm ISO/IEC 14443-A entwickelt und synthetisiert. Zur Aufgabe steht noch die Implementierung auf Softwareebene.

5. Literatur:

[1] Titze Schenk:

Halbleiterschaltungstechnik, Springer Verlag, 11. Auflage 2000

[2] Dirk Jansen:

Handbuch der Electronic Design Automation, Hanser Verlag, 1. Auflage 2001

[3] Norm: ISO/IEC 14443-A

ISO, Ausgabe 2000, www.iso.ch

[4] Klaus Finkenzeller:

RFID – Handbuch, Carl Hanser Verlag, 2. Auflage 2000

[5] Mifare: Produkt Specification Rev. 1.3,

<http://www.semiconductors.philips.com/markets/identification/customer/download/>

[6] Josef Zimmermann:

Entwicklung eines 13.56 MHz Transceivers,
Diplomarbeit 2000 ASIC DESIGN Center
Fachhochschule Offenburg, Bibliothek FHO

[7] Markus Striebel:

FHOP - Design Kit Volume 6, ASIC DESIGN Center
2002, Fachhochschule Offenburg,
<http://www.asic.fh-offenburg.de/FhopKit/index.html>

DQPSK Modulator for Inductive Data Transmission

Nidal Fawaz,
Dirk Jansen

Fachhochschule Offenburg, ASIC-Design-Center
Badstr. 24, 77652 Offenburg
Tel :0781/205-179, Fax: 0781/205-174

1. Introduction

The aim of this article is to give a general description of a DQPSK modulator built in ASIC Labor FH-Offenburg.

The idea came to develop a system from an existing one that has a limited specification and is running in different applications:

- Medical field as “ECG-Recorder” [1]
- Automotive field as “Chip in Tire”[2]
- Different fields as “Thermollogger” [3] and ...etc.

The modulator should be used as an interface module to the “FHOP Microprocessor” [4].

The neediness of higher performance led to develop a new modulator with the following specification with comparison to the running one:

- Changing from BPSK to QPSK scheme.
- Increment of the baud rate from 2400 bit/s up to 9600 bit/s or more.
- Gaussian filtering for phase transition.
- PWM technique for the modulated output signal.

The developed system has the same specification as the running one:

- Primary door frequency 11.0592 MHz.
- Carrier frequency 115.2 kHz.
- “RS232” Standard form.

The innovative step in this modulator is the combination of the “Gaussian Filter” for phase transition and the “PWM Generator” for modulation leading to a compact system that has a small chip area, low power consumption and higher performance.

2. System Design

The modulator is made up of different stages; it takes the data from the *bus*, processes it and generates at the last stage the modulated sinus signal.

The different stages are as follows:

1. SIO, Serial Input Output Interface between the *bus* and the *modulator*.
2. Serial to 2 bit parallel converter, generating 1 symbol for every 2 serial bits.
3. Differential QPSK state selector, determining the angle phase shift that must be performed $\{+45^\circ, +135^\circ, -135^\circ \text{ or } -45^\circ\}$.
4. Gaussian Filtering, producing coefficients for continuous Gaussian phase shifting.
5. NCO, Numerical Controlled Oscillator, generating sawtooth shape that determines the frequency of the modulated signal.
6. PWM generator, generating serial output bit stream mapped to the modulated signal.
7. Transponder, Bandpass filter with resonance frequency 115.2 kHz.

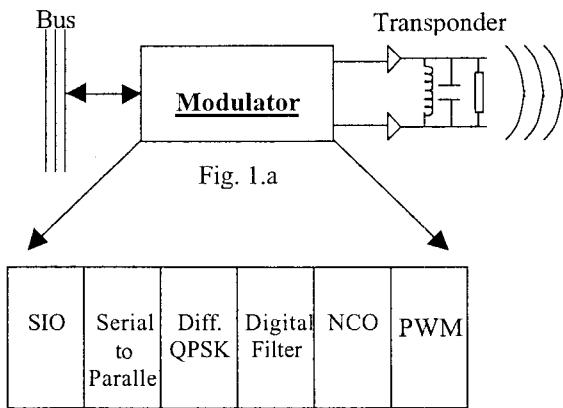


Fig. 1.b

Fig (1.a) shows the modulator interface and fig (1.b) shows the different stages that the process undergoes.

3. Description of the Stages

A comprehensive description about different stages will be described, each stage performs a special function and generates the necessary signals for the next cascaded stage. The input data will be collected from the bus, undergoes through these different stages and generates at the end the modulated sinus signal.

3.1 Serial Input Output Interface (SIO)

The Serial Input Output unit is a serial-interface that enables to communicate with external units (e.g. PC), in this application it will be an interface between a processor device 'FHOP processor' and modulator. [4]

Specification of the SIO interface:

1. Baud rate can be configured.
2. Parity can be set even or odd.
3. Interrupt-output signals for transmit and receive signals can be enabled or disabled.
4. 8 bit data transmission

5. Indicate parity-, overflow- and frame errors.
6. Half duplex.

The SIO interface contains 3 port-address registers where the following information will be saved in it:

1. Timer-register to write the requested timer-value corresponding to the wanted baud rate.
2. State Control-register to configure the divider-value, parity and interrupt enabling.
3. Send / receive -register to read and write the serial data.

It is 'RS232' compatible fig (2), asynchronous mode: 1 start bit (low), 8 data bit, 1 parity bit and 1 stop bit (high).

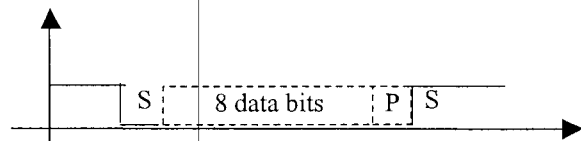


Fig. 2

The SIO interface must be configured to send and receive data according to the desired baudrate.

3.1.1 Determination of divider and timer values:

The baudrate generator can be programmed according to the following equation:

$$f_{baud} = f_{osc} / (8 * divider * (timer + 1)) \quad (\text{Eqn. 1})$$

The primary door frequency f_{osc} can be divided into {2,8,16 or 64} fig. 3, these 4 values are coded by 2 bits, it will be set in the control-register.

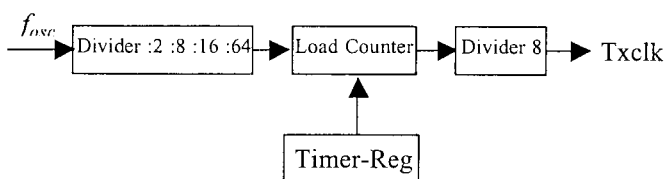


Fig. 3

Timer value will be determined and written in the timer-register.

State Control register has the following function, table 1:

Bit	State Message	Control Signal
7	Frame error	
6	Overrun-error	
5	Parity error	
4		Receive interrupt enable
3		Send interrupt enable
2		Even parity
1		Divider of baudgenerator (b1)
0		Divider of baudgenerator (b0)

Table 1

The state message contains state-information of the SIO and the control-signals are used to control it.

3.1.2 Initialisation Steps for programming:

- Timer-register will be programmed to the requested timer-value corresponding to the wanted baudrate.
- Divider value, 1 parity and interrupt-enabling are programmed and set in the state control-register.
- Transmitted data must be written in the send-register from the bus to be ready for transfer to the next stage serially according to the desired baudrate. [5]

3.2 Serial to parallel converter

To generate one symbol every 2 bits must be read from the SIO interface, the combination of these 2 bits will form one symbol,

see fig (4).

The 1st bit will be saved in a temporary register till the 2nd bit is read.

The time reading is at the half period duration of each bit.

In this application a baud rate of 9600 is used. Symbol rate=(9600 bits/s)/ 2 bits = 4800 sym/s

Bit period = 1/(9600 bit/s) =104.166µs

Symbol period = 104.166 x 2 = 208.33µs

A delay of 1½ bit occurred till the 2 bits are read.

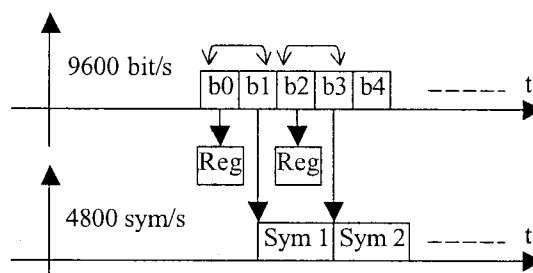


Fig. 4

3.3 Differential QPSK

3.3.1 Definition of Quadrature Phase Shift Keying

Quadrature Phase Shift Keying can be interpreted as 4-PSK, the constellation diagram of QPSK signal is shown in fig. 5.a

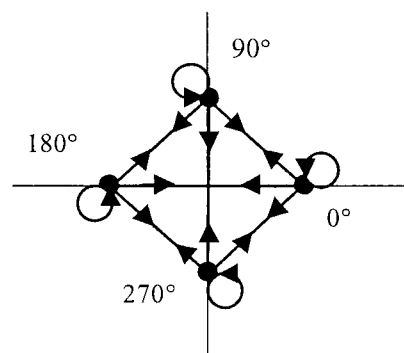


Fig. 5.a

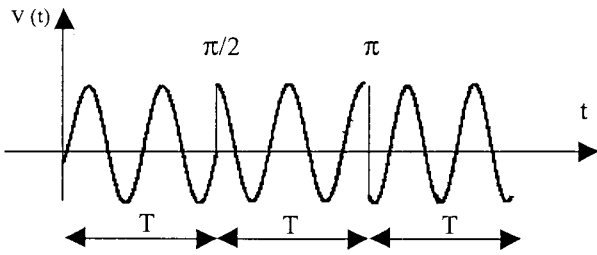


Fig. 5.b

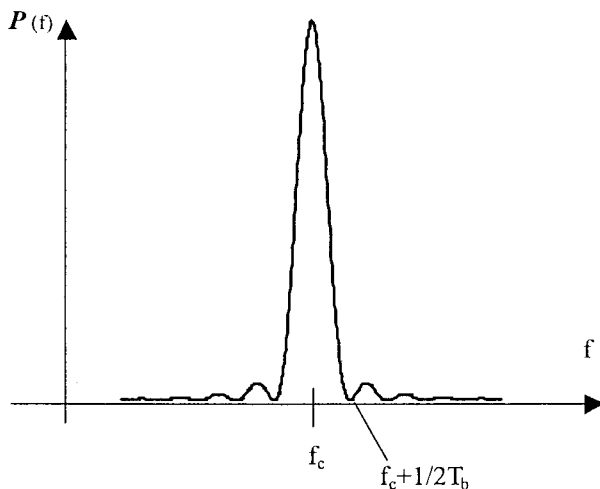


Fig. 5.c

This constellation leads to 4 phase angles $\{0^\circ, +90^\circ, +180^\circ \text{ and } +270^\circ\}$ with the possibility to shift either $\pm 90^\circ, \pm 180^\circ$ or stay on the same angle.

Fig. 5.b shows an example sequence of unfiltered QPSK data corresponding to $+\pi/2$ and $+\pi$ phase shifting, fig. 5.c shows the corresponding power spectral density derived from the following formula :

$$F(f) = T_s [\text{sinc}(\pi f_c T_s)] = l T_b [(\text{sinc}(\pi l f_c T_s))] \Rightarrow$$

$$P(f) = K \cdot [(\text{sinc}(2\pi f_c T_b))]^2 \quad (\text{Eq. 2})$$

Where $K = C / T_b$, C real positive constant, l number of bits per symbol and T_b is the bit duration.

The disadvantages of this shift keying are:

1. Power spectrum bandwidth of unfiltered QPSK is very wide
 $B_{\text{null to null}} = (1/T_b) = 9600 \text{ Hz}$.
2. Effect of 180° phase transitions will result in severe envelope fluctuation that leads to a dip in amplitude and loss of signal.

3.3.2 Why Differential QPSK ?

Most of the signal constellations have the practical problem that they are rotationally invariant typically multiples of $\pi/2$ as in QPSK, by this mean if the constellation is rotated, there is no way that the receiver can distinguish it from a valid constellation, unless it knows the actual transmitted data symbols, which it does not.

Using differential encoding can eliminate the problem, in which the information is encoded by the change in constellation position between symbols rather than absolute position.

The IS-54 standard for digital cellular radio in North America transmits two bits per symbol as a form of Quadrature PSK.

These two bits are not associated with four phases but with eight equally spaced phases as shown in fig. 6.

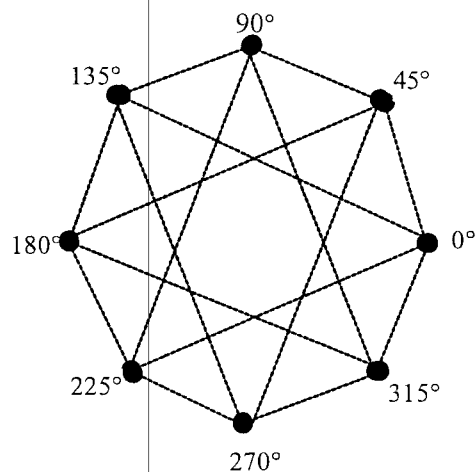


Fig. 6

The two bits are mapped into one of the four phases transitions from one symbol to the next, these transitions are shown as dashed lines for each starting phase.

The differential phase is determined from the two input information bits according to table 2:

Bit 0	Bit 1	Phase shift
0	0	+45°
0	1	+135°
1	1	-135°
1	0	- 45°

Table 2

The eight phases can be coded with three bits as eight states, see table 3:

Phase angle	State
0°	000
45°	001
90°	010
135°	011
180°	100
225°	101
270°	110
315°	111

Table 3

For each position there will be 4 possible phase shifts { ±45°, ±135° }.

The advantages of this modulation technique are:

1. There is no 180° phase shift from one state to another but either ±45° or ±135°, this avoid a severe envelope fluctuation (8 dB).
2. Repetition of the same transmitted data will not lead to stay on the same phase but it will be jumping consecutively from one state to another depending on the transmitted data.
3. Receiver depends only on the change in phase from one symbol to the other to extract the data and not on the

absolute value of the transmitted phase.

According to the previous symbol and the present one, the system will decide the next state and the corresponding phase shift that must be performed.

[6]

3.4 Sawtooth Phase Shift by using a Gaussian Filter

In this stage, a Gaussian filter will be used to carry out the smooth transition from one phase to another.

The general equation of a sinusoidal signal in the time domain is:

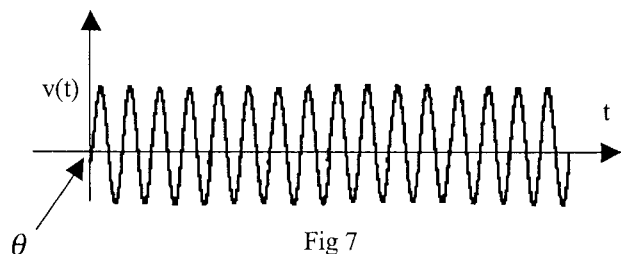
$$v(t) = A \cdot \cos[w_c \cdot t + \vartheta(t)] \quad (\text{Eq.3})$$

where A : amplitude of the wave
 w_c : angular frequency of the wave
 $\vartheta(t)$: phase shift of the wave

if $\vartheta(t)$ is constant and independent of time then it ends up with :

$$v(t) = A \cdot \cos[w_c \cdot t + \theta] \quad (\text{Eq.4})$$

This is the general equation of a carrier signal



The instantaneous frequency is defined as the frequency that is present at a particular instant of time :

$$f_i(t) = \frac{1}{2\pi} \frac{d}{dt} [w_c \cdot t + \vartheta(t)] \quad (\text{Eq. 5})$$

$$f_i(t) = f_c + \frac{1}{2\pi} \cdot \left[\frac{d\vartheta(t)}{dt} \right] \quad (\text{Eq. 6})$$

and the frequency deviation is :

$$f_d(t) = f_i(t) - f_c = \frac{1}{2\pi} \cdot \left[\frac{d\vartheta(t)}{dt} \right] \quad (\text{Eq. 7})$$

In QPSK modulation, the angle phase is shifted from one value to another in a time limit that tends to zero, fig 8.a.

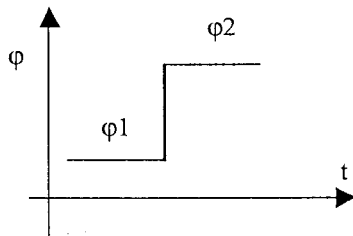


Fig 8.a

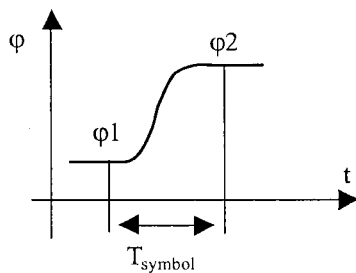


Fig 8.b

These transient shiftings lead to generate a wide spectrum with high effect side lobes, and in the case of QPSK modulation the spectrum will be a sinc shape.

To decrease these side effect lobes and increase the power within a limited bandwidth a gaussian filter will be used to smooth this transient shift in a certain period of time mainly within a symbol duration, fig 8.b.

The idea is to increase the carrier frequency and decrease it back to the original value or vice versa within a period of one symbol, this

process will lead to gain an increment or decrement of phase shift and achieved the desired phase shift value, for example $\{\pm 45^\circ \text{ or } \pm 135^\circ\}$.

The deviation of the frequency will have a Gaussian shape property, fig 9.a.

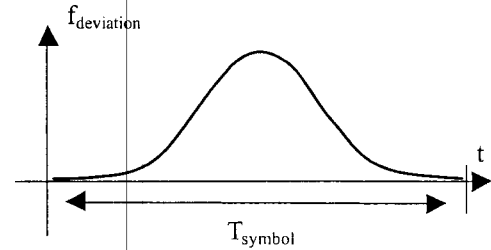


Fig 9.a

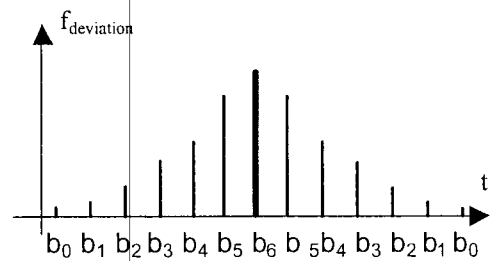


Fig 9.b

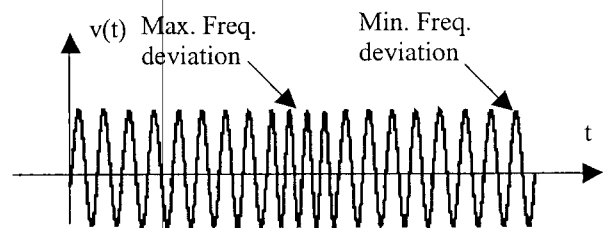


Fig 9.c

To perform the exponential of Gaussian curve, samples of Gaussian curve will be selected. The filter will have an order of 13, fig 9.b.

The effect of this Gaussian filter is shown in fig 9.c, the carrier frequency increases till maximum frequency value at the middle and then decreases back to its original value.

The frequency-time relation is described as follows:

$$f_i(t) = f_c + K_1 e^{-\left[\frac{t - \left(\frac{T_{\text{symbol}}}{2} \right)}{K_2} \right]^2} \quad (\text{Eq. 8})$$

Equating (Eq.6) and (Eq.8) :

$$f_i(t) = f_c + \frac{1}{2\pi} \cdot \left[\frac{d\vartheta(t)}{dt} \right] = f_c + K_1 e^{-\left[\frac{t - \left(\frac{T_{\text{symbol}}}{2} \right)}{K_2} \right]^2} \quad (\text{Eq. 9})$$

$$\frac{1}{2\pi} \cdot \left[\frac{d\vartheta(t)}{dt} \right] = K_1 e^{-\left[\frac{t - \left(\frac{T_{\text{symbol}}}{2} \right)}{K_2} \right]^2} \quad (\text{Eq. 10})$$

$$\vartheta(t) = 2\pi \cdot \int_{-\infty}^t K_1 e^{-\left[\frac{t - \left(\frac{T_{\text{symbol}}}{2} \right)}{K_2} \right]^2} \cdot dt \quad (\text{Eq. 11})$$

A closed form of the Gaussian integration function does not exist, math tables or numerical integration techniques must be used to evaluate it.

A definition for that is the Error function erf(x) or Q function, they are used to replace the integration form and they are known as the cumulative distribution function (CDF) for the Gaussian distribution :

$$\vartheta(t) \cong Q(t) \cong K \cdot \text{erf}(t) \quad (\text{Eq. 12})$$

The target is to achieve the desired phase shift $\vartheta(t)$ after a period of time (T_{symbol}) for one of the following values $\{\pm 45^\circ \text{ or } \pm 135^\circ\}$. These are performed after selecting the exact values of the Gaussian filter.

The coefficients are selected for a Gaussian distribution of average value:

$$m = \frac{T_{\text{symbol}}}{2}$$

and variance:

$$\sigma = \frac{1}{\sqrt{2\pi}}$$

the coefficients $\{b_6, b_5, b_4, \dots\}$ are proportional to the following Gaussian sample values : $\{1.000, 0.913, 0.697, 0.444, 0.236, 0.105, 0.03\}$

A closed form for the modulated signal can be described as :

$$v(t) = A \cdot \cos[w_c \cdot t + K \cdot \text{erf}(t)] \quad (\text{Eq. 11})$$

A comparison between an unfiltered QPSK modulated signal and QPSK modulated signal with Gaussian filtering is illustrated in fig 10. A data corresponding to a phase shift of $+45^\circ$ and -45° must be performed.

Fig 10.a shows the variation of the frequency with respect to time, in unfiltered modulation the carrier frequency doesn't change and stay constant all the time.

Fig 10.b shows the variation of the frequency with respect to time, a Gaussian shape deviation changes the instantaneous frequency then return it back to the carrier frequency value.

Fig 10.c and fig 10.d show the time domain signal of unfiltered QPSK and filtered one, discontinuity of the signal at the time phase shift appeared in unfiltered QPSK while in the other case a continuous phase frequency deviation appeared in the modulated signal. This phenomena can be interpreted as a continuous phase frequency modulation technique.

Fig 10.e and fig 10.f shows the corresponding phase deviations, a sudden jump occurred in the unfiltered signal while shifting from φ_0 to $\varphi_0 + 45^\circ$ and then dropping back to φ_0 achieving the desired phase shift values.

For the 2nd case a smooth shifting has been performed while raising up and then going back to φ_0 .

Fig 10.g shows the spectrum of the 2 signals, the effect of the filter attenuate the side lobes and gained more power within smaller bandwidth.

[7] [8]

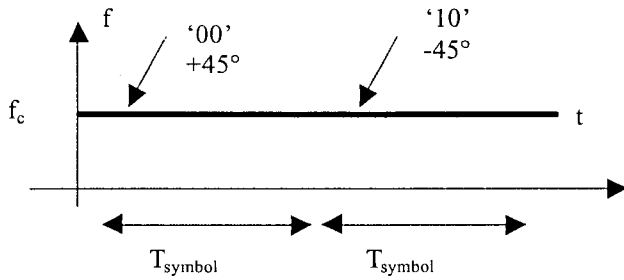


Fig 10.a

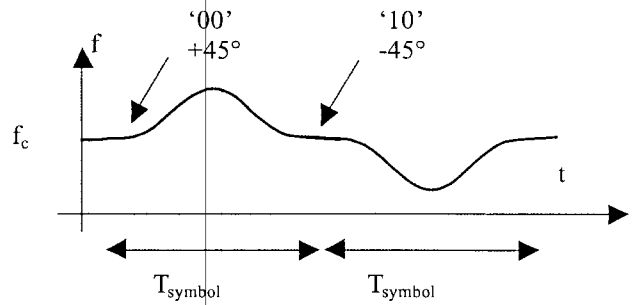


Fig 10.b

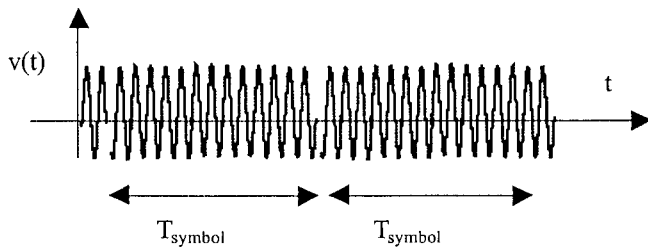


Fig 10.c

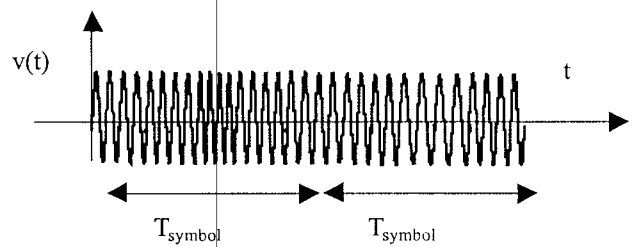


Fig 10.d

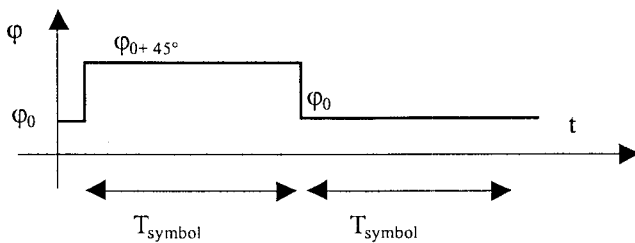


Fig 10.e

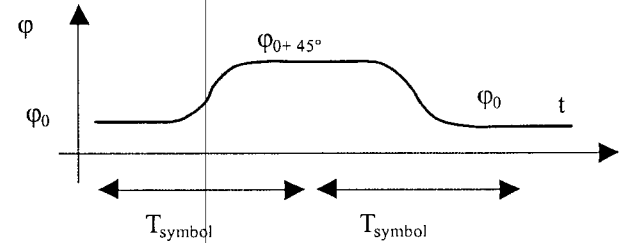


Fig 10.f

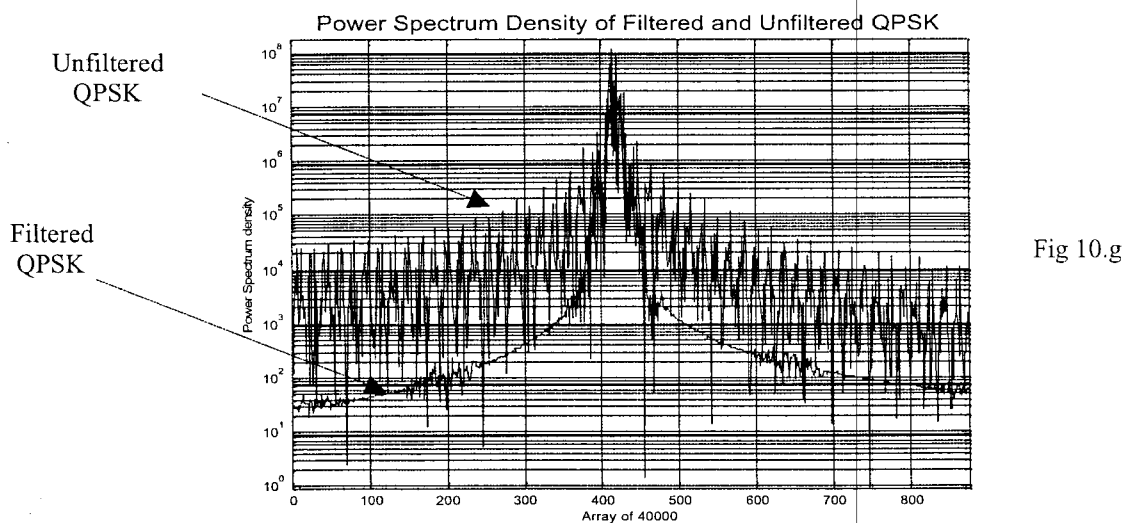


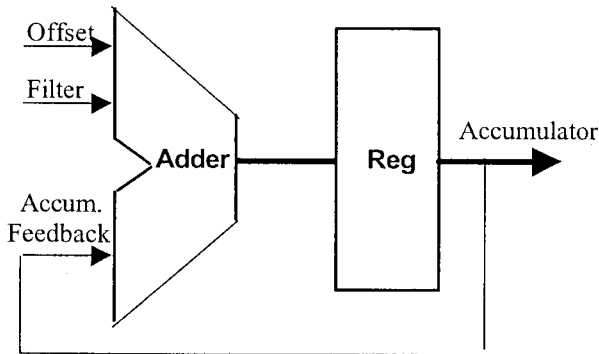
Fig 10.g

3.5 Numerical Controlled Oscillator for Generation of the Carrier Frequency

Numerical controlled oscillator is used to generate the carrier frequency as well as to perform the frequency deviation due to the Gaussian filter.

It is made up of an adder and a register, the register is clocked with frequency 11.0592 MHz.

The adder has three inputs, the offset value which corresponds to the frequency carrier value, the Gaussian filter coefficients and a feedback from the accumulator as shown in fig. 11.



The adder has 20 bits resolution which corresponds to $2^{20} = 1048576$ the max number that can be reached after addition.

The offset value that generates the carrier can be calculated as follows:

$$f_{out} = \frac{\Delta P \cdot f_{clk}}{2^{bits}} \quad (\text{Eq. 13})$$

$$\Rightarrow \Delta P = \frac{2^{bits} \cdot f_{out}}{f_{clk}} = \frac{2^{20} \cdot 115.2\text{KHz}}{11.0592\text{MHz}} = 10922$$

The value that generates a periodic signal with frequency 115.2 KHz is

$$10922 = (10101010101010)_b$$

The number of bits required to generate the carrier frequency is 14.

The NCO will generate a sawtooth signal every period of time $= 1/115.2\text{KHz} = 8.68 \mu\text{s}$. 1 bit has an influence on the frequency resolution of:

$$f_{bit} = \frac{2^1 \cdot 11.0592\text{MHz}}{2^{20}} = 21.09\text{Hz}$$

3.5.1 Filter Coefficients

The NCO has two inputs plus third as feedback from the accumulator.

The two inputs are the offset value and the filter coefficients.

When the filter has no influence on the NCO or the value of the filter coefficient is zero then the NCO will generate a sawtooth signal with periodicity $8.68 \mu\text{s}$, this sawtooth signal will stay fixed as long as there is no influence from the filter, fig 12.

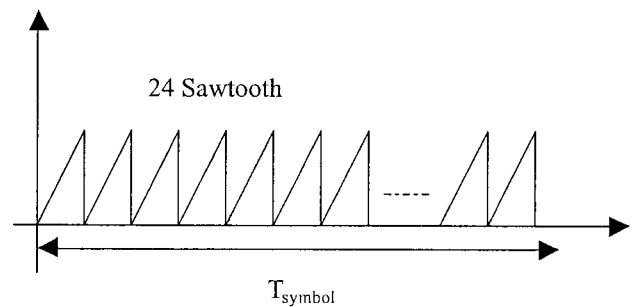


Fig 12

The filter must perform four different phase shift possibilities $\{\pm 45^\circ \text{ or } \pm 135^\circ\}$, after certain calculations the following coefficients were been assigned to achieve the desired phase shift, table 4:

Coeff.		+45°		+135°
b ₀	5	0000101	15	000001111
b ₁	13	0001101	40	000101000
b ₂	30	0011110	90	001011010
b ₃	56	0111000	169	010101001
b ₄	89	1011001	265	100001001
b ₅	116	1110100	348	101011100
b ₆	126	1111110	380	101111100

Table 4

The max value b_6 for 135° phase shift is 380 which needs 9 bits, so to achieve the max deviation 9 bits needed to perform it. On the contrary if -45° or -135° is desired, the 2th complement will be performed on the previous coefficients and will be added with the offset to achieve negative deviation to the frequency.

Since the adder perform 20 bits addition, the zeros at the left side will be inverted to ones, 'X' symbol will stand as 11 ones to the left in the following table 5.

Coeff.		-45°		-135°
-b ₀	-5	X1111011	-15	X111110001
-b ₁	-13	X1110011	-40	X111011000
-b ₂	-30	X1100010	-90	X110100110
-b ₃	-56	X1001000	-169	X101010111
-b ₄	-89	X0100111	-265	X011110111
-b ₅	-116	X0001100	-348	X010100100
-b ₆	-126	X0000010	-380	X010000100

Table 5

One symbol can be modulated with 24 sinewave cycles if the input filter was zero:

Symbol Rate = 4800 symbol/sec =>

1 symbol duration = 208.33 μs

Carrier frequency = 115.2 KHz =>

1 cycle duration = 8.680 μs

1 symbol duration/1 cycle duration = 208.33/8.68=24 cycles per symbol.

This ratio will slightly change due to the filter effect.

The following table shows the changes occurred to the ratio:

Phase Shift	Excess	Ratio	Cycles/symbol
+45°	+45°/360°	+0.125	24.125
+135°	+135°/360°	+0.375	24.375
-135°	-135°/360°	-0.375	23.625
-45°	-45°/360°	-0.125	23.875

Table 6

The modulated signal will have a continuous phase although the ratio of cycles per symbol is not fixed and changes according to the desired transmitted data.

3.6 Pulse Width Modulation generation

A pulse modulation technique is used to generate a bit stream of ones and zeros that are mapped to the modulated signal.

To generate the modulated signal in the time domain using the output of the NCO, the following equation must be performed :

$$v(n) = \sin\left(\frac{\text{output}_{NCO}(n)}{2^{20}} \cdot 2\pi\right) \quad (\text{Eq. 14})$$

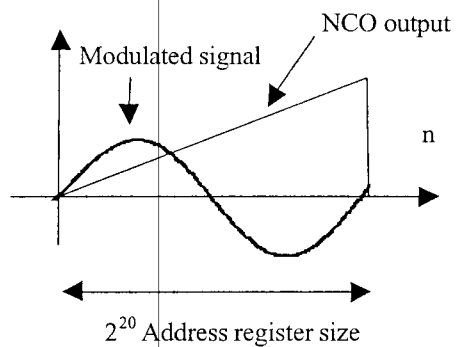


Fig 13

Fig 13 shows the direct relation between the NCO output, the modulated signal and the mapping between both signals.

To perform eqn(), a sine function generator is needed as well as multiplication process must be performed, such operation lower the efficiency of the system and to build up such system on an IC is not worthy and needs a lot of effort in time to perform it, fig 14.a.

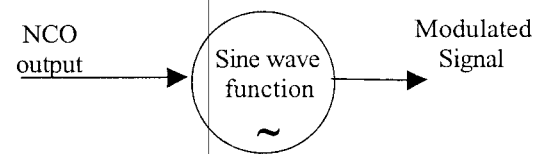


Fig 14.a

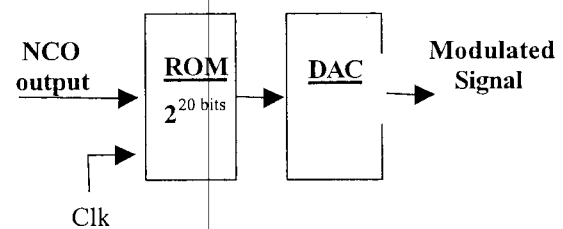


Fig 14.b

Another suggestion is to build up a ROM followed by analog to digital converter, where the samples of a sine wave will be saved in the ROM and accessed by the NCO. The output of the ROM will be discrete values clocked by the carrier frequency 11.0592 MHz as shown in fig 14.b, ADC needed to convert discrete values into continuous time signal. This solution is also not efficient and the ROM has always the disadvantage of its size as well as ADC is not worthy to be implemented.

An alternative solution is using the PWM technique. The idea behind the PWM is to map the weight of the signal $v(t)$ at an instance t and give the ratio of its value wrt to the full scale, fig 15.

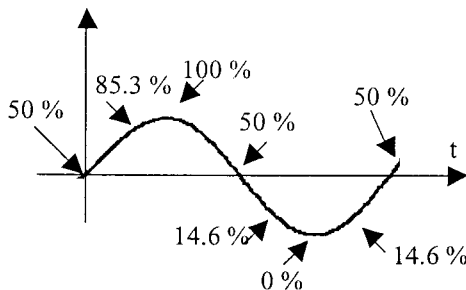


Fig 15

The ratio can be coded by a stream of bits (ones and zeros) in such a way that the number of ones wrt to zeros gives the same ratio.

A small example can clarify this point, suppose we have the following percentages (50%, 75%, 100%, 75%, 50%, 25% and 0%) and must be coded with four bit stream for each code then one of the solutions will be: (0110, 0111, 1111, 1110, 0110, 0010 & 0000). As much the number of bits and the sampling points increases as much the accuracy will increase but there must be a trade off between them for different applications. The question raised here, what is the relation between the continuous sine wave and stream of bits (ones and zeros) ?

The idea is that the sine wave that was coded by this bit stream can be later extracted from it

by mean of good filtering around the carrier frequency.

The frequency of the sine wave is internally embedded in this bit stream.

To verify that, suppose we have the following example:

A sine wave of frequency 115.2 KHz, a clock of the same frequency and a bit stream that mapped to the sine wave with a bit duration of 90.42 μ s and has a length of 256 bits, fig 16. By using matlab simulation, and performing Fourier transform to the different signals, it can be noticed that a common frequency component is found in the three spectrum. This component is the 115.2 KHz signal. In the third case, a good filtering can extract this component and generate the sine wave which stands for the modulated signal.

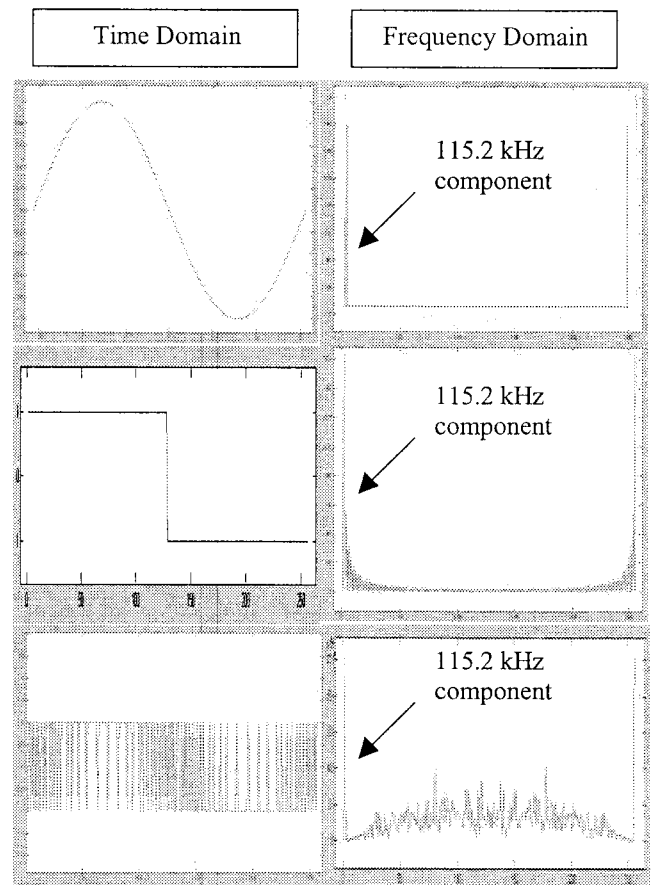


Fig 16

In our case the following specifications have been selected to perform the PWM for the modulated signal.

The NCO output has a resolution of 20 bits, the first five most significant bits will be selected as sampling points.

Each sampling point will be coded by eight bits.

5 bits => 32 sampling points for one sine wave.

Total number of bits in one stream:
32 sampling points · 8 bits = 256 bits.

Address of the sample	Position of the bit							
	000	001	010	011	100	101	110	111
00000	1	0	1	1	0	1	0	1
00001	0	1	0	1	0	1	1	0
00010	1	0	1	1	1	0	1	1
00011	0	1	1	0	1	1	0	1
00100	1	1	1	1	0	1	1	1
00101	0	1	1	1	1	1	0	1
00110	1	1	1	1	1	0	1	1
00111	1	1	1	1	1	1	1	0
01000	1	1	1	1	1	1	1	1
01001	0	1	1	1	1	1	1	1
01010	1	0	1	1	1	1	1	1
01011	1	0	1	1	1	1	0	1
01100	1	1	1	1	0	1	1	0
01101	1	1	1	0	1	1	1	0
01110	1	1	0	1	1	0	1	0
01111	1	1	0	1	0	1	0	1
10000	1	0	1	0	0	1	1	0
10001	0	1	0	1	0	1	0	0
10010	1	0	0	1	0	0	1	0
10011	0	1	0	0	1	0	0	0
10100	0	1	0	0	0	1	0	0
10101	0	0	0	1	0	0	0	0
10110	1	0	0	0	0	0	0	0
10111	0	1	0	0	0	0	0	0
11000	0	0	0	1	0	0	0	0
11001	0	0	0	0	1	0	0	0
11010	0	0	0	0	1	0	0	0
11011	0	0	0	0	1	0	0	0
11100	1	0	0	0	0	1	0	0
11101	0	1	0	0	0	1	0	0
11110	1	0	0	1	0	0	1	0
11111	1	0	0	1	0	1	0	1

Table 6

Table 6 shows the bit stream generator, every sample is coded by 8 bits and these 8 bits are numbered by 3 bits (000...111).

In other word, every bit in the table can be addressed by 8 bits (address of the sample + position of the bit)

But since there is a ratio of 96 samples per sine wave then only 96 bits will be selected from 256 bits, a trade off must be taken in consideration.

Table 7 shows an example of a selected 96 bit stream from the generation table.

This bit stream will be selected by the output of the NCO where it uses the first most significant 8 bits to select the bits.

A ratio of 96/256 must be selected which means three bits out of eight bits must be selected for every sample.

Address of the sample	Position of the bit							
	000	001	010	011	100	101	110	111
00000	1	0	1	1	0	1	0	1
00001	0	1	0	1	0	1	1	0
00010	1	0	1	1	1	0	1	1
00011	0	1	1	0	1	1	0	1
00100	1	1	1	1	0	1	1	1
00101	0	1	1	1	1	1	0	1
00110	1	1	1	1	1	0	1	1
00111	1	1	1	1	1	1	1	0
01000	1	1	1	1	1	1	1	1
01001	0	1	1	1	1	1	1	1
01010	1	1	1	1	1	0	1	1
01011	1	1	1	1	1	1	1	0
01100	1	1	1	1	1	1	1	1
01101	0	1	1	1	1	1	1	1
01110	1	0	1	1	1	1	1	1
01111	1	1	1	1	0	1	1	0
10000	1	0	1	0	0	1	1	0
10001	0	1	0	1	0	1	0	0
10010	1	0	0	1	0	0	1	0
10011	0	1	0	0	1	0	0	0
10100	0	1	0	0	0	1	0	0
10101	0	0	0	1	0	0	0	0
10110	1	0	0	0	0	0	0	0
10111	0	1	0	0	0	0	0	0
11000	0	0	0	1	0	0	0	0
11001	0	0	0	0	1	0	0	0
11010	0	0	0	0	1	0	0	0
11011	0	0	0	0	1	0	0	0
11100	1	0	0	0	0	1	0	0
11101	0	1	0	0	0	1	0	0
11110	1	0	0	1	0	0	1	0
11111	1	0	0	1	0	1	0	1

Table 7

3.7 Band Pass Filter

Last stage before transmitting the signal, a good filtering is required to filter out the sine wave component from the bit stream; the band pass filter has a resonance frequency around 115.2 KHz, the analog part is seen in fig 17, where the last stage of the modulator and the first stage of the demodulator are shown.

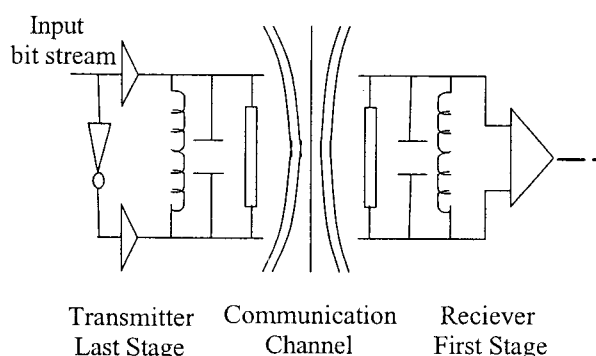


Fig 17

This stage has not been totally implemented, but simulated by P-SPICE with acceptable results.

4. Simulation Results

The digital part is totally described by VHDL language, VSIM program is used to simulate the codes.

Three simulation figures are shown using ModelSim and two figures by Matlab program for verification.

4.1 Simulation result (1)

Fig 18 shows the output of the Gaussian filter. These Gaussian curve values are added with the offset in the NCO to obtain the desired phase shift.

The time duration of the curve is synchronised with the symbol duration.

Four Gaussian curves performing phase shift of $(+45^\circ, +135^\circ, -45^\circ$ and $-135^\circ)$ correspond to four different symbols (00, 01, 10 and 11)

4.2 Simulation result (2)

In fig 19, the effect of the Gaussian filter coefficients are shown, every 24 sawtooth signals modulate one symbol.

There is a phase shift between the modulated signal and the carrier signal, the two signals were totally at the beginning in phase, after the coefficients of the filter being added to the NCO a difference in the phase can be noticed till $+135^\circ$ phase shift has been achieved.

The modulated signal increases its frequency and then goes back to the carrier frequency to obtain the desired phase shift.

The modulated signal will be further coded by "Pulse Width Modulation".

4.3 Simulation result (3)

The Pulse width modulated signal is generated according to table 6; the first eight most significant bits of the binary representation of the value are used to select either "one" or "zero".

In the positive cycle of the sine wave the density of ones are more and in the negative cycles the density of zeros are more.

Fig 20 shows the PWM of a modulated signal. A phase shift of $+135^\circ$ between the carrier and the modulated signal is shown.

4.4 Simulation result (4)

Matlab software program is being used to simulate an example of a four consecutive symbols: {01, 10, 00 and 10}.

This sequence leads to perform the following phase shifts: $\{+135^\circ, -45^\circ, +45^\circ$ and $-135^\circ\}$.

A comparison between the spectrum of unfiltered modulated signal and a Gaussian filtered modulated signal is shown in fig 21. The unfiltered signal suffers from high effect side lobes, which is a disadvantage to transfer it in a communication system.

The Gaussian filtered modulated signal has achieved attenuation up to 40 dB in its side

lobes in comparison with the other as shown in fig 21.

A 98% of the total power signal is found in a bandwidth of 9.6 KHz in the filtered signal while around 88% of the total power was found in the unfiltered signal for the same bandwidth.

This power percentage differs from one example to other, after performing different examples a difference of 10% up to 20% in the power level for the same bandwidth between both modulated signals.

The phase shift is shown in fig 22, the unfiltered signal jumps from one phase shift level to other in a discrete form while the filtered signal changes smoothly from one level to the other.

The corresponding phase shifts are as follows: $+135^\circ$, -45° , $+45^\circ$ and -135° in sequence.

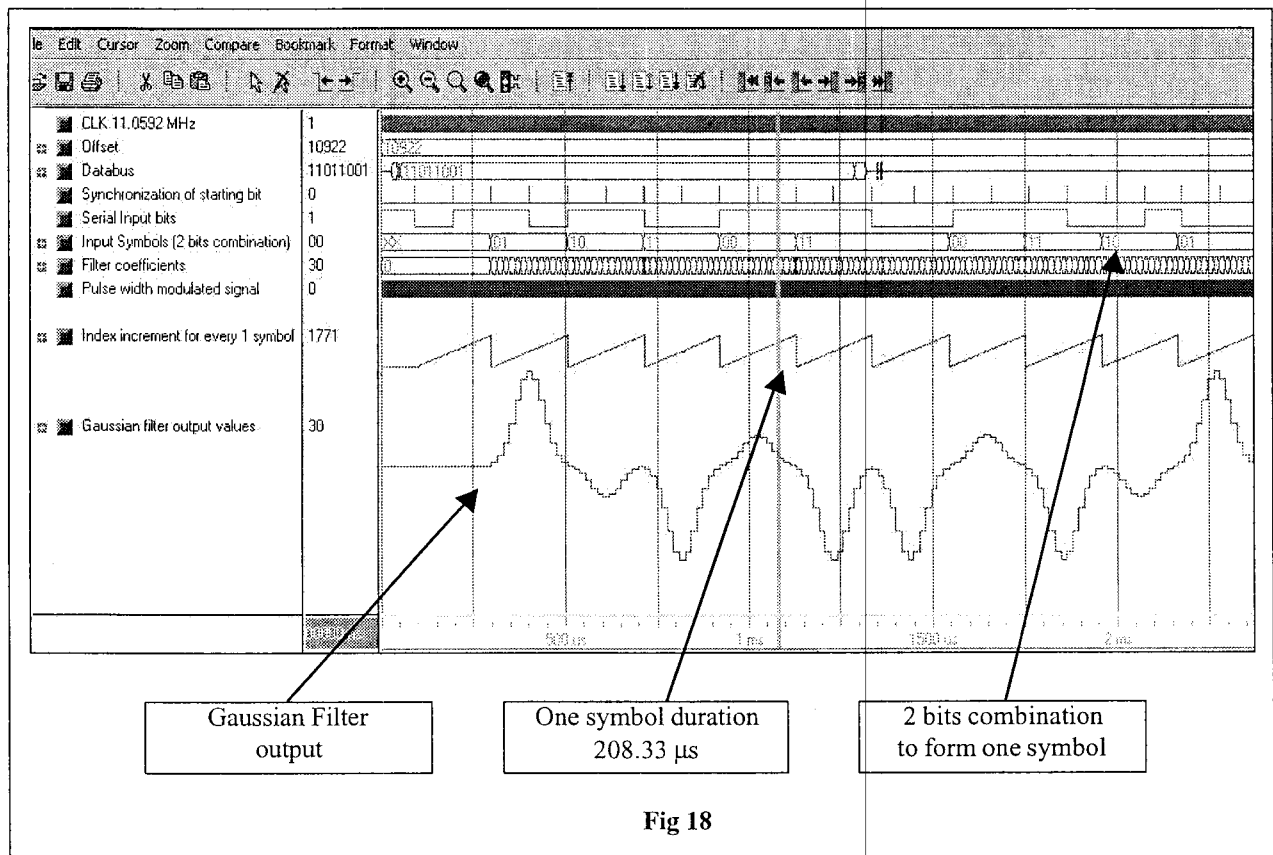
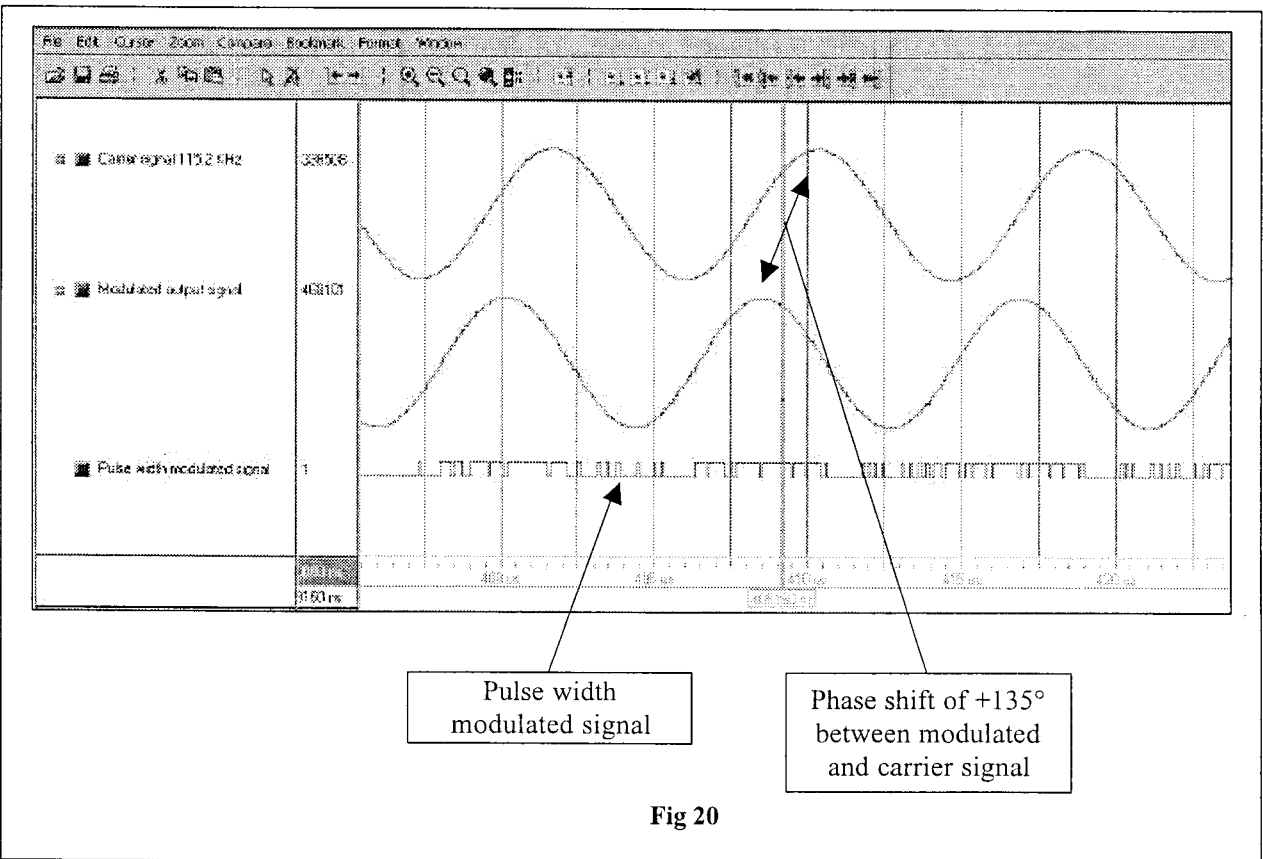
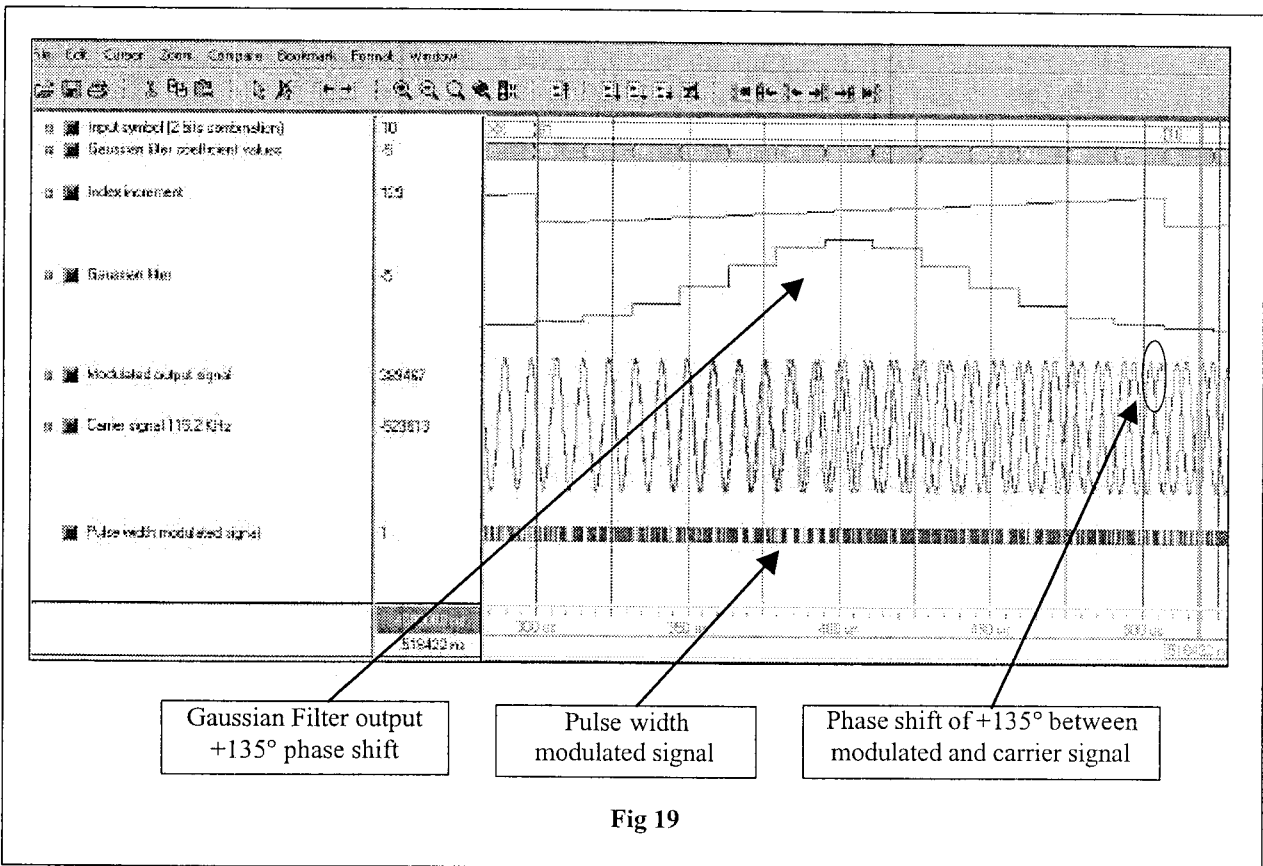
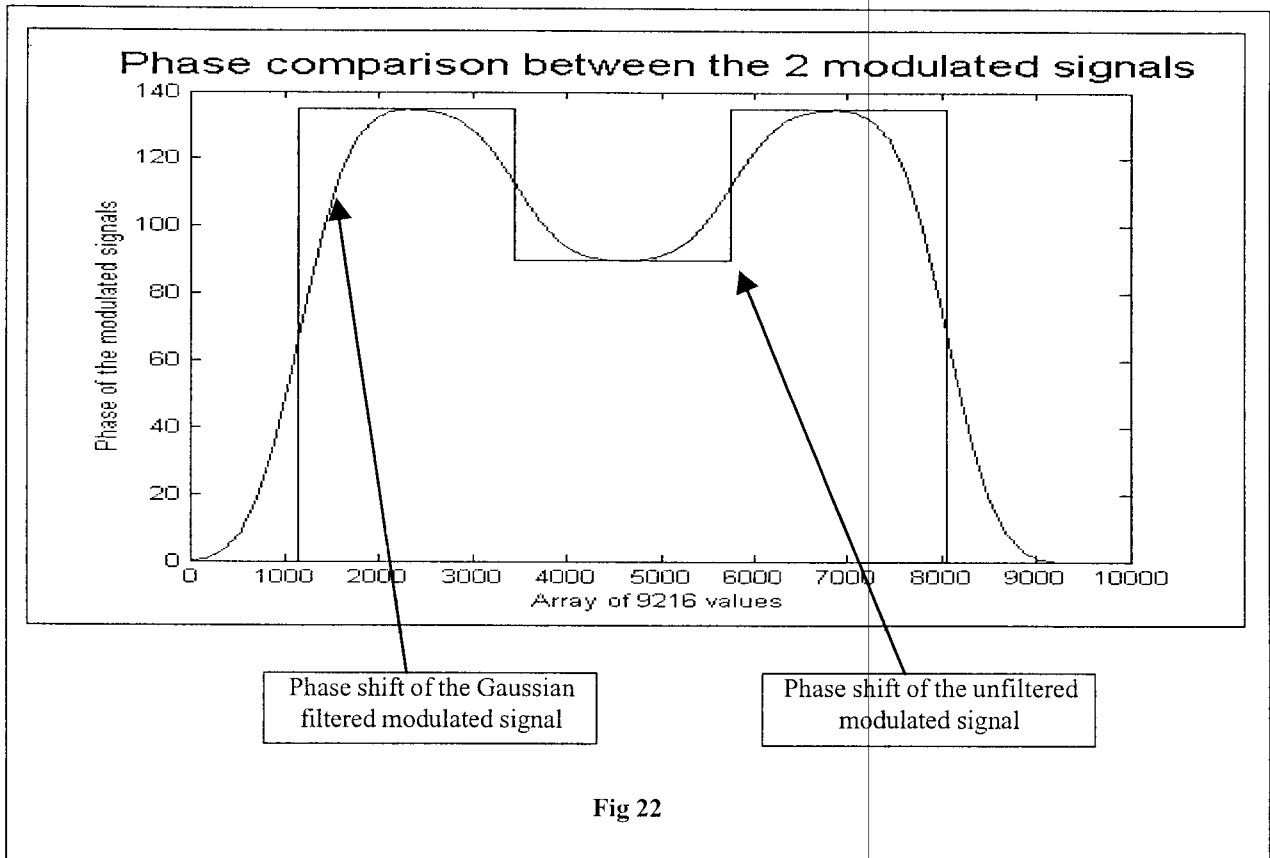
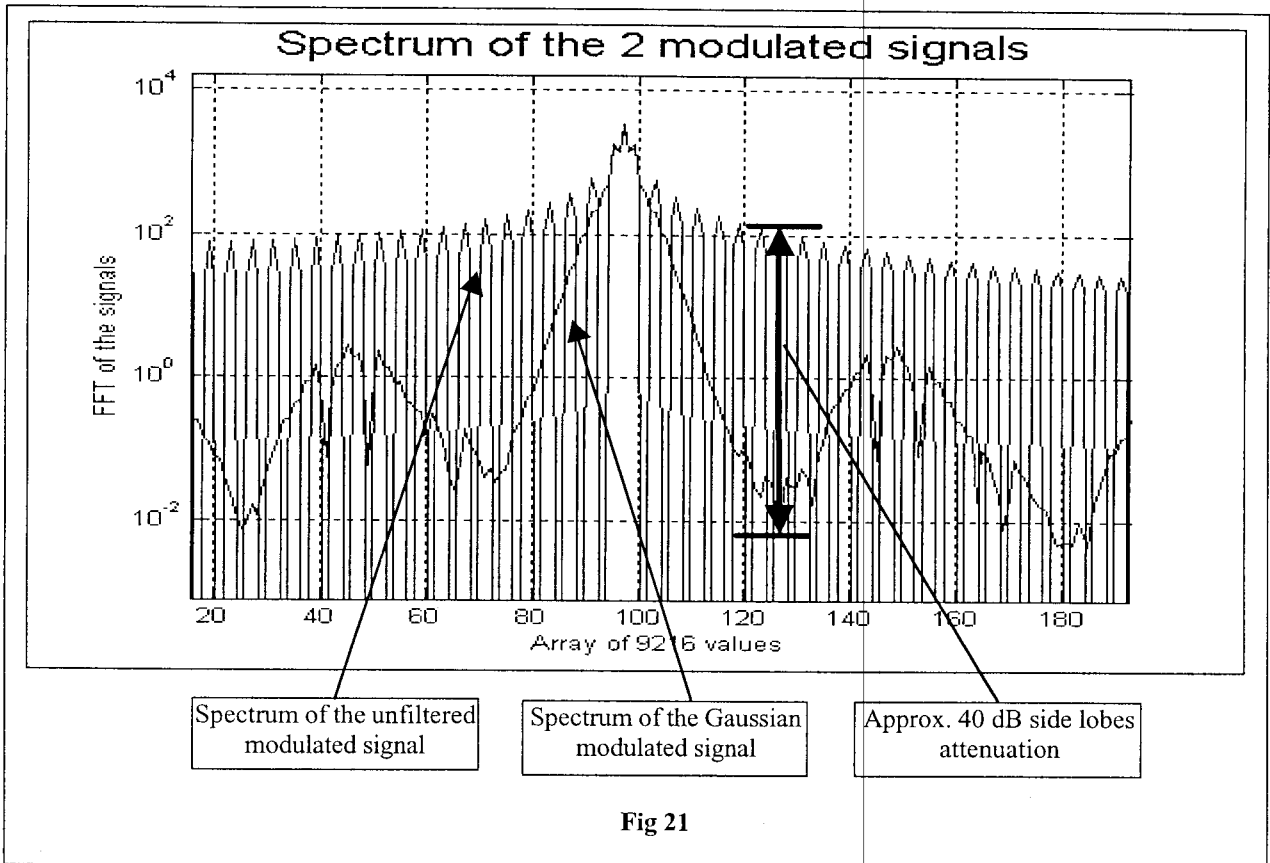


Fig 18





5. Conclusion

- The digital part of the modulator is being done.
- An attenuation up to 40 dB in the side lobes of the filtered signal has been achieved.
- A smooth transition from one phase shift to another is performed by the Gaussian filter.
- A higher power concentration is found in the filtered signal within the same bandwidth with respect to unfiltered signal.
- PWM coding is used to generate a bit stream representing the modulated signal to replace the drawback of using sine wave generator, DAC and ROM.
- The digital part is synthesized using LeonardoSpectrum.
- Further development for the demodulator will be done in the 2nd phase of the project.

6. References

- [1] "ECG-Recorder"
ASIC Design Center, FH-Offenburg
- [2] "Chip in Tire"
ASIC Design Center, FH-Offenburg
- [3] "Thermollogger"
ASIC Design Center, FH-Offenburg
- [4] "FHOP Microprocessor"
ASIC Design Center, FH-Offenburg
- [5] "SIO Interface"
ASIC Design Center, FH-Offenburg
- [6] "Digital Communication"
2nd edition, Edward Alee
- [7] "Digital & Analog Communication System"
6th edition, Leon Couch
- [8] "Digital Communications"
Glover & Grant



FPGAs entwerfen - ein Lernprogramm im Selbststudium

Dipl.-Ing. Ehrenfried Futterer
Institut für Mikroelektronik Stuttgart



Dipl.-Ing. Ehrenfried Futterer
Institut für Mikroelektronik Stuttgart

Projektdaten im Überblick

- Partner: EAZ Aalen,
Institut für Mikroelektronik Stuttgart
- Laufzeit: 1998 - 2001
- Zielgruppe: Studenten, Techniker und Meister
- Förderung: Europäische Union (ESF)
Landesgewerbeamt BW
Wirtschaftsministerium BW





Struktur des Lernprogramms

◦ Lernpfad

Digitaltechnik + Entwurf
Simulation + VHDL

◦ Tools

Software Tutorials
Mentor / Xilinx

◦ Projekte

Ampelsteuerung
Digitaluhr ...



active chips



Aufbau

- Grundlagen der Chip Entwicklung
Multimediale Lernumgebung mit Übungen
- Tutor für die Design-Software
Einstieg in komplexe Toolbedienung erleichtern
- Übungsprojekte zur Vertiefung
Eigenständige Bearbeitung als 2. Lernspirale
- Experimentierboard zur praktischen
Realisierung der Übungsprojekte



active chips



Konzepte

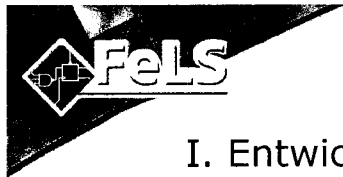
- Einfache selbsterklärende Bedienerführung
- Anschauliche Darstellung komplexer Inhalte
- Learning by doing, interaktive Beispiele
- Übungsaufgaben mit Lernkontrolle / Lernstatus



Probleme

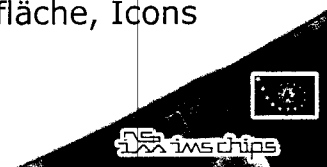
- Verschiedene Autoren - verschiedene Stile
- Aufwand bei Realisierung der Autorenvorgaben
- Auslagerung des Software Tutorials aus dem Lernsystem:
 - > häufige Updates der Design-Software
 - > separate Nutzung ermöglichen
 - > Änderungen der Hersteller





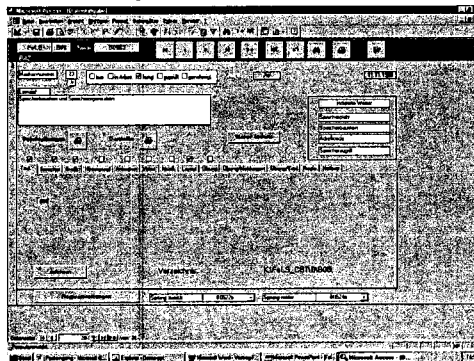
I. Entwicklungsphasen der CBT FeLS

- Grobkonzeption -
Werkzeuge, Abläufe und Didaktik
- Feinkonzeption -
Festlegung von Struktur und Inhalt
- Grundprogrammierung Menüstruktur / Features
- Layout / Grafik Benutzeroberfläche, Icons



II. Entwicklungsphasen der CBT FeLS

- Autoren erstellen das Drehbuch und
übertragen die Inhalte in die Datenbank





III. Entwicklungsphasen der CBT FeLS

- Die Redaktion überprüft die Drehbücher und Medien
- Umsetzen der Drehbücher durch die Produktion in Director-Filme
- Erprobung im Unterricht - Fehlererkennung und Optimierung



Werkzeuge

- Autoredatenbank programmiert in MS-Access
- CBT-Programmierung in Macromedia Director V7
- Software Tutor realisiert mit DemoShield V5.4

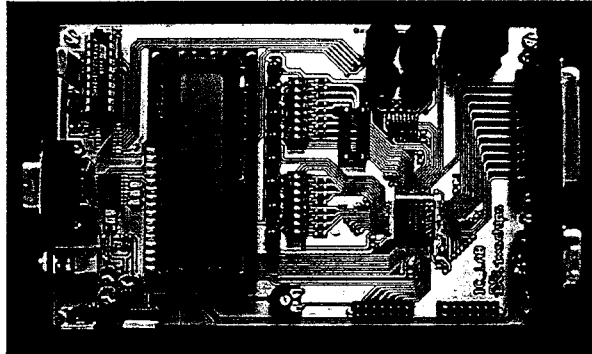




Dipl.-Ing. Ehrenfried Futterer
Institut für Mikroelektronik Stuttgart

IC_Lab Experimentierboard

zur praktischen Implementierung der FeLS Projekte:



- Digitaluhr
- Ampelanlage



imschips



Dipl.-Ing. Ehrenfried Futterer
Institut für Mikroelektronik Stuttgart

IC_Lab Features

- Reprogrammierbarer Xilinx SpartanXL FPGA
- 10.000 Gatter (600 FlipFlops)
- Programmierung per Download
- Serielle, parallele und Tastaturschnittstelle
- Taster zur Eingabe
- LEDs und LCD-Display zur Ausgabe
- Erweiterbarkeit (Quarz, Add On-Platine)
- Steckernetzteil + Dokumentation



imschips



Dipl.-Ing. Ehrenfried Futterer
Institut für Mikroelektronik Stuttgart

Ausblick

- Informationen / Demo bei:
<http://fels.ims-chips.de> oder futterer@ims-chips.de
- FeLS V2.1 F&L-Preis: **149,00 €**
(für Mentor FPGA Advantage 5.x)
- IC_Lab Experimentierboard F&L-Preis: **150,00 €**
(Place & Route / Download SpartanXL für Xilinx ISE)
- Entwicklungspartnerschaften bei
Elektronik-Lernsystemen



ESD Schutz in Deep-Sub-Micron Technologien

Dr. Harald Goßner, Infineon Technologies

Infineon Technologies, Balanstr. 73, 81541 München
Email: Harald.Gossner@infineon.com

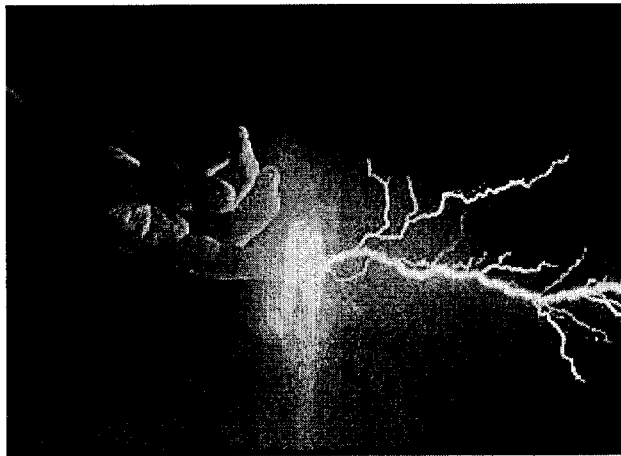
Der Schutz gegen elektrostatische Entladungen (**ElectroStaticDischarge**), die während der Handhabung der mikroelektronischen Bauteilen vor deren Einbau auf einer Platine oder in das jeweilige System auftreten können, ist eine der wesentlichen Anforderungen an das Design der IO-Zellen und Versorgungszellen eines ICs. Wird ein geeigneter ESD Schutz nicht gewährleistet, so gibt es Beispiele, dass Ausfallsraten bis zu einigen 10% beobachtet werden. Während die Größenverhältnisse der Bauelemente und damit verknüpfte elektrische Werte wie Durchbruchspannungen mit fortlaufender Entwicklung der Prozesstechnologien immer kleiner werden, skaliert die Anforderung bezüglich der ESD Robustheit der Bausteine nicht. Dies führt dazu, dass die Entwicklung von guten Schutzkonzepten immer mehr Schwierigkeiten aufwirft. Erfolgreiche Konzepte können nur bedingt aus Vorgängertechnologien übernommen werden, da gewisse Änderungen des Prozesses, wie z.B. der Umstieg von LOCOS auf ‚shallow trench insulation‘, oftmals ganze Schutzelementklassen grundsätzlich nicht mehr verwendbar machen.

Der in der Vergangenheit beschrittene Weg bestand aus dem Entwurf einer hohen Anzahl von Teststrukturen, die in den ersten Prozessläufen produziert und mittels einer detaillierten elektrischen Charakterisierung analysiert wurden. Dies erfordert jedoch einen sehr großen, weiter zunehmenden Aufwand bezüglich Fläche auf den Maskensätzen, Meßressourcen und Entwicklungszeit. Die rapide ansteigenden Maskenkosten und die geforderte schnelle Bereitstellung der ESD Schutzregeln machen diesen Ansatz in Zukunft nur noch bedingt durchführbar.

Bei Infineon wurde in den letzten Jahren ein Simulationsverfahren entwickelt, das bereits in einer Phase vor der Messung der ersten Teststrukturen ermöglicht, das Verhalten von ESD Schutzelementen während der ESD Entladung mittels Device Simulation zu ermitteln. Da während einer ESD Entladung in den betroffenen Bauelementen sehr hohe Stromdichten und Temperaturen bis zum Schmelzpunkt von Silizium auftreten, mußten dazu neue Modelle entwickelt werden und die entsprechenden Parameter geeicht werden [1]. Diese Anstrengung wurde wesentlich durch die Entwicklung eines physikalischen Analyseverfahrens unterstützt, das ermöglicht während eines ESD Pulses (mit einer typischen Dauer von 150 ns) die Temperaturverteilung und -entwicklung im Bauelement aufzunehmen. Dieses als ‚Backside Laser Interferometry‘ (BLI) bezeichnete Verfahren bestimmt interferometrisch die optische Weglänge durch einen von der Rückseite des Wafers oder Siliziumplättchens einfallenden Laserstrahl [2]. Durch die Änderung des Brechungsindex mit der Temperatur kann somit ein Temperaturintegral in der durchleuchteten Siliziumstrecke ermittelt werden. Durch die Kombination dieser Meßergebnisse mit den Möglichkeiten der Device Simulation konnten komplexe 2D und 3D Effekte während ESD Entladungen aufgedeckt werden. In den modernen CMOS Technologien von Infineon wurde die Dimensionierung der wesentlichen Designparameter für ESD Schutzelemente bereits erfolgreich auf Basis der ESD Device Simulation durchgeführt.

Für den Entwurf eines ESD Schutzkonzeptes ist jedoch nicht nur die Dimensionierung des ESD Schutzelementes von Bedeutung, sondern auch die Betrachtung der möglichen parasitären Pfade über die sonstige Schaltung und des Weges über die Metallbusse. Um dies durch eine Simulation zu erfassen, ist eine Kompaktsimulation erforderlich. Die spezifischen ESD Bedingungen erfordern die Einbeziehung von z.B. thermischen Modellen, die in den üblichen BSIM Modellen nicht enthalten sind. Mit derart modifizierten Modellen lassen sich heute mit kommerziellen Analogsimulatoren wie SABER Schaltungen bis zu einigen Dutzend Transistoren unter ESD Belastung nachstellen [3]. Letztlich bleibt es jedoch eine bis jetzt noch ungelöste Herausforderung, das Verhalten des gesamten ICs während einer ESD Entladungen zu simulieren. Dies erfordert eine Vereinfachung der extremen Anzahl von möglichen Pfaden, die der ESD Puls über den Baustein nehmen kann ohne die relevanten Stellen zu übersehen. Dieser Ansatz wird bei Infineon mit dem Konzept des ‚Virtual ESD Test‘ verfolgt.

- [1] K. Esmark, W. Stadler, H. Gossner, M. Wendel, X. Guggenmos, W. Fichtner, "Advanced 2D/3D ESD Device Simulation", ESD Symp. Proc. (2000) pp. 420-429.
- [2] C. Fürböck, N. Seliger, D. Pogany, M. Litzenberger, E. Gornik, M. Stecher, H. Gossner and W. Werner; "Backside Laserprober Characterization of Thermal Effects during High Current Stress in Smart Power ESD Protection Devices", Proceedings of the International Electron Devices Meeting IEDM'98 (1998) pp. 691-694.
- [3] H. Wolf, H. Gieser, W. Stadler, K. Esmark; "ESD Circuit Simulation for the Prevention of ESD Failures — Application to Products in a 0.18 µm Technology", Proc. IRPS 2002, Dallas, pp.



ESD Schutz in Deep-Sub-Micron Technologien

Harald Goßner, Infineon Technologies

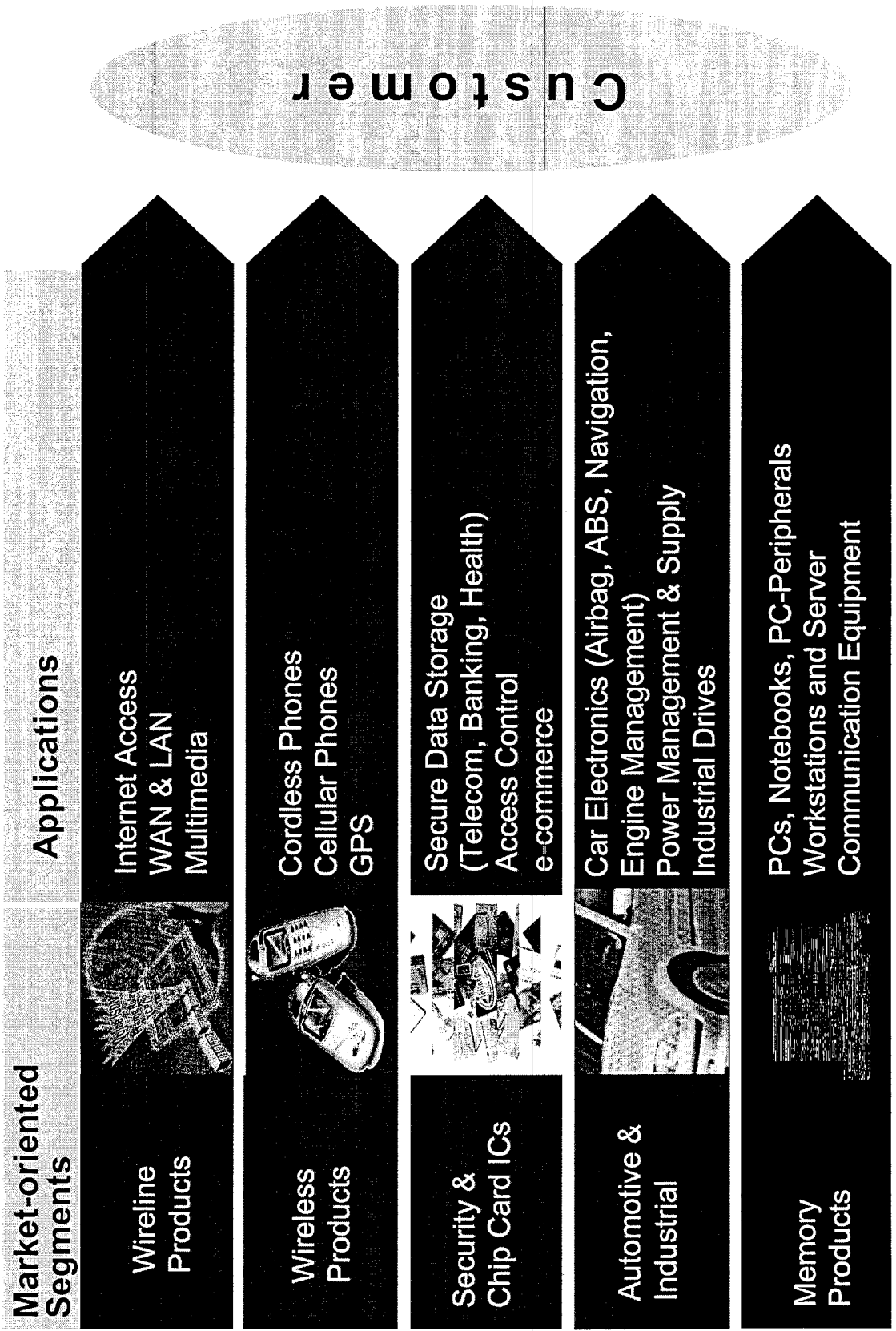
28.06.2002

Outline

- ESD & Latchup Center of Competence - Organisation
- ESD basics
- Conventional development approach - results and limitations
- ESD simulation and thermal mapping - solutions and highlights
- Outlook - Virtual ESD test
- Summary

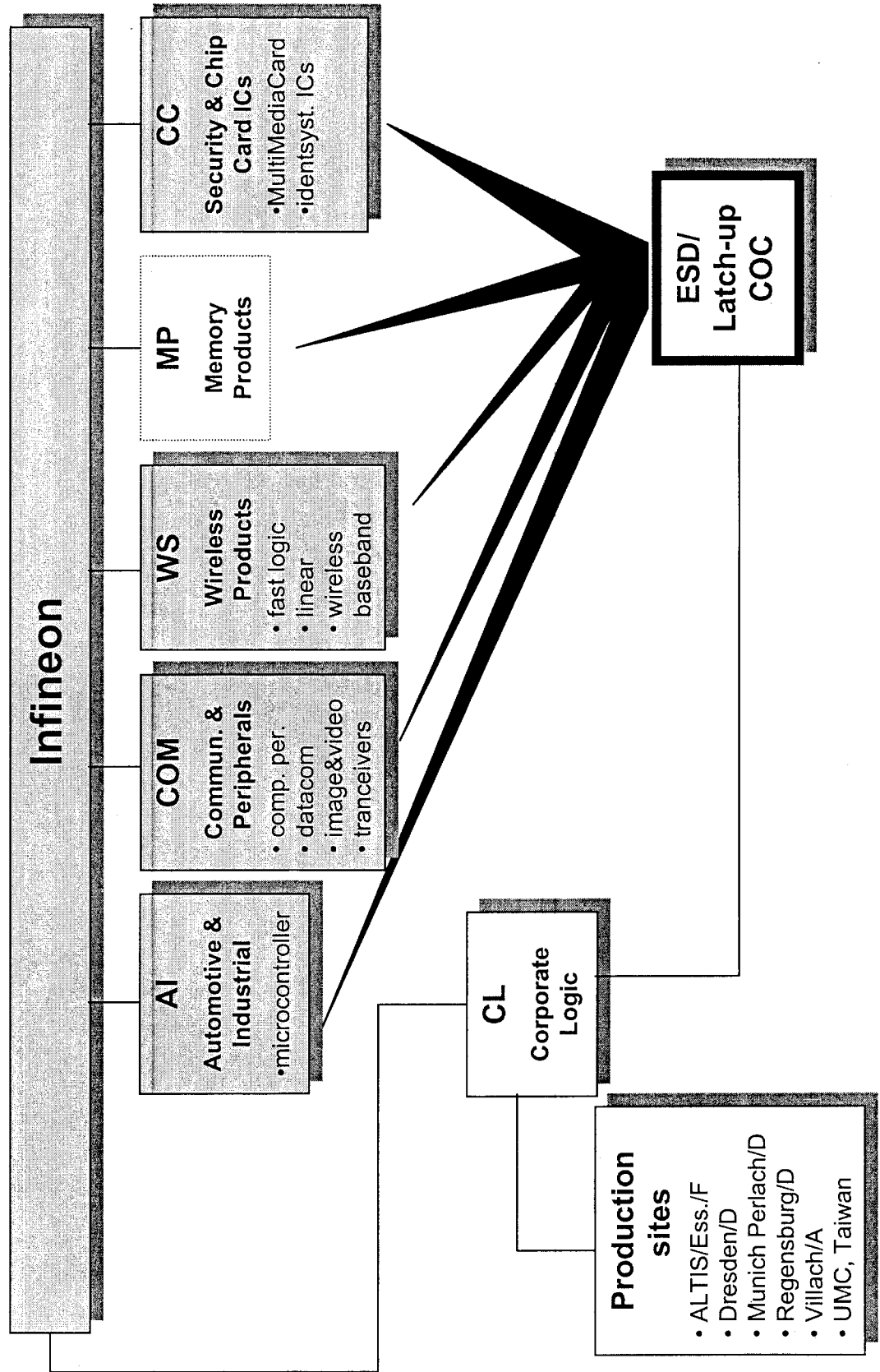


Infineon - Market-oriented Business Structure

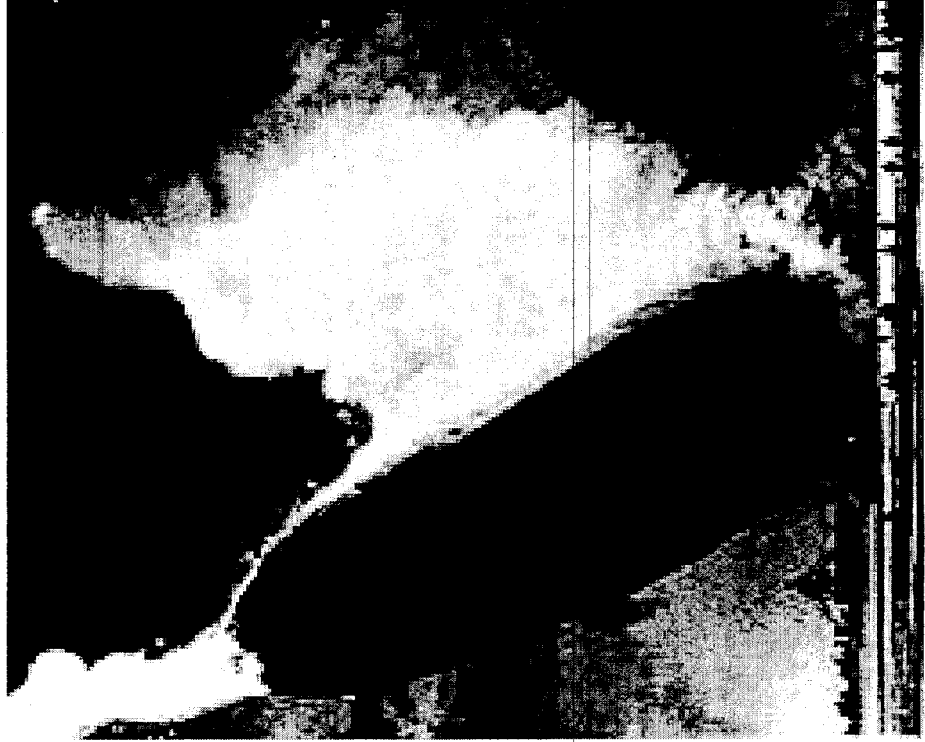


Customer

Organization of ESD/ Latch-up



What is ESD (Electrostatic Discharge)?



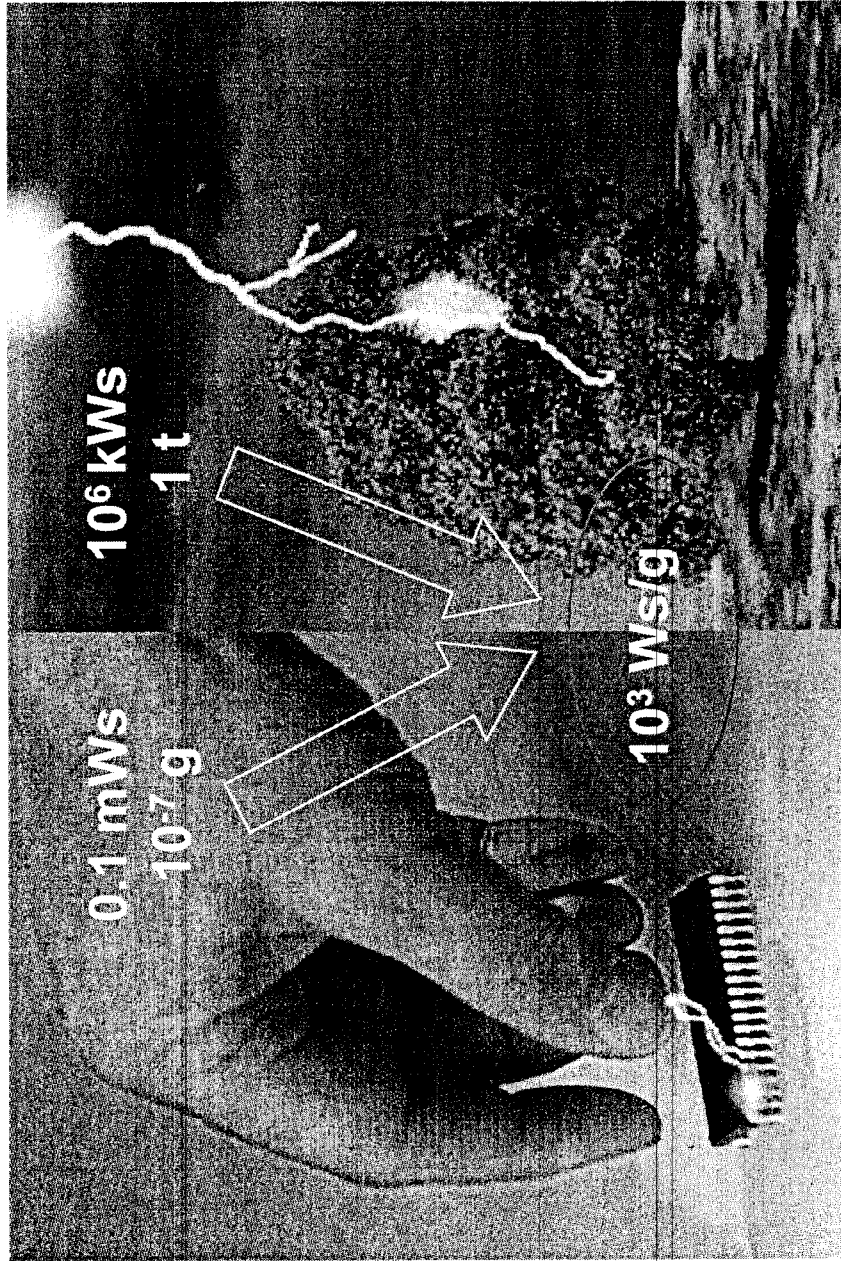
“Historic” example:
Explosion of the hydrogen
filling of the Zeppelin
“Hindenburg” during the
landing in Lake Hurst, New
Jersey (1937).
Result: known ...

Typical ESD voltages

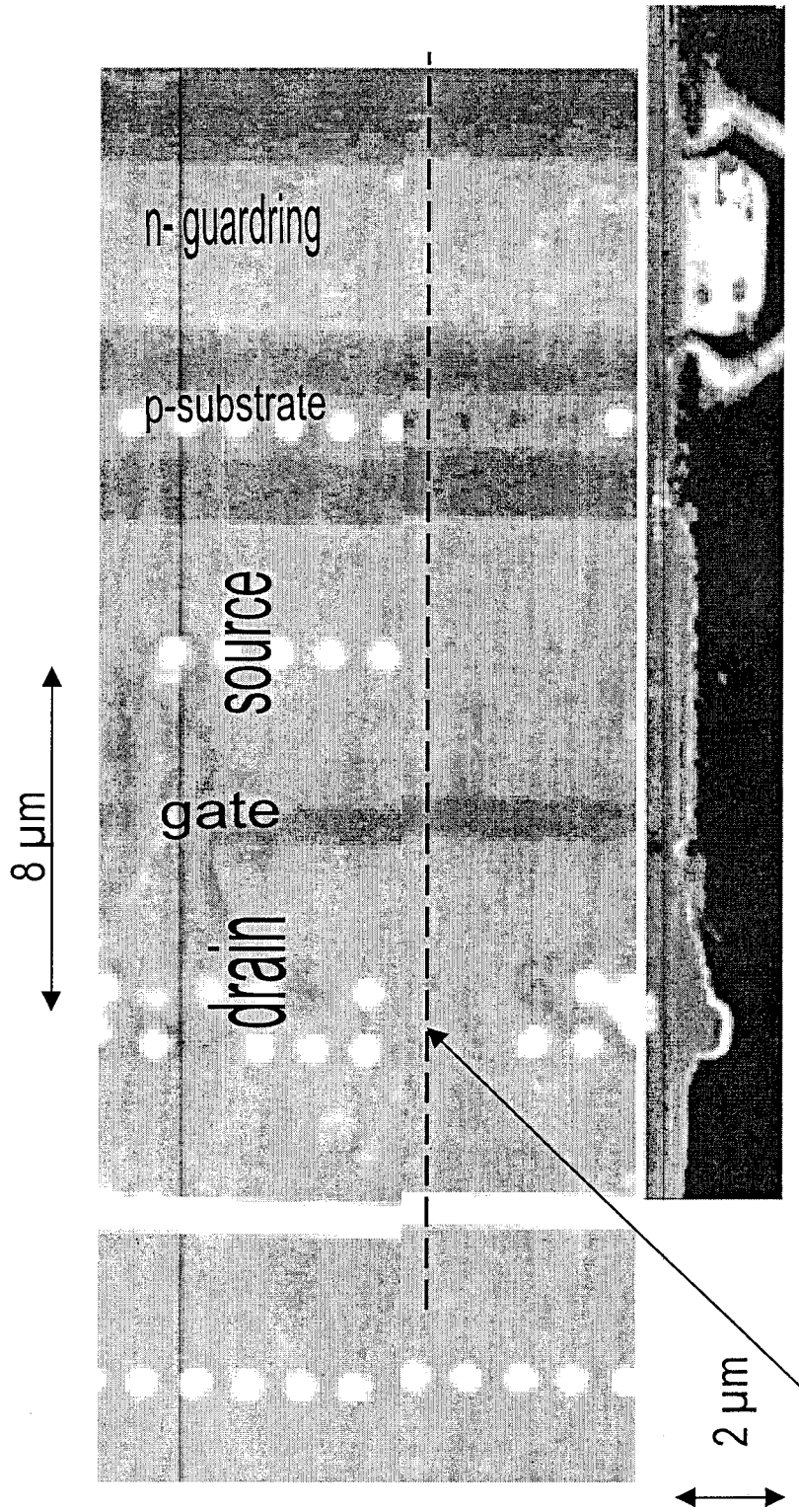
Typical electrostatic voltages / V			
Event	relative humidity		
	10%	40%	55%
Walking across a carpet	35.000	15.000	7.500
Walking across a vinyl floor	12.000	5.000	3.000
Motions of bench employee	6.000	800	400
Removing dual in-line packages (DIPs) from plastic tubes	2.000	700	400
Removing DIPs from vinyl trays	11.500	4.000	2.000
Removing DIPs from polystyrene foam	14.500	5.000	3.500
Removing bubble pack from PWBs*	26.000	20.000	7.000
Packing PWBs in foam-lined box	21.000	11.000	5.500

/Dangelmayer/

Energy Density



Look into the Silicon after an ESD event



Cutline for scanning capacitance microscopy

Technology Trends: Process Parameters

feature size (μm)	3	2	1.5	1.0	0.8	0.5	0.35	0.25	0.18	0.13	0.10
Junction depth (μm)	0.8	0.5	0.4	0.35	0.3	0.25	0.20	0.15	0.15	0.15	0.14
Oxide (nm)	50	40	30	20	15	10	7	5	3.5	2	1
LDD			X	X	X	X	X	X	X	X	X
salicide				X	X	X	X	X	X	X	X

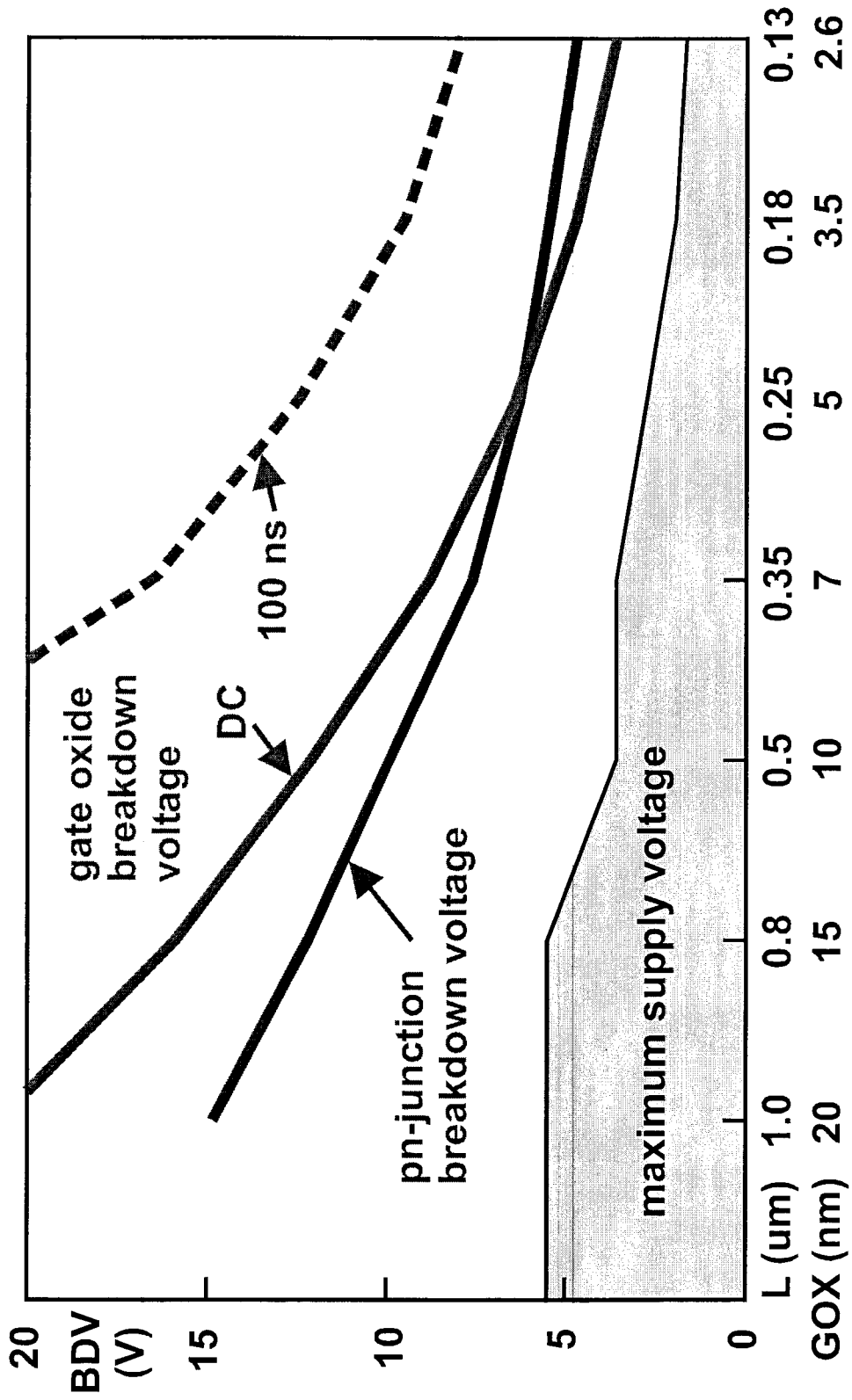
[A. Amerasekera], IFX internal data

⇒ **Reduced Gate oxide thickness**

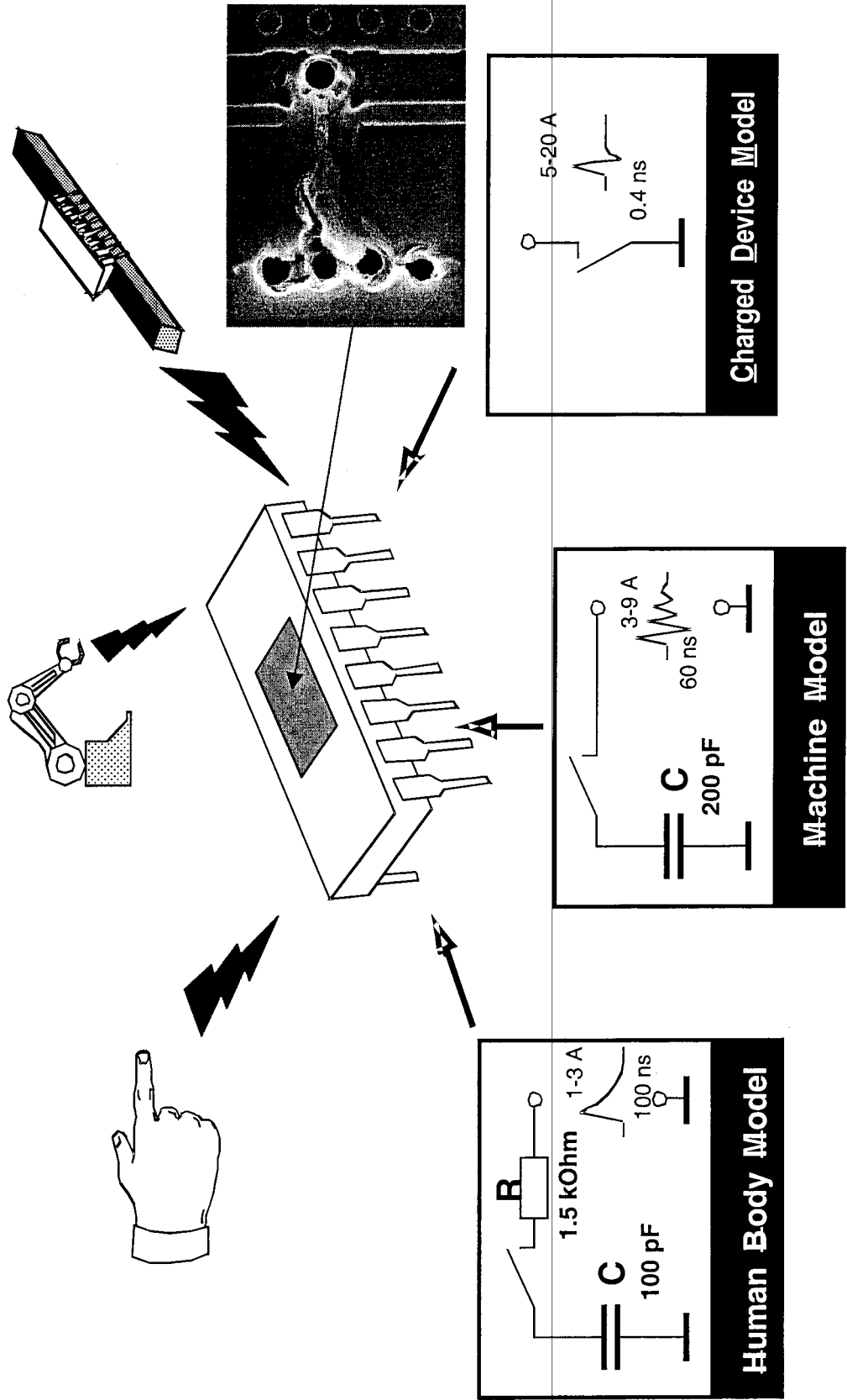
⇒ **Reduced volume for energy dissipation**

⇒ **LDD and salicide can drastically reduce the ESD robustness**

Technology Trends: Voltage Parameters



ElectroStatic Discharge - Models

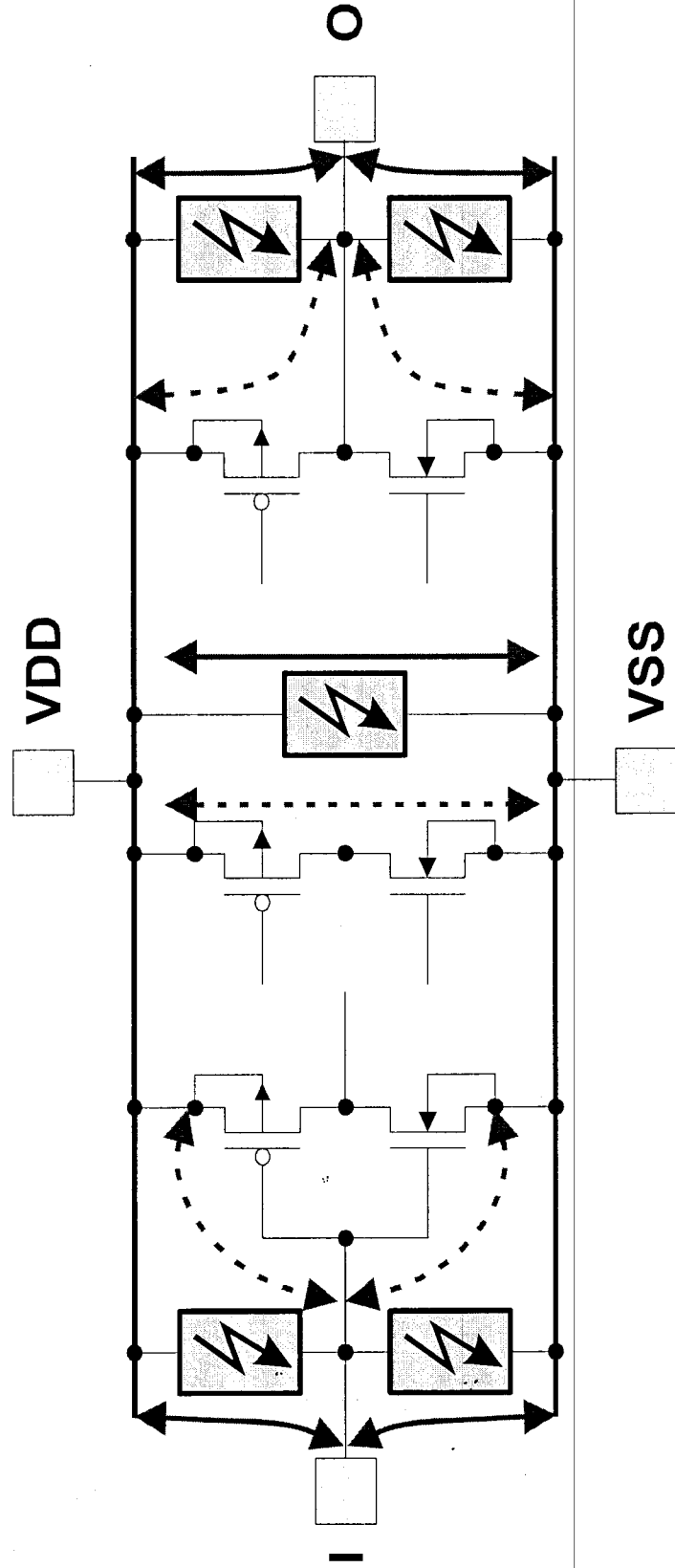


ESD Models – Pulse Parameters

Pulse Parameter	HBM	MM	CDM
Reference Voltage	2 kV	200 V	1 kV
Peak Current*	1.3 A	3.5 A	36 A
Rise time*	10 ns	10 ns	250 ps
Pulse duration*	150 ns	150 ns	700 ps

* HBM, MM: short circuit measurement; CDM: 30 pF test modul

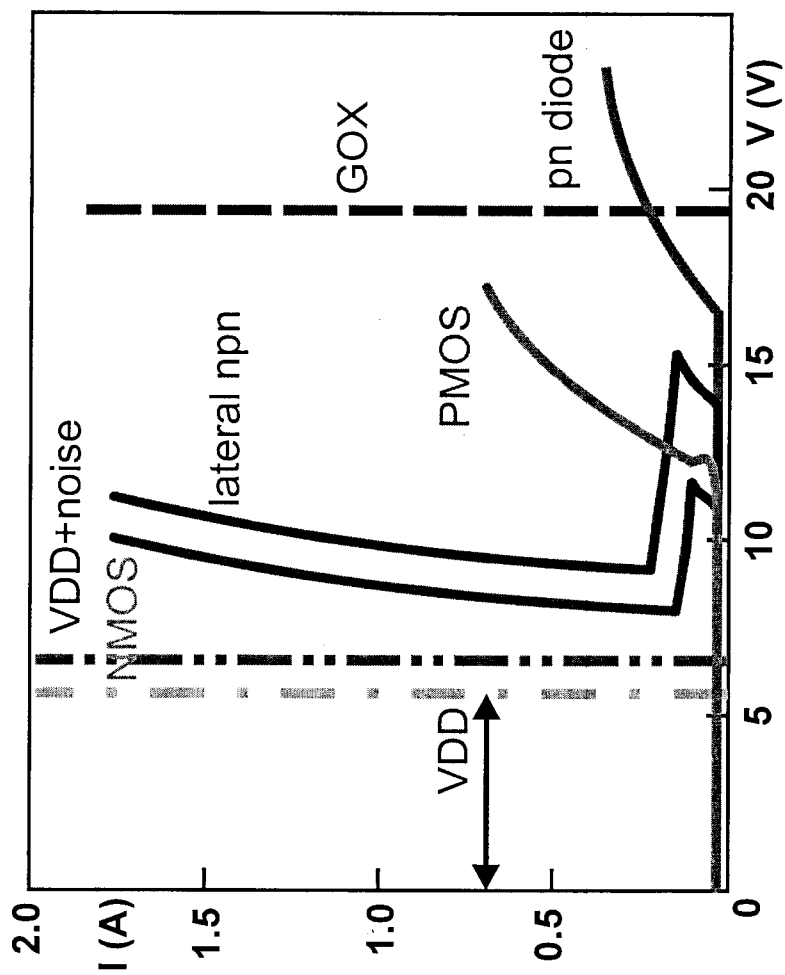
ESD Protection Concept: Discharge Paths in a Protected Circuit



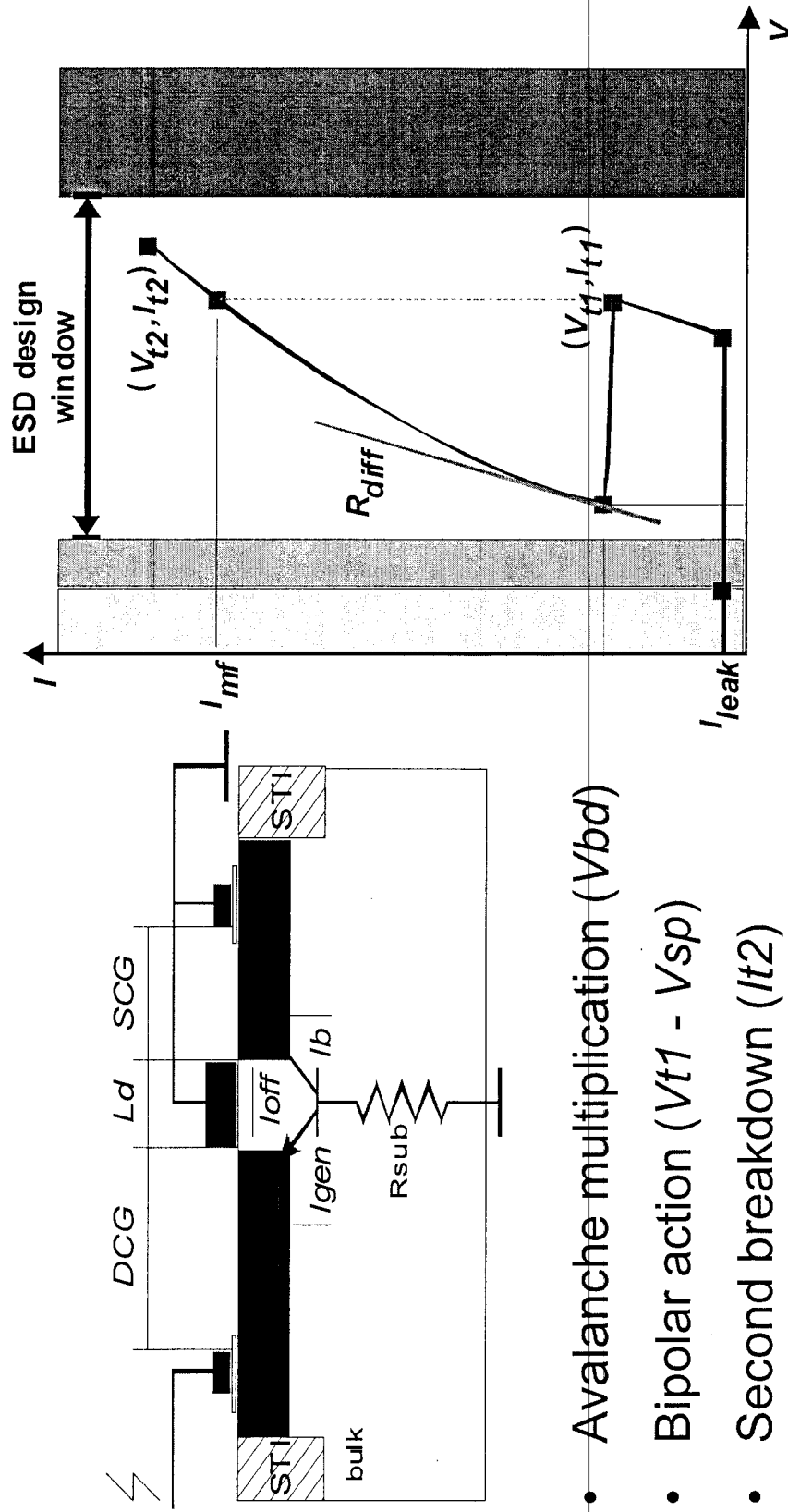
input stage internal circuit driver stage

Standard goal: 2 kV HBM and 500 V CDM

Margins for Protection Elements



A Basic Protection Device: NFET



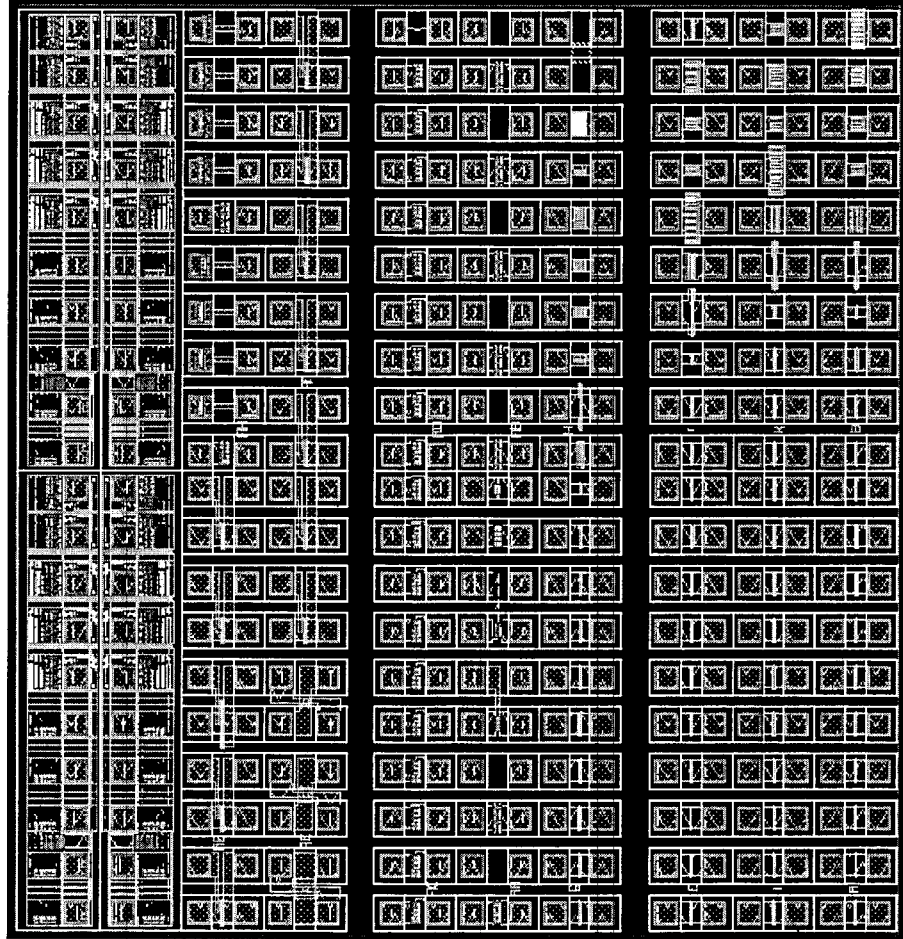
- Avalanche multiplication (V_{bd})
- Bipolar action ($V_{t1} - V_{sp}$)
- Second breakdown (I_{t2})
→ damage

General requirements for ESD protection

- Circuit requirements:
 - Signal voltage range
 - Leakage current
 - Maximum serial resistance, capacitance
- ESD requirements:
 - ESD robustness HBM, CDM
 - Voltage clamping
- Others:
 - Stability w.r.t. process changes
 - Transferability
- Possible elements:
 - NFET, PFET, bipolar, diodes, SCRs, ...

**The ESD protection concept must be optimized
for each technology!**

Empirical Approach: Test-Chip Strategy



- Based on experience from predecessor technology
- Up to 700 test structures
 - active elements (n drivers, p drivers)
 - possible ESD protection elements
 - parasitics (wiring, breakdown structures)
 - best-guess ESD library

Measurement Technique: Transmission Line Pulsing

I (A)

0.3

It2, Vt2

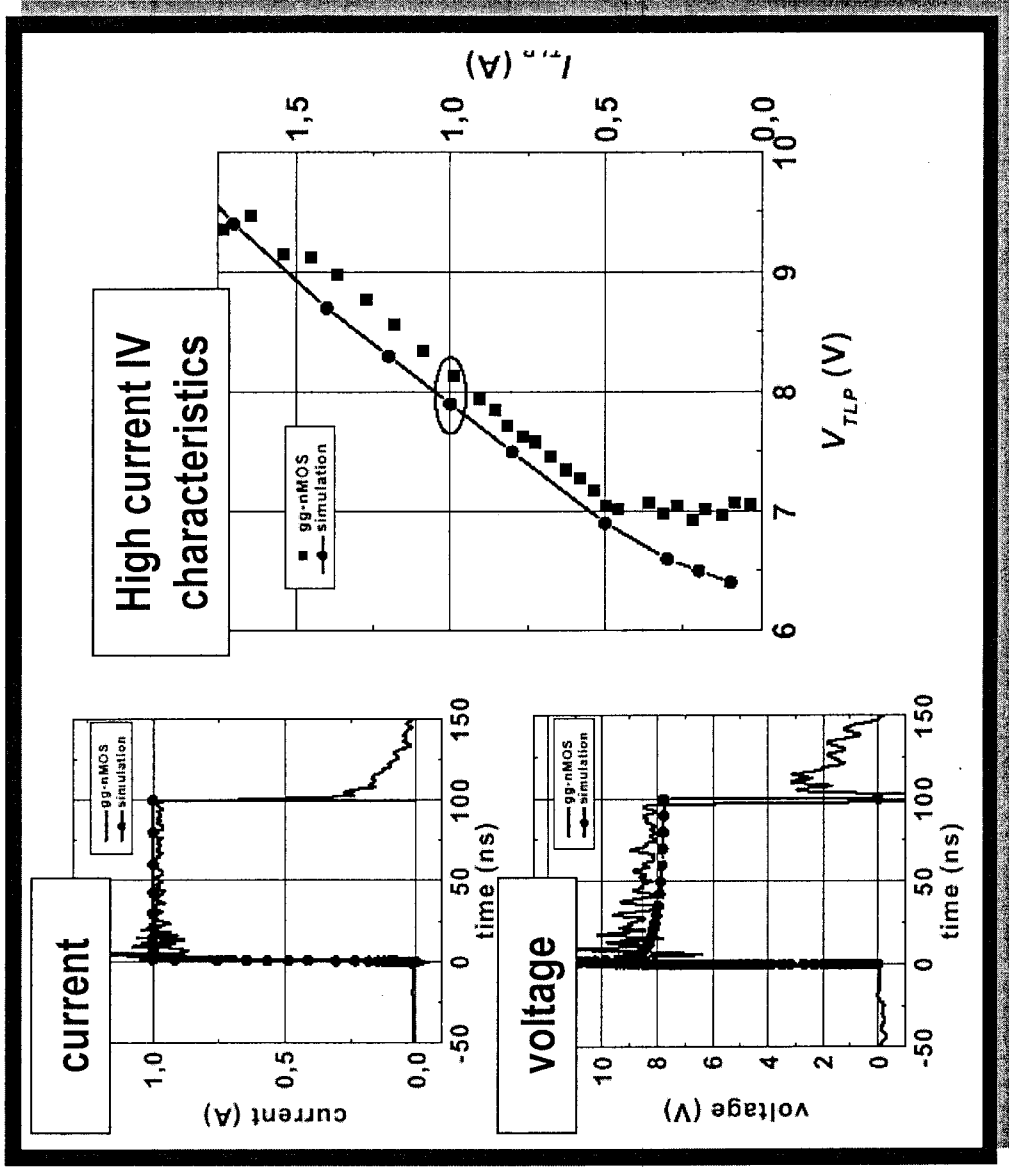
Vsp

It1, Vt1

Vsig

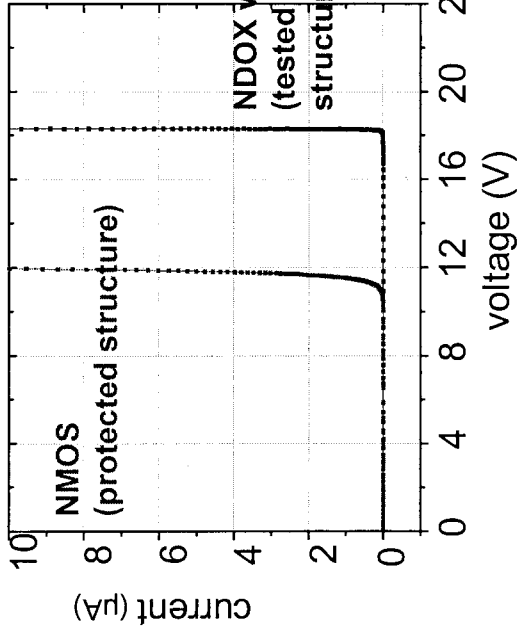
Vbd V (V)

0

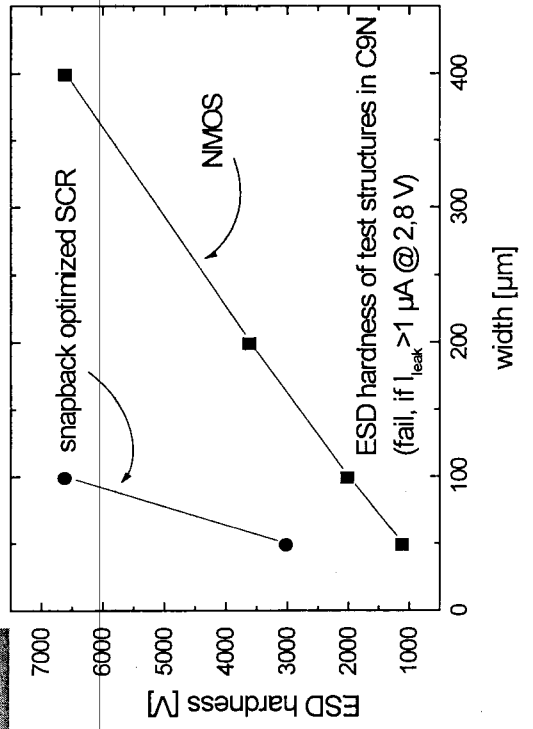


Choice of protection element

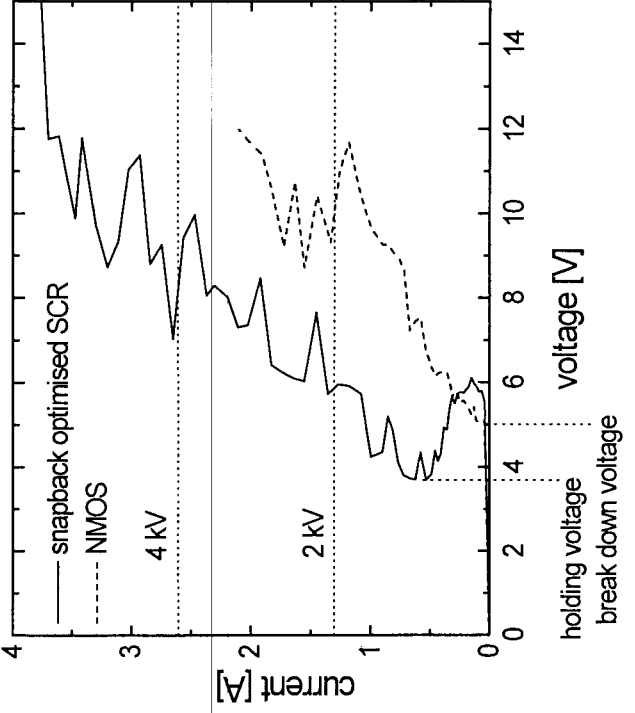
DC:



HBM:



TLP:



Empirical Approach: Conclusion

- + Experience and good new ideas lead to effective protection elements
- + Verification of selected protection elements and pad cells necessary in any case
- Pre-silicon guidelines contain great risks
 - high redesign efforts
 - concurrent engineering hindered
- Process influence hard to consider (expensive silicon splits required)
- Many elements with many geometry variations have to be evaluated
- Analysis methodology complicated and partly not sufficient for deeper understanding

ESD Device Simulation: Goals

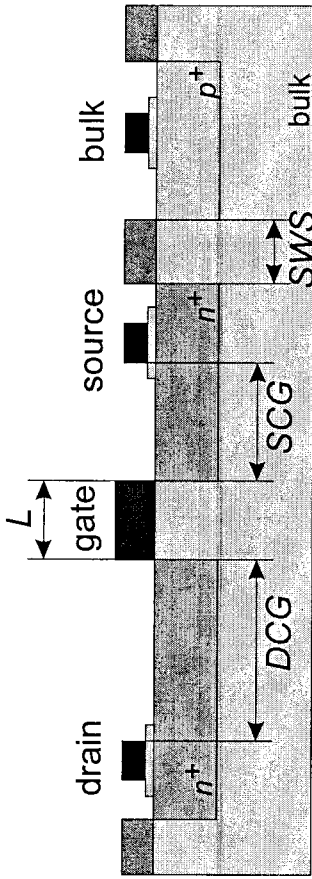
- **Optimization without silicon would be very beneficial**
 - shorten development time (concurrent engineering)
 - save costs (silicon, measurement time)

however:

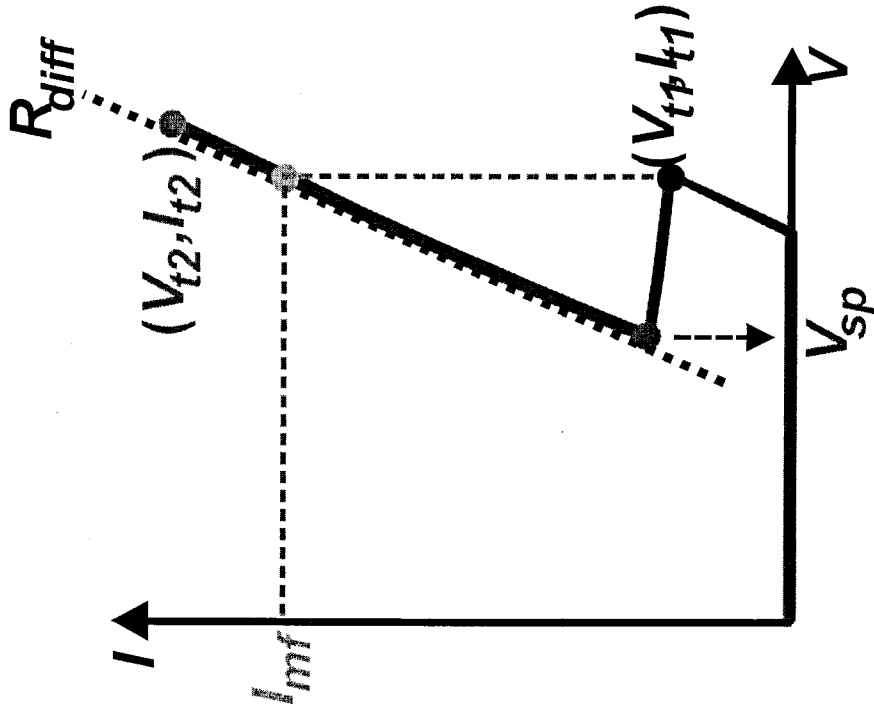
“Extreme high current densities & temperatures”

- Problems which had to be solved:
 - missing physical models
 - 2D simulation insufficient - 3D not available
 - missing numerical methodology
 - immense calculation time
 - thermal boundary conditions

Parameters requested from ESD simulation

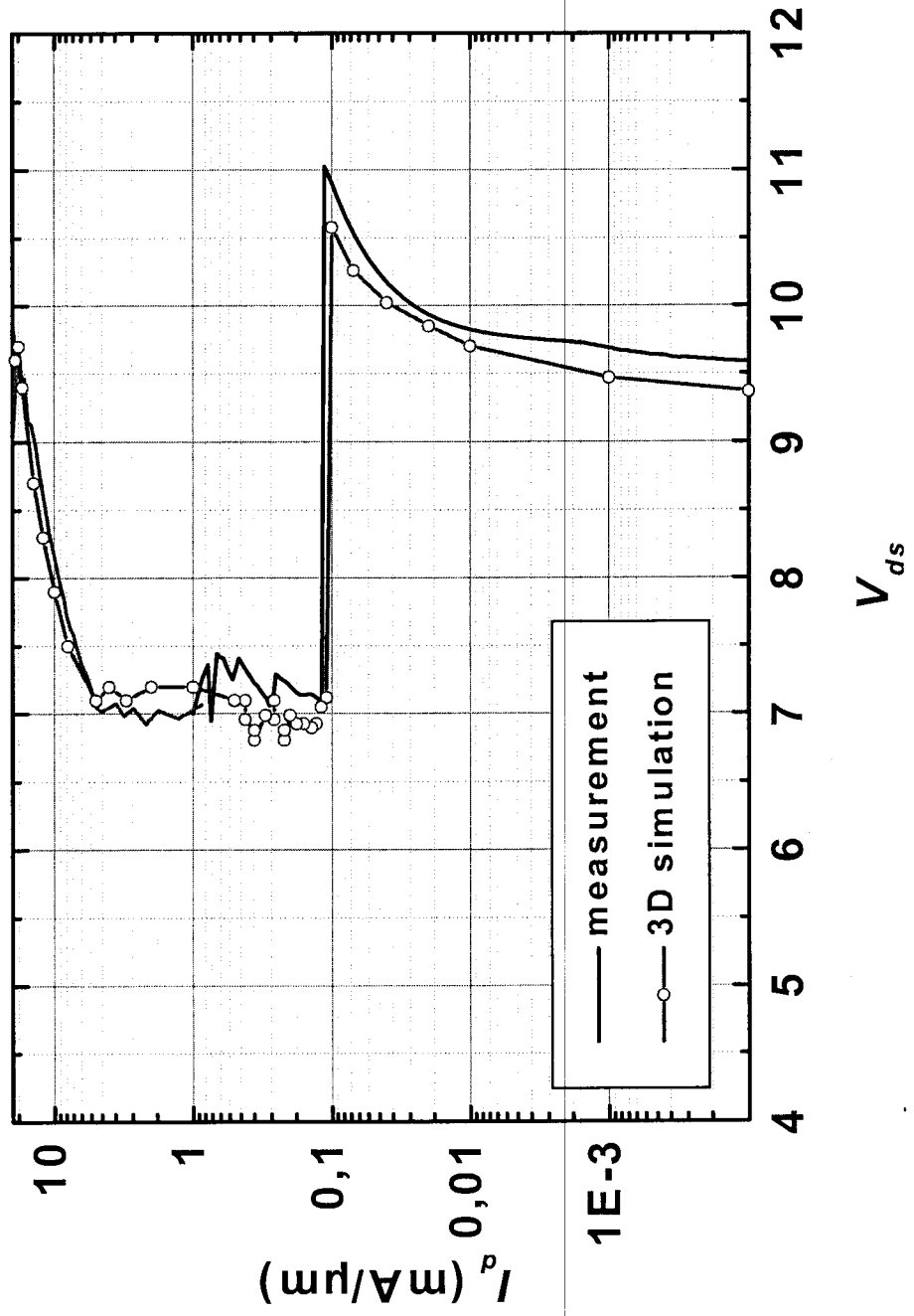


gg-nMOS



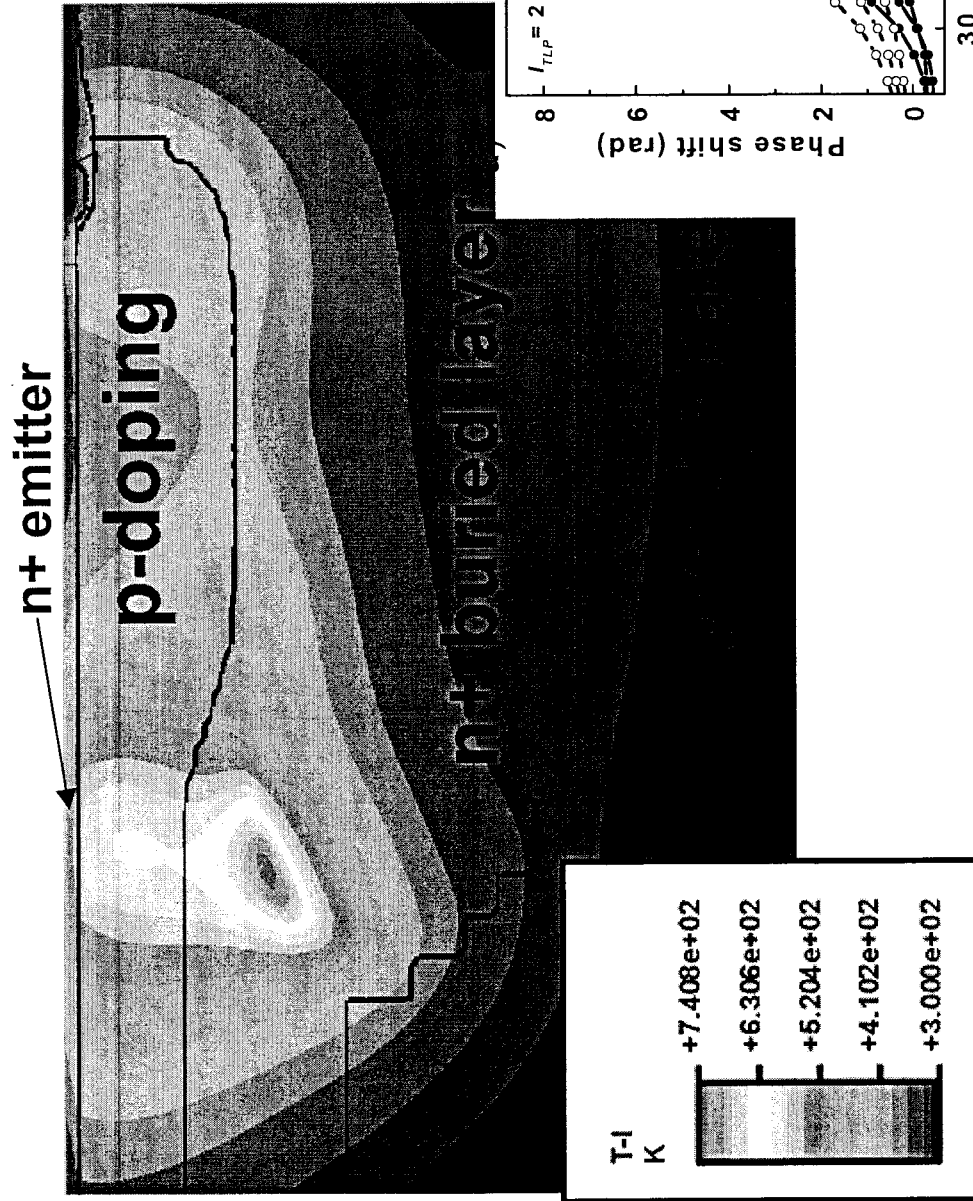
I_{t1}, V_{t1}	trigger current/voltage
V_{sp}	snapback voltage
R_{diff}	high current resistance
I_{mf}	trigger condition for multifinger structures
I_{t2}, V_{t2}	current/voltage to fail.

Results of ESD simulation - IV of 0.35 μm ggNMOS

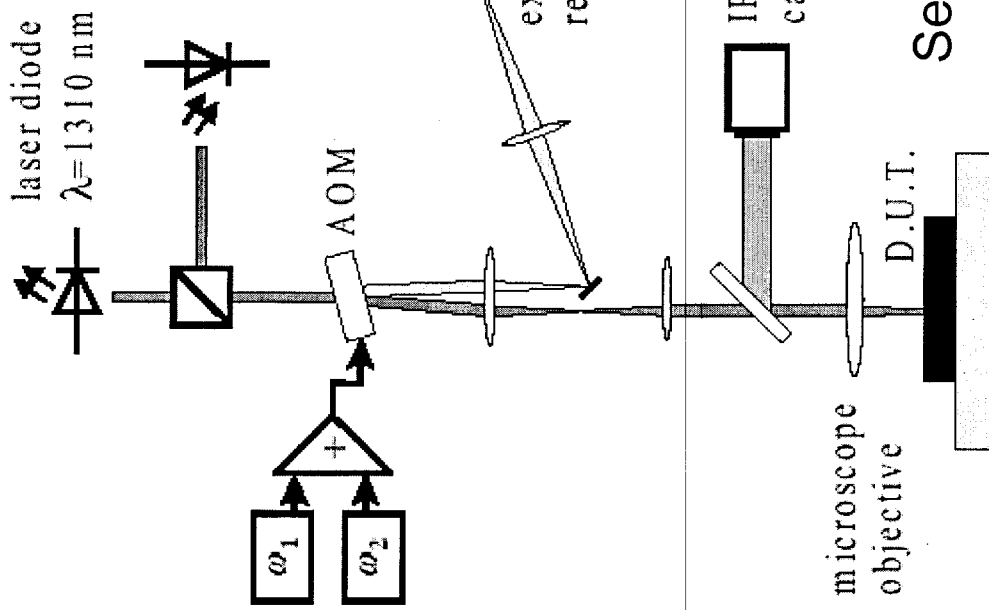


Temperature profile by ESD device simulation

Temperature profile of a vertical npn transistor during ESD



Backside thermal mapping



Spatial resolution: $\sim 1 \mu\text{m}$

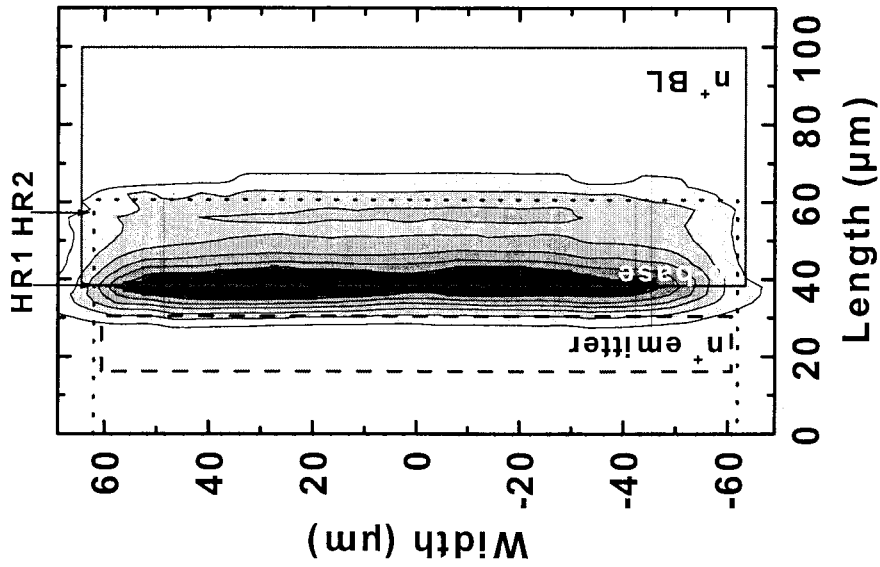
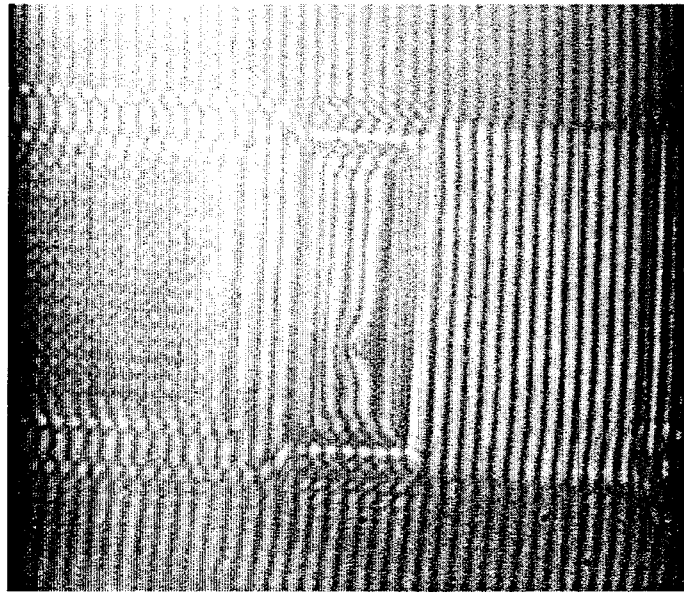
time resolution: $< 10 \text{ ns}$

temperature resolution: $\sim 1 \text{ K}$

detected power dissipation: $\sim 10 \text{ mW}$

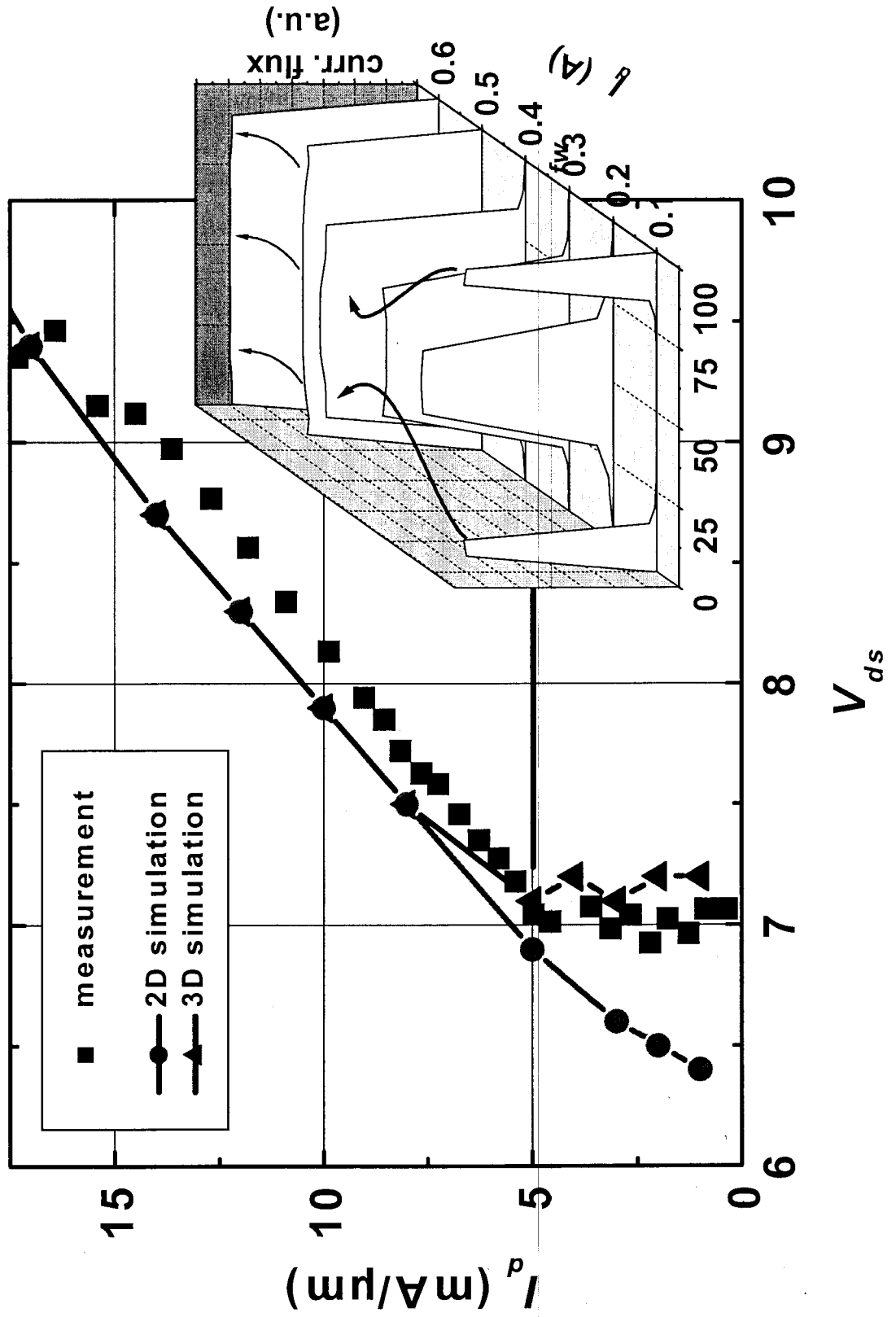
Set-up as developed at the TU Wien

2D transient temperature analysis by backside probing • (as developed by TU Wien)

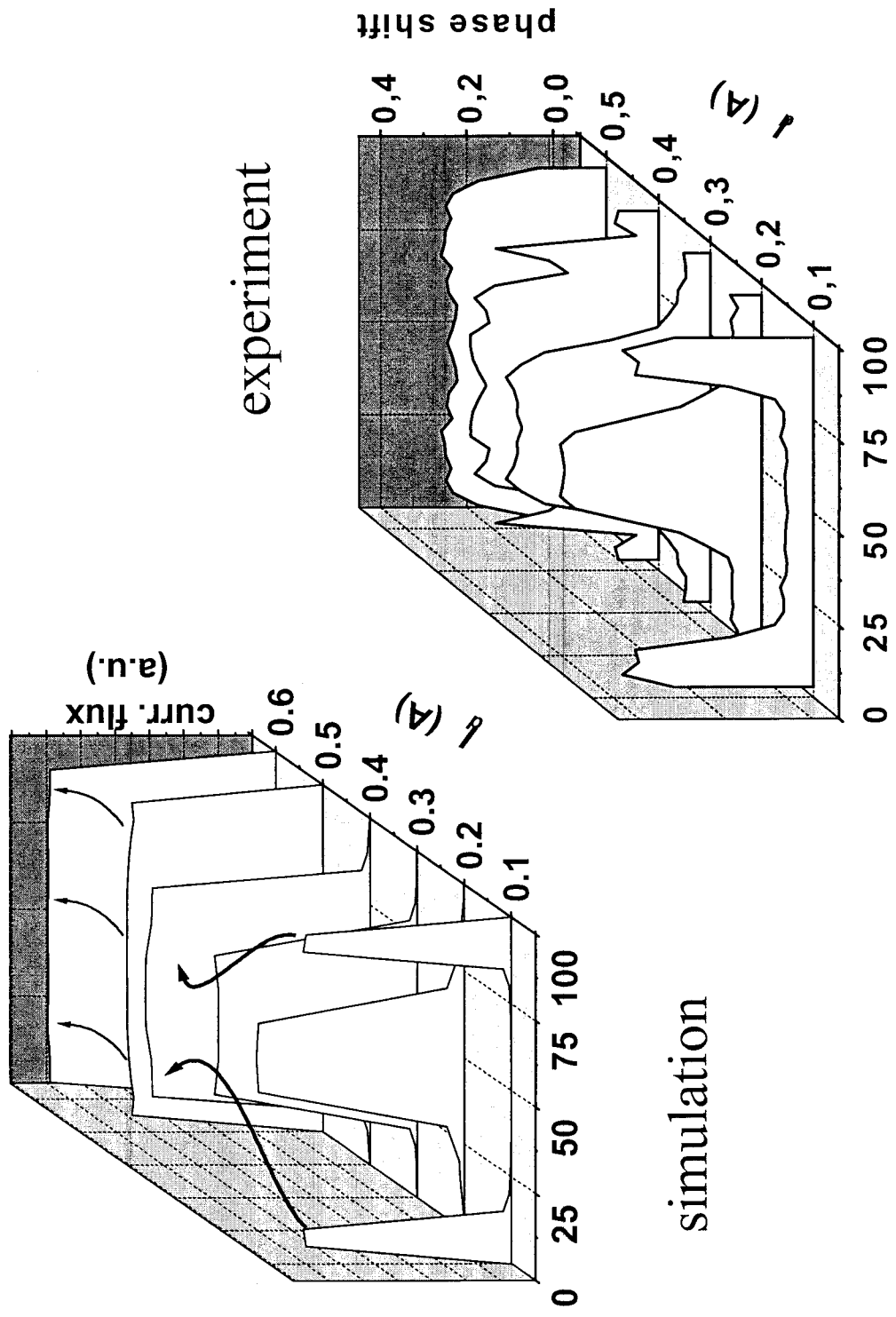


D. Pogany, M. Litzenberger,
 E. Gornik; private communication

Simulation of the high current regime of 0.35 μm ggNMOS

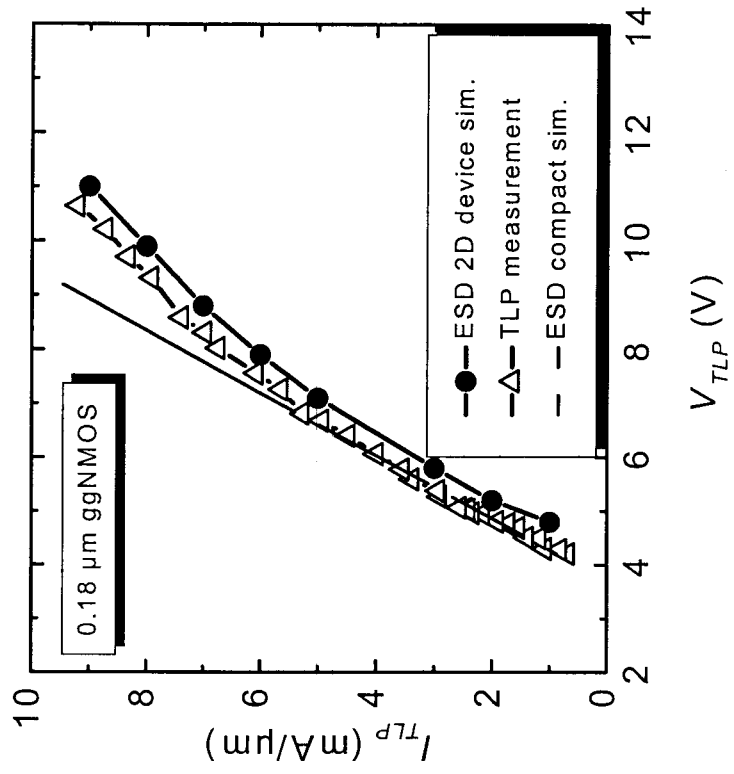
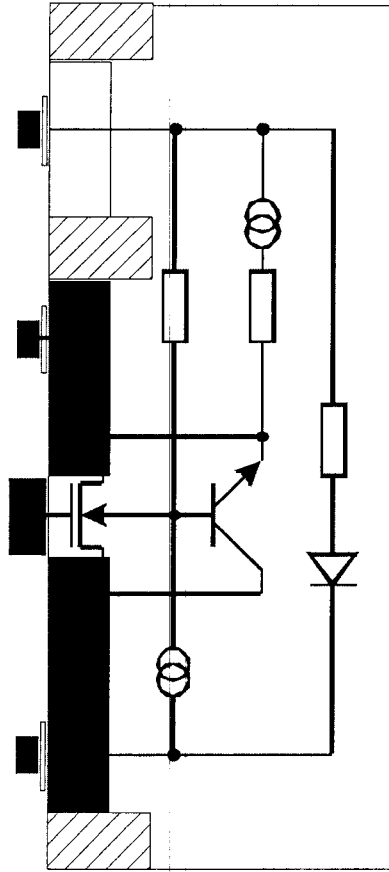


Comparison ESD device simulation vs results of backside temperature mapping



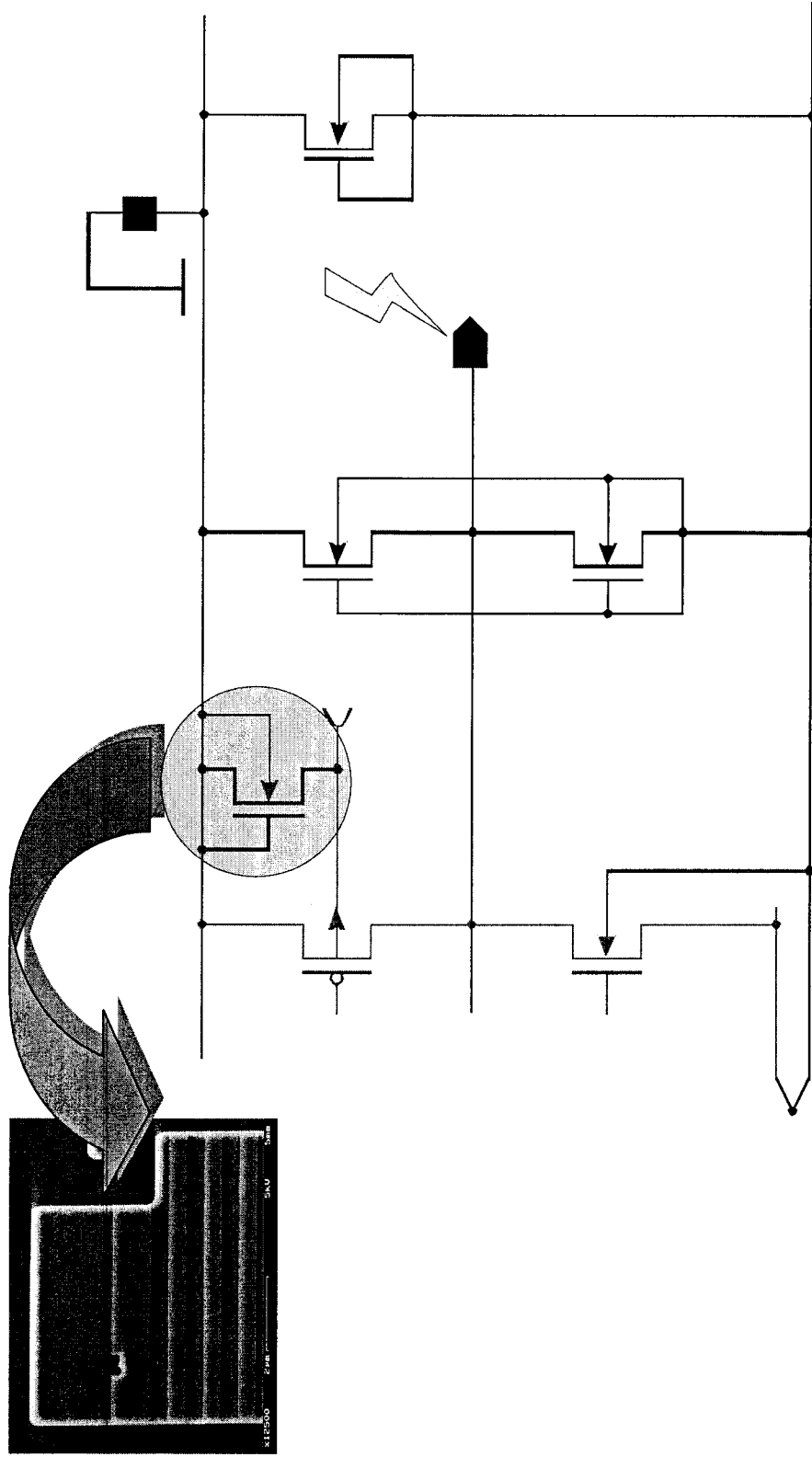
Going one step further: circuit simulation

- New complex interfaces (e. g., USB, PCI-X) require a detailed study of the interaction of protection elements and circuitry during ESD event.
- ESD device simulation is not capable to handle 50 and more devices → compact models had to be developed.



On the way to the "virtual ESD test": circuit simulation of complete IO cells

- Example: over-voltage tolerant circuits



Summary

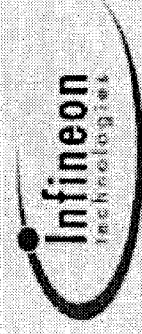
- ESD is rated among the top 3 problems of semiconductor design in the future
 - Each new technology requires a new evaluation of the ESD protection concepts
 - The accelerated 'time to market' is a big challenge for the ESD protection development
 - Infineon's answer to this challenge is
 - a well-established, unique ESD simulation flow embedded in Infineon's ESD protection development methodology
- ESD STAR**
- the development of advanced physical analysis methods for silicon based verification (backside temperature analysis)
 - 'Virtual ESD test' of product is the way to **'First time right' of product**

XXVIII. Workshop der MPC Gruppe

28. Juni 2002

Digitale Designmethodik
für die System-on-Chip Integration
in Deep-sub-Micron Technologien

Dr. Knut M. Just
Infineon Technologies AG
Wireless Solutions



Never stop thinking.

Inhalt

- Kritische Designaspekte: Ziele und Herausforderungen
- Entwicklung der Strukturbreiten in der HL-Technologie
- Trends in der Kommunikationstechnik zur Miniaturisierung
- Zeitleiste der GSM Hochintegration
- Logikentwicklung für Systemkomplexität
 - (Alter) Traditioneller Synthese -> Layout Ablauf
 - Platzierung ist der Schlüssel zum Timing
 - (Neuer) Physikalischer Synthese -> Layout Ablauf
- ... für Signalintegrität
- ... für geringen Leistungsverbrauch
 - Maßnahmen zur Reduktion des Leckstroms
 - Auswirkungen auf die Design-Tools und den Ablauf

Kritische Designaspekte

- System-on-Chip Design: Ziele
 - **Funktionalität** richtig
 - Formale Verifikation mit CVE (Siemens), Verisity ? (Bernhard fragen)
 - Simulation kritischer Abläufe; jedoch nicht komplett möglich (Falsifizierung)
 - **Timing Anforderung** erfüllt
 - Backannotation der Verzögerungszeiten in STA und Simulation
 - **Zuverlässigkeit** der Funktion unter allen Bedingungen
 - Zeitliche Prüfung unter Worst-Case (Spannung, Temperatur, Prozessvariation)
- Herausforderungen
 - **Systemkomplexität** bei Verwendung vorgegebener Blöcke
 - **Signalintegrität** der physikalischen Verbindungen
 - Spannungsabfall über den Versorgungsleitungen
 - Übersprechen z.B. in Bussen
 - **Leistungsverbrauch** der Geräte mit zunehmend portabler Verwendung

Entwicklung der Strukturbreiten in der HL-Technologie

- International Technology Roadmap for Semiconductors: ITRS-2001

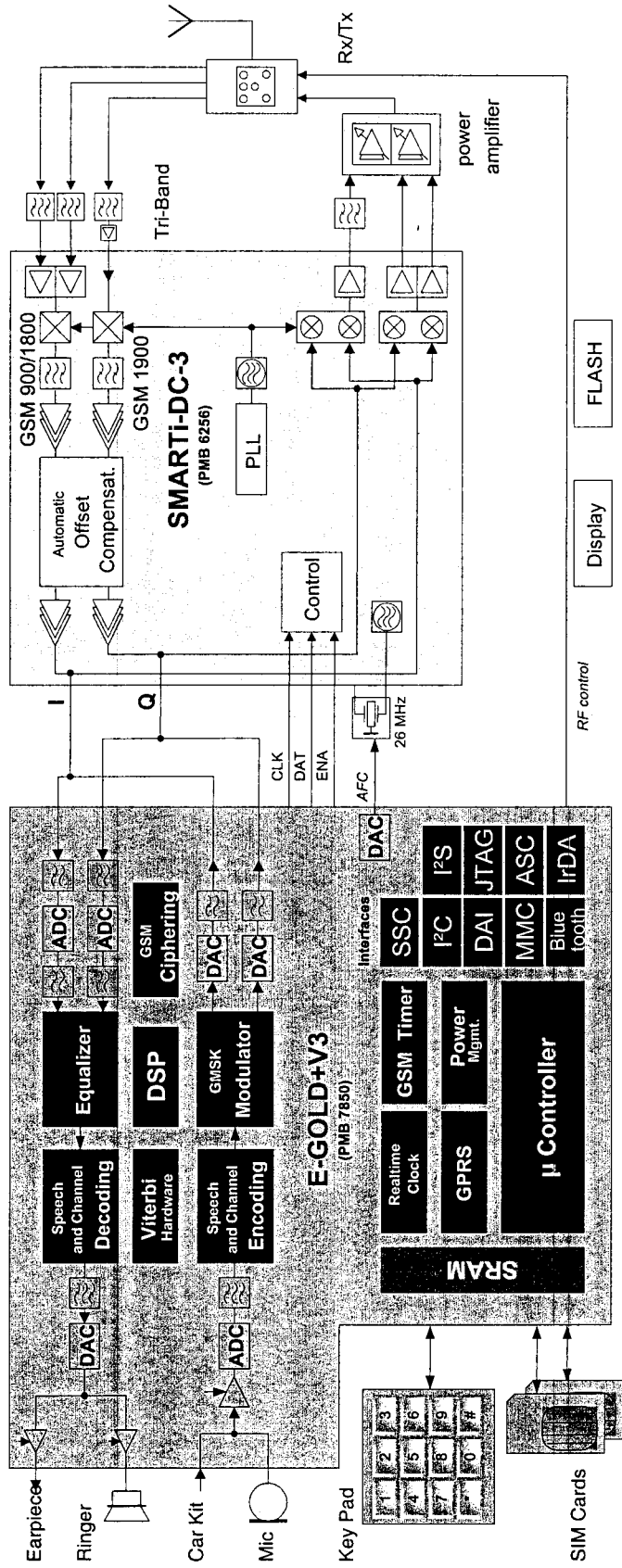
Produktionsjahr	2001	2004	2007	2010	2013	2016
DRAM 1/2 Pitch (nm)	130	90	65	45	32	22
DRAM Generation (bit)	512M	1G	4G	8G	32G	64G
Gbit / cm ²	0,42	1,15	2,35	4,75	14	28
Logik 1/2 Pitch	150	90	65	45	32	22
Phys. Gate-Länge	65	37	25	18	13	9

- DRAM 1/2 Pitch = 1/2 Abstand zweier Metallbahnen
- Tabelle bestätigt das Moore'sche Gesetz
- Verdopplung der Marktanforderungen
 - für Funktionalität (Bits, Transistoren), Performance (MHz, MIPS)
- alle 1,5 bis 2 Jahre

Trends in der Kommunikationstechnik

- Drang hin zu Miniaturisierung
- Führt zu immer größerer Integrationsdichte
- Mobilfunkgeräte der früheren Generationen bestanden aus mehr als 1000 Bauteilen
- Heutige Geräte kommen mit < 120 Bauteilen aus, incl. Akku
- Dabei werden Bausteine entwickelt die eine Komplexität erreichen die vergleichbar ist mit ersten Pentium Prozessoren
- Diese sind jedoch zusätzlich mit großem Speicher und analogen Komponenten ausgestattet sind
- Als konsequenter nächster Schritt werden auch die Hochfrequenz Komponenten integriert. (single Chip radio)

E-GOLD+ V3 - System Overview



E-GOLD+V3 (PMB 7850) and **SMARTI-DC-3 (PMB 6256)**

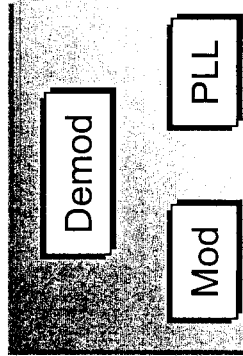
E-POWER (PMB 6810)

- Digital Parts
- Memories
- Analog Parts
- RF
- Power Supply

Zeitleiste der GSM Hochintegration

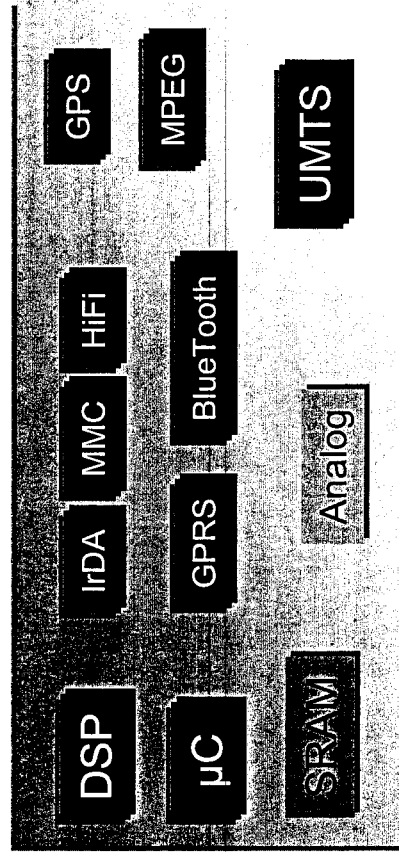


1. Muster	1995	1997	1998	2000	2002	2004
Spannung [V]	2,7	2,5	2	1,8	1,5	1,2
Frequenz MHz	13	26	52	78	104	>140
Technologie	0,5 μ CMOS	0,35	0,25	0,18	0,13	0,09



HF

M-GOLD 2
Chip-Satz
3 Chips



Basisband

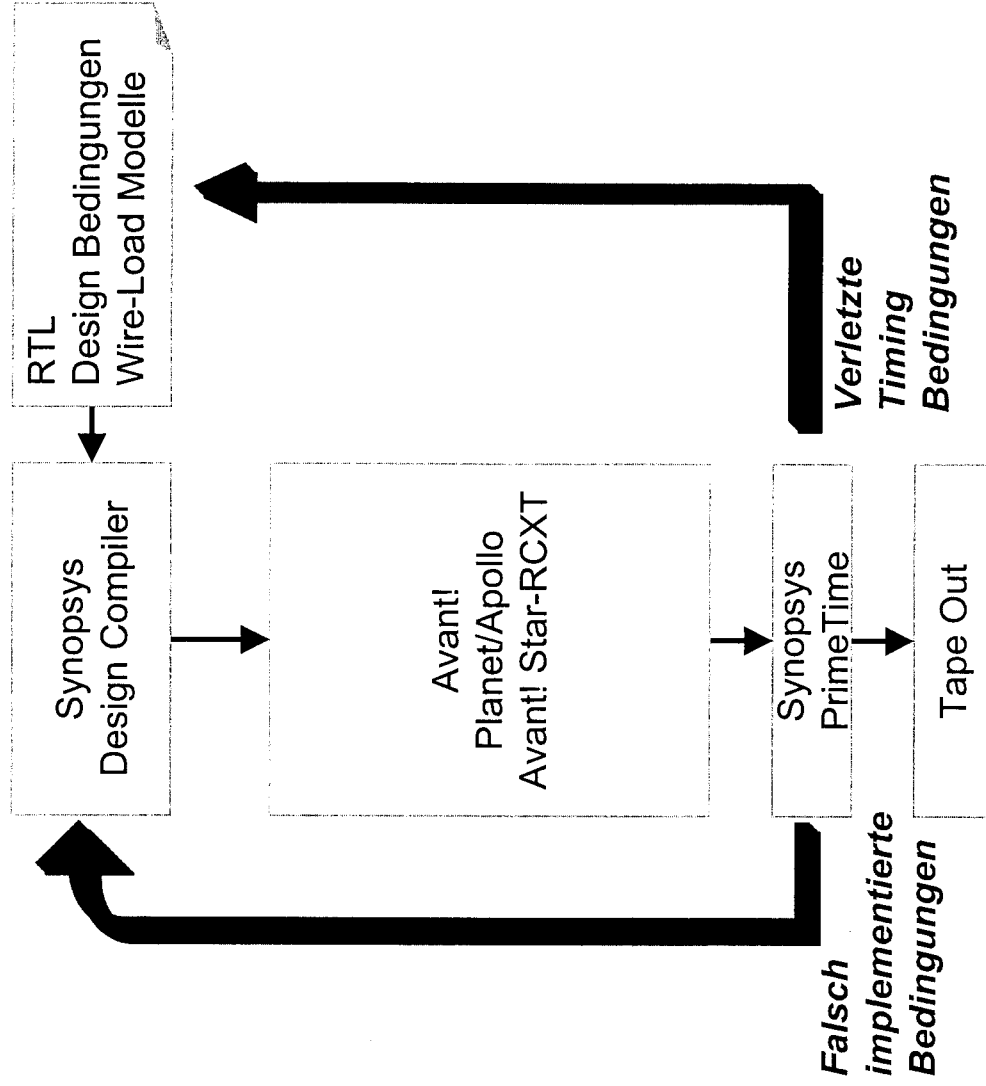
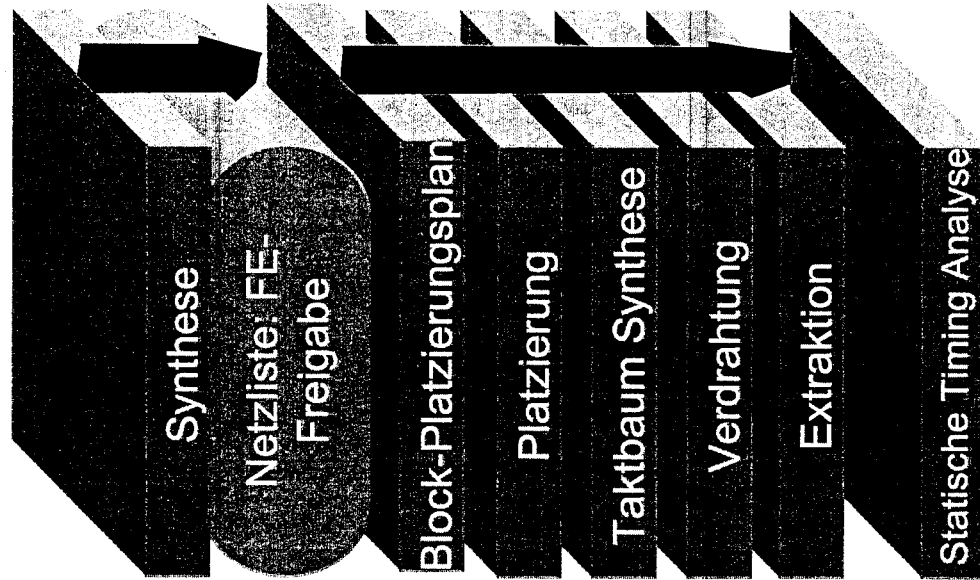
Power



Logikentwicklung für Systemkomplexität

- Minimierung der analogen Funktionalität auf AD- / DA-Wandler
- Analoge Filterung digital erledigen
- Wiederverwendung standardisierter Funktionen (Intellectual Property)
 - Kommunikation, Multimedia, Schnittstellen
 - vollständige Beschreibung für durchgängigen Designprozess nötig (Plug & Play)
- Verwendung preisgünstiger Standard-Prozessoren (μ C, DSP)
 - einfache Wiederverwendung von SW; Zukauf möglich
- optimiert auf Leistungsverbrauch in portablen Geräten

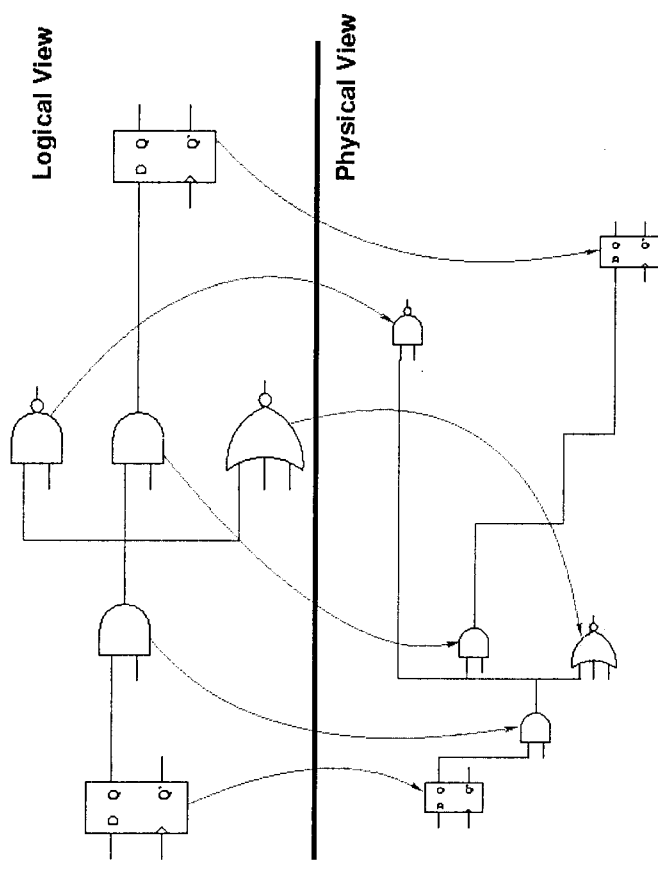
(Alter) Traditioneller Synthese -> Layout Ablauf



Platzierung ist der Schlüssel zum Timing

■ Das Netzlisten-Timing ist ungenau

- Die Verwendung von Wire-Load Modellen ergibt unrealistisches Timing
- Im traditionellen Ablauf werden Netze mit gleich vielen Senken in der Verdrahtungslaufzeit gleich behandelt
- Nach der Platzierung ist aber jedes Netz verschieden lang verdrahtet und hat seine eigene Laufzeit



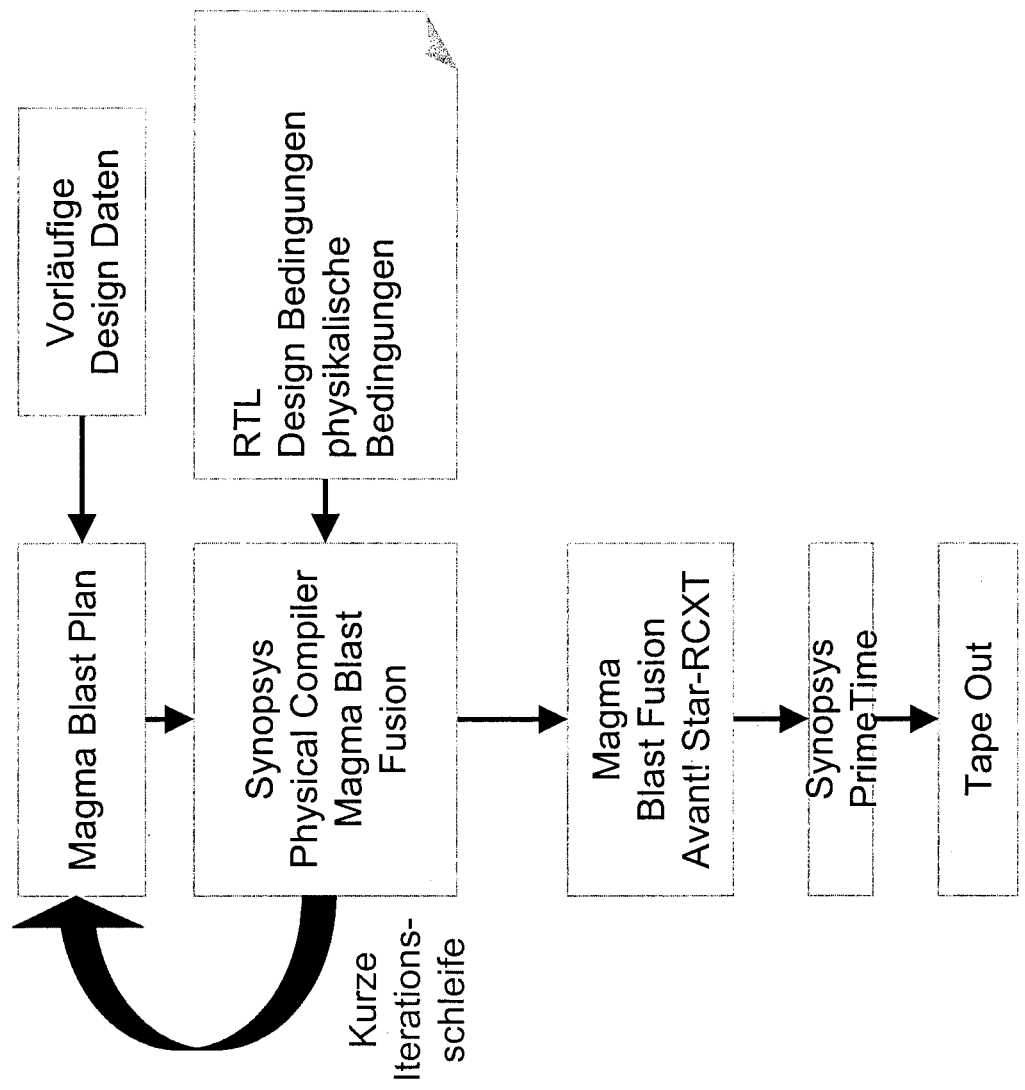
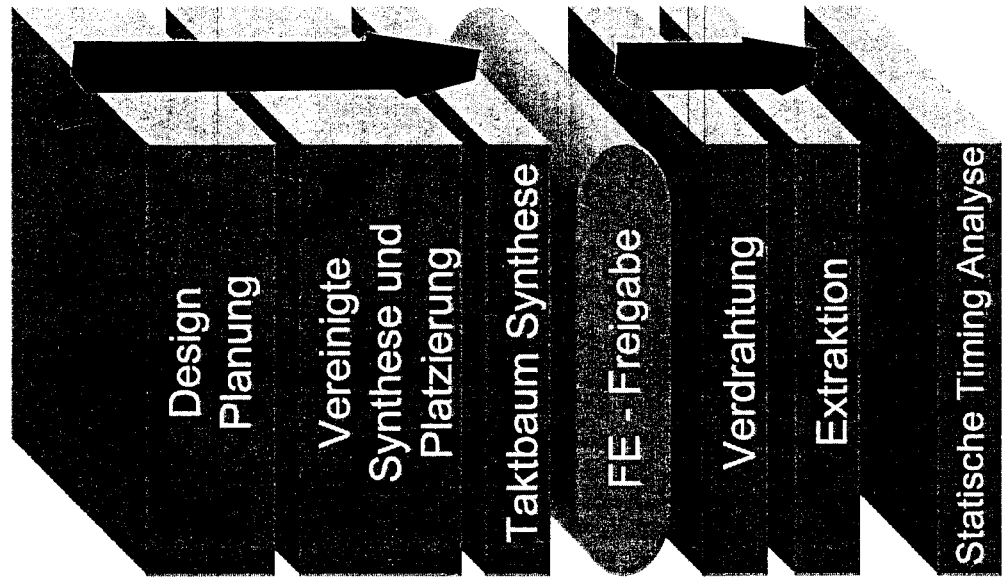
■ Ergebnis

- Timing-Abstimmung ohne Platzierungsinformation führt zu unvorhersehbarer Anzahl von Iterationsschleifen
- Timing-Abstimmung ist ein Hauptfeind der Zeitpläne heutzutage

Lösung: Synthese mit gleichzeitiger Platzierung

- Gleichzeitiges logisches und physikalisches Design
 - sieht physikalische Effekte eher
 - bestimmt Auswirkung der Verdrahtung genauer
 - bringt bessere Kontrolle der physikalischen Implementierung
 - erreicht Timing-, Flächen-, und Leistungsziele
- Bringt physikalische Schlüsselfunktionen (Block-, Zell-Platzierung) in die Logiksynthese
- FE ist verantwortlich für Platzierung und Taktbaum-Synthese
- Schiebt FE Freigabe hinter Platzierung und Taktbaum-Synthese

(Neuer) Physikalischer Synthese -> Layout Ablauf



Vorteile des physikalischen Synthese -> Layout Ablaufs


- Vorabplanung für Qualitätsvorgaben an die Synthese
- Erhöhte Sicherheit der Timing Abstimmung
- Intelligenter Erstellung der Netzliste
 - Berücksichtigung physikalischer Platzierung
 - kein 'Overconstraining' der statistischen Wire-Load Modelle
- Weniger Iterationen
- Kurze Iterationsschleife
- Verbesserte Produktivität
 - vorhersagbare BE Zykluszeit
 - 30% Reduktion der BE Zykluszeit
 - Problem der Timing-Abstimmung wird sehr früh im Designzyklus gelöst

Logikentwicklung für Signalintegrität

- Sicherheit der physikalischen Verbindungen
 - Spannungsabfall über den Versorgungsleitungen (IR-Drop) erzeugt Verzögerungen durch langsamere Logik
- Fehlerbehandlung in Bussen
 - bei Laufzeitfehlern
 - bei Übersprechen paralleler Leitbahnen
 - kann ungewollte Signalwechsel erzeugen (Glitches)
 - und gewollte Signalwechsel verschieben (Delay)
 - Im Layout Nachbarschaften wechseln
 - Fehler über redundante Codierung korrigieren mit zusätzlichen Busleitungen, Hamming Code und Decoder HW
- Fehler in Speichern (RAM)
 - Zufällige Fehler durch Strahlung (Soft Errors)

→ Ziel: Robustes Design

Logikentwicklung für geringen Leistungsverbrauch

- **Verschiedene Modi je Verarbeitungsgeschwindigkeit und Leistungsverbrauch**
 - Modi über Automaten steuern
 - Aufwachtzeit, Initialisieren und Booten von Makros
 - **Dynamische Leistung**
 - Taktabstaltung, rückgekoppelte Register „enablen“ (Clock Gating)
 - Spannungsreduktion
 - Frequenz erniedrigen
 - Aktivitätsrate reduzieren (keine Kanäle multiplexen, Korrelationen nutzen)
 - Treiber nicht überdimensionieren
 - **Statische Leistung (Leckstrom)**
 - Umschaltung zwischen verschiedenen Modi (Aktiv, Stand-by)
 - Speicher in Schlafmodus schalten (Daten halten aber kein Zugriff)
 - nicht benötigte Blöcke ganz abschalten (Power Gating)
(aber gesamte Leistungsbilanz im System betrachten!)
- 

Maßnahmen zur Reduktion des Leckstroms

- HiVt Transistor (hohe Schwellspannung V_t)
 - Gate-Leckstrom nimmt stark zu unter 100nm
 - aber bei hoher Performance-Anforderung evtl. regVt Transistoren notwendig
- Reduzierte Versorgungsspannung ($V_{dd} / 2$)
 - Mehrere Vdd Domänen; variable Vdd je nach Performance
 - Level-Shifter zur Pegelanpassung nötig
 - reduziert statische und dynamische Leistung
- Aktive Wannensteuerung
 - Vorspannung variiert V_t und damit auch den Leckstrom
- Blöcke komplett abschalten
 - ROMs, Peripherie (USB, IrDA,...), Makrozellen (power gating)
 - Schnittstellen brauchen Isolationszellen wegen der undefinierten Pegel
 - nur kurze oder keine Aktivität im Stand-By Modus

Auswirkungen auf die Design-Tools und den Ablauf

- Zellbibliothek
 - Timing der Zelle muss Abhängigkeit von Vdd usw. berücksichtigen
- Tools & Flow
 - Info über Spannungsdomäne in der Netzliste nötig
 - STA kennt den Spannungswert und ermittelt das Timing der Zellen
 - Level-Shifter zur Pegelanpassung zwischen Spannungsdomänen
 - Abschaltisolierung in der Simulation prüfen
- Layout
 - Welche Spannung bekommen neu eingefügte Zellen (im Layout)?
 - Taktbaum Synthese über Domänengrenzen
 - Verschiedene Domänen dürfen nicht verbunden sein (LVS)
 - Hat jede Zelle die richtige Spannung?

Zusammenfassung

- SoC Design-Ziele und Herausforderungen
 - Funktionalität, Timing Anforderung und Zuverlässigkeit erfüllen
 - Systemkomplexität, Signalintegrität und Leistungsverbrauch bewältigen
- Technologie und Marktanforderungen
 - Miniaturisierung, größere Integrationsdichte, weniger Bauteile
 - Trend zeigt eine Verdopplung alle 1,5 bis 2 Jahre
- Logikentwicklung mit Anforderungen an Tools & Flow
 - Systemkomplexität und kurze Entwicklungszeiten bedingen
 - Makro Wiederverwendung, Standard-Komponenten
 - Verschmelzung von Logiksynthese und Layout
 - Signalintegrität bei Spannungsabfall und Übersprechen
 - Leckstrom ist das Thema der Zukunft beim Leistungsverbrauch

Infineon E-GOLD+ V3 Eva Board als Gesamtsystem

... und darüber hinaus gibt es noch viel mehr in einem Gesamtsystem ...

