

MULTIPROJEKT CHIP-GRUPPE

BADEN - WÜRTTEMBERG

MPC-Workshop Februar 2001

Aalen



MULTIPROJEKT CHIP-GRUPPE

BADEN - WÜRTTEMBERG

MPC-Workshop Februar 2001

Aalen

Herausgeber: Fachhochschule Ulm

© 2001 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.: 07131/504-400, Fax: /252-470, Email: clauss@fh-heilbronn.de

Fachhochschule Karlsruhe

Prof. Ritzert, Postfach 2440, 76012 Karlsruhe

Tel.: 0721/925-1512, Fax: -1513, Email: ritzert@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. Klotzbücher, Postfach 1261, 88241 Weingarten

Tel.: 0751/501-630, Fax: /49240, Email: klotzbuecher@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	Seite
1. Grußwort zum 25. Jubiläums-Workshop der MPC-Gruppe D. Wagemann, Ministerium Stuttgart	7
2. Rückblick auf die Arbeit der MPC-Gruppe in Lehre und Forschung D. Jansen, FH Offenburg	9
3. Mixed Signal Design eines Sensor Interfaces zur Erfassung von Kapazitätsdifferenzen D. Jansen, J. Hauser, FH Offenburg U. Mescheder, FH Furtwagen	13
4. Integration von hochperformanten ASICs in rekonfigurierbare Systeme zur Bereitstellung zusätzlicher Multimedia-Funktionalitäten H. Blume, H.-M. Blüthgen, T. G. Noll, RWTH Aachen	19
5. Integrated CMOS-OPAMPs for Ultra-Low-Voltages K. Leitis, W. Bonath, H. Kemper, M. Weber, FH Giessen	29
6. Festkomma-Arithmetik für DSP-Anwendungen mit redundanten Zahlendarstellungen B. Hoppe, M. Loy, H. Meuth, FH Darmstadt	35
7. ASIC System Design Ausbildung mit einem RISC-IP-Core F. Kesel, FH Pforzheim	45
8. Entwurf eines CAN-Controllers mit Hilfe von grafischen VHDL-Tools S. Baur, H. Kreutzer, FH Reutlingen	51
9. Entwicklung einer Videopipeline mit digitaler Echtzeit-Bildverarbeitung C. Reichherzer, FH Aalen	57
10. Digitale Zwischenfrequenz-Empfänger für die Satelliten-Navigation mit GPS, GLONASS und GALILEO M. Zimmer, Ulm	69
Weiterer Beitrag	
11. Mikroprozessorunterstützte Signalverarbeitung eines 512*512-Pixel-Array-CMOS-Sensors K. Schmidt, T. Milcsevics, FH Furtwangen Nachdruck des entsprechenden Artikels in „Forschungsbericht 2000“ der FH Furtwangen	98

Gefertigte Bausteine	Seite
12. Interface-Chip für Photozeilensensor A. Adler, P. Escher, K. Schmidt, A. Bumüller, FH Furtwangen	100
13. MISIP: Tastatur SchnittstellenProzessor T. Strauch, A. Gauckler, W. Rülling, FH Furtwangen	101
14. Thermologger_V3 M. Fischer, C. Störk, J. Hauser, D. Jansen, FH Offenburg	102
15. Thermologger_V4 M. Fischer, C. Störk, J. Hauser, D. Jansen, FH Offenburg	103
16. Uhrenmakro mit Selbsttest H.-J. Jahn, A. Führer, FH Ulm	104
17. Laderegler für Solarsysteme M. Barchet, O. Salzmann, G. Forster, FH Ulm	105
18. Hough Transform ASIC A. Epstein, G.-U. Paul, FH Mannheim	106

XXV. Jubiläums-Workshop der MPC-Gruppe, 2001

Grußwort

von Herrn Regierungsdirektor Dirk Wagemann

Sehr geehrter Herr Rektor Prof. Dr. Henning,
sehr geehrte Damen und Herren,

ich freue mich ganz besonders, dass ich Sie heute zum **25.** Workshop der MPC-Gruppe begrüßen darf. Die Zahl 25 ist – so meine ich – ein ganz besonderes Jubiläum, das nicht nur in der Ehe besonders gewürdigt werden sollte.

Die Mikroelektronik ist heute aus unserem Leben nicht mehr wegzudenken. Sie begegnet uns täglich – z.B. im Auto, in der Küche, im Büro. So wundert es nicht, dass die Mikroelektronik auch im Bildungsangebot unserer Hochschulen einen breiten Rahmen einnimmt.

Die Mikroelektronik ist Bestandteil aller Elektrotechnik-Studiengänge, bei einigen Hochschulen gibt es sogar eigene Studiengänge für Mikroelektronik. Leider hat die Zahl der Studierenden in den letzten Jahren stark abgenommen, obwohl die von den Fachhochschulen gut und praxisnah ausgebildeten Studierenden nach wie vor hervorragende Chancen auf dem Arbeitsmarkt haben.

Die MPC-Gruppe hat sich die Aufgabe gestellt, die Forschung und Lehre an den Fachhochschulen Baden Württembergs im Arbeitsgebiet der Mikroelektronik, insbesondere im Bereich des Entwurfs Integrierter Anwenderschaltungen zu fördern. Es ist eine Kooperation aus derzeit ca. 40 Professoren aus 13 Fachhochschulen. Alle vertretenen Fachhochschulen verfügen über eine weitgehend ähnliche Ausstattung mit Rechnern und Entwurfssoftware, sodass ein gegenseitiger Austausch in Lehrmitteln, Erfahrung und Entwürfen möglich ist. Über die durchgeführten Projekte wird auf den regelmäßig 2 x pro Jahr veranstalteten Workshops öffentlich berichtet. Die Lehraktivitäten umfassen den Austausch von Vorlesungsschriften, Inhalten und Anforderungen, zuletzt ist ein gemeinsam erarbeitetes Handbuch entstanden.

Die Mitglieder bilden ein BW überspannendes Netzwerk von regionalen Ansprechpartnern im Bereich der ASIC-Entwicklung, womit auch Industrieanfragen und Projekte kompetent und regional bearbeitet werden können. Im Rahmen der Gruppe sind Spezialisten aus allen Bereichen, Analog wie Digital, Bipolar wie CMOS für alle gleichermaßen zugänglich. Erarbeitete Ergebnisse wie Entwürfe und Modelle werden in der Gruppe lizenzfrei ausgetauscht. Die Zusammenarbeit mit der Industrie verläuft sehr erfolgreich, wobei das Primat der Lehre und die Einschränkung in der Nutzung der Hochschullizenzen zu beachten sind.

Die MPC-Gruppe ist in dieser Form in Deutschland ein Vorbild, eine Ausdehnung auf weitere Bundesländer wird immer wieder diskutiert. So haben z.B. Hessen, Bayern schon ihr Interesse bekundet.

Baden-Württemberg nimmt – wie so oft in der Hochschulpolitik – auch hier in der Bundesrepublik eine führende Rolle ein.

Für die bisherige hervorragende Arbeit möchte ich der MPC-Gruppe recht herzlich danken. Ohne diese innovative, hochschulübergreifende Arbeitsgruppe hätten solche Ergebnisse nicht erreicht werden können.

Die Fachhochschulen sind in besonderem Maße auf eine moderne technische Ausstattung angewiesen. Nur so können sie die Anforderungen erfüllen, die Studierende und Wirtschaft gleichermaßen an eine zeitgemäße Ausbildung stellt. Wichtiges Hilfsmittel für Ingenieure und Wissenschaftler ist der selbstverständliche Umgang mit der Datenverarbeitung. Hochwertige Rechnerausstattungen müssen zur Verfügung stehen.

Trotz zum Teil schwieriger Finanzlage hat das Ministerium in der letzten Beschaffungsrunde der MPC-Gruppe insgesamt 4,6 Mio DM aus zentralen

Mitteln für Rechnerbeschaffungen zur Verfügung stellen können.

Nun gilt es, veraltete Rechnersysteme durch neue hochwertige zu ersetzen. Die ersten groben Kostenschätzungen ergaben hierfür einen Mittelbedarf von ca. 5,2 Mio DM. Das Ministerium wird diese Summe in Abstimmung mit der MPC-Gruppe einplanen und nach dem heutigen Stand aller Voraussicht nach auch in den nächsten Jahren finanzieren können. Der genaue Stufenplan für die Beschaffungen muss allerdings noch erarbeitet werden.

Ich wünsche der MPC-Gruppe – insbesondere auch zum Wohl unserer Studierenden – weiterhin viel Erfolg und möchte die Fachhochschulen zu dieser kompetenten Arbeitsgruppe beglückwünschen.

Vielen Dank!!

XXV. Jubiläums-Workshop der MPC-Gruppe, 2001

Rückblick auf die Arbeit der MPC-Gruppe in Lehre und Forschung

Prof. Dr.- Ing. Dirk Jansen
 Sprecher der MPC-Gruppe
 Fachhochschule Offenburg
 Badstr. 24, 77652 Offenburg
 ☎ :0781/205-267, Fax: 0781/205-174

Die Mikroelektronik ist eine Schlüsselindustrie. Nach einer Darstellung (Abbildung 1) der FH Ulm führt ein Umsatz von 6 Mrd. in der reinen Mikroelektronik sekundär zu einem 40 Mrd. großen Umsatz an Elektronikgeräten, die wiederum in Systemen wie z.B. Kraftfahrzeugen integriert sind, wobei hier über 420 Mrd. Umsatz im Raum stehen. 6 Mrd. bewirken also letztlich einen Umsatz von über 420 Mrd. Die mikroelektronischen Elemente sind heute Hirn und Nerven der industriellen Großgeräte und damit Schlüssel für moderne Produkte. Ohne Mikroelektronik sind diese Produkte in Zukunft nicht mehr verkäuflich.

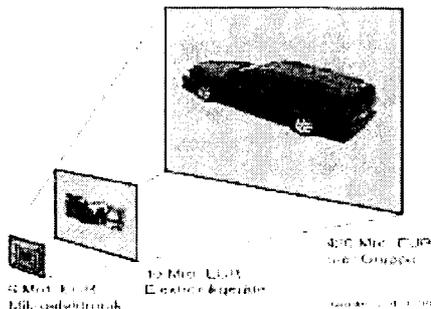


Abb. 1: Bedeutung der Mikroelektronik

Moderne Elektronik wird heute fast ausschließlich in Form von integrierten Schaltungen entwickelt. Die Kenntnisse in der Integrationstechnik, wobei digitale wie auch analoge Schaltungstechniken gleichermaßen von Bedeutung sind, sind deshalb für die Elektroingenieure von essenzieller Bedeutung. Selbst wenn der Ingenieur nicht persönlich im IC-Design engagiert ist, hat er doch ständig durch Spezifikation oder Beschaffungsvorgänge mit dem Designvorgang zu tun und muss

deshalb hier über elementare Grundkenntnisse verfügen.

Es ist Aufgabe der Hochschulen, diese Grundkenntnisse im IC-Design in der Ingenierausbildung zu vermitteln. Investitionen in diese Ausbildung verzinsen sich exponentiell in der Wirtschaft, wobei die Absolventen dieses Wissen in ihrer beruflichen Praxis verwerten. Die Hochschulen müssen deshalb professionell ausgerüstet sein und technologisch wie know-how-mäßig eine Vorreiterrolle übernehmen.

Die MPC-Gruppe hat sich die Aufgabe gestellt, für diese Ausrüstung sowie die Kenntnisse und Erfahrungen im Bereich der Mikroelektronikausbildung zu sorgen. In Baden-Württemberg wurden deshalb alle Fachhochschulen mit Elektrotechnik-Studiengängen mit geeigneten Laboratorien in einheitlicher Art und Weise ausgerüstet. In Studenten- wie in industriellen Projekten wird Erfahrung im Entwurf integrierter Schaltungen gewonnen und innerhalb der MPC-Gruppe ausgetauscht.

MPC steht für Multi Project Chip, d.h. die Aufteilung der Kosten für einen IC-Durchlauf auf mehrere Projekte, in denen die Waverkosten entsprechend von mehreren Projekten getragen werden.

Bei der Gründung 1988 stand ursprünglich ein solcher Multi-Project-Waver-Service im Vordergrund und sollte der Gruppe zu vertretbaren Kosten Zugriff auf Silizium verschaffen. Die Gruppe hatte deshalb eigene Verträge mit Halbleiterherstellern und Instituten wie auch dem IMS (Institut für Mikroelektronik Stuttgart).

Mit den europäischen Projekten Eurochip und später Europractice, in denen ein MP-Waver-Service europaweit angeboten wird, konnte auf den eigenen MP-Service verzichtet werden. Geblieben

ist ein intensiver Erfahrungsaustausch, ein regelmäßiges Treffen zu Workshops, die rotierend an den beteiligten Hochschulen durchgeführt werden, sowie ein intensiver Studenten- und Lehrmittelaustausch. In gemeinsamen Lehrgängen, die auch aus der Gruppe heraus organisiert werden, soweit sie nicht auf dem Markt oder von Europractice angeboten werden, werden Spezialkenntnisse im IC-Design vermittelt.

Abbildung 2 zeigt das über ganz Baden-Württemberg gespannte Netz der Standorte. MPC ist überall regional vertreten und Ansprechpartner für die örtliche Industrie.

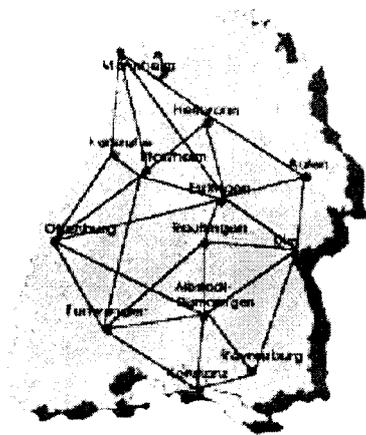


Abb. 2: Standorte und Mitglieder

Die Hochschullaboratorien stellen ASIC-Kompetenz-Center dar (ASIC steht für Application Specific Integrated Circuits), die Beratungsfunktionen aber auch konkrete Industrieentwicklungen für KMU's zu übernehmen bereit sind. Durch die Vernetzung der Gruppenpartner kann an jedem Standort die gesamte Palette des Know-hows im ASIC-Bereich angeboten werden, da innerhalb der MPC-Gruppe Spezialisten für analoge wie digitale Entwurfsprozesse verfügbar sind.

Organisatorisch ist die MPC-Gruppe ein Zusammenschluss von Professoren der Fachhochschulen Baden-Württembergs mit dem Lehrgebiet „Entwurf, Herstellung und Test integrierter Schaltungen“. Gegründet von den Kollegen Schmidt, FH Furtwangen und Führer, FH Ulm 1988 sind die Beziehungen und Rechte der Mitglieder in einer eigenen Satzung seit 25.01.1996 geregelt. Organe sind die Vollversammlung aller Mitglieder, der MPC-Ausschuss, in dem jede beteiligte Hochschule mit einer Stimme vertreten ist, und der MPC-Sprecher, der sich um die Abwicklung und Ausführung der Beschlüsse kümmert. Sprecher und

Stellvertreter sind für zwei Jahre gewählt. Hinzu kommen Kollegen, die Sonderaufgaben wie Finanzen, Herausgabe des Berichtsheftes usw. übernommen haben. Für Sonderaufgaben werden Ad-hoc-Ausschüsse gebildet, wie z.B. für die Ausschreibung CAE-E im Zusammenhang mit der Neubeschaffung der Laborausrüstungen.

Die MPC-Gruppe tritt mit gemeinsamem einheitlichen Logo in der Öffentlichkeit auf, stellt eigene Messestände zusammen, tritt bei Symposien und Konferenzen mit Vorträgen hervor und hat inzwischen mit dem „EDA-Handbuch“¹ ein gemeinsames Buchprojekt realisiert.

Wichtige Aufgabe ist die Schaffung, der Erhalt und die Pflege einer modernen Ausrüstung, siehe hierzu Abbildung 3. Durch Rahmenverträge mit Hewlett Packard und Mentor Graphics wurde inzwischen ein weitgehend einheitlicher technischer Stand in Hard- und Software erreicht. Die Ausrüstung mit modernen Computer-Workstations und zugehörigen Software-Lizenzen wurde in HBFV-Anträgen beschafft.

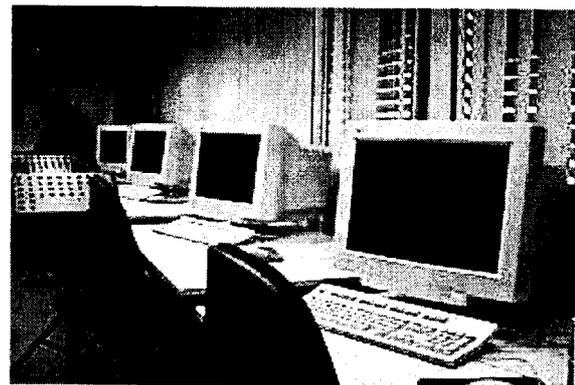


Abb. 3: Schaffung, Erhalt und Pflege einer modernen Ausrüstung

Inzwischen wurden zwei Beschaffungsrunden erfolgreich in der nun zwölfjährigen Geschichte abgewickelt. Eine neue Runde steht derzeit an. Die einheitliche Ausstattung schafft nicht nur einen Synergieeffekt bei der Beschaffung, sondern bedeutet auch eine wertvolle Hilfe bei Software- und Installationsproblemen. Das gemeinsame Auftreten gegenüber den Lizenzgebern ermöglicht zudem die Erzielung von Sonderkonditionen, die andernfalls unerreichbar wären. Dies betrifft sowohl die wirtschaftliche Seite wie auch die Beschaffung von Spezialsoftware, die andernfalls eine einzelne Hochschule nicht bekommen hätte.

¹ Das Handbuch der Electronic Design Automation, ISBN 3-446-21288-4, Hanser-Verlag, Leipzig

Innerhalb der MPC-Gruppe wurde inzwischen in über 120 in der Gruppe entwickelte und gefertigte Schaltungen und Technologien unmittelbare Erfahrung gewonnen. Die meisten dieser Projekte sind Studentenprojekte, wobei die Komplexität von einfachen Logikschaltungen bis hin zu 16 Bit-Prozessoren, von Operationsverstärkerschaltungen bis hin zu Mixed-Signal-SOC's reicht.

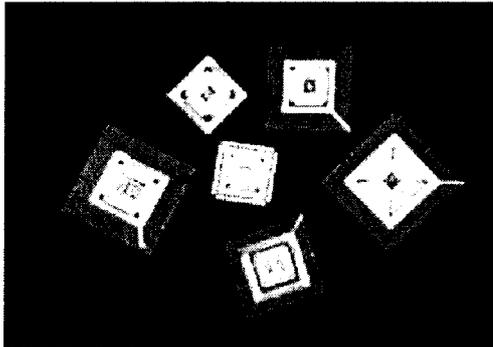


Abb. 4: Chip-Projekte der MPC-Gruppe

In der Gruppe sind Bibliotheken, CMOS- von 0,35 μm , Bipolar-, BICMOS-, Gate-Array- und Analog-Gate-Array – Technologien vorhanden. Es bestehen auch umfangreiche Erfahrungen im Bereich von FPGA-Logik, hauptsächlich zu Emulationszwecken. Die MPC-Gruppe deckt damit die gesamte Bandbreite moderner Mikroelektronik in Lehre und Forschung, soweit sie an Fachhochschulen anzubieten ist, ab. Beispiele für Entwürfe

sind ein Sigma-Delta-Wandler 16 Bit und ein Mixed-Signal-ASIC DSWPC, mit Prozessor und Signalverarbeitung integriert, beides Entwürfe der Fachhochschule Offenburg (s. hierzu die folgenden Abbildungen 5 und 6).

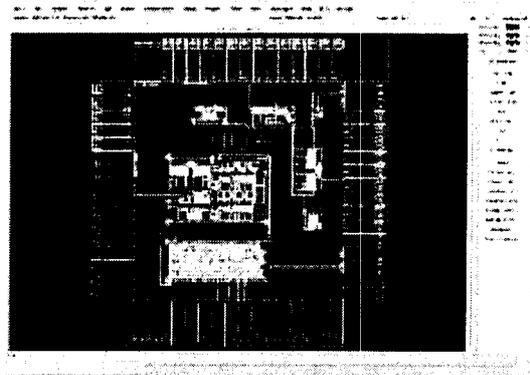


Abb. 5: Sigma-Delta-Wandler 16 Bit

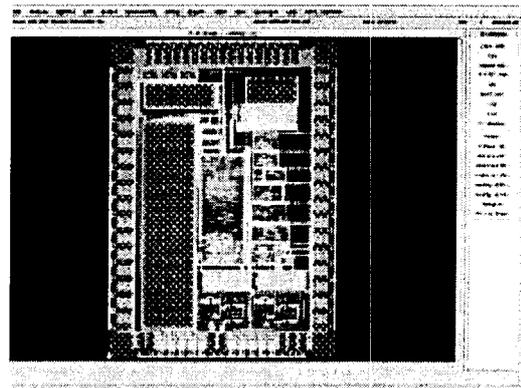


Abb. 6: System on Chip: DSWPC-ASIC

Abbildung 7 enthält weitere Beispielchips der MPC-Gruppe. Die Bilder sagen mehr aus als umfangreiche Beschreibungen und demonstrieren den Stand der Elektronikausbildung und letztlich den Erfolg der MPC-Gruppe.

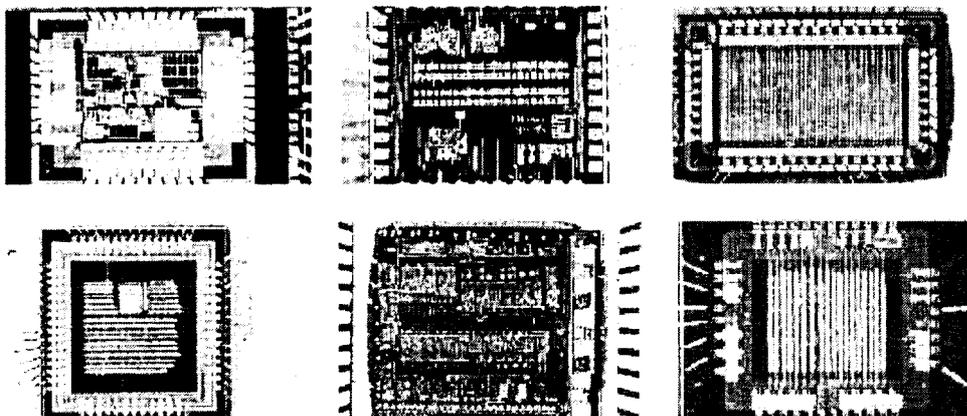


Abb. 7: Weitere in der MPC-Gruppe entwickelte Chips

XXV. Jubiläums-Workshop der MPC-Gruppe, 2001

Mixed Signal Design eines Sensor Interfaces zur Erfassung von Kapazitätsdifferenzen

Das Projekt „Chip im Reifen“ (CiR)

Prof. Dr.- Ing. Dirk Jansen
 Dipl.-Ing. (FH) Jürgen Hauser
 Prof. Dr. U. Mescheder, FH Furtwangen
 Fachhochschule Offenburg
 ASIC Design Center
 Badstr. 24, 77652 Offenburg
 ☎ :0781/205-267, Fax: 0781/205-174

Es wurde eine Interface-Elektronik zur Messung kleinerer Kapazitätsdifferenzen entworfen, wie sie zur Auswertung der Signale eines mikromechanischen Absolutdruckaufnehmers erforderlich ist. Messkapazität und Referenzkapazität bilden die Frequenz bestimmenden Elemente jeweils eines digitalen Oszillators, der im Frequenzbereich von 6.....10 MHz schwingt. Die Frequenzen werden über einen Zeitintervall ausgezählt und die Differenz in einem Rechner gebildet. Der Zeitintervall ist konfigurierbar. Durch die digitale Weiterverarbeitung ist das Interface auf einen großen Bereich ähnlich konzipierter Drucksensoren anpassbar. Der Design fügt sich als Modul in das Konzept des FHOP-Kits ein. Den Nachweis der Funktion muss noch ein Testchip erbringen.

Abbildung 1 beschreibt das Konzept. Die Druckmessung erfolgt durch einen mikromechanisch hergestellten Druckaufnehmer, die elektronische Auswertung und Datenübertragung erfolgt durch einen elektronischen Chip, der induktiv über eine in der Reifenwange angeordnete Antenne mit einem fahrzeugfesten Aufnehmer kommuniziert. Hierbei können neben Druck und Temperatur auch noch reifenspezifische Werte wie Seriennummer, Hersteller, Montagedaten, Typ und zulässige Nutzwerte übertragen werden, so dass der Transponder auch für logistische Zwecke eingesetzt werden kann. Damit unterscheidet sich das Konzept signifikant von Varianten, bei denen nur die physikalische Übertragung der Sensorsignale realisiert ist.

1. Das Projekt CiR

Im Automobilbereich gewinnt die Sicherheitstechnik zunehmend an Bedeutung. Neben Motor- und Bremskontrolle ist ein wesentlicher sicherheitsrelevanter Bereich die Kontrolle der Reifen, wobei die Betriebsparameter Druck, Temperatur und Verschleiß zu überwachen sind. Die existierenden Systeme verwenden hierbei Sensoren in der Radnabe, der Reifen ist rein passiv. Im Rahmen des innovativen Projektes Chip im Reifen, eine Kooperation des IAF's der FH Offenburg mit der FH Furtwangen, wird nun untersucht, wie weit durch einen mikromechanischen, elektronischen Transponder, der in den Reifen einvulkanisiert ist, eine zuverlässige Kontrolle von Reifendruck und Temperatur möglich ist.

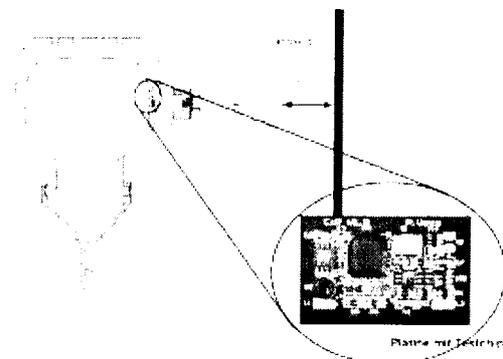


Abb. 1: Konzept des innovativen Projekts
Chip im Reifen

Die Entwicklung der Transponderelektronik erfolgt bei der FH Offenburg, wobei auf die umfangreichen Erfahrungen mit ähnlichen Systemen im Mittelfrequenzbereich zurückgegriffen werden kann. Die in Abbildung 1 dargestellte Testplatine realisiert die Kommunikation und Messtechnik, allerdings noch nicht in voll integrierter Form.

2. Der mikromechanische Drucksensor

Die Fachhochschule Furtwangen entwickelt in diesem Projekt den mikromechanischen Sensor, wobei hier eine Neuentwicklung wegen der extremen Umweltbelastung, die im Reifen während des Betriebes auftritt, erforderlich wurde. Der MEMS-Sensor wurde inzwischen von Furtwangen realisiert, er misst 1.....6 bar absolut und beruht auf einem kapazitiven Messprinzip. Der Druck bewirkt eine Änderung der Kapazität in einem kleinen Feld von noppenförmigen Drucksensorelementen, wobei ein ähnliches Feld als Referenzkapazität nicht druckabhängig ist. Durch die kleinen Sensornoppen ist diese Anordnung auch bei hohen Querschleunigungen und Vibrationen extrem robust, in CMOS-kompatibler Mikromechanik relativ leicht herstellbar und durch Skalierung der Noppenzahl auf viele Anwendungen anpassbar. Das Messsignal erscheint hier als eine Änderung der Messkapazität gegenüber der Referenzkapazität, wobei die Auflösung ca. 100 fF/bar beträgt auf einer Basis von 8 pF des gesamten Arrays.

3. Auswertung der Kapazitätsdifferenzen

Die Auswertung dieser kleinen Kapazitätsdifferenzen muss in einer Interface-Logik erfolgen, die als Modul in die Transponder-Elektronik eingefügt ist.

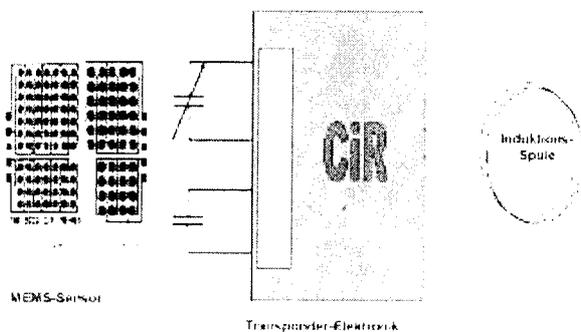


Abb. 2: Transponder

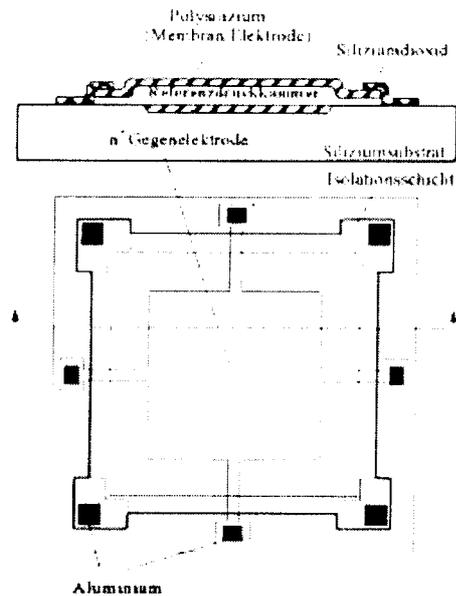


Abb. 3: Mikromechanischer Drucksensor (FH Furtwangen, Prof. Dr. Mescheder)

Es kommen die in Abbildung 4 vorgestellten Auswerteprozesse in Frage. Eine direkte Impedanzmessung liefert nicht die erforderliche Genauigkeit. Besser ist da schon das Prinzip der Kapazitätsbrücke, wobei die Messkapazität und die Referenzkapazität quasi gegeneinander geschaltet sind.

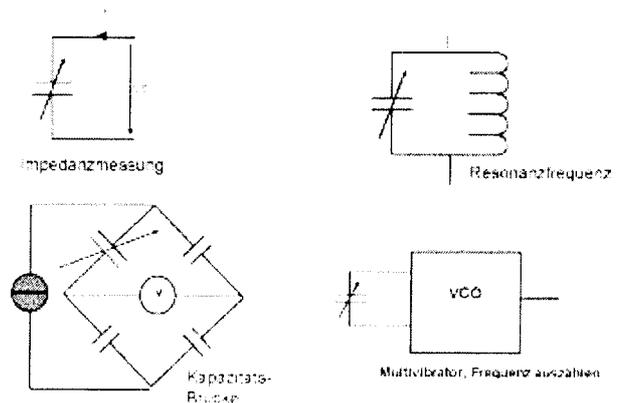


Abb. 4: Auswerteprozesse

Die Realisierung einer integrierten Elektronik zur Auswertung einer Brückenmessung ist aber sehr aufwendig, da umfangreiche Filter, ein Synchrongleichrichter und ein zusätzlicher A/D-Wandler benötigt werden. Diese Baugruppen benötigen

relativ viel Strom und Siliziumfläche und sind deshalb nur aufwendig zu integrieren.

Ein weiteres Prinzip zur Messung von Kapazitäten ist die Auswertung der Resonanzfrequenz, wenn man die Kapazität mit einer Induktivität zu einem Schwingkreis kombiniert. Eine solche Messung liefert prinzipiell schon ein digitales Ergebnis (Frequenz = Häufigkeit). Anstelle der Induktivität kann vorteilhaft auch eine geeignete elektronische Ersatzschaltung, im einfachsten Fall ein Multivibrator verwendet werden. Letztere lassen sich sehr klein integrieren, arbeiten quasi digital und können extrem leistungsarm ausgelegt werden. Die Messkapazität bildet in dieser Anordnung als das die Frequenz bestimmende Element eines Oszillators. Die Auswertung erfolgt durch Auszählen der Impulse über einen vorgegebenen Zeitintervall. (Abbildung 5).

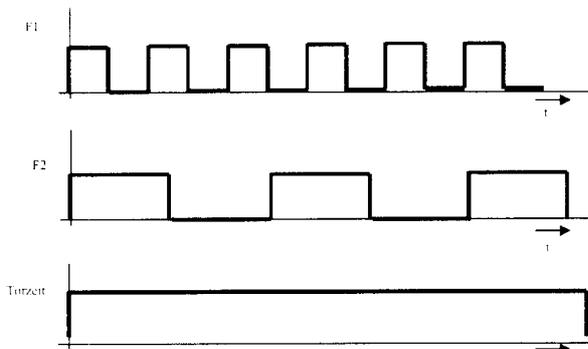


Abb. 5: Zeitdiagramm für Frequenzdifferenzbewertung

Das Differenzprinzip kann auf unterschiedliche Art und Weise auch hier realisiert werden, z.B. indem mit der Messkapazität ein Zähler herauf gezählt wird, mit der Referenzkapazität der Zähler wieder herab gezählt wird. Der verbliebene Zählerwert repräsentiert dann die unterschiedliche Frequenz und damit die Kapazitätsdifferenz. Eine weitere Möglichkeit ist die hier realisierte (Abbildung 6), dass für Referenz- und Messkapazität jeweils ein eigener Oszillator verwendet wird, der auch jeweils einen eigenen Zähler bedient. Die Zählerstände können vom Rechner (hier ein FHOP-Prozessor) ausgelesen werden. Diese etwas aufwendigere Ausführung hat den Vorteil, dass durch den Prozessor umfangreichere Filtermaßnahmen, die der Programmierung zugänglich sind, durchgeführt werden können. Der Modul enthält einen eigenen Zeitgeber (Timer), der ebenfalls programmierbar die Auswertzeit zu kontrollieren erlaubt. Damit ist das System extrem flexibel und kann sehr unterschiedlichen messtechnischen Bedingungen angepasst werden. Dies ist auch notwendig, da die

absolute Größe der Mess- und Referenzkapazität sowie möglicher Streukapazitäten zum jetzigen Zeitpunkt nicht genügend bekannt ist.

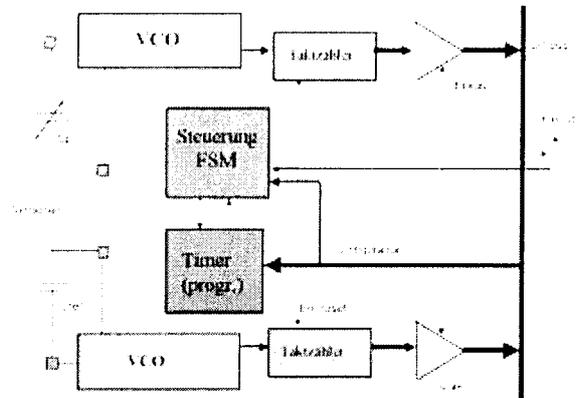


Abb. 6: CapSens-Interface Unit

4. Ausführung des Oszillators

Der verwendete Oszillator basiert auf dem Multivibratorprinzip und geht auf einen VCO-Design zurück, der sich in einer PLL-Schaltung der FH Offenburg [1] bewährt hat. Die Messkapazität C_x wird durch eine Stromquelle aufgeladen und durch eine zweite Stromquelle entladen.

Beide Stromquellen sind symmetrisch, so dass sich ein sägezahnförmiges Spannungssignal C_x ergibt. Erreicht die Spannung die positive Referenzspannung, schaltet ein Komparator, genau so wie bei der negativen Referenzspannung. Die beiden Komparatoren triggern wiederum ein Flip-Flop, welches die Umschaltung der zuvor erwähnten Stromquellen steuert. Der Prozess ist wegen dieser Rückführung selbst erregend, und ein Anlauf ist über einen sehr großen Kapazitätsbereich unproblematisch (Abbildung 7).

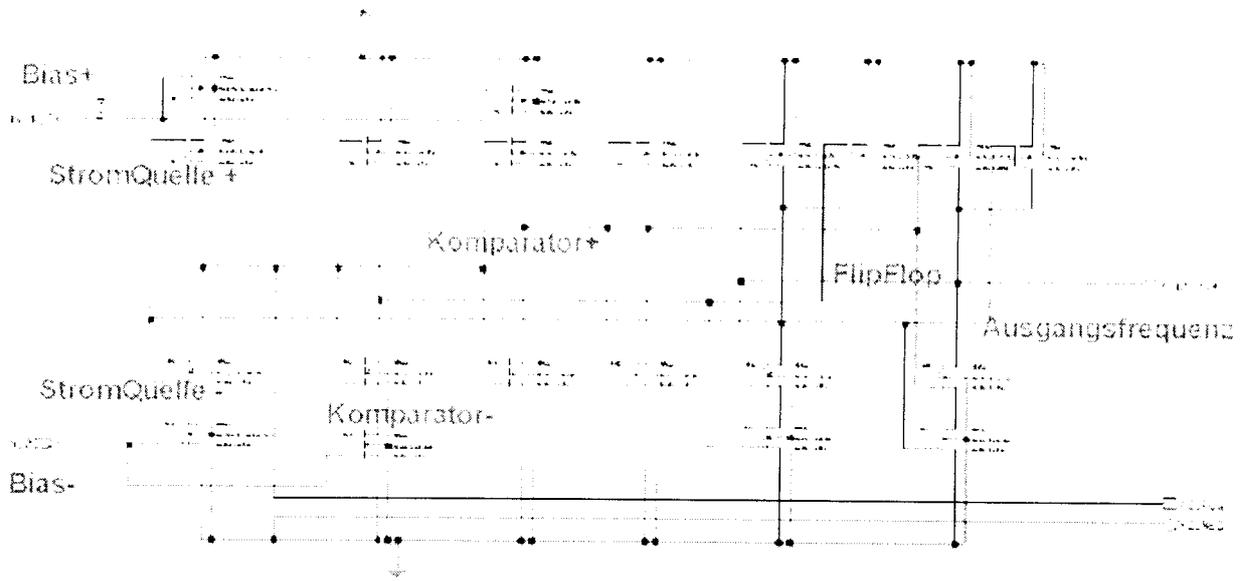


Abb. 7: VCO

Abbildung 8 zeigt das Frequenzverhalten des VCO's (simuliert) im interessierenden Kapazitätsbereich von 6 – 10 $\mu\text{F}4 C_x$. Das Gesamtverhalten ist naturgemäß nicht linear und entspricht einer $1/x$ – Funktion. Es wird jedoch noch ein sehr kleiner Bereich der Kennlinie für die Kapazitätsmessung verwendet, so dass eine Linealisierung im Arbeitspunkt möglich ist.

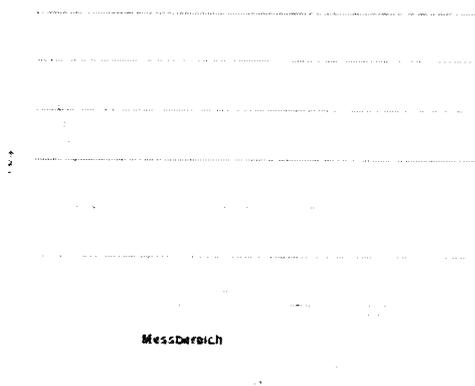


Abb. 8: Frequenzverhalten des VCO

Allerdings sollten C_x und C_{ref} nicht zu sehr voneinander abweichen. Die Frequenzen, die durch Wahl der Stromquellenparameter beeinflusst werden können, liegen somit zwischen 6 und 10 MHz, damit dürfte eine 12 Bit-Auflösung bei einer

Messhäufigkeit von ca. 10 Messungen/sec. zu erwarten sein. Ein Layout des VCO's zeigt Abbildung 9. Es entspricht weitgehend dem in der zuvor erwähnten Literaturstelle PLL-VCO.

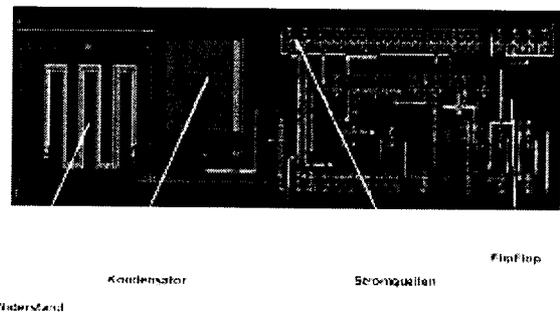


Abb. 9: Layout VCO

5. Diskussion des gewählten verfahrens

Das gewählte Konzept erlaubt eine unmittelbare digitale Auswertung und ist anpassbar an einen großen Bereich von Eingangskapazitäten. Die Auflösung ist über die Integrationszeit kontrollierbar, der digitale Aufwand ist relativ niedrig. Die verwendeten Baugruppen, insbesondere der VCO sind erprobt und ihre Eigenschaften bekannt. Physikalische Einflüsse auf

den Messwert kann nur der Lade- und Entladestrom im VCO haben, der möglicherweise temperaturabhängig ist. Weiterhin sind Toleranzen durch die Fertigung in diesem Bereich denkbar. Da der Chip über eine Spannungsreferenz (Bandgap) sowie einen Temperatursensor verfügt, sind die Temperatureinflüsse weitgehend kompensierbar, was teilweise unmittelbar elektronisch erfolgen kann. Darüber hinaus ist noch eine rechnerische Kompensation und Eichung des Moduls durch die digitale Prozessierung der Signale möglich. Wie weit das notwendig ist, kann erst die Messung zeigen und hängt nicht zuletzt von den Anforderungen ab.

Die Empfindlichkeit der Anordnung und des Auswerteverfahrens ist nicht zuletzt von der Streukapazität abhängig als auch von der Symmetrie der Messkapazität zur Referenzkapazität des MEMS-Sensors. Ein weiteres Risiko besteht durch die Verwendung von zwei VCO's, die mit nur wenig voneinander differierender Frequenz auf dem gleichen Chip betrieben werden. In einer solchen Anordnung besteht immer die Gefahr der Mitkopplung, was sich als Totzone im Messbereich auswirken würde. Wie weit solche Risiken eintreten, muss letztlich ein realisierter und vermessener Chip zeigen.

6. Geplanter Chip im Reifen-ASIC

Ein Kompletchip für das Projekt Chip im Reifen wurde wegen zahlreicher noch zu entwickelnder Details bisher nicht realisiert und ist auch über die zur Verfügung stehenden Mittel der Vorstudien nicht machbar.

Abbildung 10 zeigt das Konzept eines solchen Chips, wobei auch in der nächsten Phase noch ein Hybridenaufbau aus Sensorchip und digitalem Transponderchip vorgesehen ist. Erst in einem Serienprodukt lohnt sich der Aufwand für die Integration von Mikromechanik und CMOS-Technologie.

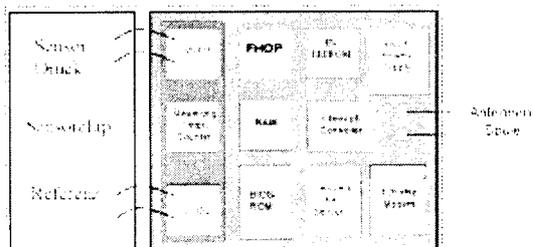


Abb. 10: Blockbild des geplanten CiR-ASIC

In einem realistischen Transponder muss die Energieversorgung aus der Antennenspule erfolgen. Dieser Teil wurde bisher noch nicht entwickelt. Wegen der geringen zur Verfügung stehenden Leistung ist zudem eine leistungsarme Auslegung aller übrigen Baugruppen essenziell. Der Chip enthält neben dem Interface zum Drucksensor auch einen Temperatursensor, die induktive Transponderelektronik (125 kHz-Modem) sowie einen FHOP-Controller zur Steuerung und Prozessierung der Daten. Für Identifikationszwecke sollte der Chip auch einen EE-Prom enthalten, was allerdings in der derzeitig zur Verfügung stehenden Technologie nicht möglich ist. Die Gesamtintegration wird deshalb erst angegangen, wenn die übrigen Komponenten des Systems CiR so weit entwickelt und definiert sind, dass der Chip vollständig spezifiziert werden kann. Das kann erst in einer nächsten Projektphase erfolgen.

7. Literatur

- [1] Entwicklung eines PLL zur Generierung eines 11 MHz-Taktes aus einem 32 kHz-Takt, Projektbericht, Dipl.-Ing. (FH) Jürgen Hauser, Okt. 99.
- [2] Mikrosystemtechnik, Prof. Mescheder, Teubner Verlag, 2000.
- [3] Patent CiR, Offenlegungsschrift DE 199 24 830 A 1, 30.11.2000

Integration von hochperformanten ASICs in rekonfigurierbare Systeme zur Bereitstellung zusätzlicher Multimedia-Funktionalitäten

*Blume, H.; Blüthgen, H.-M.; Noll, T. G.
Lehrstuhl für Allgemeine Elektrotechnik und Datenverarbeitungssysteme
RWTH Aachen, Schinkelstraße 2
52062 Aachen
Email: blume@eecs.rwth-aachen.de*

Abstract

Die erforderliche Rechenleistung vieler zukünftiger Multimedia-Anwendungen liegt weit über der, die heutige Systeme zur Verfügung stellen. Daher kommt der Integration hochleistungsfähiger Multimedia-Komponenten eine sehr große Bedeutung zu. Dieser Beitrag stellt die Integration solcher Multimedia-Komponenten in Rechner-Systeme vor, wobei rekonfigurierbare Coprozessor-Boards verwendet werden. Das Ziel rekonfigurierbarer Plattformen, die für verschiedene Anwendungen angepasst werden können und die digitale Signalprozessoren, Control- und Speicher-Bausteine sowie dedizierte Multimedia-ASICs enthalten, wird in diesem Beitrag herausgearbeitet. Auf dem Weg zu solch einer Plattform werden vier ASICs für die Bild- und Textverarbeitung vorgestellt. Die Integration der Bausteine in mobile Rechner-Systeme unter Verwendung von CardBus-Coprozessor-Boards wird vorgestellt. Als eine beispielhafte Anwendung für diese Coprozessor-Boards wird eine scheckkartengroße Karte zur approximativen Volltextsuche in Datenbanken, wie z.B. Enzyklopädien, die aus dem Internet geladen und aktualisiert werden können und die dann mit kürzesten Antwortzeiten durchsucht werden können, vorgestellt und diskutiert.

1. Einleitung

Zukünftige Multimedia-Anwendungen wie mobile Video-Kommunikation oder multimediales Information-Retrieval werden eine große Rechenleistung erfordern. Diese Rechenleistung liegt deutlich über der, die in heutigen (mobilen) Systemen durch General-Purpose- oder Signalprozessoren zur Verfügung gestellt wird. Daher kommt der Integration von hochleistungsfähigen Multimedia-Komponenten, die diese Rechenleistung zur Verfügung stellen, eine sehr große Bedeutung zu.

Um existierenden Systemen neue Funktionalitäten hinzuzufügen, müssen diese Systeme mit dedizierten anwendungsspezifischen Bausteinen (ASICs), die diese hohe Rechenleistung bieten, ausgestattet werden. Diese Bausteine führen dann Basisoperationen mit einem sehr hohen Datendurchsatz aus. Daher werden diese Coprozessoren z.B. in Coprozessor-Boards eingefügt, die leicht in existierende Systeme integriert werden können.

Da es nicht möglich ist, ein dediziertes Coprozessor-Board für jede Anwendung zur Verfügung zu stellen, müssen diese Systeme rekonfigurierbar im Sinne einer Anpassung auf einen ganzen Bereich von Anwendungen bzw. mindestens adaptierbar auf einen größeren Satz von Systemparametern wie Filterkoeffizienten, Schwellwerte, etc., sein. Dieses Konzept wird im Weiteren als „Rekonfigurierbares Computing“ bezeichnet. Eine ideale rekonfigurierbare Plattform sollte dabei Rechenkomponenten wie digitale Signalprozessoren (DSPs) genauso wie

schwach programmierbare Coprozessoren für Multimedia-Aufgaben mit sehr hohem Datendurchsatz enthalten. Weiterhin müssen diese Plattformen Speicherbausteine sowie Controller-Bausteine enthalten. Die Control-Funktion kann dabei z.B. durch Field Programmable Gate Arrays (FPGAs) oder Embedded Programmable Logic Devices (EPLDs) erfolgen, die eine Programmierbarkeit im System bieten („In-System Programmability“). Bild 1 stellt die Basisstruktur solch einer rekonfigurierbaren Plattform dar. Der Anwendungsbereich von FPGAs und EPLDs ist jedoch nicht auf Control-Funktionen beschränkt. Diese Bausteine gewinnen in den letzten Jahren zunehmend mehr Aufmerksamkeit, seit Untersuchungen zeigen, dass sie, bezogen auf die Silizium-Fläche, eine höhere Rechenleistung aufweisen als General-Purpose-Prozessoren [1]. Deshalb spielen FPGAs und EPLDs bei der Entwicklung rekonfigurierbarer Plattformen eine wichtige Rolle [2].

Auf dem Weg zu solch einer rekonfigurierbaren Plattform stellen wir in diesem Beitrag die Entwicklung hochleistungsfähiger ASICs zur Text- und Bildverarbeitung und ihre Integration in Computer-Systeme mittels Coprozessor-Boards vor.

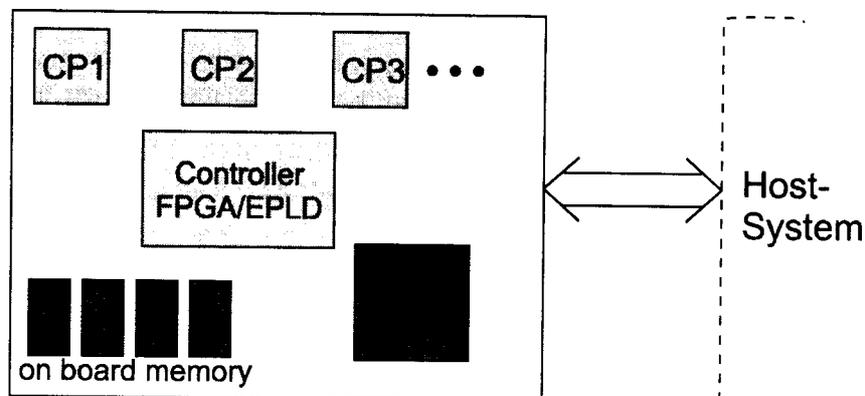


Bild 1: Basis-Struktur eines rekonfigurierbaren Coprozessor-Boards

Eine beispielhafte Anwendung dieser hybriden rekonfigurierbaren Coprozessor-Boards stammt aus dem Information-Retrieval-Bereich. In diesem Bereich sind hochleistungsfähige und intelligente Formen von inhaltsbasierter On-line-Suche erwünscht. Hier müssen Anfragen in sehr großen Datenbanken multimedialer Dokumente bearbeitet werden. Neben Text- und Audio-Datenbanken existieren dort insbesondere Bild-Datenbanken, die bearbeitet werden müssen. Die Bearbeitung dieser Datenbanken erfordert daher Bildverarbeitungsmodul mit der Fähigkeit zu höchstem Datendurchsatz.

Die rekonfigurierbaren Coprozessor-Boards, die in diesem Beitrag vorgestellt werden, wurden u.a. im sogenannten High-Performance-Query-Server (HPQS, [3]) eingesetzt. Das Ziel der Entwicklung dieses Servers war, dass via Internet natürlichsprachliche Anfragen an den HPQS gerichtet werden können, die dann in die Sprache der adressierten Zieldatenbank transformiert werden. Die Dokumente werden semantisch bzgl. ihres Informationsgehalt analysiert und die extrahierten Informationen werden in kognitiv-adäquaten Ergebnisdokumenten abgelegt. Da keine Pre-Indexierung der Dokumente erfolgt, die im System abgelegt werden, wird ein sehr hoher Flexibilitätsgrad erhalten, was aber auch bedeutet, dass für jede Anfrage sehr große Datenmengen verarbeitet werden müssen. Um akzeptable Antwortzeiten zu erzielen, ist es daher unbedingt erforderlich, dass diese Systeme mit dedizierten Hardware-Prozessoren, die einen sehr großen Datendurchsatz bieten, ausgestattet werden. Bild 2 stellt die Grundstruktur des HPQS dar.

Um diese Anwendung zu unterstützen, wurden ein Histogrammprozessor, ein programmierbares zweidimensionales Transversalfilter ein programmierbares zweidimensionales gewichtetes Medianfilter zur Bildverarbeitung und ein Textprozessor-ASIC zur fehlertoleranten Volltextsuche entwickelt.

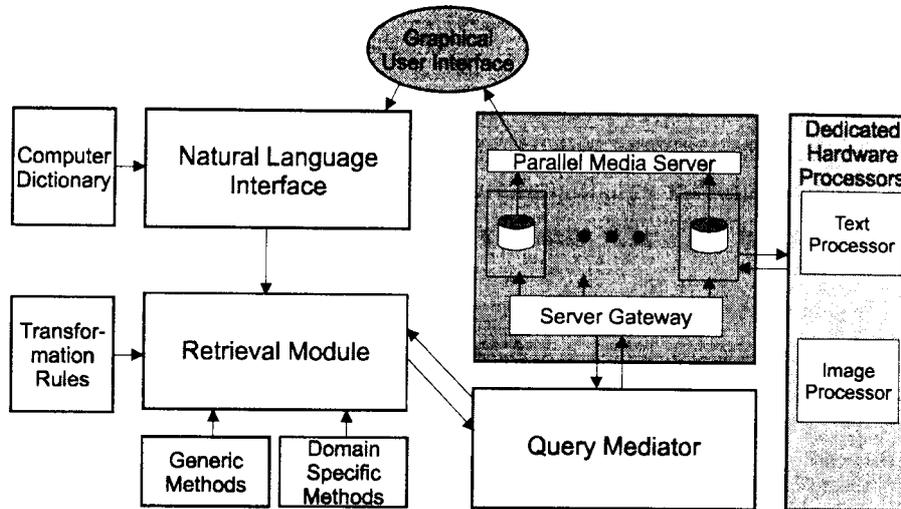


Bild 2: Grundstruktur des High-Performance-Query-Servers [3]

Als einen ersten Schritt hin zu rekonfigurierbaren Plattformen, die auch in mobilen Systemen eingesetzt werden können, präsentieren wir in diesem Beitrag ein Coprozessor-Board, das nur einen einzigen Coprozessor enthält und nur eine einzige Multimedia-Funktionalität unterstützt, dafür aber, bedingt durch die extrem kleinen Abmessungen dieses Coprozessor-Boards, auch für mobile Anwendungen geeignet ist. Die Funktionalität dieses Boards kann verändert werden, indem der Coprozessor, der auf diesem Board angebracht ist, ausgetauscht wird und das EPLD, das zu Steuerzwecken verwendet wird, im System umprogrammiert wird. Um das Coprozessor-Board mit mobilen Host-Systemen zu verbinden, wurde für dieses Board die moderne 32-Bit CardBus-Schnittstelle gewählt. Das CardBus-Interface ist direkt mit dem PCI-Bus verbunden und erlaubt daher einen Datentransfer von bis zu 132 Mbyte/s. Weiterhin ermöglicht diese Schnittstelle eine dynamische Rekonfiguration der Systemressourcen und erlaubt somit das Einfügen einer CardBus-Karte in ein laufendes System („Hot-Plug-in“). CardBus-Karten besitzen einen extrem kleinen Formfaktor (Scheckkarten-Größe) und sind somit ideal für mobile Systeme (Einsatz z.B. in Laptops).

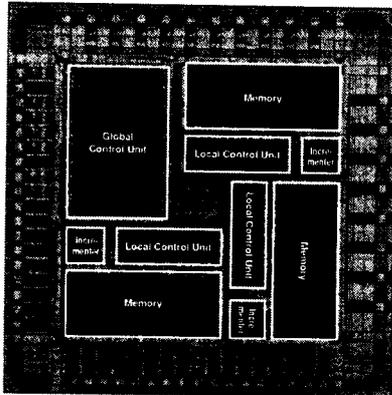
Das zuvor genannte CardBus-Coprozessor-Board, das die dedizierten ASICs enthält, wird durch eine komfortable graphische Benutzeroberfläche unter dem Betriebssystem Linux gesteuert. Die entwickelten ASICs, ihre Integration in ein CardBus-Coprozessor-Board und erste Resultate bei der Anwendung dieses Boards werden in den folgenden Kapitel diskutiert. Die erzielten Datentransferraten werden dabei mit denen alternativer Hardware- und Software-Implementierungen verglichen. Weiterhin wird ein Ausblick auf die nächsten Board-Generationen gegeben.

2. ASICs zur Bildverarbeitung

2.1 Histogrammprozessor-ASIC

Die Berechnung von Histogrammen ist eine wichtige Basisoperation der digitalen Bildverarbeitung. Hier soll die Implementierung eines verlustleistungsarmen Histogrammprozessor-

ASICs vorgestellt werden. Der Prozessor wurde für Grauwertbilder mit einer Auflösung von bis zu 4000 x 4000 Pixeln und einer Pixel-Wortbreite von 8 Bit implementiert. Er wurde so spezifiziert, dass Histogramm-Operationen mit einer Datenrate von 132 Mpx/s ausgeführt werden können. Die Architektur des Histogrammprozessors wird in [4] vorgestellt. Bild 3 zeigt ein Chipfoto und die technischen Daten des Histogrammprozessors.



Silizium-Fläche	ca. 13 mm ²
Max. Taktfrequenz (5 V; 3,3 V)	70 MHz; 60 MHz
Verlustleistungsaufnahme (5 V)	0,49 W bei 70 MHz
Verlustleistungsaufnahme (3,3 V)	0,17 W bei 60 MHz
Transistoranzahl	ca. 50.000
Technologie	0,8 µm CMOS

Bild 3: Chipfoto des technische Daten des Histogramm-Prozessors¹

2.2 Zweidimensionales Transversalfilter

Eine parametrisierbare Architektur wurde für ein zweidimensionales Transversalfilter entwickelt. Diese Architektur ist sowohl parametrisierbar bezüglich der Wortbreite der Abtastwerte und der Koeffizienten als auch bezüglich der Größe des Filterfensters. Die Skalierbarkeit dieser Architektur ermöglicht einen effizienten Abtausch zwischen der Durchsatzrate und der erforderlichen Chipfläche indem das Geschwindigkeitspotenzial moderner Sub-µ-Technologien durch Timesharing von Schaltungsteilen ausgenutzt wird.

Für ein Filter mit einer variablen Fenstergröße von 3 x 3 bis 17 x 17 bei einer Wortbreite von 8 Bit wurde ein Transversalfilter mit sechs MAC-Einheiten (MAC = Multiply and Accumulate) implementiert. In einer 0,5-µm-CMOS-Technologie beträgt die Siliziumfläche des Datenpfades einschließlich des Synchronisationsnetzwerkes und der Addierer zur Ausnutzung der Kernel-Symmetrien ca. 3 mm². Der Zeilenspeicher für 16 Zeilen mit jeweils 768 Abtastwerten belegt eine Siliziumfläche von 6 mm². Das Filter arbeitet bei einer Taktfrequenz von 165 MHz. Bild 4 zeigt das Layout des zweidimensionalen Transversalfilters. Die Architektur dieses Filters wird in [5] beschrieben.

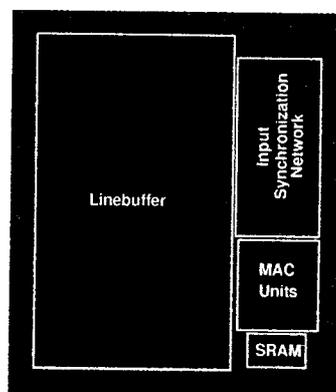
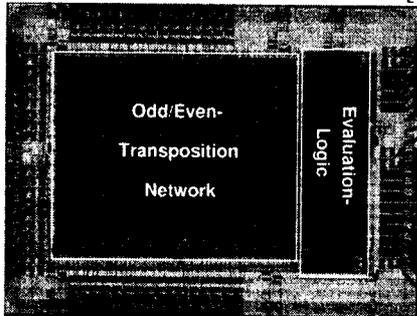


Bild 4: Layout des zweidimensionalen Transversalfilters

¹ Für alle in diesem Beitrag vorgestellten ICs gilt, dass sie auf Grund des vorhandenen Test-Equipments nur bis zu einer Taktfrequenz von ca. 60-70 MHz getestet werden konnten. Simulationen zeigten aber, dass sie auch bei der spezifizierten Frequenz korrekt arbeiten.

2.3 Zweidimensionales gewichtetes Medianfilter

Es wurde ein Sortiernetzwerk für ein gewichtetes Medianfilter mit einer maximalen Anzahl von 25 Eingangswerten, einer Wortbreite von 8 Bit/Pixel und 3 Bit Wortbreite pro Gewicht realisiert. Das ASIC wurde in einer 1- μ m-CMOS-Technologie gefertigt. Bild 5 zeigt ein Chipfoto dieses ASICs und gibt die technischen Daten zu diesem Baustein an. Die Architektur dieses Medianfilters wird in [5] beschrieben.



Silizium-Fläche	3,5 x 2,3 mm ²
Max. Taktfrequenz (5 V)	60 MHz
Max. Taktfrequenz (3,3 V)	55 MHz
Verlustleistungsaufnahme (5 V)	0,77 W bei 60 MHz
Verlustleistungsaufnahme (3,3 V)	0,26 W bei 55 MHz
Transistoranzahl	ca. 46.000
Technologie	1 μ m CMOS

Bild 5: Chipfoto und technische Daten des Medianfilters

3. Textprozessor-ASIC

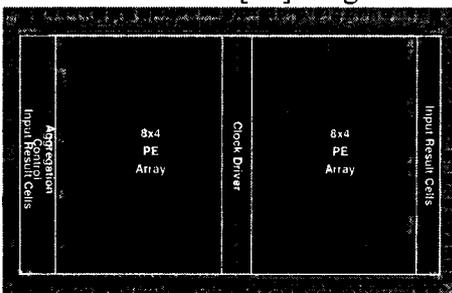
3.1 ASIC-Implementierung

Zur Anwendung im bereits erwähnten HPQS wurde ein dedizierter Prozessor zur approximativen Textsuche entworfen und implementiert. Die maximale Durchsatzrate bei einer parallelen Suche nach acht Suchworten wurde dabei auf die maximale Durchsatzrate des PCI-Interfaces, die 132 Mio. Zeichen pro Sekunde beträgt, begrenzt. Der implementierte Algorithmus basiert auf dem Prinzip der dynamischen Programmierung nach [8] und beinhaltet die folgenden Erweiterungen:

- Behandlung von Wildcards
- Umcodierung des Textes, um spezielle idiomatische Konstrukte zu behandeln (z.B. Umlaute, landesspez. Sonderzeichen, æ, etc.)
- Integration eines „Multi-Token Matching“

Das Prinzip der Berechnung der sogenannten „Edit-Distance“, das in diesem Prozessor realisiert wurde, wird in [9] beschrieben.

Bild 6 zeigt ein Chipfoto und die gemessenen technischen Daten des produzierten Textprozessor-ASICs, der ein Array von 8 x 8 Prozessorelementen enthält. Die Architektur dieses Prozessors wird in [10] vorgestellt.



Silizium-Fläche	ca. 53 mm ²
Max. Taktfrequenz (5 V)	60 MHz
Verlustleistungsaufnahme (5 V)	1,8 W bei 60 MHz
Transistoranzahl	ca. 340.000
Technologie	0,6 μ m CMOS

Bild 6: Chipfoto und technische Daten des Textprozessor-ASICs

3.2 Performance

Um die verschiedenen in der Literatur vorgestellten Lösungen zur approximativen Textsuche bezüglich ihrer Performance zu vergleichen und ihre Antwortzeiten abzuschätzen, wurde der folgende Benchmarking-Test durchgeführt (siehe auch [9]):

Ein Text-File der Größe 1 MByte wurde nach acht Suchworten, die jeweils aus acht Zeichen bestehen, durchsucht. Die Suche in diesem Text liefert dabei 500 Treffer. Alle Lösungen, die analysiert wurden, basieren dabei auf dem Prinzip der dynamischen Programmierung. Daher besitzt die maximal zugelassene Anzahl von Fehlern (innerhalb sinnvoller Grenzen) keinen Einfluß auf die Durchsatzrate. Diese Bewertung berücksichtigt die folgenden Systeme:

- Eine General-Purpose-Prozessor-basierte Lösung, wie sie in Computer-basierten Multimedia-Systemen eingesetzt wird. In diesem Experiment wurde eine Implementierung auf einer SUN Workstation mit einem Ultra SPARC I Prozessor, der mit 167 MHz Taktfrequenz arbeitet, betrachtet (**SPARC**).
- Ein VLIW (Very Long Instruction Word) basierter Signalprozessor, wie er in Multimedia-Systemen für verschiedene Aufgaben eingesetzt wird. Der Prozessor, der für diese Bewertung verwendet wurde, war ein Philips Trimedia TM-1000 Signalprozessor [14] mit einer Taktfrequenz von 100 MHz. Die Software-Implementierung auf diesem Prozessor erzielte dabei einen Parallelitätsgrad (ILP = Instruction Level Parallelism) von 3 (**TM 1**).
- Ein System, basierend auf einem Signalprozessor der nächsten Generation. Hierzu wurde die Performance der TM-1000 Implementierung auf die angekündigten nächsten Generationen mit einer 64-Bit Architektur und einer Taktfrequenz von 166 MHz skaliert. Es wurde ein maximaler ILP von 5 angenommen (**TM 2**).
- Ein der Literatur entnommener Wert für ein PLD-basiertes System [13], das zur Suche in DNA-Sequenzen in Gen-Datenbanken verwendet wird. In jedem PLD werden bei diesem System vier Prozessorelemente implementiert und eine Zusatzkarte für ein Rechnersystem enthält vier PLDs, die bei einer Taktfrequenz von 25 MHz arbeiten (**PLD**).
- Ein System basierend auf dem im Abschnitt zuvor dargestellten Textprozessor-ASIC (**ASIC**).
- Zu einem weiteren Vergleich wurde die Durchsatzrate einer optimierten Software-Lösung nach [7] herangezogen. Die Werte wurden dabei skaliert auf eine Suche nach acht Suchbegriffen mit jeweils acht Zeichen pro Suchwort (**Opt. SW**).

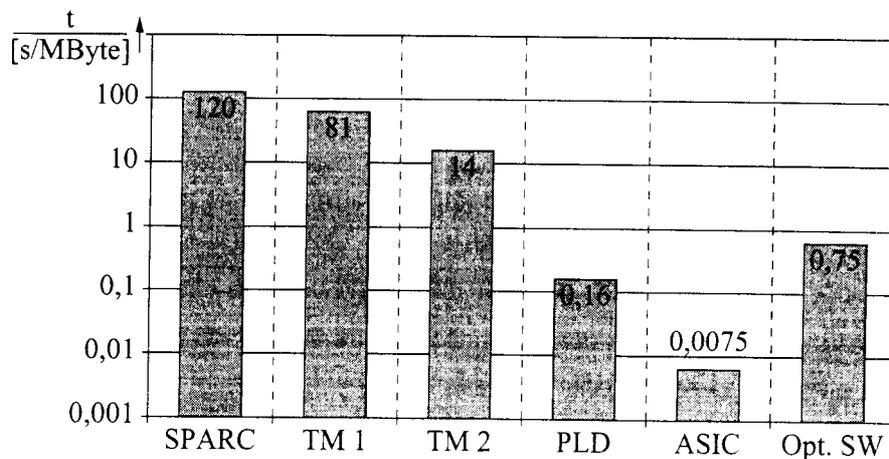


Bild 7: Ausführungszeit der Textsuche auf verschiedenen Testsystemen

Bild 7 stellt die Ausführungszeit der exemplarischen Textsuche auf den zuvor genannten Systemen dar. Weder die Antwortzeiten der General-Purpose- noch der Signalprozessor-basierten Lösung stellen akzeptable Werte für einen Anwender dar. Dieses gilt auch für die PLD-basierte Lösung, wenn Dateigrößen von mehreren 10 MByte verwendet werden. Eine weitergehende Analyse ist [9] zu entnehmen.

4. Coprozessor-Boards

4.1 Flexibles CardBus-basiertes Evaluationssystem

Einen ersten Schritt hin zu einer rekonfigurierbaren Rechenplattform bildete die Implementierung eines *flexiblen* Coprozessor-Boards, das die Integration der zuvor diskutierten ASICs in ein PCI-basiertes System gestattet. Die Systemarchitektur des entwickelten Coprozessor-Boards ist in Bild 8 dargestellt. Es wird hier die 32 Bit breite CardBus-Schnittstelle verwendet, um die Control- und Datenübertragung zwischen dem Host-System und dem Board zu realisieren. Da das CardBus-Interface ein synchrones Protokoll mit einer Taktfrequenz von 33 MHz verwendet, sind Datenübertragungen bis zu einer Datenrate von 132 MByte/s möglich. Ein CMOS-SRAM-basiertes Embedded Programmable Logic Device (EPLD) dient hier als Controller-Baustein für das Coprozessor-Board und steuert den Datenaustausch zwischen dem CardBus und den auf dem Board integrierten Komponenten. Die Konfiguration des EPLD wird in einem separaten Flash Memory Device abgespeichert, welches im System über eine JTAG-Verbindung programmiert werden kann.

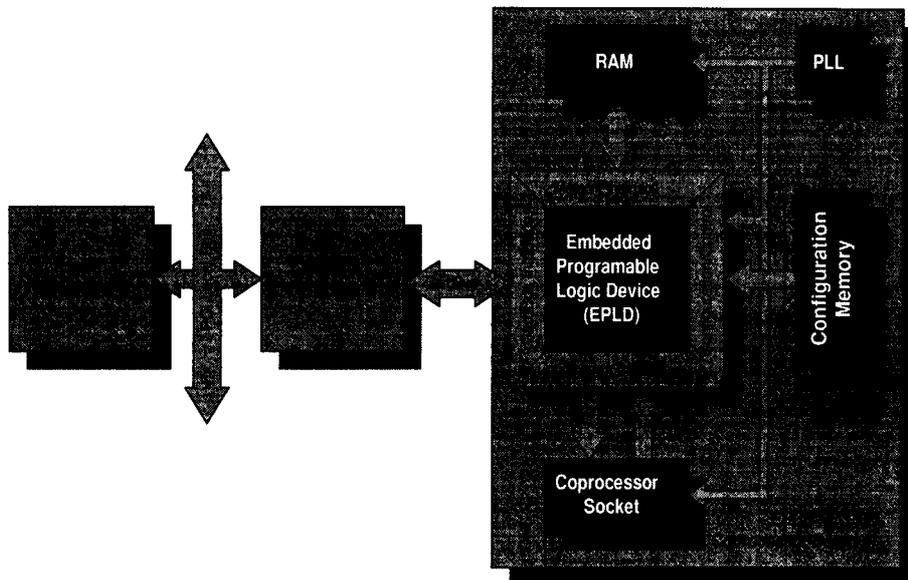


Bild 8: Architektur des entwickelten Coprozessor-Boards

Das Coprozessor-Board ist weiterhin mit 32 MByte SDRAM ausgestattet, die als On-board-Datenspeicher und Ergebnisspeicher dienen. Dieser Speicher kann dazu verwendet werden, das ASIC mit einer konstanten Datenrate zu versorgen, die über der maximalen Datenrate des CardBus liegt.

Das ASIC auf dem Coprozessor-Board ist in der ersten Version der Karte (Evaluationssystem) auf einem Sockel montiert, sodass die zuvor erwähnten Coprozessoren ausgetauscht werden können und die Bild- und Textverarbeitungsfunktionalitäten durch Austausch der

Prozessoren und Umprogrammierung des EPLD auf einem Evaluationsboard realisiert werden können.

Tabelle 1 stellt die technischen Daten des flexiblen Coprozessor-Boards zusammen.

Interface	CardBus PC Card 32 Bit gemultiplexer Address/ Daten-Bus 33 MHz synchr. Protokoll, Max. Bandbreite: 132 MByte/s
Controller	CMOS SRAM basierter EPLD, bis zu 158.000 nutzbare Gatter 24 kbit integriertes SRAM, 189 User I/Os
Speicher	32 MByte SDRAM, 2x2.097.152x4 Bänkex16 Bit
Taktfrequenz	33 MHz
Versorgungsspannung	5 V extern oder 3,3 V CardBus
Verlustleistungsaufnahme ²	2,3 W

Tabelle 1: Technische Daten des entwickelten Coprozessor-Evaluations-Boards

4.2 CardBus-basierte Spezialkarte für mobile Systeme

Als Anwendungsbeispiel für eine *dedizierte* Coprozessor-Karte, die lediglich eine Funktionalität aufweist, wurde das flexible Evaluationssystem weiter integriert und der Textprozessor fest auf diesem System installiert. Um sehr kleine Abmessungen der Coprozessor-Karte zu erzielen, die den Einsatz dieser Karte in mobilen Systemen ermöglichen, wurden moderne Gehäuseformen wie z.B. ein Ball Grid Array (BGA) für den Control-Baustein und ein sechslagiger Platinaufbau gewählt und die Karte somit nahezu auf Scheckkartengröße reduziert. Bild 9 zeigt ein Foto dieser Karte.

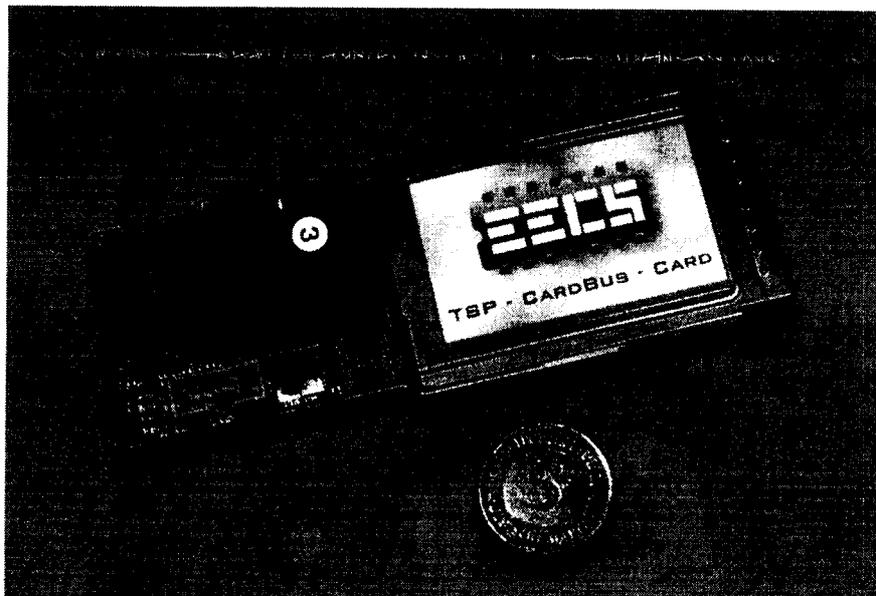


Bild 9: Board-Foto der Textprozessor-Karte

² mit dem Textprozessor als Coprozessor auf dem Board installiert

Ein Anwender kann z.B. ganze Enzyklopädien auf dieser Karte speichern und sie via Internet aktualisieren und dann mit sehr hohem Datendurchsatz fehlertolerant durchsuchen. Bild 10 zeigt die hierzu entwickelte graphische Benutzeroberfläche, die unter dem Betriebssystem Linux eine komfortable Steuerung der Textprozessorkarte und somit eine komfortable Bearbeitung großer Textdatenbanken ermöglicht.

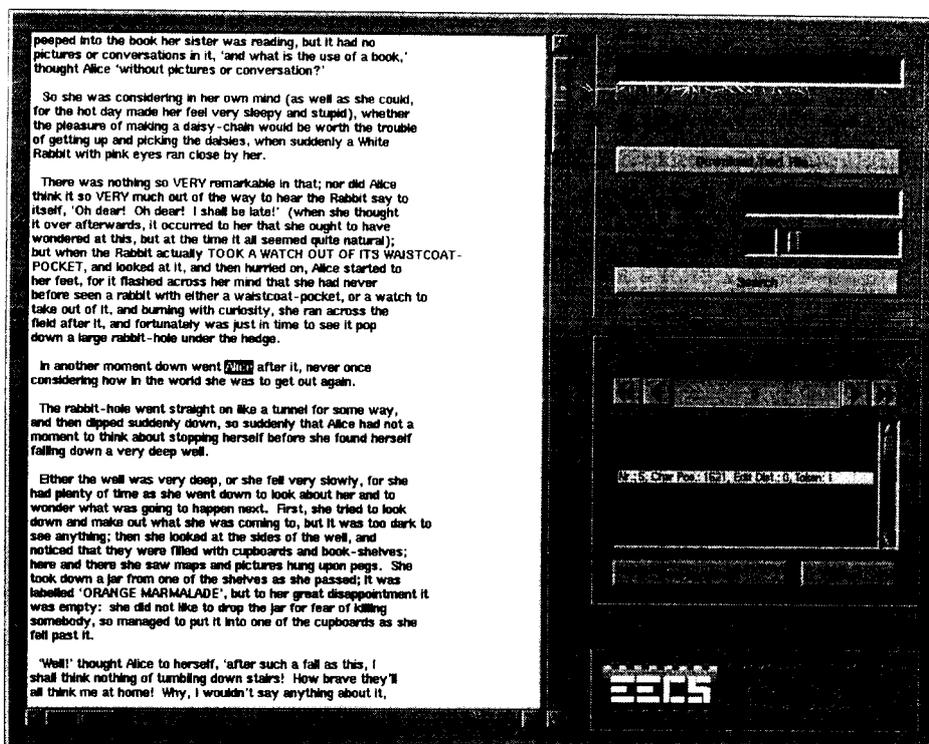


Bild 10: Benutzeroberfläche der Textprozessor-Karte

Das Ziel der zuvor geschilderten Arbeiten ist die eingangs erläuterte hybride rekonfigurierbare Multimedia-Plattform. Diese Plattform wird neben den dedizierten Coprozessoren zur Bild- und Textverarbeitung auch im System programmierbare EPLDs und in Hochsprache programmierbare DSPs enthalten. In solch einer Architektur können die dedizierten ASICs dann dazu eingesetzt werden, um zeitaufwendige Basis-Operationen wie eine Bildfilterung oder eine Histogrammbildung, die für nahezu alle Video-Algorithmen erforderlich sind, aus dem DSP auszulagern. Diese so entlasteten DSPs können dann dazu verwendet werden, effizienter weitere Multimedia-Funktionalitäten zu realisieren. Die integrierten EPLDs können auf Grund ihrer heute verfügbaren großen Komplexität sowohl dazu verwendet werden, die ASICs und die DSPs sowie den On-board Datenfluß zu steuern, als auch, um selber Basis-Operationen wie z.B. Farbraumtransformationen (z.B. YUV→RGB) auszuführen.

5. Zusammenfassung

Für viele zukünftige Multimedia-Anwendungen werden zukünftig rekonfigurierbare Hardware-Plattformen, die dedizierte ASICs mit ausreichender Rechenleistung zur Verfügung stellen, erforderlich sein. In diesem Beitrag wurde das Grundkonzept solcher hybrider rekonfigurierbarer Architekturen vorgestellt und diskutiert. Auf dem Weg zu diesen Plattformen wurden vier ASICs zur Bild- und Textverarbeitung präsentiert, die in diese Plattformen integriert werden können. Als ein erstes Anwendungsbeispiel wurde ein rekonfigurierbares Card-Bus-basiertes Coprozessor-Board vorgestellt, das aus Speicherbausteinen, einem EPLD für

Steuerzwecke sowie einem dedizierten Coprozessor besteht. Durch den Austausch des Coprozessors und die Umprogrammierung des Controller-Bausteins kann dieses Board an verschiedene Anwendungen angepasst werden. Eine zweite Version dieser Boards beweist, dass für mobile Anwendungen diese Funktionalitäten auch mit einem sehr kleinen Formfaktor realisiert werden können. Es wurde eine scheckkartengroße CardBus-Karte vorgestellt, die eine fehlertolerante Volltextsuche in abgespeicherten Textdatenbanken wie z.B. Enzyklopädien ermöglicht. Die weiteren Schritte hin zu einer leistungsfähigen rekonfigurierbaren Architektur wurden aufgezeigt.

Literatur

- [1] *DeHon, A.:* "The Density Advantage of Configurable Computing ", IEEE Computer, pp. 41-49, 2000
- [2] *Hauck, S.:* "The Roles of FPGAs in Reprogrammable Systems ", Proceedings of the IEEE, vol. 86, Nr. 4, pp. 615-638, 1998
- [3] *Knoll, A. et al.:* "An Integrated Approach to Semantic Evaluation and Content-Based Retrieval of Multimedia Documents ", Proceedings of ECDL 1998, LNCS, vol. 1513, pp. 409-428, Sept. 1998
- [4] *Blüthgen, H.-M.; Mallok, J.; Noll, T. G. :* "A Scalable Architecture for Low-Power and High-Throughput Histogramming Applications", Proc. of Poster 98, Prag 1998
- [5] *Henning, C.; Noll, T. G. :* "Scalable Architectures and VLSI Implementations of High-Performance Image Processing Algorithms", Proc. of the International Conference on Imaging Science, Systems and Technology, Las Vegas 1999, pp. 191-197
- [6] *Wu, S.; Manber, U. :* "Fast Text searching allowing errors", Communications of the ACM, vol. 35, Oct. 1992, pp. 83-91
- [7] *Baeza-Yates, R.; Navarro, G. :* "Multiple Approximate String Matching", Proceedings of WADS 1997, LNCS, vol. 1272, pp. 174-184
- [8] *Wagner, R.; Fischer, A. J. :* "The string-to-string correction problem", Journal of the ACM, vol. 21, Jan. 1974, pp. 168-173
- [9] *Blüthgen, H.-M.; Blume, H.; Noll, T. G. :* "Hardware-Implementierung für die approximative Textsuche in multimedialen Anwendungen" ITG-Fachtagung Multimedia: Anwendungen, Technologie, Systeme, 27.-29.9.1999, Dortmund, pp. 229-235
- [10] *Blüthgen, H.-M.; Noll, T. G. :* "A Programmable Processor for Approximate String Matching With High Throughput Rate, zur Veröffentlichung angenommen auf der International Conference on Application-specific Systems, Architectures and Processors, Boston, Massachusetts, 10.-12. Juli 2000, USA
- [11] *Hirschberg, J. D.; Hughey, R.; Karplus, K.; Speck, D. :* "Kestrel: A programmable array for sequence analysis", Proceedings of ASAP 1996, vol. 21, 1996, pp. 25-34
- [12] *Sastry, R.; Ranganathan, N.; Remedios, K. :* "CASN: A VLSI chip for approximate string matching", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, Aug. 1995, pp. 824-830
- [13] *Compugen :* "BioXLP, technical data", <http://www.cgen.com>, 1999
- [14] *Rathnam, S.; Slavenburg, G. :* "An Architecture overview of the programmable multimedia processor TM1000", Proc. Compcon, IEEE CS Press, 1996, pp. 319-326

Integrated CMOS-OPAMPs for Ultra-Low-Voltages

Fachhochschule Giessen – University of Applied Sciences Giessen

K. Leitis, W. Bonath, Holger Kemper, Matthias Weber

Kurzfassung

In diesem Beitrag wird die Entwicklung eines Differenzverstärkers dargestellt, der mit 1V Versorgungsspannung arbeitet. Es handelt sich hier um einen ersten Entwicklungsschritt. Die Schaltung wurde mit BSIM 3V3 Modellen simuliert, wobei die Transistoren im Sub-Threshold-Bereich betrieben werden.

1. Einleitung

Low-Voltage- und Low-Power-Anwendungen werden in zunehmendem Maße in elektronischen Geräten und Systemen eingesetzt. Hauptsächlich sind folgende Anwendungsgebiete zu nennen:

- Batteriebetriebene Systeme
- Energieautarke Systeme
- Spezialanwendungen (z.B. ein magnetoresistives Sensorsystem, bei dem der Signalrauschabstand bei einer Versorgungsspannung von 1 V besonders günstig ist)

Meist resultiert die Anforderung nach einer niedrigen Versorgungsspannung aus der Forderung nach einer reduzierten Verlustleistung. Bei digitalen Schaltungen wird die Leistungsaufnahme durch folgenden Zusammenhang ausgedrückt:

$$P \sim CV^2f$$

Die Verlustleistung P ist nicht nur von der Kapazität C und der Frequenz f , sondern insbesondere quadratisch von der Versorgungsspannung V abhängig. Dieses ist der wichtigste Grund für die Herabsetzung der Versorgungsspannung.

Folgendes Diagramm stellt den Entwicklungstrend in den vergangenen und in den kommenden Jahren dar:

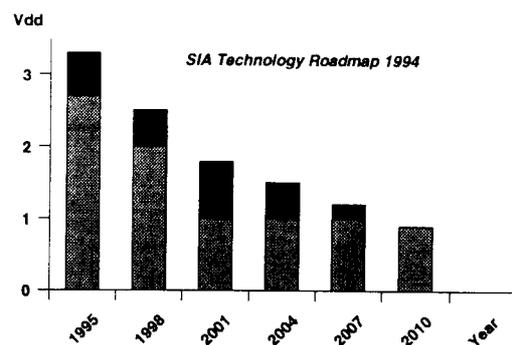


Abbildung 1: Reduzierung der Versorgungsspannung bei integrierten Schaltungen

Abbildung 1 stellt die unterschiedlichen Versorgungsspannungen für Analog- und Digital Schaltkreise dar, wobei die digitalen Schaltkreise zu einem früheren Zeitpunkt mit geringeren Versorgungsspannungen arbeiten werden. Einhergehend mit der Reduktion der Versorgungsspannung werden auch im Laufe der Zeit die Strukturabmessungen geringer. Jedoch kann die Zuverlässigkeit einer Schaltung sinken, wenn die Spannung an den Schaltungselementen zu groß wird. Transistoren können beschädigt oder zerstört werden.

Für Mixed-Signal-IC's ist eine einheitliche Versorgungsspannung vorteilhaft. Deshalb wird in jedem Fall der Betrieb von Mixed-Signal-ASICs bei geringen Versorgungsspannungen notwendig werden.

2. Problemstellung

Beim Entwurf von Analogschaltkreisen mit geringen Versorgungsspannungen treten

Probleme auf, die mit der herkömmlichen Schaltungstechnik nicht gelöst werden können. Deshalb sind neue Schaltungskonzepte für die analoge Signalverarbeitung notwendig. Das gewichtigste Problem stellt die, in bezug auf die Versorgungsspannung, relativ hohe Schwellspannung von Transistoren dar. Beispielsweise beträgt die Schwellspannung bei einem 2μ CMOS Prozeß $\pm 0,9V$. Bei einer Betriebsspannung von $5V$ ergibt sich ein Verhältnis von Betriebsspannung zur Schwellspannung von $5,5$. Neuere Prozesse wie z.B. ein $0,5\mu$ CMOS Prozeß hat eine Schwellspannung um $\pm 0,6V$. Bei einer Versorgungsspannung von $1V$ ergibt sich ein Quotient von $1,6$. Das Verhältnis verschlechtert sich demnach signifikant.

Meistens ist eine hohe Verstärkung bei OPAMPs wünschenswert, was in einer Vielzahl der Realisierungen mit Kaskodenschaltungen erreicht wird. Die Anzahl der Kaskodenstufen wird jedoch beim Low-Voltage-Design durch die Drain/Source-Sättigungsspannung und die Gate/Source-Spannung begrenzt. Für eine vergleichbar hohe Verstärkung werden mehrstufige Verstärker eingesetzt.

Wird die Versorgungsspannung reduziert, so reduziert sich auch der Eingangsarbeitsbereich von Verstärkern. Um diesen Arbeitsbereich so groß wie möglich zu gestalten, wird typischerweise eine komplementäre Eingangsstufe, bestehend aus parallel geschalteten PMOS/NMOS-Differenzverstärkern, eingesetzt. Die Steilheit einer komplementären Eingangsstufe ist in der Summe nicht konstant [1]. Deshalb sind auch hier neue Schaltungskonzepte notwendig, wie sie beispielsweise in [1] vorgestellt werden.

Der Gleichtakt-Arbeitsbereich ist je nach Operationsverstärkerbeschaltung unterschiedlich. So ist für einen invertierenden Verstärker nur eine geringe Gleichtaktaussteuerung um den Arbeitspunkt notwendig; hingegen benötigt ein Spannungsfolger einen großen Gleichtakt-Arbeitsbereich. Beim

Schaltungsentwurf ist deshalb der spätere Einsatz im ASIC zu berücksichtigen. Aufgrund der niedrigen Versorgungsspannung ist nicht nur der $0,9$ Eingangsspannungsarbeitsbereich geringer, sondern auch der Ausgangsspannungsarbeitsbereich. Um einen möglichst großen Aussteuerbereich zu erhalten, wird eine Rail-to-Rail-Ausgangsstufe benötigt. Besonders gut geeignet sind Verstärkerstufen im AB-Betrieb, die eine geringe Signalverzerrung aufweisen und welche hohe Lasten treiben können. Für Switched-Capacitor-Anwendungen sind auch Verstärkerstufen im A-Betrieb geeignet, welche allerdings einen relativ hohen Ruhestrom im Arbeitspunkt aufweisen.

Der Eingangsspannungsoffset von CMOS-Differenzverstärkern wird mit neueren Technologien und geringeren Versorgungsspannungen nicht kleiner. Das Verhältnis von Ausgangsaussteuerbereich zum Eingangsoffset wird ungünstiger. Deshalb empfiehlt sich der Einsatz von Switched-Capacitor-Schaltungen, um den Eingangsspannungsoffset weitgehend zu kompensieren.

Ein weiteres Problem stellen Analogschalter dar, deren klassisches Schaltungskonzept (zwei parallelgeschaltete NMOS/PMOS-Transistoren) für $1V$ Versorgungsspannung ungeeignet sind.

Aus den genannten Problemstellungen lassen sich einige Forderungen zum Schaltungsentwurf von Low-Voltage-CMOS-OPAMPs herleiten.

- Rail-to-Rail Eingangsstufe
- kontinuierlicher Gleichtakteingangsspannungsbereich
- Versorgungsspannung so gering wie möglich
- Großes Verstärkungsbandbreiteprodukt
- Steuerschaltung zur Linearisierung der Steilheit einer komplementären Rail-to-Rail-Eingangsstufe
- Rail-to-Rail-Ausgangverstärker im AB-Betrieb
- Kompensation des Operationsverstärkeroffsets mit SC-Schaltungstechnik

3. Grundlagen zum Entwurf

Sättigung der Transistoren

Die Transistoren eines Verstärkers werden gesättigt betrieben. Folgende Darstellung zeigt das Kennlinienfeld eines NMOS-Transistors bei dem der Drainstrom I_D (Ordinate) als Funktion der Drain/Source-Spannung V_{DS} (Abszisse) aufgetragen ist. Als Parameter wurde die Gate/Source-Spannung V_{GS} eingesetzt.

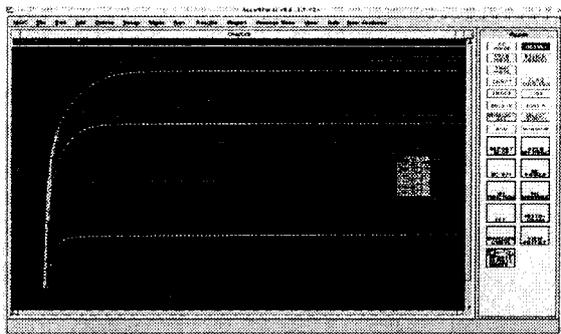


Abbildung 2: $I_D = f(V_{DS}, V_{GS})$

Aus Abbildung 2 geht die Sättigungsspannung hervor. Diese ist im Subthreshold- bzw. Weak-Inversion-Bereich kleiner als im Strong-Inversion-Bereich. Daher ist der Weak-Inversion-Bereich für Schaltungen gut geeignet, wenn die Schaltung mit geringen Drain/Source-Sättigungsspannungen arbeiten soll, wie es bei geringen Versorgungsspannungen der Fall ist.

Strong-Inversion

Die Sättigungsspannung V_{DS} im Strong-Inversion-Bereich ergibt sich aus folgendem Zusammenhang (1):

$$V_{DS} > V_{GS} - V_t \quad (1)$$

Die Schwellspannung bzw. Threshold-Voltage V_t setzt sich zusammen aus dem technologieabhängigen Schwellspannungswert V_{t0} und einer additiven Komponente, dem sog. Body-Effekt. Dieser Zusammenhang wird in Gl. (2) beschrieben.

$$V_t = V_{t0} + \gamma (\sqrt{V_{SB} + 2|\phi_f|} - \sqrt{2|\phi_f|}) \quad (2)$$

In Gl. (2) ist γ die Body-Effekt-Konstante und ϕ_f das Fermipotential des Halbleiters. Für den Fall das die Source/Bulk-Spannungen $V_{SB}=0$ ist gilt:

$$V_t = V_{t0} \big|_{V_{SB}=0} \quad (3)$$

Deshalb sollte für den Entwurf von Schaltungen mit geringen Versorgungsspannungen zum einen eine Technologie gewählt werden, welche ein geringes V_{t0} aufweist, und zum anderen sollte der Body-Effekt vermieden werden.

Der Strom I_d ist eine Funktion der effektiven Gate/Source-Spannung $V_{GS,eff}$. Wobei gilt: $V_{GS,eff} = V_{GS} - V_t$. Im Strong-Inversion-Bereich wird I_d durch folgende Gleichung (4) ausgedrückt

$$I_d = \frac{1}{2} \mu C_{ox} \frac{W}{L} V_{GS,eff}^2 \quad (4)$$

Weak-Inversion

Im Weak-Inversion-Bereich ist das Transistorverhalten anders. Transistoren gehen in die Sättigung, wenn folgende Bedingung (5) erfüllt ist:

$$V_{DS} > 3 \text{ bis } 4 V_T \quad (5)$$

Die Spannung V_T ist die Temperaturspannung, welche sich aus Gleichung (6) ergibt.

$$V_T = \frac{kT}{q} \quad (6)$$

In Gleichung (6) ist die Boltzmannkonstante k , die Temperatur T und die Elektronenladung q enthalten. Anders als im Strong-Inversion-Bereich (vgl. Gl. (1)) ergibt sich keine Abhängigkeit von V_{GS} und der Schwellspannung V_t . Bei Raumtemperatur tritt die Sättigung bei ca. 100 mV ein.

Der Drain-Strom I_d im Weak-Inversion-Bereich ist in Gleichung (7) dargestellt.

$$I_D = k_x \cdot \frac{W}{L} e^{\frac{V_{gs}}{nV_T}} \cdot \left(1 - e^{-\frac{V_{ds}}{V_T}}\right) \quad (7)$$

Bulk-gesteuerte Schaltungstechnik

1987 veröffentlichten Guzinski u. a. [2] eine Methode über Bulk-gesteuerte Differenzverstärker. Ballock setzte dieselbe Technik für einen Low-Voltage-CMOS Operationsverstärker ein [3]. Das Prinzip eines bulk-gesteuerten MOSFET ist in folgender Abbildung 3 dargestellt.

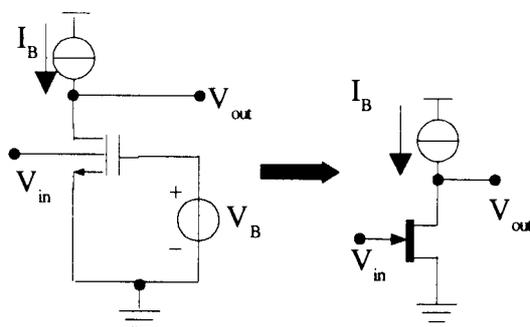


Abbildung 3: Bulk-gesteuerter MOSFET (links), Bulk-gesteuerter MOSFET verhält sich ähnlich wie einem JFET (rechts)

Abbildung 3 zeigt einerseits die Beschaltung eines bulk-gesteuerten MOSFETs und andererseits die elektrische Verhaltensähnlichkeit eines bulk-gesteuerten MOSFETs mit einem JFET oder einem selbstleitenden Depletion-MOSFET.

Vorteilhaft für die Low-Voltage-Schaltungstechnik ist, dass bulk-gesteuerte MOSFETs selbstleitend sind. Die Schwellspannung tritt in einem nicht relevanten Bereich der Kennlinie auf. Dadurch kann die Versorgungsspannung auf ein Minimum reduziert werden, ohne an die Grenzen der Schwellspannung zu stoßen.

Mit dieser Technik gehen jedoch auch in bezug auf gate-gesteuerten Transistoren Nachteile einher:

- Die Transistorsteilheit ist geringer
- Verstärkungsbandbreite ist geringer
- Eingangsrauschen ist größer
- Für einen P- (N-) well Prozeß ist nur ein N- (P-) Kanal-Transistor mit Bulk-Steuerung verfügbar
- Schaltungen neigen zum Latch-up

Gleichtakt-Arbeitsbereich von Differenzeingangsstufen

Der Gleichtakt-Arbeitsbereich von P-Kanal- bzw. N-Kanal- Differenz-CMOS-Eingangsstufen ist unterschiedlich. Folgende Abbildung 4 stellt die Unterschiede dar:

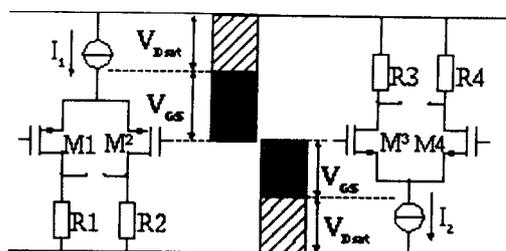


Abbildung 4: Gleichtakt-Arbeitsbereich

Die Differenzeingangsstufen in Abbildung 4 haben einen Gleichtaktbereich der durch die Gate/Source-Spannung V_{GS} und die Drain/Source-Sättigungsspannung V_{Dsat} eingeschränkt wird. Werden beide Differenzeingangsstufen gleichzeitig eingesetzt, wird ein Gleichtaktbereich von der negativen bis positiven Versorgungsspannung möglich. Überschneiden sich jedoch die Bereiche der Gleichtaktbereichseinschränkung von der PMOS- und der NMOS-Differenzeingangsstufe, dann tritt eine Lücke im Gleichtaktbereich auf. Differenzeingangsstufen die einen Eingangsgleichtaktbereich aufweisen, der von der negativen Versorgungsspannung bis zur positiven Versorgungsspannung reicht, werden Rail-to-Rail Eingangsstufen genannt.

Die minimale Versorgungsspannung $V_{supply,min}$ läßt sich gem. Gleichung 8 berechnen [1]:

$$V_{supply,min} = 2(V_{GS} + V_{Dsat}) \quad (8)$$

Wird nur eine Eingangsstufe von Abbildung 4 eingesetzt, dann reduziert sich die minimal mögliche Versorgungsspannung gem. Gleichung 9 auf:

$$V_{supply_{min}} = V_{GS} + V_{Dsat} \quad (9)$$

Schaltungen mit extremen Anforderungen an eine niedrige Versorgungsspannungen benötigen demnach gem. Gleichung 9 nur die Hälfte an Versorgungsspannung.

Rail-to-Rail Eingangstufen werden aus zwei komplementären Stufen aufgebaut, deren Ströme in einer folgenden Stufe addiert werden. Die Steilheit der Gesamteingangsstufe ist je nach Betriebsart unterschiedlich. Bei Betrieb im Weak-Inversion-Bereich berechnet sich die Steilheit gem. folgender Gleichung 10 [3].

$$g_{mi,weak} = \frac{I_p}{2n_p V_{th}} + \frac{I_n}{2n_n V_{th}} \quad (10)$$

Die Steilheit gem. Gl. 10 ist neben den Steigungsfaktoren (Weak-Inversion) n_p und n_n sowie der Schwellspannung insbesondere von der Strömen durch die Differenzstufen I_p und I_n abhängig. Für eine große Steilheit sind möglichst große Ströme zu wählen. Werden die Ströme jedoch zu groß, wechselt die Betriebsart in den Strong-Inversion-Bereich. Die funktionale Abhängigkeit der Steilheit im Strong-Inversion-Bereich ist in Gleichung 11 [3]

dargestellt.

$$g_{mi,strong} = \sqrt{\mu_p C_{ox} \left(\frac{W}{L}\right)_p I_p} + \sqrt{\mu_n C_{ox} \left(\frac{W}{L}\right)_n I_n} \quad (11)$$

Wird Gleichung 11 mit Gleichung 10 verglichen, so zeigt sich, daß im Strong-Inversion-Bereich über die Transistorgeometrie (W/L) und über die Ströme durch die Differenzstufe (I_p, I_n) auf die Steilheit Einfluß genommen werden kann. Die Löcher- und Elektronenbeweglichkeit μ_p und μ_n sowie die Kapazitätsdichte C_{ox} sind Technologieparameter auf die beim Entwurf kein Einfluß genommen werden kann.

4. Realisierter Entwurf

Realisiert wurde ein Differenzverstärker mit einer komplementären PMOS/NMOS-Differenzeingangsstufe, um einen Rail-to-Rail-Gleichtaktarbeitsbereich zu erhalten. Die Architektur des Verstärkers ist in Abbildung 5 dargestellt.

Als Last wird bei der Differenzverstärkerstufe keine aktive Stromspiegellast eingesetzt, sondern eine gefaltete Kaskodenstufe, um Rail-to-Rail-Eigenschaften erhalten zu können. Denn Stromspiegellasten reduzieren den Gleichtaktbereich drastisch.

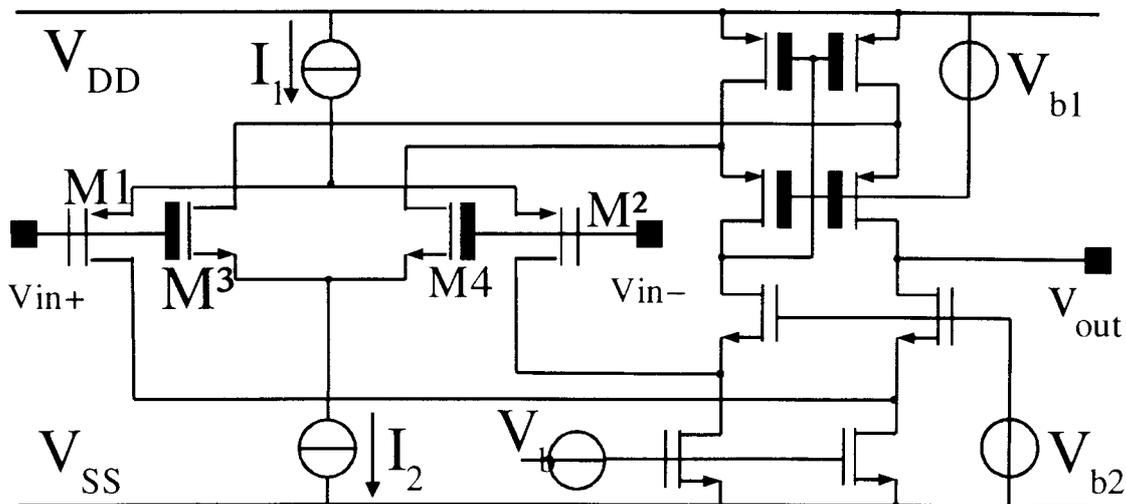


Abbildung 5: Realisierter OPAMP

Die gefaltete Kaskodenschaltung summiert die Ströme der beiden Eingangsstufen. Die Transistoren, die mit V_b vorgespannt werden, arbeiten als Stromquellen. Mit V_{b2} erhalten die Kaskodentransistoren eine Vorspannung, so dass bzgl. der Differenzstufenquerströme der doppelte Strom durch die Stromquellen der Kaskodenschaltung fließt. Der Low-Voltage Stromspiegel addiert die Ströme aus der Eingangstufe.

Für die Simulation wurden BSIM 3V3-Modelle der Transistoren eingesetzt. Als Simulationsergebnis ergibt sich das in Abbildung 6 dargestellte Übertragungsverhalten.

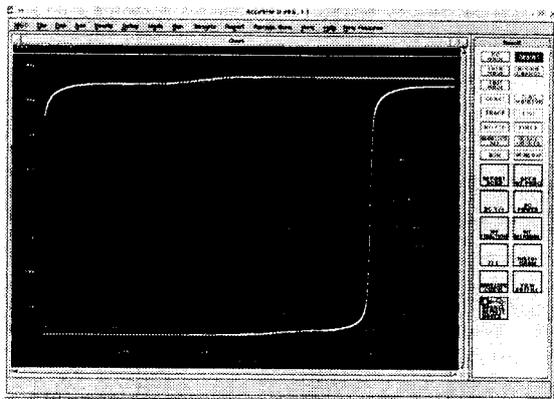


Abbildung 6: Übertragungsverhalten des Low-Voltage OPAMP

Erkennbar ist in Abbildung 6, dass der Eingangsgleichtaktarbeitsbereich bis zu beiden Versorgungsspannungen ($V_{ss}=0V$, $V_{dd}=1V$) reicht. Jedoch erreicht die Ausgangsspannung nicht die Versorgungsspannungsgrenzen, weil noch keine Rail-to-Rail-Ausgangsstufe integriert ist. Eine Nichtlinearität überlagert das Ausgangssignal, welche an unterschiedlichen Steilheiten und uneinheitlichen Rails der Kennlinien deutlich wird. Mit einer Steuerschaltung kann die Steilheit der Eingangsstufe konstant gehalten werden, welche hier noch nicht integriert ist.

5. Zusammenfassung

Dieser Beitrag gibt einen Überblick für den Entwurf elektronischer Schaltungen, welche mit sehr niedriger Versorgungsspannung arbeiten. Dazu wird das Problem dargestellt und eine Anleitung für den Entwurf gegeben.

Ein erster OPAMP-Entwurf mit einer Rail-to-Rail-Eingangsstufe und einer gefalteten Kaskodenstufe zeigt den Betrieb der Eingangsstufe im Subthresholdbereich bei einer Betriebsspannung von 1 V.

Im nächsten Schritt ist eine Steuerschaltung für eine konstante Steilheit der Eingangsstufe und eine Rail-to-Rail-Ausgangsstufe zu implementieren. Ebenso sind bulk-gesteuerte Verstärker in der Entwicklung.

6. Literatur

- [1] R. Hogervorst, J.H. Huijsing, „Design of Low-Voltage, Low-Power Operational Amplifier Cells, Kluwer Academic Publishers, Dordrecht, 1996
- [2] A. Guzinski, M. Bialko, and J.C. Matheau, „Body-driven differential amplifiers for application in continuous-time active-C filter“, Proc. European Conf. Circuit Theory and Design (ECCTD '87), pp.315–320, 1987
- [3] B.J. Blalock, „Designing 1-V Op Amps Using Standard Digital CMOS Technology, IEEE Transactions on Circuits and Systems-II, Vol.45, No. 7, July 1998

Festkomma-Arithmetik für DSP-Anwendungen mit redundanten Zahlendarstellungen

B. Hoppe, M. Loy und H. Meuth

Fachbereich Elektrotechnik, FH Darmstadt,
Schöfferstr. 3, 64295 Darmstadt, F. R. G.,
Tel.: +49 6151 168 322, FAX: +49 6151 168 934
e-mail: hoppe@fh-darmstadt.de

Zusammenfassung

In diesem Artikel werden neuere Entwicklungen bei der Verwendung von *Signed Digit* (SD)-Zahlen vorgestellt. Es wird gezeigt, daß in VHDL beschriebene SD-Arithmetikblöcke deutliche Vorteile im Vergleich zu Gatterimplementierungen von etablierten binären Hochgeschwindigkeitsarchitekturen aufweisen. Als Fallstudien werden Addierer/Subtrahierer, 3-Summandenaddierer, Multiplizierer, Akkumulatoren und $\Delta\Sigma$ -Stufen präsentiert. Die Vorteile von solchen redundanten Lösungen zeigt sich, wenn mehrere SD-Operationen verwendet werden können, um eine komplexere Signalverarbeitungsaufgabe zu lösen. Der Geschwindigkeitsvorteil liegt bei 20 - 30%, wobei der Zuwachs der Chipfläche moderat bleibt. In einigen Fällen ergibt sich sogar ein kleiner Flächenvorteil. Diese Aussagen gelten inklusive der Konversionsblöcke zwischen binärer und redundanter Zahlendarstellung.

1. Einleitung

Eine Zahl hat einen Wert und eine bestimmte Darstellung (Binärdarstellung, Dezimaldarstellung, etc.). Es gibt verschiedene Zahlenfamilien (ganze Zahlen, rationale Zahlen, komplexe Zahlen, etc.). Im folgenden werden wir uns auf die ganzen binären Zahlen konzentrieren, die durchgängig für schnelle Signalverarbeitungsalgorithmen in festverdrahteten CMOS-Schaltkreisen Anwendung finden. Ganze Zahlen werden mit Ziffern dargestellt, die je nach Position und Darstellung bestimmte Wertigkeiten haben. Der Wert Z der Zahl ist dann die Summe der Ziffern multipliziert mit den jeweiligen Wertigkeiten. Der Wert einer positiven Binärzahl $A = a_{n-1} a_{n-2} \dots a_1 a_0$, $a_i \in (0,1)$, ist beispielsweise

$$A = \sum_{i=0}^{n-1} a_i \cdot 2^i.$$

Die bekannten Zahlendarstellung (binär, dezimal) sind eindeutig und nicht redundant, d. h. sie kommen mit einer minimalen Menge an Ziffern aus. Die Ziffern einer Zahl sind positiv. Negative Zahlen werden über ein Vorzeichen dargestellt oder bei den Zweierkomplementzahlen über ein Vorzeichenbit mit entsprechendem Gewicht. Bei redundanten Zahlen fehlt diese Eindeutigkeit und hier sind bereits die einzelnen Ziffern vorzeichenbehaftet. Man spricht deshalb von der *Signed-Digit* Darstellung (SD-Zahlen).

Bekanntermaßen lassen sich alle arithmetische Operationen mit Festkomma-Binärzahlen auf Additionen zurück führen. Bei den heute üblichen längeren Zahlenformaten (16 Bit, 32 Bit, 64 Bit) bestimmen die Laufzeiten für die Überträge die Dauer des Additionsvorgangs und damit den Datendurchsatz in arithmetischen Modulen. Wie bei dezimalen Zahlen können diese Überträge im ungünstigsten Fall auch bei Binärzahlen von der niederwertigsten bis zur höchstwertigsten Stelle propagieren. Die besondere Bedeutung von SD-Zahlen liegt nun nach Avizienis [1] darin, daß sich in SD-Darstellung potentielle Überträge bei einem Additionsvorgang auf die nächste Stelle beschränken lassen (total parallele Addition/Subtraktion), siehe Bild 1.

Bevor aber diese günstige Eigenschaft genutzt werden kann, müssen die in der Regel binär vorliegenden Zahlen aus der Zweierkomplementdarstellung in die redundante SD-Darstellung gewandelt werden. Nach Abschluß der Rechenoperation sind dann die Ziffern des SD-Ergebnisses in die Binärdarstellung zurück zu wandeln, damit diese Daten weiterverarbeitet werden können. Dieser Konversionsaufwand erschwert die Nutzung von SD-Algorithmen. Deshalb hat sich dieses Rechenverfahren noch nicht im erwarteten Maß durchgesetzt. Es kommt daher besonders darauf an, effiziente Konversionsverfahren zu verwenden und bei der Darstellung der SD-Ziffern mit möglichst wenig Bits auszukommen.

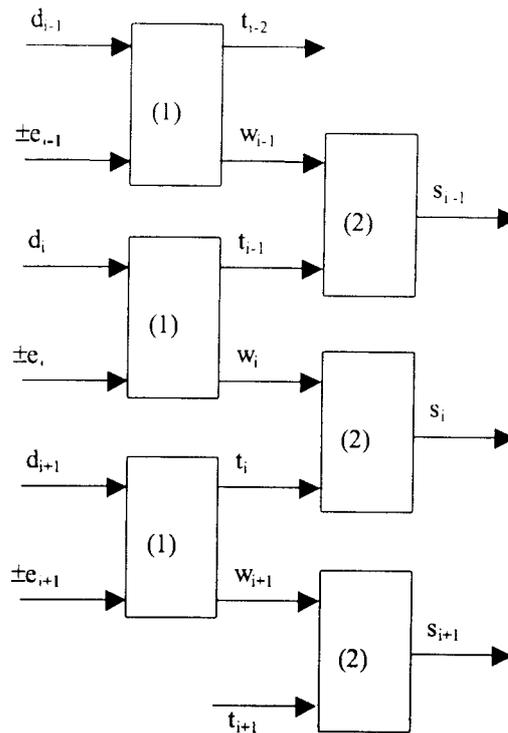


Bild 1 Total parallele Addition/Subtraktion von drei benachbarten Ziffern der SD-Zahlen $E = e_{n-1} \dots e_0$ und $D = d_{n-1} \dots d_0$. Überträge t_{i+j} propagieren nur an die nächste Stelle. Summen/Differenzen $S = s_{n-1} \dots s_0$ können daher parallel berechnet werden. w_j bezeichnen Zwischensummenziffern (siehe Text).

Zur konsistenten Bewertung der Vor- und Nachteile von Algorithmen mit redundanten Zahlen benutzen wir im folgenden den Vergleich mit kommerziell verfügbaren Intellectual Property Zellen (IP-Soft Cores) für Hochgeschwindigkeitsanwendungen aus der *Design Ware* (DW) Bibliothek des CAE-Software-Anbieters *Synopsys*. SD-Blöcke werden mit VHDL modelliert, dann auf eine $0,7\mu\text{m}$ CMOS Zieltechnologie durch Logiksynthese abgebildet und mit den auf die gleiche Technologie hin synthetisierten DW-Blöcken verglichen, auf der Basis von Synopsys Design Compiler Flächen- und Laufzeit- Reports. Neben elementaren arithmetischen Operationen, wie z. B. Addition/Subtraktion, Multiplikation usw. werden auch größere Funktionsblöcke (Multiply Accumulate Stufen, MAC) oder Dreisummanden-Addierer betrachtet. Abschließend untersuchen wir komplexe DSP-Funktionen, wie Filter oder $\Delta\Sigma$ -Modulatoren.

2. Konversion und Rückkonversion

In einer konventionellen Zahlendarstellung zur Basis $r > 1$ kann jede Ziffer genau r positive Werte annehmen ($0, 1, 2, \dots, r-1$), in einer *SD-Darstellung* hingegen sind für jede Ziffer positive und negative Werte möglich. Der maximale Betrag pro Ziffer ist $r-1$, und damit gibt es $r-1$ zusätzliche Möglichkeiten jede Ziffer zu wählen. Deshalb sind die SD-Zahlen redundant.

Um mit SD-Zahlen in Hardware zu rechnen, sind die einzelnen Ziffern mit Bits darzustellen. Die üblichen Zahlenbasen r (Radices) sind Potenzen von 2, weil sich dann die SD-Ziffern effizient als Zweierkomplementzahlen für eine binäre Verarbeitung darstellen lassen. Für die Basis $r = 2^n$ sind $n+1$ Bits pro Stelle erforderlich. Weicht man von Zweierpotenzen ab, dann ist die binäre Darstellung der SD-Ziffern nicht effizient. In den letzten Jahren [2-6] hat sich insbesondere gezeigt, daß $r = 2$ im Vergleich zu höheren Radices die besten Ergebnisse liefert, weil dann 2 bits pro Stelle zur Darstellung einer SD - Zahl genügen und keine logischen Verknüpfungen zur Konversion in die SD-Darstellung nötig sind. Wir werden uns daher im folgenden auf diese $r = 2$ -SD-Zahlen konzentrieren.

2.1 SD-Zahlen zur Basis $r = 2$

Jede Ziffer d_i einer ($r = 2$)-SD-Zahl $D = [d_{n-1} \dots d_1 d_0]$ mit dem dezimalen Wert $Z = d_{n-1} * 2^{n-1} + \dots + d_0 * 2^0$ nimmt Werte aus der Menge $\{-1, 0, 1\}$ an. Die Elemente dieser Menge können dann mit Zweibitvektoren (x, y) dargestellt werden:

$$(1) \quad (x, y) = -x_i + y_i \in (10, 00, 01), [6, 4].$$

Eine transformierte Binärzahl $A = a_{n-1} a_{n-2} \dots a_1 a_0$, $a_i \in (0, 1)$, ergibt sich damit zu $D = d_{n-1} (0 a_{n-2}) \dots (0 a_1) (0 a_0)$, wobei für die führende Ziffer im Fall $a_{n-1} = 0$ $d_{n-1} = 00$ zu setzen ist, und $d_{n-1} = 10$ gilt, wenn $a_{n-1} = 1$ gilt, also eine negative Zweierkomplementzahl vorliegt. Damit können alle binären Bits a_i von A , außer dem MSB, unmittelbar als das y -Bit der binären Darstellung der entsprechenden SD-Ziffern d_i aufgefaßt werden. Die gewandelte Binärzahl $1001 = -7_{\text{dez}}$ liest sich also beispielsweise $\{(10) (00) (00) (01)\}$. In der $r=2$ -Zahlendarstellung können also die Bits der Binärzahl ohne logische Verknüpfungen und damit laufzeitneutral in SD-Zahlen gewandelt werden.

Die Rückkonversion ist komplizierter: Im Prinzip müssen zur Rückwandlung die positiven und negativen Ziffern separat und mit korrekter Wertigkeit von einander subtrahiert werden: Zwar sind gewisse Bitkombinationen ausgeschlossen, was die logischen Gleichungen vereinfacht, aber trotzdem können dabei Überträge entstehen, die vom niederwertigsten bis zum höchstwertigsten Bit propagieren. Jedes Bit a_i der resultierenden Binärzahl ergibt sich iterativ beginnend bei $i = 0$ aus folgenden Gleichungen:

$$(2) \quad \begin{aligned} a_i &= (x_i + y_i) \oplus \beta_i, \\ \beta_{i+1} &= x_i + \overline{y_i} \cdot \beta_i. \end{aligned}$$

Hier bezeichnet $\overline{y_i}$ die logische Inversion von y_i . Bei größeren Wortlängen (16 Bit oder 32 Bit) können über Carry-Select-Verfahren, bei denen das Carry-Bit β_i mehrfach berechnet wird, die Wandelgeschwindigkeiten erhöht werden.

2.2 Elementare Arithmetische Operationen mit SD - Zahlen

2.2.1 Addition

Um zwei SD-Zahlen D und E zu addieren, wird jede Ziffer d_i zur entsprechenden Ziffer e_i addiert. Das Ergebnis ist eine Zwischensumme w_i und eine Übertragziffer t_i [1]. Die Ziffern des Ergebnisses $S = s_{n-1} \dots s_0$ ergeben sich aus:

$$(3) \quad d_i + e_i = 2 t_i + w_i$$

$$(4) \quad s_i = w_i + t_{i-1}.$$

Negative (positive) Überträge von der vorherigen Stelle, t_{i-1} , sind nur dann problematisch, wenn **eine** der beiden zu addierenden Ziffern d_i bzw. e_i negativ (positiv) ist. Ein negativer (positiver) Übertrag t_{i-1} führt dann zu $s_i = -2$ ($s_i = 2$), was einer Bereichsüberschreitung entspricht. Wenn aber die Aufteilung in Carry-Ziffer und Zwischensumme in Glg. (3) nach Maßgabe der Vorzeichen der SD-Ziffern der vorherigen Stelle ($i-1$) durchgeführt wird, läßt sich die Akkumulation von Überträgen vermeiden [6], (Tabelle I). Das Wertepaar für die Summe $t_i + w_i$ wird geeignet gewählt auf der Basis einer Kontrollziffer h_{i-1} , die die Vorzeichen von d_{i-1} und e_{i-1} , also der Ziffern auf der vorherigen Position erfaßt: $h_{i-1} = 1$, wenn d_{i-1} oder e_{i-1} negativ ist und $h_{i-1} = -1$, wenn d_{i-1} und e_{i-1} positiv sind.

2.2.2 Subtraktion

Wenn die SD-Zahl D von der Zahl E subtrahiert werden soll, dann wird dies durch die Addition von $-D$ dargestellt: $E + (-D)$, wobei $-D$ durch die Vorzeichenumkehr jeder einzelnen Ziffer d_i entsteht. Jedes einzelne Bit jeder Ziffer wird zur Vorzeichenumkehr invertiert, mit einer Ausnahme: Bei $d_i = (00)$ bleibt die Ziffer unverändert. Im Gegensatz zum Vorzeichenwechsel bei Zweierkomplementzahlen tritt dabei aber kein Übertrag auf und alle Ziffern können zeitlich parallel gewandelt werden.

2.2.3 Hybride Addition/Subtraktion

SD-Zahlen mit $r = 2$ bieten als besonderen Vorteil die Möglichkeit, auch sog. *hybride Additionen* auszuführen [4]: $S = A + D$, wobei A eine Binärzahl und S , sowie D SD-Zahlen sind. Dies ergibt einen erheblichen Geschwindigkeitsvorteil, denn wie unsere Studien zeigen, liegt das Ergebnis nach ca. der halben Verzögerungszeit einer echten SD-Operation vor. Die Binärzahl wird dabei als SD-Zahl mit einem beschränkten Wertevorrat $\{0, 1\}$ aufgefaßt. Da jedes Bit a_i von A nicht negativ ist, ergibt sich für die unkorrigierte Zwischensumme $a_i + d_i = v_i \in \{-1, 0, 1, 2\}$. Wenn $v_i \geq 1$ gilt, dann wird ein positives Carry - Bit $t_i \in \{0, 1\}$ erzeugt:

$$(5) \quad w_i = v_i - 2 t_i$$

$$(6) \quad s_i = w_i + t_{i-1}$$

wobei t_{i-1} der Übertrag von der vorherigen Stelle ist und $w_i \in \{-1, 0\}$, wie üblich, die Zwischensumme. Da keine negativen Überträge auftreten können, vereinfacht sich die Operation: t_i läßt sich als **Bit** darstellen und die Kontrollziffer h_i muß nicht ausgewertet werden. Die notwendigen Schaltungen vereinfachen sich also und die Addition kann schneller durchgeführt werden.

Die Vorteile der hybriden Addition übertragen sich auch auf die hybride Subtraktion:

$$(7) \quad D - A = D + (-A) = D + \overline{A} + 1.$$

\overline{A} ist die Binärzahl mit invertierten Bits. Die Addition der 1 erfolgt durch Setzen von t_0 , der niederwertigsten Übertragsziffer, die sonst nicht benutzt wird. Da im SD-Format addiert wird, erfolgt die Inkrementierung total parallel.

h_{i-1}	d_i	e_i	t_{i-1}	w_i	t_i	s_i
d.c.	0	0	0	0	0	0
d.c.	0	0	1	0	0	1
d.c.	0	0	-1	0	0	-1
0	1	0	0	-1	1	-1
0	1	0	1	-1	1	0
1	1	0	0	1	0	1
1	1	0	-1	1	0	0
0	-1	0	0	-1	0	-1
0	-1	0	1	-1	1	0
1	-1	0	0	1	-1	1
1	-1	0	-1	0	-1	-1
d.c.	1	1	0	0	1	0
0	1	1	1	0	1	1
1	1	1	-1	0	1	-1
d.c.	-1	-1	0	0	-1	0
1	-1	-1	-1	0	-1	-1
0	-1	-1	1	0	-1	1

Tabelle I: Regeln für die Kollisionsvermeidung von Überträgen bei der Addition von SD - Ziffern d_i, e_i , sowie der Übertragsziffer t_{i-1} . h_{i-1}, s_i, w_i , und t_i sind im Text definiert. "d.c." bezeichnet Fälle, in denen der Wert von h_{i-1} keinen Einfluß auf das Ergebnis hat. Die Fälle mit Kommutationsymmetrie ist nur eine der beiden Kombinationen von d_i und e_i berücksichtigt.

2.2.4 Vorzeichenbestimmung bei SD- Zahlen

Aufgrund der Redundanz ist es nicht wie bei Zweierkomplementzahlen möglich, das Vorzeichen einer SD - Zahl aus dem Wert des höchstwertigsten Bits (MSB) zu bestimmen. Es ist vielmehr nötig, die höchstwertigste Stelle mit Betrag ungleich Null zu finden, denn deren Vorzeichen bestimmt das Vorzeichen der gesamten Zahl [1]. Gleichzeitig entfällt aber die Notwendigkeit bei Vergrößerung der Wortbreite vorzeichenrichtig zu erweitern. Zur Vorzeichenbestimmung wird ein Prioritätendekoder für alle Stellen benötigt. Hier haben kleine Radices (wie z.B. $r = 2$) Nachteile gegenüber größeren r-Werten.

Die Vorzeichenproblematik überträgt sich auf Vergleichsoperationen, die über Subtraktion mit anschließender Vorzeichenbestimmung implementiert werden. Dies ist ein Nachteil bei bestimmten iterativ arbeitenden Algorithmen, wie dem CORDIC-Verfahren.

2.2.5 Multiplikation

Die Multiplikation bei $r = 2$ -Zahlen erfolgt, wie bei gewöhnlichen Binärzahlen, mit Schiebeoperationen und Additionen der Partialprodukte. Zu beachten ist aber generell, daß die Verschiebung um eine Stelle nur dann der Multiplikation mit dem Radix (hier $r = 2$) entspricht, wenn Radices verwendet werden, die sich durch eine

Zweierpotenz darstellen lassen. Folgendes Beispiel zeigt, wie zwei $r = 2$ -SD-Zahlen, 1001 und 0001010, die dezimal den Zahlen 9 und 6 entsprechen, multipliziert werden (unterstrichene Zahlen sind negativ):

$$\begin{array}{r}
 1001 * 00010\underline{1}0 \\
 \hline
 0000 \\
 \underline{1001} \\
 0000 \\
 1001 \\
 \hline
 10\underline{1}10\underline{1}0
 \end{array}$$

Wie bei der binären Multiplikation können auch bei SD - Zahlen Algorithmen eingesetzt werden, die die Multiplikationszeit verkürzen. Die sind zum einen die Verwendung eines Wallace-Trees zur parallelen Summation der Bitscheiben der Partialprodukte, und zum anderen die Booth-Kodierung des Multiplikators zur Reduktion der Zahl der Partialprodukte.

3. SD-Arithmetik im Vergleich

Die Vorteile von SD-Algorithmen, die in der Literatur an exemplarischen Einzelfällen immer wieder betont wurden, sind nicht einfach zu bewerten. Wegen des Konversionsaufwandes und der höheren Komplexität im Vergleich mit etablierten binären Algorithmen ist der Einsatz von SD-Zahlen nur bei erheblichem Geschwindigkeitsgewinn sinnvoll. Um hier zu objektiven Aussagen zu kommen, haben wir SD-Operationen als parametrisierbare VHDL-Modelle beschrieben, die mit dem Synopsys Design Compiler optimiert und auf eine Standard-CMOS-Technologie abgebildet wurden. Aus den Area-Reports und den Timing-Reports lassen sich Gatterzahlen und Laufzeiten ablesen, die mit den entsprechenden Werten von auf die gleiche Technologie synthetisierten binären Implementierungen der High Performance Varianten aus der Design Ware IP-Bibliothek verglichen werden können.

Für die Synthese wurden die *Operating Conditions* auf *Worst Case Commercial* gesetzt, der *Optimization Effort* auf *Medium* und als *Wire Load Model* haben wir ein 8k Gate Design gewählt.

3.1 Elementare Operationen und Multiplizierer

Ergebnisse für die elementaren arithmetischen Operationen zeigt Tabelle II. Die reinen SD-Implementierungen der Addierer, Subtrahierer usw. sind erheblich schneller als die entsprechenden *Design Ware*-Komponenten. Insbesondere bieten die hybriden Varianten besondere Vorteile. Der Gatterbedarf für SD-Versionen unterscheidet sich nicht wesentlich von dem der binären Lösungen. Interessanterweise benötigen einige der SD-Blöcke sogar weniger Gatterfläche, als die entsprechenden binären Varianten. Hervorzuheben ist auch, daß - wie erwartet - die Laufzeit der SD-Blöcke nicht von der Wortlänge der Operanden abhängt. Wie aus der Tabelle aber auch hervorgeht, verdoppelt sich die Chipfläche und die Datendurchlaufzeit, wenn die Rückkonversion mit betrachtet wird. Daher werden sich mit SD-Algorithmen nur dann die Leistungsdaten von DSP-Systemen verbessern lassen, wenn größere Segmente des Datenpfads $r=2$ -kodierte Zahlen verwenden.

Da SD-Zahlen insbesondere bei Additionen Vorteile aufweisen, und bei der Multiplikation viele Additionen durchzuführen sind, wurden Multiplizierer als erstes Beispiel für komplexere DSB-Blöcke untersucht (Tabelle III). Der Geschwindigkeitsgewinn mit SD-Zahlen im Vergleich zu *Design Ware*-Multiplizierern ist aber gering. Dies hängt damit zusammen, daß die Booth-Kodierung bei binären Multiplizierern bereits eine redundante SD-Darstellung des Multiplikators darstellt, und folglich die binäre Multiplikation im Prinzip als hybride SD-Operation ausgeführt wird, ohne daß dies der Nutzer, der rein binäre Daten zuführt und abliest, wahrnimmt. SD-Implementierungen bieten also keinen echten inhärenten Architekturvorteil, der den zeitlichen Zusatzaufwand für die Rückkonversion überkompensiert. Diese Ergebnisse sind in Übereinstimmung mit der Literatur. Es wurden z. B. verschiedene 54x54-Multiplizierer mit und ohne SD - Zahlen vorgestellt, die sich kaum in Laufzeit und Flächenbedarf unterscheiden, wenn man die verschiedenen Technologiestände nur entsprechend wichtet [6-8].

3.2 Komplexere Funktionen: Dreifach Addition und Akkumulatoren (MAC)-Stufen

Eine in vielen DSP-Funktionen benötigte Funktion ist die gleichzeitige Addition von 3 Summanden. Wie bereits in [1] ausgeführt, ist diese Operationen einstufig redundant implementierbar, wenn nur der Radix groß genug gewählt wird ($r \geq 5$). Große Radices haben aber, wie bereits angesprochen, den Nachteil, daß die einfachen Konversionsmöglichkeiten wie bei $r = 2$ nicht mehr bestehen. Deshalb bietet es sich an, die Dreifachoperationen als

kaskadierte Schaltung aus zwei binären oder $r = 2$ -SD-Addierer/Subtrahierer-Blöcken aufzubauen. Zwei hybride SD-Addierer/Subtrahierer sind hier besonders schnell, wie Tabelle IV entnommen werden kann.

<i>Block</i>	Wortlänge	Fläche/Gatteräquivalente	Laufzeit/ns
DW Schneller Carry Look Ahead Addierer/Subtrahierer	8	268	4,29
	16	592	5,09
	32	1234	5,77
DW Schneller Carry Look Ahead Vergleicher	8	193	3,35
	16	282	4,17
	32	556	4,77
SDN Addierer	8	250	2,62
	16	484	2,72
	32	1248	2,88
SDN Addierer/Subtrahierer	8	315	3,43
	16	624	3,51
	32	1477	3,64
Hybrider Binärer Addierer	8	291	1,70
	16	407	1,71
	32	902	1,73
Hybrider Binärer Addierer/Subtrahierer	8	284	2,31
	16	582	2,23
	32	1143	2,31
SD nach Binär Konversion 2 Borrow Select Stufen 4 Borrow Select Stufen	8	102	3,06
	16	601	3,68
	32	824	4,41

Tabelle II: Laufzeit und Gatterfläche für verschiedene elementare Arithmetikblöcke mit SD- bzw. binären Operanden und verschiedenen Wortlängen. Die Wortlänge für Binärzahlen bezieht sich auf Bits, die Angaben für SD-Zahlen auf Ziffern, was dem gleichen Wertebereich für die Operanden entspricht.

<i>Block</i>	Wortlänge	Fläche/Gatteräquivalente	Laufzeit/ns
Synthese des binären Verhaltensmodells A*B	16x16	2135	27,55
SD - Multiplizierer: Wallace Tree, Booth Encoded, inkl. Rückkonversion	16x16	4658	13,11
		5754	17,65
DW Multiplizierer: Wallace Tree, Booth Encoded	16x16	3506	14,64

Tabelle III: Laufzeit und Gatterfläche für 16x16 Multiplizierer.

<i>Block</i>	Wortlänge Eingang/Ausgang	Fläche/Gatteräquivalente	Laufzeit/ns
Synthese des Verhaltensmodells A + B + C	32/33	1234	11,49
Kaskadierte Addition mit DW Fast Carry Look - Ahead Addierern	32/33	1865	9,8
Kaskadierte Addition von 3 SD - Zahlen mit Rückkonversion	32/33	1702	5,65
		2442	10,35
Kaskadierte hybride Addition von 3 SD - Zahlen mit Rückkonversion	32/33	710 1638	3,89 7,94

Tabelle IV: Laufzeit und Gatterfläche für Dreifach Addierer/Subtrahierer

Eine Analyse der Flächenwerte der verschiedenen Lösungen zeigt, daß die Fläche für den 32 Bit-Hybrid-Entwurf inklusive Rückkonversion nur 1638 Gatteräquivalente beträgt, was mit den 1865 Gatteräquivalenten der Design Ware Lösung zu vergleichen ist. Statt eines erwarteten Flächennachteils zeigt hier die SD-Lösung einen Flächen-vorteil von 14 %. Ohne die Rückkonversion fällt mit 710 statt 1865 Gatterflächen der Vergleich noch günstiger aus. Dies ist dadurch zu erklären, daß schnelle Binäraddierer zur Vermeidung großer Carrylaufzeiten redundante Hardware vorsehen, was den Flächenvorteil der binären Zahlendarstellung aufheben und sogar überkompensieren kann.

Ein weiterer typischer Block in DSP-Schaltungen sind Multiplikations-Akkumulationsstufen (MAC) oder zweifache Multiplikations-Akkumulationsstufen (MAC2). Diese Blöcke wurden für eine binäre Eingabewortlänge von 16 bit untersucht. Tabelle V zeigt Ergebnisse. Auch hier haben SD-Lösungen Vorteile, wenn nicht rückgewandelt werden muß.

Block	Wortlänge	Fläche/Gatteräquivalente	Laufzeit/ns
MAC			
Synthese des Verhaltensmodells $A*B + C$	16	4368	31,90
DW MAC	16	4656	20,50
SD-MAC mit SD-Ergebnis	16	5635	16,27
MAC2			
Synthese des Verhaltensmodells $A*B + C*D + E$	16	8587	36,13
DW MAC – Module als kaskadiertes Design	16	8870	25,30
SD-MAC2 mit SD-Ergebnis	16	9360	19,74

Tabelle V: Laufzeit und Gatterfläche für MAC- und MAC2-Stufen.

4. Systemstudien

Wie in den vorangehenden Abschnitten diskutiert, sind meßbare Vorteile für SD-Implementierungen nur in Anwendungen zu erwarten, in denen der komplette Datenpfad oder zumindestens ein kritischer Teil dieses Pfades $r=2$ -kodiert ausgeführt werden kann, ohne daß Konversionen in die binäre Darstellung notwendig sind. Zusätzliche Vorteile ergeben sich, wenn, hybride arithmetische Blöcke eingesetzt werden können. Dies erfordert aber eine angepaßte Zerlegung des Datenpfades in binäre und hybride Bereiche. Als Beispielanwendungen werden im folgenden Digitale Filter und Sigma-Delta-Modulatoren diskutiert.

4.1 Filter

Bild 2 zeigt ein digitales IIR-Filter. Hier wird das Eingangssignal X und das Signal Y_{sd} im Rückkoppelpfad mit MAC2-Modulen mit Filterkoeffizienten b_n bzw. a_m multipliziert. Die Ergebnisse der Multiplikationen werden mit zu den verzögerten MAC2-Ergebnissen der vorhergehenden Stufen addiert. Um hybrid arbeiten zu können, ist es vorteilhaft, den Rückkoppeldatenfluß SD-kodiert und den direkten Pfad binär zu belassen. Die MAC2-Stufen werden dann mit einem hybriden Multiplizierer für SD-Zahlen und einem zweiten hybriden Multiplizierer für binäre Eingänge (X und b_n) aufgebaut. Ein 3fach-Addierer für SD-Zahlen akkumuliert die Multiplikationsergebnisse. Die Rückkonversion in das binäre Zahlenformat erfolgt erst nach der Wiederherstellung der Eingangswortlänge durch Abschneiden außerhalb der Rückkoppelschleife, um den Konversionsaufwand zu minimieren.

Die minimale Taktperiode für das Filter ergibt sich aus der Verzögerung der MAC2-Stufe zuzüglich einer Registerlaufzeit für das Verzögerungsglied zu insgesamt $(19.57 + 1.02)$ ns. Die binäre aus DW-Komponenten aufgebaute Struktur hingegen weist eine minimale Taktperiode von $(25.30 + 1.02)$ ns. Die Taktfrequenz der hybriden Version kann daher um 27% gegenüber der binären Implementierung erhöht werden. Der Zusatzaufwand an Chipfläche fällt deutlich geringer aus: Die hybride Version ist nur 8% größer.

4.2 Sigma-Delta-Modulatoren

Eine weitere vielversprechende Anwendung für SD-Datenpfade ist die Delta-Sigma-Modulation ($\Delta\Sigma$), die für die Signalsynthese in DSP Systemen eingesetzt wird [9-11]. Bild 3 zeigt einen Modulator zweiter Ordnung. Die Schaltung wandelt ein W -Bit breites binäres Eingangssignal X in ein pulsdichte-moduliertes einbittiges Ausgangssignal Y um. Der Wert von Y wird durch einen Komparator ermittelt, der das Vorwärtssignal mit einem Schwellwert vergleicht (meist "0" für Zweierkomplementeingaben). Folglich kann der gesamte Datenpfad mit SD-Blöcken aufgebaut werden, ohne daß eine explizite Rückkonversion erforderlich wäre. Die maximale Taktfrequenz des Modulators wird von der Additions-/Subtraktionsstufe zuzüglich der Laufzeit der Schwellwert-schaltung bestimmt, die sich hier auf eine Vorzeichenbestimmung reduziert. Die erste Additions-/Subtraktionsstufe kann als hybride Subtraktion mit einem SD-Addierer und die zweite Additions-/Subtraktionsstufe als kaskadierte SD-Additions-/Subtraktionsstufe mit SD Addierwerk aufgebaut werden. Der Einfachheit halber wird eine einheitliche Wortlänge von 16 bit angenommen, mit Ausnahme des Ausgangssignals Y (1 Bit).

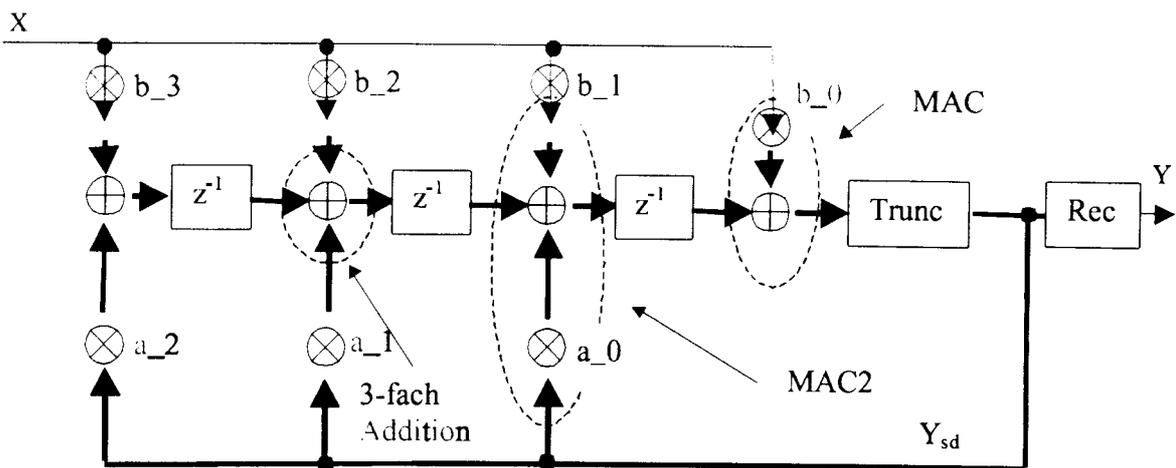


Bild 2 Hybride Implementierung eines IIR-Filters. b_n und a_m sind Filter-Koeffizienten. z^{-1} kennzeichnet ein Verzögerungsglied, das um eine Taktperiode verzögert. SD-Datenpfade sind fett gezeichnet, binäre Datenflüsse dünn. Die Blöcke *Trunc* und *Rec* führen die Abschneideoperation bzw. die Rückkonversion der SD-Ergebnisse in Binärformat aus.

Nach der Synthese findet man für die SD-Implementierung 13,22 ns Durchlaufzeit im Vorwärtspfad bis zum Schwellwertvergleich, der 2,78 ns in Anspruch nimmt. Die gesamte Durchlaufzeit addiert sich zu 16,0 ns. Die binäre Version hat eine Laufzeit von 20,18 ns plus einer kleinen Verzögerung für die vorzeichenrichtige Erweiterung im D/D-Block. Die Bandbreite ist deshalb gegenüber der SD-Variante um 26% niedriger. Die Chipfläche der 16 Bit Hybrid-Schaltung beträgt 2900 Gatteräquivalente, die binäre Lösung ist um 20% kompakter (2337 Gatteräquivalente).

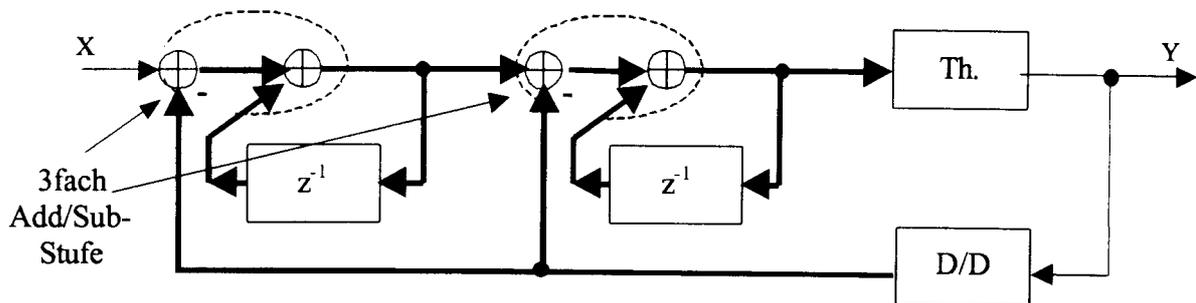


Bild 3 Delta Sigma Modulator 2ter Ordnung: X ist das Eingangssignal (16 Bit), Y ist das Ausgangssignal (1 Bit). Der Block *Th.* bestimmt das Vorzeichen des 16 Bit Wortes im Vorwärtspfad und reduziert so die Wortlänge auf 1 Bit. Der Digital-Digital-Konverter (D/D) erweitert die Wortlänge des Vorzeichenbits auf 16 Bit. Die eingekreisten Komponenten 3fach Add/Sub (dreifach Additions-/Subtraktionsstufe) bestimmen die Durchlaufzeit. SD-Datenpfade sind fett gezeichnet.

5. Schlußfolgerungen

In dieser Arbeit wurden Radix-2 redundante Implementierungen von arithmetischen Operationen für DSP-Algorithmen vorgestellt. Es zeigt sich, daß Additionen, Subtraktionen und Multiplikationen zwar schneller als mit binären Verfahren ausgeführt werden können. Der Aufwand für die Rückkonversion des SD - Ergebnisses in die binäre Zahlendarstellung (über-)kompensiert aber den Geschwindigkeitsgewinn und erhöht den Chipflächenaufwand gegenüber der binären Lösung. Wenn jedoch mehrere SD-Operationen zur Darstellung einer DSP-Funktion zusammengesetzt werden können und nur das Endergebnis gewandelt werden muß, dann zeigen SD-Algorithmen substantielle Vorteile: Die Gesamtlaufzeit läßt sich bis zu 30% gegenüber binären Lösungen reduzieren, wobei der Chipflächenzuwachs im Vergleich zu *high performance* binärer Implementierungen vergleichsweise gering ist. In bestimmten Fällen kommt ein SD-System sogar mit weniger Chipfläche aus, wie dies die 3fach-Addition zeigt. Die Vorteile von SD-Zahlen steigen mit der Wortlänge der zu verarbeitenden Daten und mit der Komplexität der DSP-Funktion, also z. B. mit der Filter- oder der Modulatorordnung. Wie unsere Untersu-

chungen zeigen, führt der Einsatz von *hybriden* Blöcken, die gleichzeitig binäre und SD-Daten verarbeiten können, zu den größten Geschwindigkeitsvorteilen.

Es ist hervorzuheben, daß die Leistungsdaten der SD-Blöcke über die Synthese von Verhaltens-VHDL-Modellen gewonnen wurden. Die binären Algorithmen, die als Referenz dienten, waren hingegen komplexe Entwürfe auf Gatterebene, die sich nicht auf die gleiche einfache Weise an gegebene Systemanforderungen anpassen lassen und weniger Spielraum für zukünftige Optimierungen beinhalten.

Da die Gatterzahlen für redundante und nicht redundante DSP-Strukturen fast gleich sind, redundante Lösungen aber ein stärker kurzreichweitig orientiertes Verdrahtungsschema aufweisen als binäre Lösungen, sind positive Auswirkungen auf die Verlustleistung von SD-Designs zu erwarten, denn kürzere Leitungen reduzieren die benötigten Ladeströme für die Leitungskapazitäten. Lokalisierte Verdrahtungen begünstigen auch FPGA-Implementierungen. Diese Fragestellungen sollen in weiteren Untersuchungen angegangen werden.

6. Literatur

- [1] A. Avizienis, "Signed digit number representation for fast parallel arithmetic", IRE Trans. Electron. Comput., vol. EC - 10, pp. 389 - 400, 1961
- [2] A. Vandemeulebroecke, E. Vanzieleghem, T. Denayer, and P. G. A. Jespers: "A New Carry-Free Division Algorithm and its Application to a Single-Chip 1024-b RSA Processor", IEEE J. of Solid State Circuits, vol. 25, No. 3, pp. 748 - 756, 1990
- [3] S. M. Yen, C. S. Lai, C. H. Chen, and J. Y. Lee: "An Efficient Redundant-Binary Number to Binary Number Converter", IEEE J. of Solid State Circuits, vol. 27, No. 1, pp. 109 -112, 1992
- [4] H. R. Srinivas, and K. K. Parhi: "A Fast VLSI Adder Architecture", IEEE J. of Solid State Circuits, vol. 27, No. 5, pp. 761 - 767, 1992
- [5] T. N. Rajashekhara and A. S. Nale: "Conversion from signed-digit to radix complement representation", Int. J. of Electronics, vol. 69, pp. 717 - 721, 1990
- [6] H. Makino, Y. Nakase, H. Suzuki, H. Morinaka, H. Shinohara, and K. Mashiko: "An 8.8 ns 54 x 54 Bit Multiplier with High Speed Redundant Binary Architecture", IEEE J. of Solid State Circuits, vol. 31, No. 6, pp. 773 - 783, 1996
- [7] J. Mori, M. Nagamatsu, M. Hirano, S. Tanaka, M. Noda, Y. Toyoshima, K. Hashimoto, H. Hayashida, and K. Maeguchi: "A 10 ns 54 x 54 - bit parallel structured full array multiplier with 0.5 μ m CMOS technology", IEEE J. of Solid State Circuits, vol. 26, No. 4, pp. 600 - 605, 1991
- [8] N. Ohkubo, M. Suzuki, T. Shinbo, T. Yamanaka, A. Shimizu, K. Sasaki, Y. Nakagome: "A 4.4 ns 54 x 54 - bit multiplier using pass - transistor multiplexer", IEEE J. of Solid State Circuits, vol. 30, No. 3, pp. 251 - 257, 1995
- [9] T. A. D. Riley, M. A. Copeland, and T. A. Kwasniewski: "Delta - Sigma Modulation in Fractional - N Frequency Synthesis", IEEE J. of Solid State Circuits, vol. 28, No. 5, pp. 553 - 559, 1993
- [10] N. Tan and S. Eriksson, "Second Order Delta - Sigma Modulator with Two 1 Bit Quantisers", Electronics Letters, Vol. 28 No. 16, pp. 1486 ff., 1992
- [11] G. W. Roberts and A. K. Lu. "Analog Signal Generation for Built -In - Self - Test of Mixed - Signal Integrated Circuits", Boston/Dordrecht/London, Kluwer Academic Publishers, 1996



ASIC System Design Ausbildung mit einem RISC-IP-Core

Frank Kesel

Fachhochschule Pforzheim
Tiefenbronner Str. 65, 75175 Pforzheim
Tel.: 07231 / 28-6567, Email: kesel@fh-pforzheim.de

Kurzfassung - Bei der Entwicklung von "Systems-on-Chip" ist sowohl die Kenntnis des HW-Entwurfs (z.B. mit VHDL) wichtig als auch die Kenntnis der Entwicklung eingebetteter Software (z.B. mit C). An der FH Pforzheim entwickeln die Studierenden daher im Rahmen eines Vertiefungslabors des Studiengangs Elektro-/Informationstechnik einen Peripherieblock für einen RISC-Mikrocontroller in VHDL und verifizieren ihren Entwurf mit Hilfe eines Testprogramms für den Controller in C. Das entstehende Mikrocontrollersystem wird dann auf ein FPGA abgebildet und auf einem Evaluationsboard getestet (Prototyping). Im letzten Schritt wird das System dann auf eine Alcatel-0.5um-CMOS-Technologie abgebildet und ein Maskenlayout entworfen. Der Beitrag berichtet über den an der FH Pforzheim selbstentwickelten RISC-Core, der zur Microchip-PIC-Familie kompatibel ist, und berichtet über die verwendeten Werkzeuge und die gemachten Erfahrungen.

1 Einleitung

Moderne elektronische Hardware/Software-Systeme, wie beispielsweise Mobiltelefone, sind nur durch den Einsatz von höchstintegrierten ASICs realisierbar, häufig auch als "Systems-on-Chip" bezeichnet. Diese bestehen in der Regel aus Mikrocontroller- und Signalprozessorsystemen. Beim Entwurf solcher Systeme muß komplexe Hardware und Software entwickelt werden und beides in einem System integriert und getestet werden. Da die Herstellung der ASICs in modernen Submikrometerprozessen sehr teuer ist und einige Zeit benötigt, geht man zunehmend dazu über, zunächst Prototypen der ASICs mit FPGAs zu realisieren. Hier bietet eine technologieunabhängige Beschreibung der Schaltung mit einer Hardwarebeschreibungssprache wie VHDL große Vorteile. Bei der Entwicklung der eingebetteten Software für diese Systeme wird, neben der klassischen Assemblerprogrammierung, vermehrt auch C/C++ eingesetzt.

Benötigt werden hierzu Entwickler, die in der Lage sind, Spezifikationen, unter Einsatz moderner Entwicklungswerkzeuge und -methodiken, sowohl in Hard- als auch in Software umzusetzen. Aufgrund immer kürzerer Entwicklungs- und Marktzyklen wird auch gefordert die Anzahl der Fehler und Redesigns zu verringern. Daher kommt es immer stärker darauf an, Hard- und Software systematisch und gut dokumentiert zu entwickeln, um eine Wiederverwendbarkeit (Re-Use) und auch Vermarktung als sogenannten IP-Block (IP : Intellectual Property) zu ermöglichen.

Die Fachhochschule Pforzheim bietet daher im Rahmen des Elektro/Informationstechnikstudiums ein Vertiefungslabor an, welches Kenntnisse in der systematischen Entwicklung solcher Systeme vermitteln soll. Aufbauend auf Grundlagenkenntnissen in Digitaltechnik, Mikroelektronik, Mikroprozessortechnik, C und VHDL sollen die Studierenden für ein Mikroprozessorsystem eine Timer-Peripherieinheit in VHDL entwickeln und mit selbstgeschriebener Software in C verifizieren. Sinnvoll erschien hierfür ein möglichst einfacher Prozessorkern, der aber dennoch industrienahe sein sollte. Gewählt wurde daher eine 8-Bit-RISC-Architektur, die beispielsweise von den Firmen Microchip /1/ und Uvicom /4/ verwendet wird, und nur 33 Befehle umfaßt, siehe /2/. Bei Microchip ist diese Architektur als PIC16C5X-Familie implementiert. Es handelt sich dabei um eine einfache 1-Adress- oder Akkumulatormaschine ohne Interrupts mit einer zweistufigen Pipeline und einer Harvard-Architektur. Diese Architektur wurde an der FH Pforzheim in VHDL als Prozessorkern "PortUC12" implementiert.

2 Der "PortUC12"-Kern

Ziel des "PortUC12"-Kerns ist es, den flexiblen und modularen Aufbau von Mikrocontroller-Systemen, bestehend aus Programm- und Datenspeicher sowie verschiedener Peripherieinheiten, in VHDL zu ermöglichen, sowie

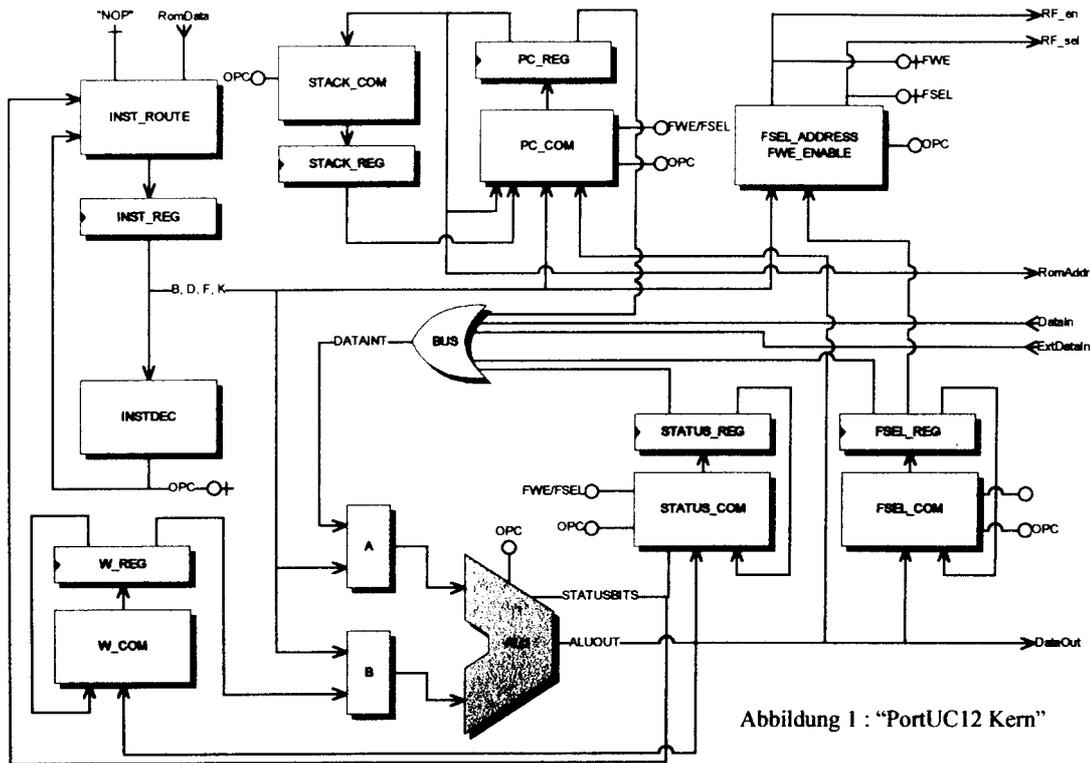


Abbildung 1 : "PortUC12 Kern"

deren einfache Portierbarkeit auf unterschiedliche Zieltechnologien (z.B. ASIC, FPGA) sicherzustellen.

Der "PortUC12"-Kern weicht vom PIC16C5X in einigen Punkten ab, insbesondere was das Zeitverhalten und den Takt angeht, sowie bei der Datenspeicheraufteilung und dem Bus-Design. Jedoch ist der Kern kompatibel mit dem in /2/ beschriebenen Instruktionssatz, so daß die für den PIC16C5X teilweise kostenlos erhältlichen Werkzeuge für die Softwareentwicklung, wie C-Compiler, Assembler oder Simulator, eingesetzt werden können.

Die Modularisierung wird dadurch erreicht, daß nur die für einen Prozessorkern wesentlichen Einheiten und Register zum eigentlichen PortUC12-Kern zusammengefaßt werden. Insbesondere gehören die Peripherieeinheiten, wie der Timer und die Ports, sowie Daten- und Programmspeicher *nicht* zum Kern. Zur Modularisierung trägt eine einheitliche Schnittstelle des Kerns zum Datenspeicher (General Purpose Register) und den in den Adreßraum des Datenspeichers abgebildeten SFRs (Special Function Registers) bei, die eine einfache Erweiterbarkeit um zusätzliche Peripherieeinheiten sicherstellt. Hierdurch ist es insbesondere auch möglich, zusätzliche Peripherieeinheiten, die in der PIC16C5X-Familie nicht verfügbar sind, einzubauen. Weiterhin ist es möglich, Programm- und Datenspeichergröße anzupassen. Die Aufteilung des Datenspeichers (General Purpose Register File, SFRs) kann

extern vorgenommen werden und das Kern-Design muß hierzu nicht verändert werden. Der VHDL-Code des Kerns ist sehr kompakt und besteht aus einer einzigen VHDL-Entity/Architecture-Einheit. In Abbildung 1 ist in schematischer Darstellung der Aufbau des VHDL-Codes des Kerns gezeigt. Die Bezeichnungen in Abbildung 1 benennen entsprechende Prozesse im VHDL-Code. Der Kern besteht aus speichernden Registern und zugeordneten Schaltnetzen sowie dem Rechenwerk (ALU), welches ebenfalls ein Schaltnetz ist. Die Funktionen der Register entsprechen weitgehend den in /2/ beschriebenen Registern des PIC16C5x. Sämtliche in Abbildung 1 gezeigten Register (Endung _REG) werden mit einem Einphasentakt getaktet. Die kombinatorischen Überführungs- und Ausgabeschaltnetze werden in den mit "_COM" bezeichneten Prozessen implementiert.

Das Instruktionsregister (INST_REG) speichert die vom Programmspeicher geholten Befehlscodes. Die Daten des Programmspeichers liegen an RomData in Abbildung 1 an. Bei Befehlen, die den Programmzähler direkt verändern, wie Sprüngen und Unterprogrammaufrufen, muß die zwei-stufige Fließbandverarbeitung des Prozessors angehalten werden und ein "NOP"-Befehl eingefügt werden, da der nächste, schon geholte Befehl nicht ausgeführt werden darf. Hierzu fügt der Multiplexer INST_ROUTE, in Abhängigkeit von Instruktion und Statusbits, den "NOP"-Befehl ein.



Der Instruktionsdekoder INSTDEC erzeugt aus den unterschiedlich langen PIC16C5X-Opcodes einen internen Opcode *OPC* einheitlicher Länge und führt dieses Signal *OPC* den einzelnen Verarbeitungseinheiten (Register, ALU) zu. Das Signal *OPC* ist ein VHDL-Aufzählungstyp¹. Jede Verarbeitungseinheit dekodiert *lokal OPC* und leitet daraus die für die Ausführung des Befehls notwendigen Aktionen ab. Dies steht im Kontrast zur üblichen Vorgehensweise, im Instruktionsdekoder einzelne Steuersignale für die Verarbeitungseinheiten *zentral* zu dekodieren. Das W-Register (W_REG) stellt den Akkumulator dar.

Der Programmzähler (PC_REG) besitzt in der vorliegenden Version eine Bitbreite von 11 Bit und kann damit maximal 2k Worte des Programmspeichers adressieren (Signal *RomAddr*). Der PC wird entweder ständig inkrementiert oder bei bestimmten Befehlen (z.B. CALL, GOTO, RETLW) mit dem Sprungziel *K* oder dem Inhalt des Stack neu geladen. *K* wird im PIC-Instruktionssatz als Literal bezeichnet.

Der Stack (STACK_REG) ist in der PIC16C5X-Architektur nicht für die Speicherung von Daten verwendbar, sondern dient nur der Ablage der Rücksprungadressen des PC. In der vorliegenden Version hat der Stack eine Tiefe von 2 Ebenen. Ein Überlauf des Stack wird nicht angezeigt. Der Stack läßt sich auf einfache Weise im VHDL-Code auf mehr als 2 Ebenen erweitern.

Das Statusregister (STATUS_REG) gehört zu den SFRs mit 8 Bit Breite und speichert im wesentlichen den Status der ALU (Zero-, Carry-Flags, etc.). Das FSR-Register (FSEL_REG) gehört ebenfalls zu den SFRs und dient, in Verbindung mit dem "virtuellen" Register INDF, der indirekten Adressierung des Datenspeichers (SFRs und File-Register). Die Adressierung des Datenspeichers erfolgt mit den Signalen *FSEL* (Adresse) und *FWE* (Write/Read).

Wie schon erwähnt, unterscheidet sich das Taktschema vom PIC16C5X. Dieser untersetzt den von außen zugeführten Takt um den Faktor 4 und generiert einen internen 4-Phasen-Takt mit den Phasen Q1 bis Q4, siehe auch /2/. Um eine einfache Integration des PortUC12 zu ermöglichen, wurde er als flankengesteuertes Einphasendesign implementiert. Dies ermöglicht auch eine einfache Testbarkeit des Kerns. Der Einbau von Testhilfen für ASIC-Technologien (z.B. Scan-Pfad) und die Generierung von Testvektoren kann bei oder nach der Synthese der ASIC-Gatternetzliste implementiert wer-

1. "enumeration type", siehe /3/.

den. Der Maschinenzyklus des Prozessors wurde in zwei Phasen, entsprechend zwei Taktzyklen des Einphasentaktes *Clk*, aufgeteilt, siehe Abbildung 2. Dies war im wesentlichen deshalb erforderlich, um eine Verwendung von synchronen SRAM-Speichern für das General-Purpose Register-File zu ermöglichen. In Phase 1 erfolgt in Pipelinestufe 2 (Decode/Execute) die Dekodierung des Befehls sowie der Lesezugriff auf die Register, in Phase 2 erfolgt die Ausführung des Befehls und das Zurückschreiben des Resultats in die Register, wie in Abbildung 2 gezeigt.

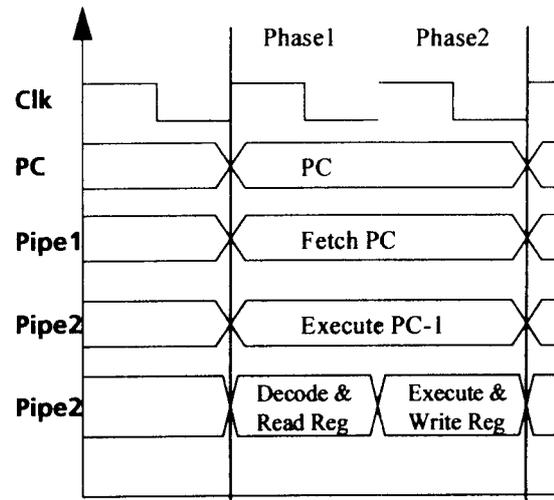


Abbildung 2 : Taktschema

3 Bus- und Speicherdesign, Aufbau eines Systems

Einen Unterschied zum PIC16C5X gibt es auch beim eingesetzten Datenspeicher-Bus-Design, der die Operanden der ALU zuführt und das Ergebnis der ALU-Operation auf ein Zielregister zurückschreibt und über den auch die Peripherieeinheiten (SFRs) angeschlossen werden. Für den PortUC12 wird ein sogenannter "logischer ODER-Bus", oder kurz "OR-Bus" verwendet.

Der Grund für die Verwendung des "OR-Busses" ist, daß ein Tristate-Bus-Design im Hinblick auf zwei Aspekte problematisch ist. Zum einen können bei Buskonflikten Querströme entstehen und zum anderen sind diese Bus-Designs unhandlicher für Synthese und Test, speziell bei der Realisierung als ASIC. Ein "OR-Bus" stellt die Busfunktionalität, die beim Tristate-Bus durch Tristate-Buffer und die Busleitung dargestellt wird, durch eine Kombination von logischen UND-ODER-Einheiten dar, siehe Abbildung 3.

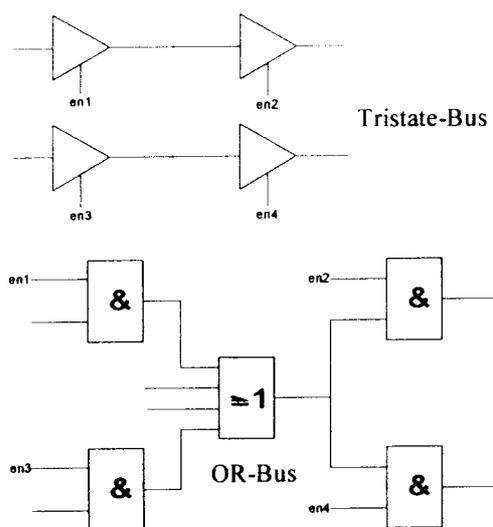


Abbildung 3 : OR-Bus vs. Tristate-Bus

Für die Realisierung des Programmspeichers und des General Purpose Register File (Datenspeicher) stehen im VHDL Code verschiedene effiziente Möglichkeiten zur Verfügung. ASIC- und FPGA-Technologien stellen in der Regel synchrone RAM- und ROM-Makros zur Verfügung. Werden diese Makros als VHDL-Modelle in den VHDL-Code eingebunden, so können sie über die VHDL-Synthese und die jeweiligen "Place&Route"-Werkzeuge für ASIC oder FPGA effizient auf die physikalischen Makrozellen der jeweiligen Technologie abgebildet werden. Nachteilig ist, daß hier zwangsläufig eine Technologieabhängigkeit des RTL-Codes (Register-Transfer-Level) entsteht.

Darüberhinaus sind auch synthetisierbare, technologie-unabhängige Implementierungen für beide Speicher verfügbar. Alle Implementierungen der Speicher werden durch eine "Verpackung" mittels unterschiedlicher Architekturen einer VHDL-Entity zugeordnet, welche damit eine einheitliche Schnittstelle für die Systemintegration darstellt.

Der Datenspeicher ist im PIC16C5X in vier Bänke aufgeteilt, der Zugriff wird über FSR[6:5] gesteuert. Jede Bank umfaßt 32 Register wobei die unteren 16 Register, in denen sich die SFRs befinden, in alle 4 Bänke gespiegelt sind. Die Register OPTION (Steuerregister des Timers), TRIS (Port-Steuerregister) und WDT (Watchdog) gehören im PortUC12 zu den CPU-externen Peripherieregistern. Im PIC16C5X, siehe /2/, werden diese Register *nicht* als zum Adreßraum des Datenspeichers gehörige SFRs angesprochen, sondern ein (Schreib)Zugriff erfolgt über spezielle Instruktio-

nen (OPTION, TRIS, CLRWDT). Dies erschwert eine saubere modulare Trennung in Peripherie und CPU, da nun entsprechende zusätzliche Steuerleitungen notwendig würden. Im PortUC12 wird dies so gelöst, daß diesen Registern eine Adresse in einer speziellen, fünften, Bank des Datenspeichers zugeteilt wird, die über das "freie" Bit FSR[7] adressiert wird (erweiterter SFR-Bereich). In der CPU wird bei Dekodierung der oben erwähnten Befehle die zugehörige Adresse auf den Adreßbus FSEL/RF_sel gelegt. Als Besonderheit ergibt sich hierdurch, daß diese Register auch über die normalen Datentransportbefehle angesprochen werden können. Weitere Peripherieblöcke können durch Belegen entsprechender Adressen im erweiterten SFR-Bereich hinzugefügt werden.

Ein Mikrocontrollersystem mit dem PortUC12-Kern benötigt mindestens folgende Komponenten:

- PortUC12-Kern (siehe Abbildung 1)
- General Purpose Register-File
- Programmspeicher

Hinzu können noch Peripherieeinheiten kommen, vorhanden sind hier beispielsweise die Ports, der Timer und ein UART. Zu bemerken ist, daß der PortUC12, ebenso wie die ganze PIC16C5X-Familie, nicht über Interrupts verfügt. Dies bedeutet, daß die Peripherie nur mittels Polling bedient werden kann.

Für die Simulation ist eine VHDL-Testbench vorhanden. Der Inhalt des Programmspeichers, die Software, wird über den C-Compiler oder den Assembler erzeugt. Es wird ein Hexfile für das verwendete Speichermodell generiert, welches auch für die FPGA-Hardware verwendet wird, und das Gesamtsystem kann mit einem VHDL-Simulator simuliert werden.

4 Entwurfsablauf

Die Studierenden erhalten zu Beginn des Labors eine Spezifikation der Timereinheit, nebst Dokumentationen zum PortUC12-Kern und dem Entwurfsablauf. Zunächst soll dann die Spezifikation auf Papier in ein Blockschaltbild umgesetzt werden, welches die Gliederung des Entwurfs in Prozesse, und für jeden Block die Funktion, z.B. als Pseudocode, zeigen soll. Dies ist auch die Grundlage für die abzuliefernde Dokumentation. In dieser Phase sollen auch Fragen zur Spezifikation im Sinne eines "Kunden-Entwickler-Dialogs" geklärt werden. Zu dieser Phase gehört natürlich auch die Einarbeitung in die Prozessorarchitektur und die Klärung der Schnittstellen des Timers zum System. Weiterhin sollen sich die Studierenden über-



legen, wie sie den Timer testen wollen und wie die Synchronisierung des externen Eingangs des Timers implementiert werden soll.

Nach dieser initialen Phase erfolgt die Eingabe der Schaltung als VHDL-RTL-Code mit Hilfe des Werkzeugs "RENOIR" von Mentor Graphics, wie in Abbildung 4 gezeigt. Zusammen mit den weiteren Blöcken des Designs und der Testbench wird dann ein Simulationsmodell für den VHDL-Simulator "MODELSIM", ebenfalls von Mentor, kompiliert. Die Software für den Test des Timers wird in C geschrieben und mit einer Freeware-Version eines C-Compilers für den PIC16C5X kompiliert. Nun erfolgt die Simulation mit MODELSIM und die Phase der Fehlersuche ("Debugging") beginnt. Erfahrungsgemäß erfordert diese Phase den größten Zeitaufwand des Labors.

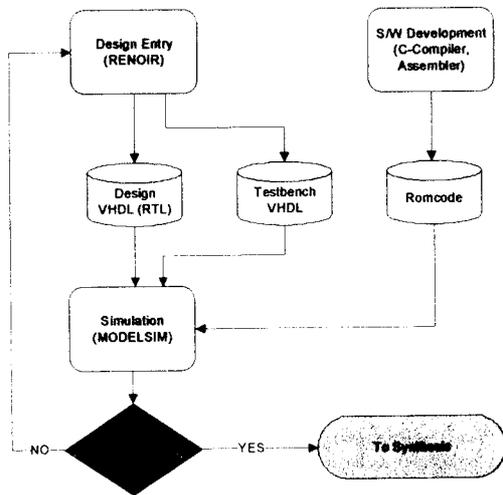


Abbildung 4 : Schaltungseingabe und Simulation

Zur Überprüfung der Synthesefähigkeit des entwickelten VHDL-Codes erfolgt schon während der Schaltungseingabe von Zeit zu Zeit ein Einlesen des Codes in das Synthesewerkzeug, hier LEONARDO von Mentor. Ist die Schaltung nach Meinung des Studierenden ausreichend simuliert worden, so erfolgt nun die Abbildung des gesamten Systems mit LEONARDO auf die FPGA-Zielbibliothek, hier ein Altera Flex10k Baustein (EPF10k10LC84-4). Mit Hilfe der Altera-Werkzeuge wird daraus eine Konfigurationsdatei für den Baustein erzeugt und auf den Baustein geladen. Hierfür wird noch die Hex-Datei für die Testsoftware des Programmspeichers benötigt, die der Studierende ebenfalls entwickeln soll. Dann erfolgt der Test auf einem Evaluationsboard des Altera-FPGAs, der insbesondere auch dem externen Eingang des Timers gelten soll.

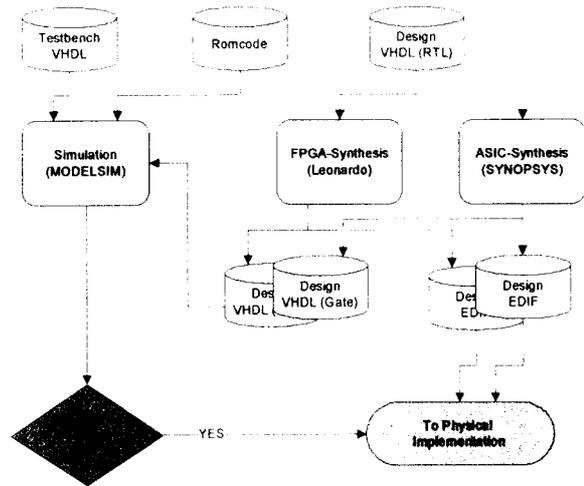


Abbildung 5 : Synthese und Resimulation

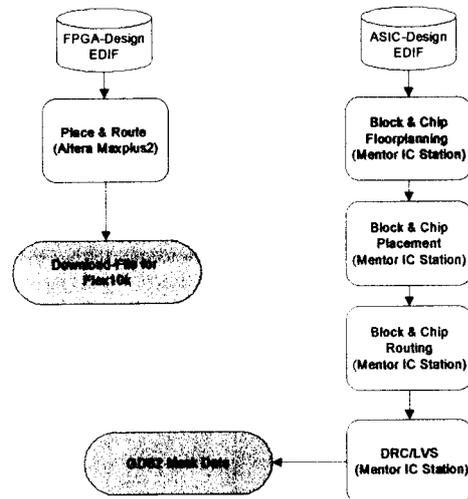


Abbildung 6 : Physikalische Realisierung

Nach erfolgreichem Test des Timers soll das ganze System in ein Standardzellen-ASIC-Layout mit Hilfe der IC-STATION von Mentor umgesetzt werden. Hierfür muß zunächst eine Netzliste durch einen weiteren Syntheseschritt mit dem DESIGN COMPILER von Synopsys erzeugt werden. Verwendet wird hier eine 0.5µm-CMOS-Technologie von Alcatel (C05M-D). Die Standardzellen des Timers werden zunächst als Block platziert und verdrahtet und danach mit den schon fertigen weiteren Blöcken des Systems und den Padzellen ebenfalls platziert und verdrahtet. Mit Hilfe von LVS (Layout-Versus-Schematic) und DRC (Design Rule Check) wird die Korrektheit des Designs überprüft.



Diese Schritte sind durch entsprechende Skriptdateien weitgehend automatisiert, so daß die Studierenden die Werkzeuge mit den Skriptdateien aufrufen müssen und die Ergebnisse verstehen sollen.

In Tabelle 1 sind Ergebnisse der physikalischen Realisierung für die FPGA- und die ASIC-Technologie zusammengefaßt. Der Ressourcenverbrauch bezieht sich dabei nur auf den Prozessorkern und die Peripherieblöcke (Timer und Ports) und beinhaltet nicht die Speicherblöcke. Die maximale Taktfrequenz bei der ASIC-Realisierung sind Werte aus der Timinganalyse, die Werte für das FPGA wurden gemessen.

Technologie	ASIC (C05M-D)	FPGA (EPF10k10LC 84-4)
Ressourcenverbrauch	2290 NAND2	486 Logic Elements
max. Taktfrequenz	60 MHz (= 30 MIPS bei 3V, 85° C)	10 MHz (= 5 MIPS bei 5V, 25° C)

Tabelle 1: Ergebnisse der phys. Realisierung

Die Programmspeichergröße für die FPGA-Realisierung mußte aufgrund der begrenzten Ressourcen für eingebettete Speicher auf 256 Worte (x 12 Bit) beschränkt werden, der Datenspeicher wurde mit 128 Worten (x 8 Bit) realisiert. Damit sind die eingebetteten Speicher des FPGAs zu 66% und die Logikblöcke (Logic Elements) zu 79 % ausgenutzt. Beim ASIC wurde für den Programmspeicher ein Makroblock mit 1024 Worten verwendet und der Datenspeicher synthetisiert.

5 Zusammenfassung

Gezeigt wurde in diesem Beitrag ein Vertiefungslabor welches im Rahmen des Elektro-/Informationstechnikstudiums an der FH Pforzheim belegt werden kann. Es vermittelt Kenntnisse im systematischen Entwurf von komplexen Hardware-/Softwaresystemen unter Verwendung von professionellen Entwicklungswerkzeugen, wie sie auch in der Industrie Verwendung finden.

Basis des Labors ist ein in VHDL selbstentwickelter RISC-IP-Core. Die Studierenden entwickeln für diesen Prozessorkern eine Timer-Peripherieeinheit als VHDL-RTL-Modell und realisieren diese als FPGA-Hardware und als ASIC-Layout.

Hierdurch bekommen die Studierenden einen vollständigen Überblick über eine ASIC-Entwicklung von der Spezifikation über das Prototyping mit einem FPGA bis zum fertigen ASIC-Layout inklusive der Softwareentwicklung. Obwohl das Labor, verglichen mit komplexeren Systemen wie beispielsweise einem Mobilfunk-IC, natürlich im "verkleinerten Maßstab" abläuft, zeigen sich doch die typischen Probleme einer solchen Entwicklung. Hierzu gehören die Vielzahl der Werkzeuge und die Probleme ihrer Benutzung. Weiterhin zeigt sich, ebenso realistisch, daß der Hauptteil der Arbeit im Zyklus Schaltungseingabe - Simulation - Schaltungseingabe verbracht wird und daß sich zwischen der RTL-Simulation und der Hardware Unterschiede ergeben können. Ebenso erkennen die Studierenden, daß das genaue Verständnis der Spezifikation essentiell für das Gelingen eines Projektes ist und die fehlerfreie Umsetzung derselben in eine VHDL-RTL-Beschreibung einer der wesentlichsten Schritte ist. Aber es wird auch erkannt, daß man durch eine systematische und saubere Arbeitsweise Problemstellungen beherrschen kann.

6 Literatur und weitere Informationen

- /1/ <http://www.microchip.com>
- /2/ PIC16C5X Datasheet, DS30453B; Microchip Technology Inc., 1998
- /3/ The Designers Guide to VHDL; P.J.Ashenden; Morgan Kaufman; 1994
- /4/ <http://www.ubicom.com>



Entwurf eines CAN-Controllers

mit Hilfe von grafischen VHDL-Tools

Stephan Baur,
Prof. Dr. Hans Kreuzer
Fachhochschule Reutlingen

Da oftmals bei verschiedenen Projekten eine CAN-Busanbindung erwünscht ist, man aber nicht immer auf kommerzielle Controller ausweichen wollte, lag es nahe, einen CAN-Controller zu entwickeln. Dieser sollte bestimmte Eigenschaften mitbringen, die es ermöglichen den Controller sowohl als eigenständigen Baustein, als auch als Modul in einem IC-Design verwenden zu können. Des weiteren wurden hohe Ansprüche an die Wartbarkeit und damit Übersichtlichkeit gestellt. Damit kann der Baustein einfach weiterentwickelt und zu Lernzwecken verwendet werden.

1. Einführung

1.1. FPGA-Advantage

Um eine Entwicklung eines solch komplexen Controllers im Rahmen einer Diplomarbeit zu ermöglichen, mußten sehr leistungsfähige Tools verwendet werden. Auch sollte ein großes Maß an Übersichtlichkeit erreicht werden. Um ein komplexes Gebilde übersichtlich darstellen zu können sind am besten Grafiken und Diagramme geeignet. Für die Darstellung von Automaten hat sich beispielsweise in der Digitaltechnik das „Zustand-Übergangs-Diagramm“ bewährt. Mit dem Programm „Renoir“, das im Programmpaket FPGA-Advantage von Mentor Graphics enthalten ist, bekommt man solche Hilfsmittel an die Hand. Es besteht die Möglichkeit mit Hilfe von Tabellen, Flußdiagrammen und Zustand-Übergangs-Diagrammen Funktionen darzustellen und mittels Blockschaltbildern zu verknüpfen. Aus diesen Diagrammen und Schaltbildern kann dann ein VHDL-Quelltext erzeugt werden. Das Programmpaket ermöglicht auch die Simulation und Synthese des Designs. Ein großer Vorteil von VHDL ist die Möglichkeit bis zur Synthese weitgehend unabhängig von der Zieltechnologie zu bleiben. So kann das Design einfach auf die unterschiedlichsten

Technologien portiert werden und bleibt somit auch zukunftssicher.

1.2. CAN

Der CAN-Bus (Controller Area Network) ist ein Bussystem, das vorwiegend im PKW-Bereich und in der Automatisierungstechnik eingesetzt wird. Er wird zur Kommunikation zwischen Sensoren, Aktoren und Steuereinheiten verwendet. Die Übertragung erfolgt nachrichtenorientiert, d.h. nicht die einzelnen Bausteine, sondern die zu übertragenden Nachrichtentypen bekommen eine Adresse. Zum Beispiel hat die Nachricht „Fahrzeuggeschwindigkeit“ eine eigene Adresse. Wenn nun ein beliebiger Baustein diese Information benötigt, muß er nur warten bis die Nachricht mit der entsprechenden Adresse übertragen wird, oder diese selbst anfordern. Hierbei muß dem Baustein lediglich die Adresse für die benötigte Nachricht bekannt sein, nicht aber der Baustein, der diese Nachricht erzeugt.

Die Topologie eines CAN-Netzwerkes ist in der Regel linienförmig. Durch den Einsatz von Routern, Sternverteilern oder Repeatern sind auch andere Strukturen möglich.

Die Anzahl der Teilnehmer wird durch die verwendeten Leitungstreiberbausteine bestimmt. Da das Protokoll nachrichtenorientiert arbeitet, ist die Anzahl der möglichen Nachrichten, nicht aber die Anzahl der Bausteine durch die Adressierung begrenzt.

Die maximale Ausdehnung eines CAN-Netzwerkes hängt von den Laufzeiten auf dem jeweiligen Übertragungsmedium ab. Neben Zweidraht-Cu-Leitungen kommen auch Kunststoff- oder Glasfaser-Lichtwellenleiter zum Einsatz.

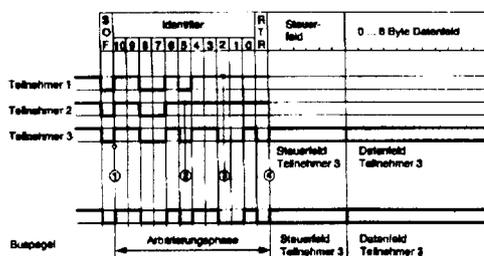
Die nachrichtenorientierte Adressierung legt zugleich auch die Priorität der Nachricht fest. So kann sichergestellt werden, daß besonders wichtige Nachrichten auch bei hoher Busbelastung bevorzugt behandelt werden und so den Zugang zum Bus mit kurzer Latenzzeit erlangen.



Das Bussystem arbeitet nach einem Multimasterprinzip. Jeder Teilnehmer darf selbständig zu senden beginnen, ohne erst das Zugriffsrecht von einer höheren Instanz erlangen zu müssen.

Die Priorisierung erfolgt durch den Arbitrierungsprozess. Hierbei sendet jeder Teilnehmer, der eine Nachricht senden will die Adresse der entsprechenden Nachricht bitweise. Die Buspegel sind so definiert, daß es dominante und rezessive Zustände gibt, wobei ein dominanter Zustand einen rezessiven „überschreibt“. Stellt nun ein sendender Teilnehmer, der einen rezessiven Pegel im Arbitrierungs- bzw. Adressfeld gesendet hat, fest, daß auf dem Bus ein dominanter Zustand vorhanden ist, so kann er erkennen, daß ein anderer Teilnehmer eine Nachricht mit höherer Priorität senden will. Er muß deshalb das Senden bis zum nächsten Telegramm einstellen und darf dann einen erneuten Versuch starten. Die höchste Priorität hat also die Adresse, die nur aus dominanten Zuständen besteht und die niedrigste demzufolge diejenige, die nur aus rezessiven Pegeln besteht.

Obwohl mehrere Teilnehmer gleichzeitig zu senden beginnen können, ist keine Zerstörung und damit wiederholtes Senden der Nachricht nötig, da nach der Arbitrierungsphase nur der Teilnehmer mit der am höchsten priorisierten Nachricht mit senden fortfährt.



Beispiel eines Arbitrierungsvorgangs im CAN-Protokoll. Die Teilnehmer 1, 2 und 3 beginnen gleichzeitig einen Arbitrierungsversuch (1). Teilnehmer 2 verliert zum Zeitpunkt (2). Teilnehmer 1 zum Zeitpunkt (3) das Buszugriffsrecht. Beide Teilnehmer gehen damit in den Empfangszustand; am Ende der Arbitrierungsphase (4) besitzt nur noch Teilnehmer 3 das Buszugriffsrecht. Nur dieser Teilnehmer sendet seine Nachricht über den Bus.

Abb. 1.2.1 Arbitrierung

Die Telegrammtypen unterteilen sich in Anforderungsrahmen, Datenrahmen, erweiterter Datenrahmen, Fehlerrahmen und Überlastrahmen.

Mit Hilfe des Fehlerrahmens kann signalisiert werden, daß ein Fehler erkannt wurde.

In einem CAN-System können verschiedene Fehlerzustände erfaßt und signalisiert werden. Für den Bereich vom Start-Of-Frame (SOF) bis zum CRC-Feld (ohne Stuffbits) wird eine zyklische Blockprüfung (Cyclic Redundancy Check, CRC) vorgenommen und die gewonnene CRC-Sequenz wird im CRC-Feld übertragen. Der Empfänger kann mit Hilfe

dieser Sequenz feststellen, ob ein Fehler (CRC-Fehler) bei der Übertragung aufgetreten ist, wobei die Restfehlerwahrscheinlichkeit sehr gering ist.

Die Empfänger quittieren den erfolgreichen Empfang einer Nachricht mit dem dominanten Acknowledge-Bit (ACK-Bit). Bleibt dieses Bit aus, hat kein Teilnehmer die Nachricht korrekt empfangen (ACK-Fehler).

Alle Teilnehmer überwachen ständig die Einhaltung der festgelegten Form der Rahmen und zeigen eine Nichteinhaltung mit einem Formfehler an.

Ebenfalls überwacht wird das Bitstuffing. Beim Bitstuffing wird nach 5 gleichen Bits ein entgegen gepoltes Bit eingefügt, um die Taktrückgewinnung zu ermöglichen. Werden im Bitstuffingbereich mehr als 5 gleiche Bits erkannt, so wird ein Bitstuffingfehler signalisiert.

Die Sender kontrollieren ob die gesendeten Pegel mit dem tatsächlichen Pegel auf dem Bus übereinstimmen. Ist dies nicht der Fall, wird ein Bit-Fehler erkannt (Ausnahme ist z.B. das Arbitrierungsfeld oder der ACK-Slot).

Um zu verhindern, daß fehlerhaft arbeitende CAN-Knoten den Bus durch Fehlerrahmen blockieren, gibt es unterschiedliche Zustände des Controllers. Es gibt den fehleraktiven Zustand, bei dem ein gesendeter Fehlerrahmen die Nachricht eines Senders zerstören kann, um einen Fehler zu signalisieren. Ist der CAN-Knoten hingegen im fehlerpassiven Zustand kann er lediglich einen modifizierten Fehlerrahmen senden, der nicht in der Lage ist, eine Nachricht eines Senders zu zerstören aber bei eigenen Sendungen schnellstmöglich den Fehlerfall signalisieren kann. Der Bus-Off-Zustand schaltet den Knoten komplett vom Bus.

In welchem Zustand sich der jeweilige Knoten befindet wird anhand von Sende- und Empfangs-Fehlerzählern ermittelt, wobei diese Zähler nach einem komplexen Regelwerk in- bzw. dekrementiert werden.

Der Anforderungsrahmen wird eingesetzt, wenn eine bestimmte Nachricht bzw. bestimmte Daten benötigt werden. Mit ihm kann man einen anderen Baustein veranlassen, die gewünschte Nachricht zu übermitteln.

Der Datenrahmen wird zur Übertragung einer Nachricht mit zugehörigen Daten eingesetzt.

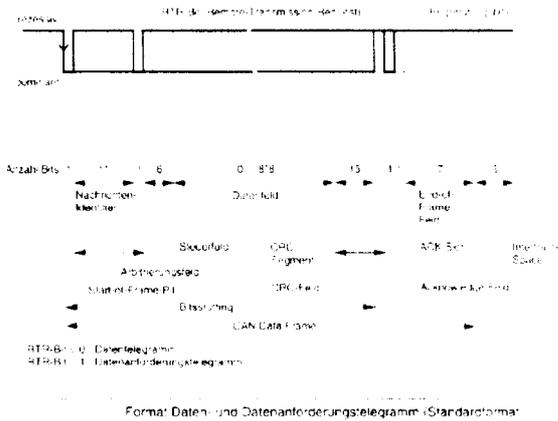


Abb. 1.2.2 CAN-Standardrahmen

Das Aussehen der weiteren möglichen Übertragungsrahmen soll hier nicht näher beschrieben werden. Alle folgenden Ausführungen stützen sich auf das Standardformat.

2. Struktur des CAN-Controllers

In diesem Kapitel sollen die Strukturen und die Abläufe innerhalb des Controllers beispielhaft erläutert werden.

2.1. Hauptblattschaltbild (TOP)

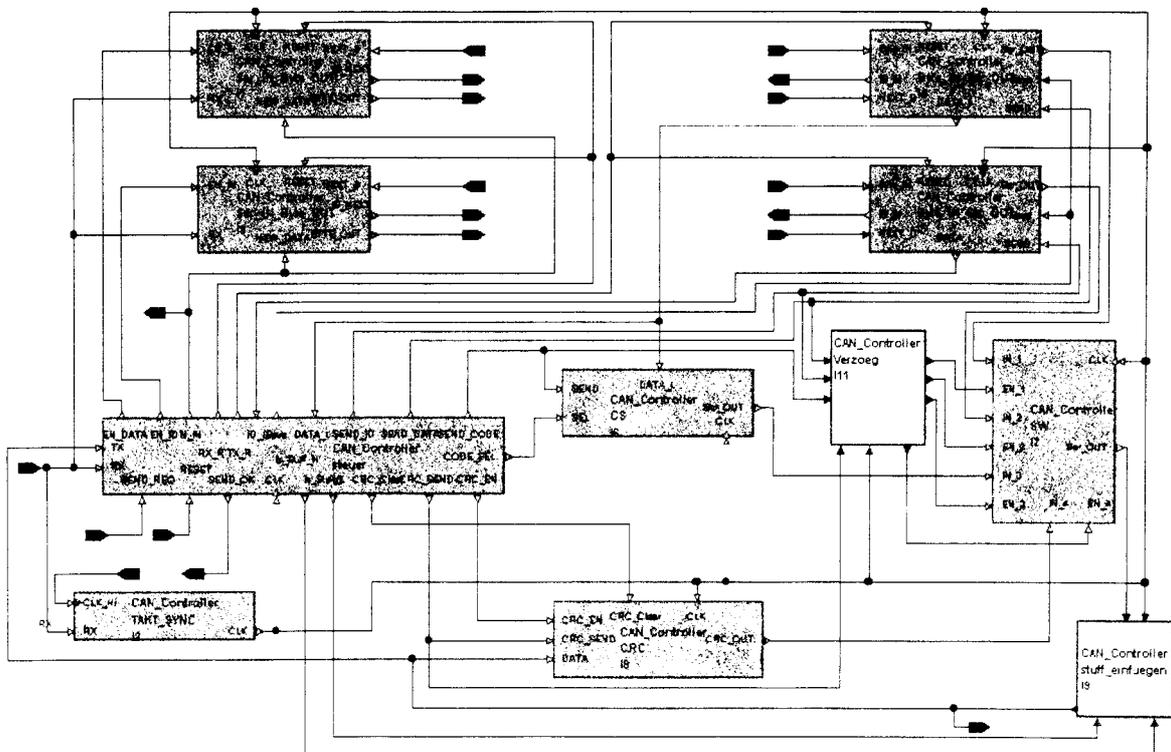


Abb. 2.1.1 Blattschaltbild (TOP)

Dies ist die Schicht, mit deren Schnittstellen nach außen der Anwender des CAN-Controllers nur die Grundfunktionalität beinhaltet, um flexibel für die unterschiedlichsten Anwendungen zu bleiben, ist es denkbar, daß über diese Schicht eine weitere eingefügt wird, die die Funktionalität des CAN-Controllers um beispielsweise Empfangs- und Sendepuffer erweitert.

Diese Schicht (TOP) enthält eine Schnittstelle zur BUS-Seite bzw. zum Leitungstreiber die aus einer Sendeleitung (TX) und einer Empfangsleitung (RX) besteht. Als „übergeordnete Geräte“ sind entweder Mikrocontroller oder andere Logik-Bausteine möglich. Aber auch ein Einsatz des CAN-Controllers als Modul, also Bestandteil eines IC-Designs ist denkbar. Zu diesen „übergeordneten Geräten“ bietet der CAN-Controller eine Schnittstelle, die je einen 8 Bit breiten Bus für das Laden der Adresse, das Speichern der Adresse, das Laden der Daten und das Speichern der Daten beinhaltet. Für die Signalisierung von verschiedenen Zuständen gibt es weitere Signalleitungen. Um Komponenten mit vom Bittakt verschiedene und nicht synchrone Taktraten anbinden zu können, ist ein sogenanntes Handshake-Protokoll für die verschiedenen Aktionen nötig, das ausschließt, daß Daten verloren gehen oder falsch interpretiert werden.



Prinzip eines Handshake-Protokolls:



Abb. 2.1.1

Ablauf des Sendevorgangs:

Zunächst werden die Byte_In_Ser_Out-Blöcke mit den zu übertragenden Daten, der Adresse und dem Anforderungsbit (RTR) geladen. Danach wird überprüft, ob das Send_OK-Signal eine logische Null führt. Ist dies der Fall, kann das Send_Req-Signal gegeben werden ('1'). Hiermit sind die nötigen Vorbereitungen, die von Außen vorgenommen werden müssen, abgeschlossen und die Steuerung übernimmt alle nötigen Schritte einen der Norm entsprechenden Rahmen zu generieren und zu übertragen. Hierzu wird zunächst der CodeSender (CS) über die entsprechenden Signale veranlaßt, das Start-Of-Frame-Bit (SOF) zu generieren und zu senden. Eine Taktperiode später liegt das Bit am Ausgang des CS an. Damit zu diesem Zeitpunkt der Umschalter (SW) dieses SOF-Bit durchschaltet ist eine Verzögerung im zugehörigen Send-Signal notwendig. Dies übernimmt der Verzögerungsblock (Verzoeg). Das SOF-Bit wird noch einmal im Block „stuff_einfuegen“ zwischengespeichert. Da aber zu diesem Zeitpunkt noch kein Stuffbit nötig ist, wird das gespeicherte Bit während der nächsten Taktperiode gesendet. Die Steuerung hat inzwischen schon den Byte_In_Ser_Out-Block für die Adressen veranlaßt, die Adresse (ID) mit dem RTR-Bit im Bittakt zu versenden. Wie schon zuvor beim SOF-Bit wird auch hier das zugehörige Send-Signal im „Verzoeg“-Block verzögert und das gesendete Bit im „stuff_einfuegen“-Block zwischengespeichert, so daß der Rahmen unterbrechungsfrei in der richtigen Abfolge gesendet wird. Als Nächstes wird das Steuerfeld mit der aus dem „Byte_In_Ser_Out“-Block für die Daten erhaltenen Datenlängenangabe (Data_L) über den CS versendet. Darauf folgen die Daten aus dem „Byte_In_Ser_Out“-Block für Daten, die CRC-Sequenz aus dem „CRC“-Block und zuletzt folgen die „CRC_Del“- „ACK_Slot“- und „ACK_Del“-Bits aus dem CS. Auch hier schaltet der Umschalter (SW) die jeweiligen Leitungen dank der Verzögerung des „Verzoeg“-Blocks zum richtigen Zeitpunkt zum „stuff_einfuegen“-Block durch. Sollte es nötig werden, ein Stuffbit zu senden, erhält der „stuff_einfuegen“-Block die Anweisung hierzu

ebenfalls von der Steuerung. Wurde die Nachricht komplett, ohne Fehlermeldung (entweder eigene, oder von einer entfernten Station) versendet, wird das „Send_OK“-Signal gegeben.

Auch das Senden von Sonderrahmen, wie Fehlerrahmen oder Überlastrahmen, ist mit Hilfe des Codesenders (CS) möglich.

• Ablauf des Empfangsvorgangs:

Der Empfangsvorgang ist im Gegensatz zum Sendevorgang immer aktiv. Auch Nachrichten vom eigenen Sender werden zur Kontrolle empfangen.

Beim Empfangen werden die meisten Informationen, die neben den Daten in einem CAN-Übertragungsrahmen enthalten sind, direkt in der Steuerung verarbeitet. Nur die eigentlichen Nutzdaten und die zugehörige Adresse werden in den „Ser_In_Byte_Out“-Blöcken abgespeichert und können später ausgelesen werden. Wenn bei der Übertragung die Bits, die zur Adresse oder zum Datenfeld gehören, übertragen werden, wird der Eingang des entsprechenden Blocks freigegeben und die Daten im Bittakt eingelesen. Sollte ein Stuffbit enthalten sein, so sperrt die Steuerung den Eingang, so daß diese nicht mit eingelesen werden. Wenn die Stelle des „ACK“-Bits erreicht ist und kein Fehler im bisher gesendeten Rahmen gefunden wurde, so veranlaßt die Steuerung den Sender, die Nachricht mit einem dominanten „ACK-Bit“ zu quittieren, es sei denn, die Nachricht wurde vom eigenen Sender übermittelt. Ist die Übertragung fehlerfrei beendet, so wird die Nachricht mit dem „N_IN“-Signal freigegeben. Dieses Signal wird auch nach Außen weitergegeben, um dem übergeordneten Gerät die Ankunft einer Nachricht zu signalisieren. Es ist aber darauf zu achten, daß die Nachricht erst nach einem weiteren Bittaktzyklus tatsächlich bereitsteht!

2.2. Beispiele für die Realisierung einzelner Blöcke

In diesem Kapitel soll anhand von einfachen Beispielen gezeigt werden, wie man die einzelnen Blöcke bzw. deren Funktionen mit Hilfe der im FPGA-Advantage-Programmpaket enthaltenen Tools umsetzen kann.



Beispiel 1:
„oder“-Funktion mit Hilfe einer Wahrheitstabelle:

	A	B	C	D	E	F
1	BIT_Fehler	ACK_Fehler	Stufing_Fehler	CRC_Fehler	Formatfehler	
2						1
3		1				1
4			1			1
5				1		1
6					1	1
7	1	1	1	1	1	1

Abb. 2.2.1 „oder“-Tabelle

Hierbei wird den verschiedenen Eingangszuständen der zugehörige Ausgangszustand zugeordnet.

Beispiel 2:
Die Funktion des Code-Sender-Blocks (CS) wurde mit Hilfe eines Flußdiagramms realisiert:

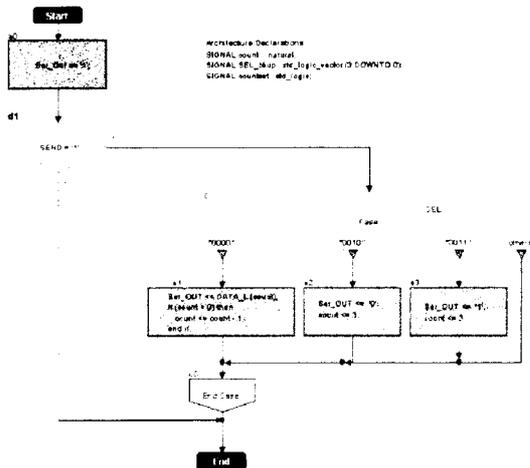


Abb. 2.2.2 Flußdiagramm (CS)

Beispiel 3:
Mit Hilfe eines Zustand-Übergangs-Diagramms wurde die „Stuff-Einfügen“-Funktion (SE) beschrieben.

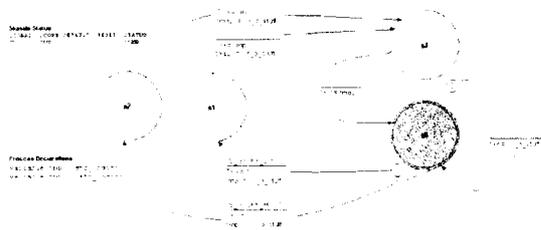


Abb. 2.2.3 Zustand-Übergangs-Diagramm (SE)

3. Zusammenfassung und Ausblick

Durch die vielfältigen und leicht verständlichen Funktionen des FPGA-Advantage-Programmpaketes ist es gelungen in relativ kurzer Zeit einen funktionierenden CAN-

Controller zu entwickeln, der Dank seines übersichtlichen Aufbaus leicht bis zur Serienreife weiterentwickelt werden kann. Da bei diesem Projekt aber weniger die Serientauglichkeit, als vielmehr der Wert des Entwurfs zu Lernzwecken im Vordergrund stand, ist zu erwarten, daß in der Zukunft noch viele Projekte, basierend auf diesem Entwurf entstehen werden. Auch ein ausführlicher Test mit einer dafür entworfenen Testumgebung konnte aus Zeitmangel noch nicht durchgeführt werden und steht demzufolge noch aus.

Literatur:

1. Konrad Etschberger
Controller-Area-Network
Grundlagen, Protokolle, Bausteine, Anwendungen
ISBN 3-446-19431-2
Abb. 1.2.1 und 1.2.2 sind aus diesem Buch entnommen
2. Horst Engels
CAN-BUS
CAN-BUS-Technik einfach, anschaulich und praxisnah vorgestellt
ISBN 3-7723-5145-X
3. International Standard ISO 11898
4. Prof. Dr. H. Kreutzer
Dipl.-Ing. F. Stockmayer
CAE Vorlesungsunterlagen
5. Gunther Lehmann, Bernhard Wunder, Manfred Selz
Schaltungsdesign mit VHDL
6. Can-Bus Einführung
http://www.htl2.asn-linz.ac.at/intranet/elektronik/can_bus/Can_einf.html
7. Bosch CAN-Website
<http://www.can.bosch.com>

Entwicklung einer Videopipeline mit digitaler Echtzeit-Bildverarbeitung

Diplomarbeit von Christian Reichherzer

Fachhochschule Aalen

Fachbereich Elektronik/Technische Informatik

Beethovenstrasse 1, 73430 Aalen

In erster Linie dient die von mir entwickelte Videopipeline zur Bildmanipulation monochromer Videosignale in Echtzeit. Die Manipulation beinhaltet die Gewichtung des digitalisierten Videosignals anhand einer Gamma-Übertragungskennlinie, die Implementierung digitaler Filter sowie die optionale Darstellung von Monitortestbildern. Die Videopipeline ist als Stand-Alone-System konzipiert, d.h. sie arbeitet völlig autark und ist auf keinerlei externe Synchronisiersignale angewiesen. Es sind lediglich eine BNC-Videoeingangs- und BNC-Videoausgangsbuchse vorhanden, über die das BAS-Videosignal eingespeist und am Ausgang an nachfolgende Videosysteme gefiltert ausgegeben wird. Aufgrund der einfachen Handhabung der Videopipeline und der Möglichkeit zur Manipulation von Helligkeit und Kontrast, findet die Videopipeline im Bereich der bildgebenden Verfahren in der radiologischen Skelettdiagnostik sicherlich ihre Anwendung. Die Videopipeline bietet die Möglichkeit Tiefpass-, Rechteck- und Medianfilter zur Abflachung von Kanten und Oszillationen, Unterdrückung von Rauschen und Verschmierung von Störungen zu integrieren. Die an diese Diplomarbeit anknüpfende Studienarbeit beschäftigt sich daher ausgehend mit der Kantendetektion durch Differenzfilter und der Kombination von Kantendetektion und Rauschunterdrückung.

Aufgabenstellung

In dieser Diplomarbeit soll unter Einsatz eines FPGAs eine Videopipeline realisiert werden. Dazu muss ein der PAL/NTSC-Norm entsprechendes Videosignal digitalisiert und dem FPGA zugeführt werden. Der FPGA führt eine digitale Filterung der Eingabe-Bilder durch. Die gefilterten Bilder sollen wieder als analoges Videosignal am Ausgang der Videopipeline bereitgestellt werden. Ziel dieser Arbeit soll es sein, ein autarkes System aufzubauen, welches auf einfache Weise zwischen einem Bildaufnahmegerät und z.B. einem Monitor geschaltet wird. Die Bilder sollen mit unterschiedlichen digitalen Filtern in Echtzeit bearbeitet werden.

Es sollen folgende Entwicklungswerkzeuge eingesetzt werden:

- Orcad 9.1 von Cadence für die Schaltungseingabe, Simulation und Layoutentwicklung
- Programmpaket FPGA Advantage von Mentor Graphics sowie Alliance von Xilinx für die FPGA-Entwicklung

Die Videopipeline im Überblick

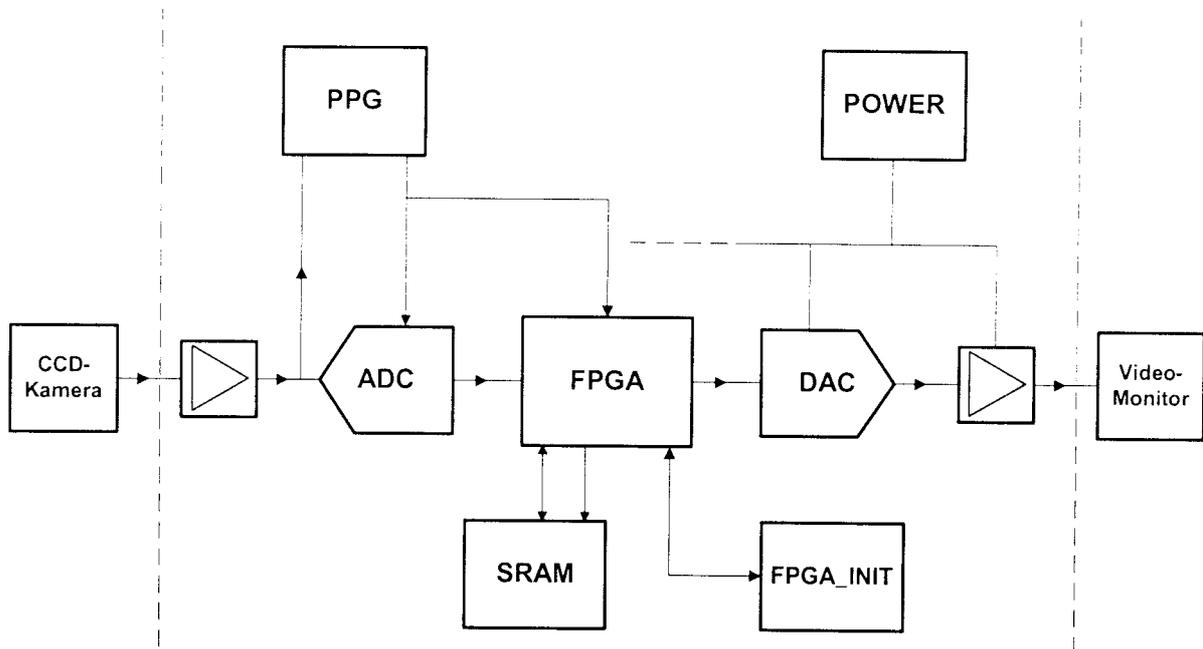


Abbildung 1: Übersichtsblockschaltbild der Videopipeline

Platzierung der Blöcke

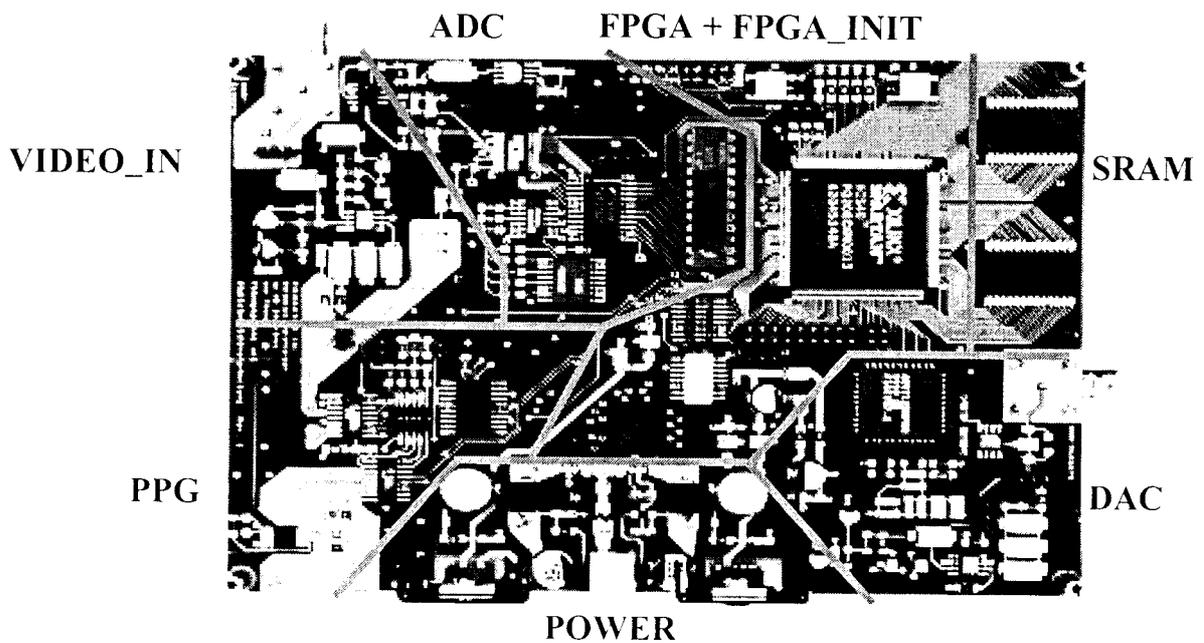


Abbildung 2: Aufteilung der Blöcke auf der Platine

Spannungsversorgungs- und Massekonzept

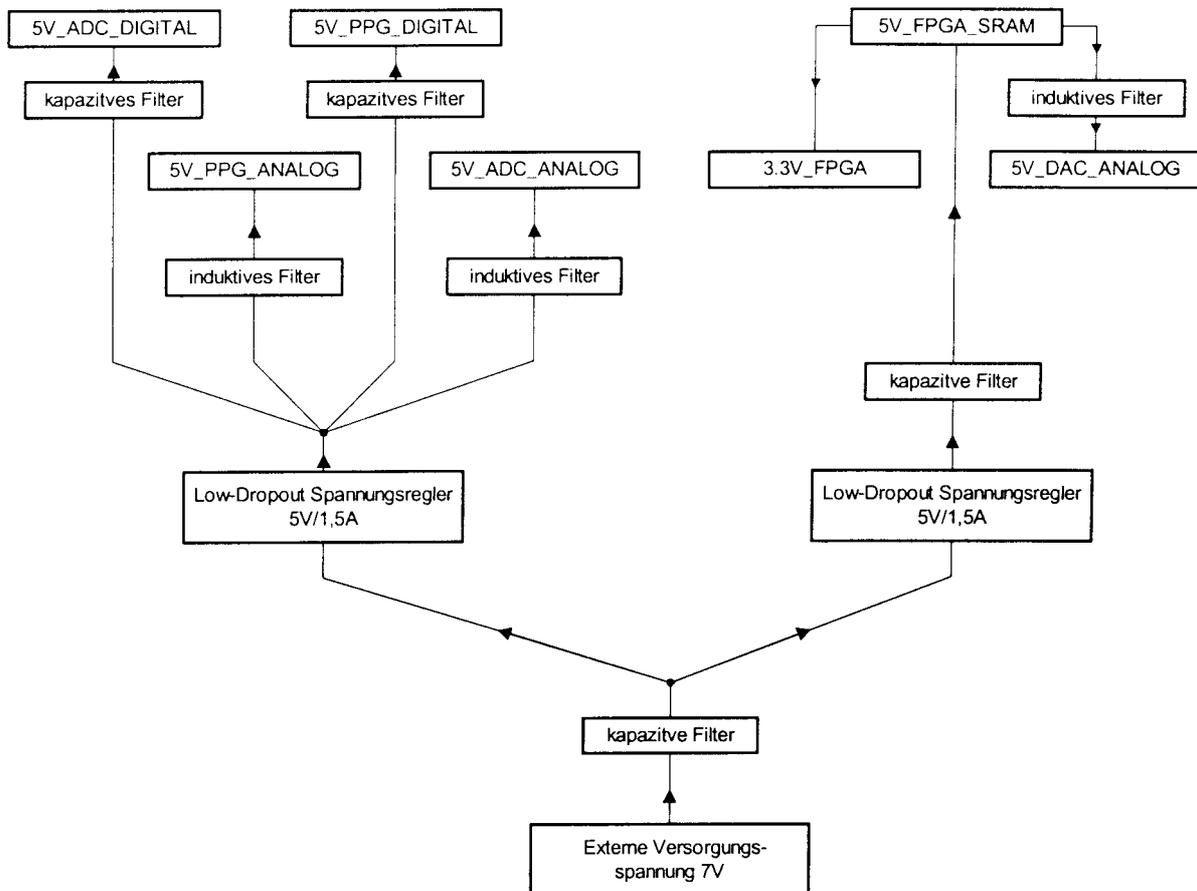


Abbildung 3: Spannungsversorgungskonzept der Videopipeline

Auf der Videopipeline werden insgesamt 7 unterschiedliche Versorgungsspannungen zur Verfügung gestellt, wobei diese wiederum in Digitale und Analoge unterteilt sind. Die digitalen Spannungspfade werden mit den kapazitiven Filtern beschaltet. Die analogen Pfade werden dagegen mit induktiven HF-Filtern, die sich besonders für den Einsatz in Videosystemen eignen, realisiert.

Neben der aufwendigen Filterung der Versorgungsspannungen, müssen auch die Auswirkungen von Störeinkopplungen auf der Platine selbst minimiert werden. Dafür bieten sich 2 Strategien an. Zum einen müssen analoge und digitale Masseflächen physikalisch voneinander getrennt werden und nur an einem gemeinsamen Punkt, dem sogenannten Massesternpunkt, zusammengeführt werden (siehe Abbildung 4). Zum anderen werden die signalführenden Lagen an der Ober- und Unterseite der Platine durch die Masse- und Versorgungsspannungslage im Inneren voneinander kapazitiv entkoppelt. Um zudem noch eine Abschirmung von Störungen aus der Außenwelt zu gewährleisten, wäre anstatt einer 4-lagigen eine 6-lagige Platine vorzuziehen, um die signalführenden Schichten ganz im Inneren der Platine zu isolieren.

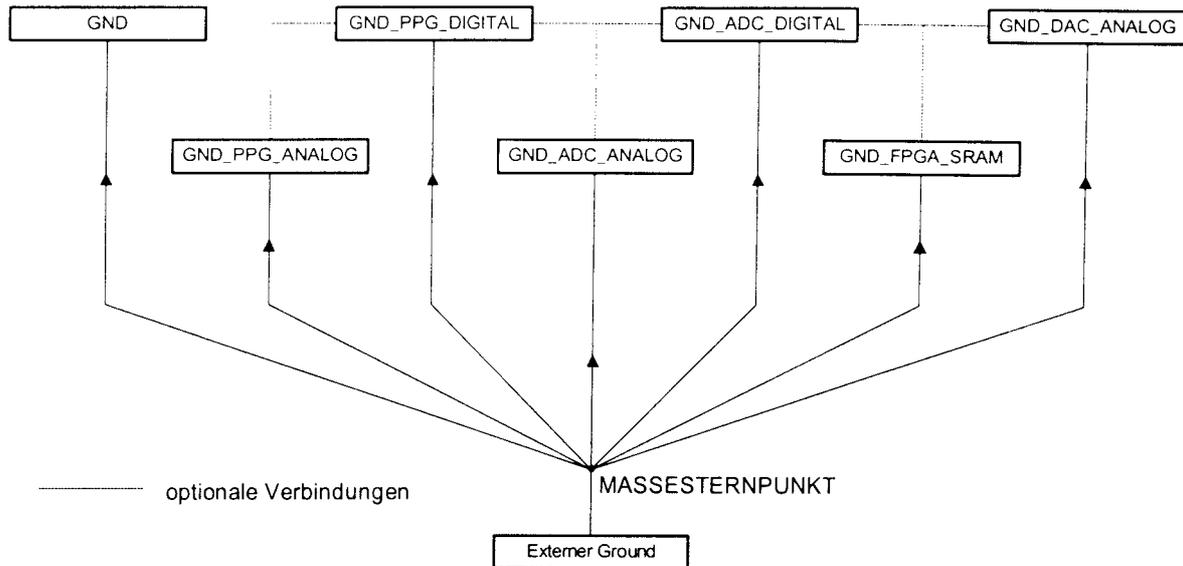


Abbildung 4: Sternförmiges Massekonzept mit optionalen Verbindungen

Zusammenfassend bedeutet dies für das Grundkonzept der Videopipeline, dass keine Störungen auf der Platine erzeugt, von der Platine heruntergelassen und auf die Platine heraufgelassen werden. Erreicht werden diese Forderungen durch die schon erwähnten Maßnahmen wie zum Beispiel die Verwendung von langsamen Bauteilen (z.B. induktive Filter), die Schaffung von größtmöglichen Masseflächen, die Gruppenbildung nach Störverhalten (digitale und analoge Massen) und die Breitbandentkopplung der Spannungsversorgung.

Puls Pattern Generator (PPG)

Der Puls Pattern Generator PPG dient zur Regenerierung der Synchronisiersignale und des Pixeltaktes aus dem BAS-Signal. Es werden Signale Composite Sync CSYNC, Horizontal Sync HSYNC, Vertical Sync VSYNC und ODD/EVEN für die Bildverarbeitung im FPGA bereitgestellt. Neben diesen Synchronisiersignalen werden hier auch die einfachen Pixeltakte CLK_FPGA, CLK_ADC und CLK_LATCH sowie der doppelte Pixeltakt DOUBLE_CLK_FPGA erzeugt. Für das PAL-Verfahren nach CCIR601-Norm haben die Takte eine Frequenz von 13,5MHz bzw. 27MHz. Beide dienen als Systemtakte für das FPGA. Zur besseren Veranschaulichung für die nun folgenden Betrachtungen soll das folgende Blockschaltbild des PPGs dienen.

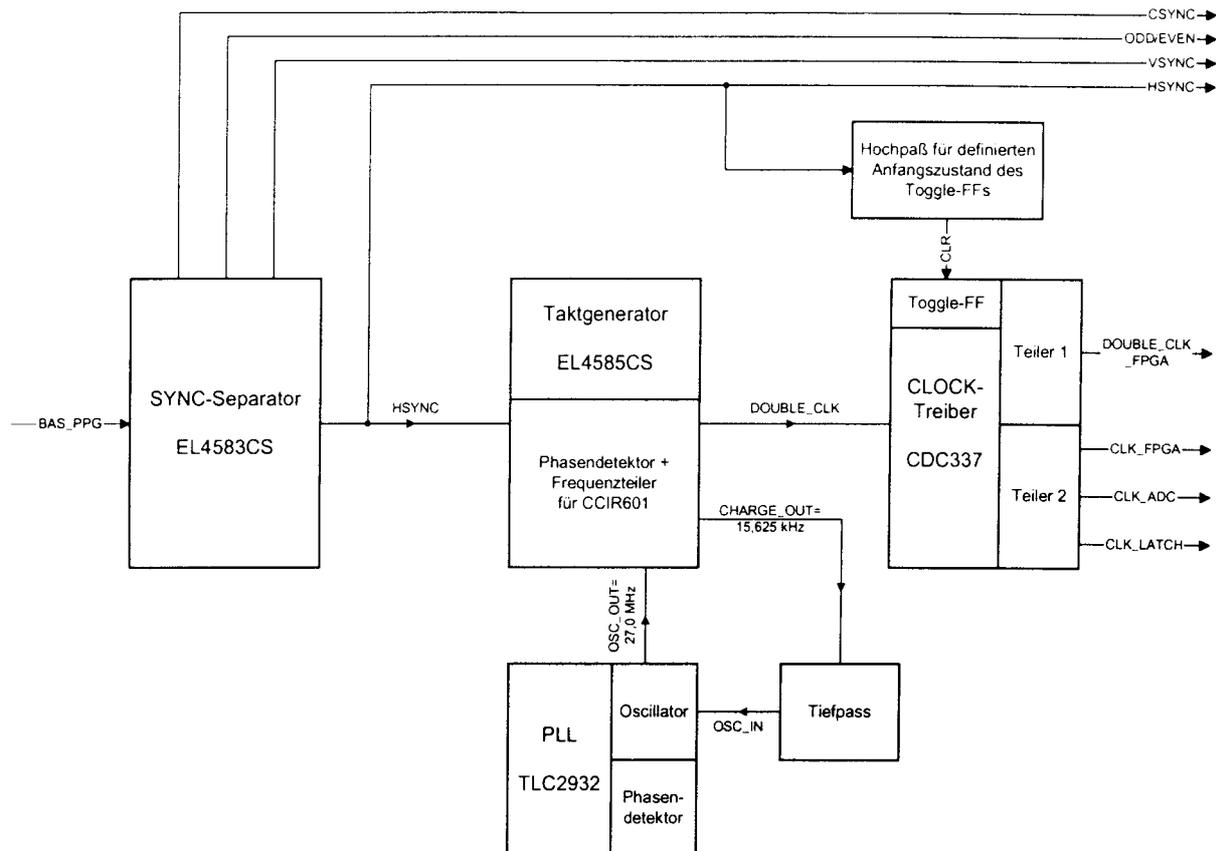


Abbildung 5: Blockschaltbild des Puls Pattern Generator

Sync-Separator EL4383CS

Der EL4583CS übernimmt in der Videopipeline eine entscheidende Rolle bei der Restauration des Videosignals BAS_PPG. Er generiert die Signale HSYNC, CSYNC, VSYNC, ODD/EVEN und CLAMP_ELANTEC aus dem Videosignal BAS_PPG heraus und bildet zugleich die Vorstufe für die Pixeltakterzeugung.

Taktgenerator EL4585CS mit Phase Locked Loop TLC2932

Als Referenzsignal für den EL4585CS dient das horizontale Synchronisiersignal HSYNC des Sync-Separators EL4583CS. Mit einem externen Schalter werden die Frequenzteilverhältnisse zwischen dem Pixeltakt $f_{CLKOUT} = 27,0\text{MHz}$ und der Horizontalfrequenz $f_{HSYNC} = 15,625\text{kHz}$ eingestellt. Daraus ergibt sich ein Teilungsverhältnis von 1728. Es kann also theoretisch eine Auflösung von 1728 Pixel pro Zeile realisiert werden.

Die PLL TLC2932 bietet eine flexible Einbindung in die Peripherie des EL4585CS, da der Phasendetektor und der spannungsgesteuerte Oszillator VCO (voltage controlled oscillator) getrennt verwendet werden können.

Analog/Digital-Wandlung mit optionaler Gammakorrektur (ADC)

In der ADC-Teilschaltung (siehe Abbildung 6) wird das von der Videoeingangsstufe kommende Signal `BAS_ADC` von einem 10-bit Analog/Digital-Wandler digitalisiert und anschließend über ein 10-bit breites Übernahmeregister entweder direkt oder über ein Gamma-EPROM an das FPGA geführt. Außerdem beinhaltet der ADC-Block einerseits eine Spannungsreferenzschaltung und andererseits eine Verzögerungsschaltung die für das Wechselspiel zwischen A/D-Wandler und Übernahmeregister verantwortlich ist.

Das Herzstück der ADC-Teilschaltung bildet der A/D-Wandler AD9200ARS von ANALOG DEVICES. Dabei handelt es sich um einen auf CMOS-Technologie basierenden 10-bit A/D-Wandler mit einer bis zu 20 MSPS (mega samples per second) reichenden Dateneingangsrate. Was den Baustein aber besonders für den Einsatz zur Digitalisierung von Videosignalen auszeichnet, ist die interne Eingangsklemmstufe zur Restauration von Videosignalen. Diese ermöglicht es, wechselspannungsgekoppelte Analogsignale auf eine definierte Spannung festzuklemmen.

Zeitliches Verhalten der ADC-Teilschaltung

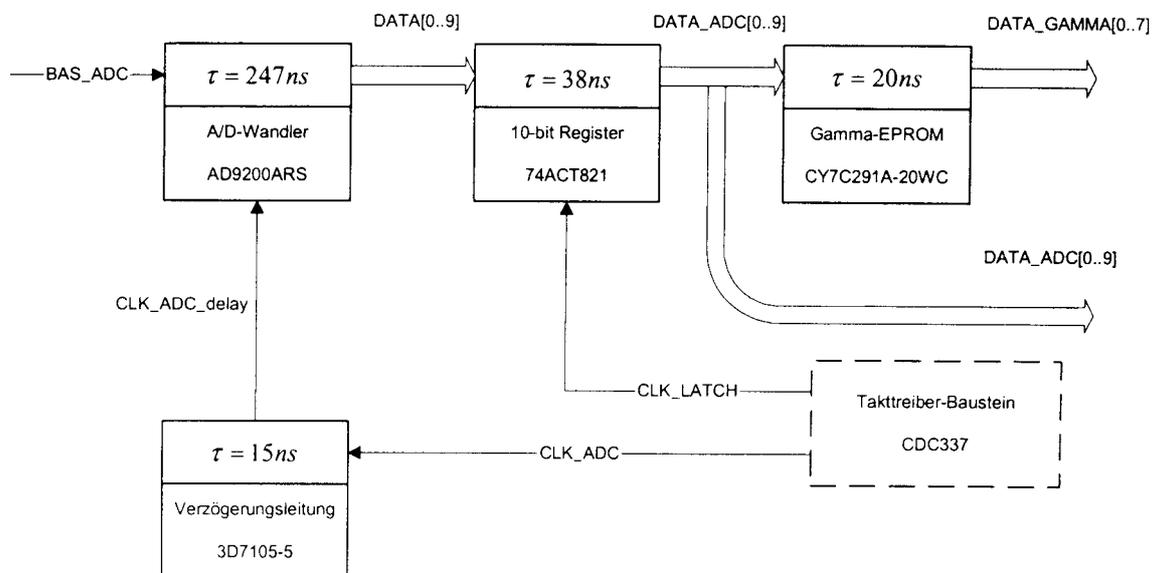


Abbildung 6: Blockschaltbild für das Verzögerungsprinzip der ADC-Teilschaltung

Im Vorfeld des FPGAs sind die zeitlichen Verhalten des Puls Pattern Generators und der ADC-Teilschaltung von enormer Bedeutung, da durch diese der FPGA-interne Abtastzeitpunkt für die jeweilige sichtbare Zeile des entsprechenden Halbbildes festgelegt wird. Zum prinzipiellen Verständnis des Verzögerungsprinzips sind in Abbildung 6 die jeweiligen Logikblöcke der ADC-Teilschaltung herausgearbeitet.

Field Programmable Gate Array mit ISP (FPGA und FPGA INIT)

FPGA XCS40

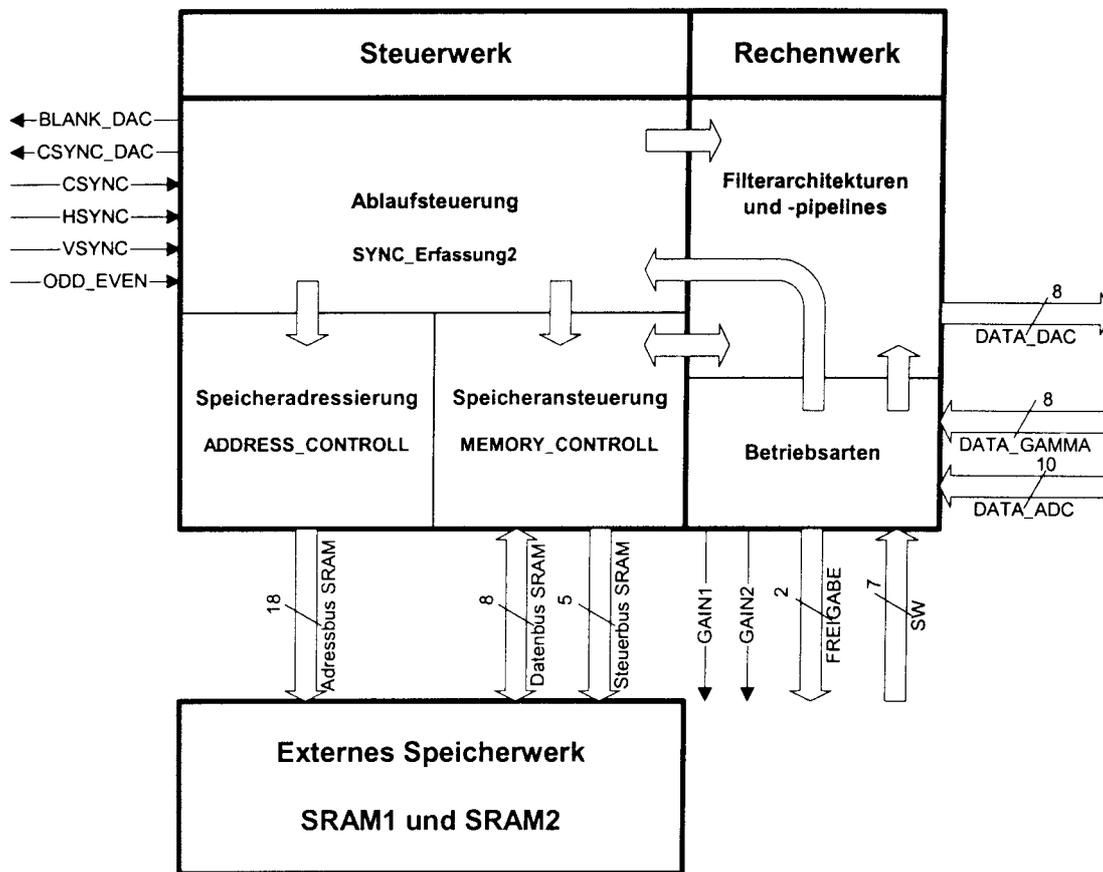


Abbildung 7: Die Bildverarbeitung im FPGA XCS40-4 PQ208

Die Manipulation der digitalen Bilddaten wird durch das FPGA XCS40 aus der Spartan Familie der Firma XILINX bewerkstelligt. Die Betriebsarteneinstellungen des FPGAs erfolgen über einen externen Schalter. Um das zeitliche Verhalten der Eingangssignale am FPGA nachzubilden, wurden die zeitlichen Verhalten der ADC- und PPG-Teilschaltung kombiniert (siehe Abbildung 8).

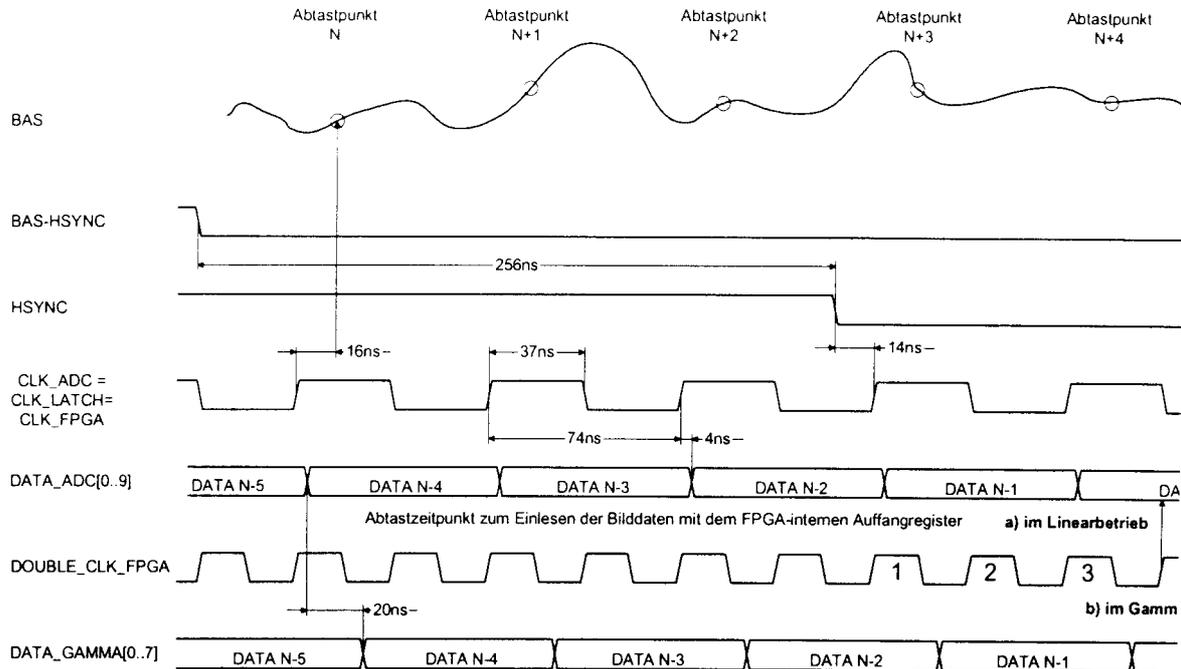


Abbildung 8: Zeitliche Darstellung der Eingangssignale am FPGA

ISP XC18V512 (FPGA_INIT)

Der ISP XC18V512 wurde für den Einsatz in einer Boundary-Scan-Kette mit anderen JTAG-kompatiblen Geräten (hier XCS40) entwickelt. Der ISP (in system programmable configuration PROM) wird aufgrund der flexiblen Re-Programmierbarkeit speziell im frühen Entwicklungsstadium von Prototypen eingesetzt. Ist das Produkt serienreif, so kann der XC18V512 jederzeit durch den billigeren OTP-Baustein (one time programmable PROM) der Serie XC1700 ersetzt werden. Die Bausteine der Serien XC18V00 und XC1700 sind im SOIC-20-, VQFP-44- und PLCC-44-Gehäuse pinkompatibel. Der Programmierpin V_{PP} der XC1700-Serie korrespondiert mit dem Betriebsspannungspin V_{CC} der XC18V00-Serie. Im Normalbetrieb liegt am Programmierpin V_{PP} der XC1700-Serie ebenfalls die Betriebsspannung V_{CC} an.

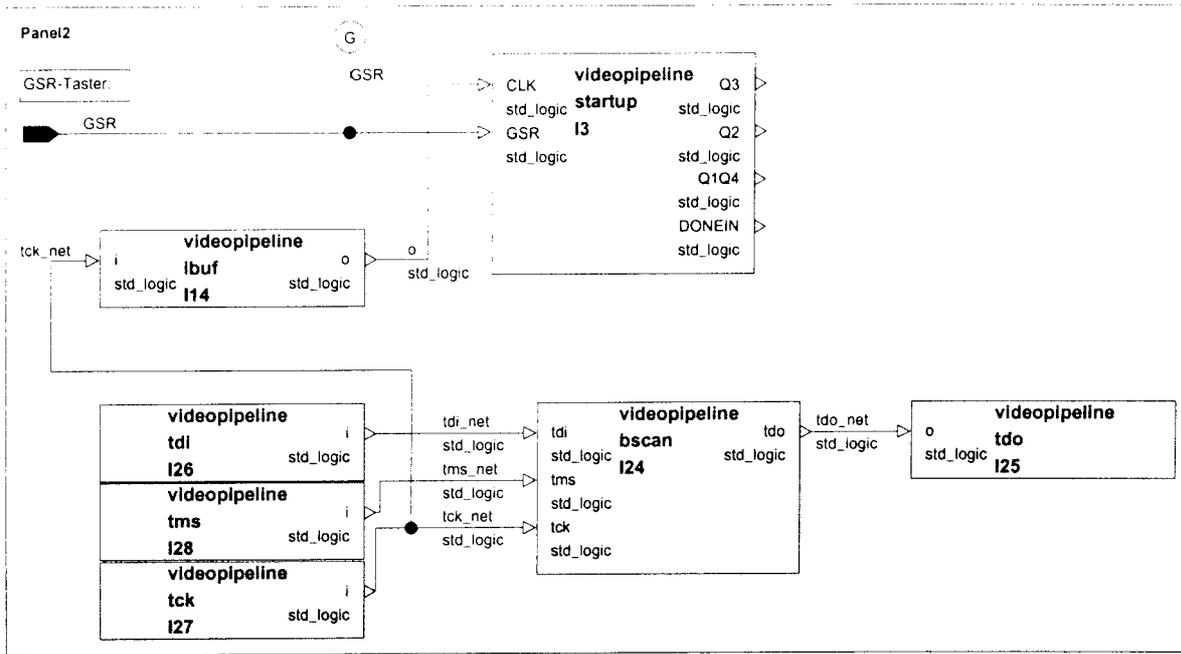


Abbildung 9: Instantiierung des Boundary-Scan- und Startup-Blocks

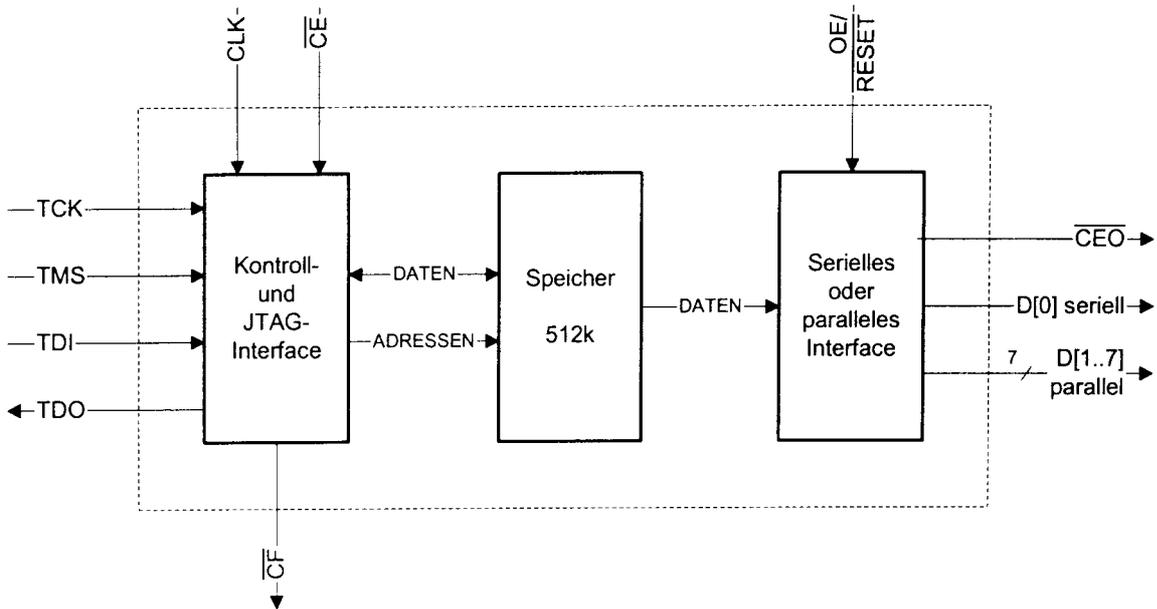


Abbildung 10: Blockschaltbild des ISP XC18V512

Halbbildspeicher CY7C1041-12 (SRAM)

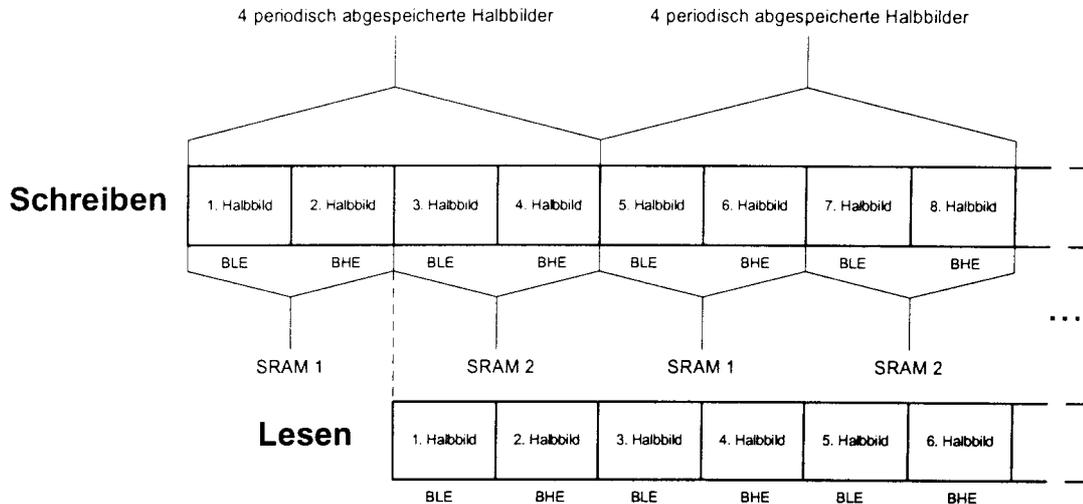


Abbildung 11: SRAM-Speicherprinzip

Bei den zur Bildmanipulation verwendeten Speicherbausteinen handelt es sich um asynchrone SRAMs mit einer Speicherkapazität von 256kBit x 16. Die Adressierung der Speicherzellen erfolgt über den 18-Bit-breiten Adressbus A_{0-17} , die Ansteuerung mit den Signalen \overline{BHE} , \overline{BLE} , \overline{CE} , \overline{OE} , \overline{WE} und der Datentransfer über den bidirektionalen Datenbus I/O_{0-15} . Die bidirektionalen Schreib-/Leseports I/O_{0-15} können in ein nieder- und höherwertiges Byte I/O_{0-7} und I/O_{8-15} aufgeteilt werden. Diese Tatsache erlaubt es, jeweils 2 Halbbilder in einem SRAM zwischenzuspeichern.

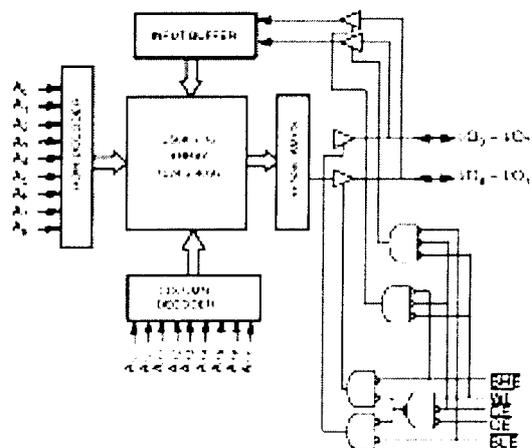


Abbildung 12: Blockschaubild des asynchronen SRAMs CY7C1041B-12

Digital/Analog-Wandlung mit Ausgangstreiberstufe (DAC)

Nachdem die Bilddaten den FPGA manipuliert oder auch unbehandelt durchlaufen haben, werden sie nun mit dem Video-DAC ADV7120KP50 wieder in ein analoges BAS-Signal zurücktransformiert, über einen Cauer-Tiefpass gefiltert und letztendlich mit einer Videoausgangstreiberstufe kapazitiv entkoppelt und für nachfolgende Videosysteme zur Verfügung gestellt.

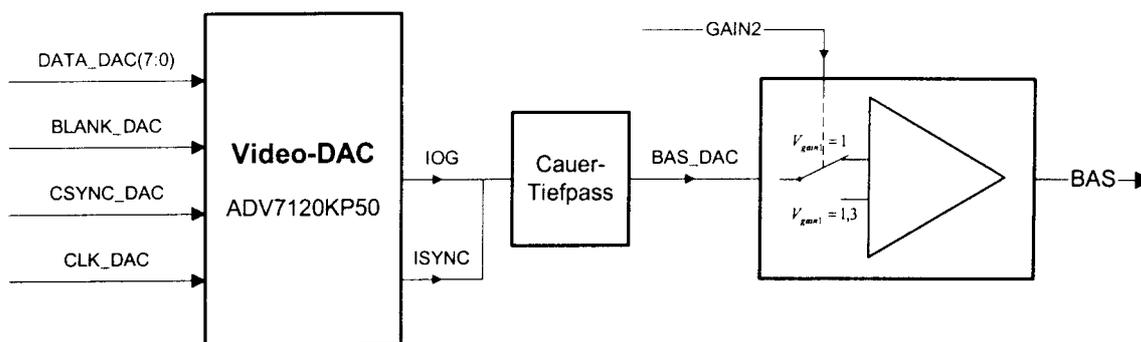


Abbildung 13: Blockschaltbild D/A-Wandlung mit Ausgangstreiberstufe

Der ADV7120KP50 ist ein Video-Digital/Analog-Wandler, der speziell für hochauflösende Farbgrafiken und Videosysteme entwickelt wurde. Der DAC hat drei Hochgeschwindigkeits-DACs für die drei Farben Rot, Grün, Blau (RGB). Jeder DAC ist mit einem 8-bit Eingang verbunden. Für die externe Videokontrollsignale sind die Eingänge für den Pixeltakt CLK_DAC=DOUBLE_CLK_FPGA ($f_{CLK_DAC} = 27\text{MHz}$), das Austastsignal BLANK_DAC und das Synchronisierungssignal CSYNC_DAC vorgesehen. Sämtliche Steuersignale und Daten werden vom FPGA zur Verfügung gestellt. Der REF-WHITE-Eingang wird auf Masse (GND_FPGA_SRAM) gelegt, da für die Videopipeline kein Weißabgleich vorgesehen ist. Die ungenutzten 8-bit Dateneingänge für Rot und Blau werden auf die Masse gelegt.

Digitale Zwischenfrequenz-Empfänger für die Satelliten-Navigation mit GPS, GLONASS und GALILEO

Zusammenfassung:

Zunächst wird eine kurze Gegenüberstellung des neuen, digitalen ZF-Stufen-Konzeptes zu den bekannten Bandpaß- und Tiefpaßsystemen zur digitalen Verarbeitung und Abwärtsmischung von Hochfrequenz-Empfangssignalen in eine ZF-Lage (Digitaler ZF-Empfänger) gegeben. Dann erfolgt für das neue Konzept die Beschreibung der mehrkanaligen, digitalen Filterung der ZF-Signale mit speziellen Tiefpaßfiltern, die vor allem für "Spread-Spectrum-Signale" geeignet sind und sich mit nur sehr geringem technischen Aufwand realisieren lassen. Hierzu werden außerdem einfache, technische Ausführungsbeispiele für die erste und zweite Quadratur-Mischstufe dieser digitalen Abwärtsmischer angegeben. Anschließend erfolgt noch eine genaue Beschreibung der digitalen Wortbegrenzung des ZF-Stufen-Ausgangssignals mit einem sogenannten 2 bit "Gray-Code-Formatierer", der besonders für die Anwendungen bei kombinierten Navigationsempfängern, wie z.B. für GALILEO-GPS oder GPS-GLONASS, geeignet ist.

Inhaltsverzeichnis	Seite
1. Einleitung	2
2. Der neue, digitale Abwärtsmischer im Vergleich zu den bekannten ZF-Stufen-Konzepten	3
3. Die technische Ausführung des neuen, mehrkanaligen, digitalen Abwärtsmischers	6
3.1 Das digitale Tiefpaß-Filter des Abwärtsmischers	7
3.2 Die erste, frequenzprogrammierbare Mischstufe	10
3.3 Die zweite Festfrequenz-Mischstufe	13
4. Wortbreiten-Reduktion des ZF-Stufen-Ausgangssignals mit dem Gray-Code-Formatierer	15
Literaturverzeichnis	17
Anhang : Bilder	18

1. Einleitung

Die modernen Kommunikationsdienste, wie z. B. UMTS, GSM, DCS1800, PDC, DECT usw., sowie die zur Zeit im Aufbau befindlichen "Mobile Satellite Services" (MSS) mit den Satelliten-Systemen: GLOBALSTAR, IRIDIUM und INMARSAT verwenden alle digitale Modulationsverfahren die auf "Phase Shift Keying" (PSK) und/oder "Frequency Shift Keying" (FSK) beruhen. Hierbei werden die Nachrichten (Sprache, Bilder und Daten) vorwiegend im "Time Division Multiple Access" (TDMA), im "Code Division Multiple Access" (CDMA) und im "Frequency Division Multiple Access" (FDMA) auf dem Funkkanal übertragen. Ebenfalls verwendet die Satelliten-Navigation mit den Satelliten-Systemen: GPS (Global Positioning System, U.S.A) und GLONASS (Global Navigation Satellite System, Rußland) eine CDMA-Codierung und bei GLONASS zusätzlich noch eine FDMA-Codierung für die Nachrichten-Übertragung. Dies gilt gleichermaßen auch für das geplante Satelliten-Navigationssystem GALILEO der Europäischen Gemeinschaft, das im oberen/unteren L-Band mit Ausnahme des C-Bandes, Fig. 1, mehrere bereits benutzte Frequenzbereiche überlagern wird. Alle diese über den Funkweg übertragenen Signale besitzen somit spezielle, breitbandige Signal-Spektren und werden deshalb allgemein als sogenannte "Spread Spectrum Signale" (SSS) bezeichnet.

Dies erfordert weitgehend digitale Hochfrequenz-Empfänger, Fig. 2, die diese Signale einer entsprechend breitbandigen Signal-Filterung im RF-FRONTEND unterziehen, um sie dann dem Basisband-Prozessor, der meist mit einem speziellen, Mikroprozessor (μ P) realisiert wird, über die Zwischenfrequenz-Stufen (ZF-Stufen) zur relativ breitbandigen, digitalen Signalweiter- und Endverarbeitung (Demodulation, Korrelation, Daten/Bit-Synchronisation, Daten-Auswertung usw.) zu zuführen. Zusätzlich zu diesen Anforderungen, die an diese Hochfrequenz-Empfänger zu stellen sind, wird ein ausreichend guter, linearer Phasengang der Signale durch die Mischstufen und Filter des RF-FRONTEND's und der weiteren ZF-Stufen benötigt.

Insbesondere bei mehrkanaligen Satelliten-Navigations-Empfängern, z.B. für GLONASS, GALILEO und in Kombination mit GPS [1, 2], ist diese Linearphasigkeit der Signalübertragung (konstante Gruppenlaufzeit) der Filter und Mischstufen unabdingbar, da dies andernfalls zu erheblichen Fehlern in den Navigationsergebnissen führen kann. Außerdem sind bei den GNSS-Empfängern harte Anforderungen an die Interferenz-Störfestigkeit bezüglich der benachbarten Frequenzkanäle der Satellitenfunkdienste MSS, SATCOM, GLOBALSTAR usw. zu stellen (bereits schon jetzt eine teilweise Überbelegung des L-Bandes vorhanden).

Nur ein weitgehend digitaler Hochfrequenz-Empfänger kann diese Kriterien der modernen (Satelliten)-Kommunikation und (Satelliten)-Navigation erfüllen. Jedoch benötigt jeder Hochfrequenz-Empfänger, Fig. 2, einschließlich der Antenne eine analoge Empfängerbaugruppe, das RF-FRONTEND, das die extrem schwachen Empfangssignale von der Antenne ausreichend verstärkt, vorfiltert und für die nachfolgende digitale Signalverarbeitung entsprechend aufbereitet (Abwärtsmischung, Signal-Abtastung, A/D-Umsetzung, usw.). Eine technisch besonders günstige Lösung (Minimalaufwand) für ein solches RF-FRONTEND ergibt sich nach Fig. 2, wenn die Empfangssignale bereits in einer sehr hohen ZF-Lage digitalisiert werden und zur Weiterverarbeitung in einen nachfolgenden voll digitalen ZF-Empfänger eingespeist werden. Da der Basisband-Prozessor des ZF-Empfängers in der Regel nicht direkt diese Breitbandsignale in der hohen ZF-Lage in Echtzeit [2] (Bandbreite, Taktrate, Wortbreite usw.) verarbeiten kann, muß noch eine entsprechende digitale Abwärtsmischung, digitale Filterung und Abstraten-Reduktion (Dezimation) mittels einer mehrkanaligen, digitalen ZF-Stufe (Digitaler Abwärtsmischer) erfolgen. Nachfolgend wird für einen derartigen, digitalen ZF-Empfänger von "Spread Spectrum Signalen" eine technisch besonders vorteilhafte Lösung eines neuen mehrkanaligen, digitalen Abwärtsmischers mit speziellen digitalen Tiefpaßfiltern vorgeschlagen, der (GLONASS-ASIC) bereits sehr erfolgreich im kombinierten GPS-GLONASS-Empfänger ASN-22 eingesetzt wird.

2. Der neue, digitale Abwärtsmischer im Vergleich zu den bekannten ZF-Stufen-Konzepten

Alle Konzepte moderner digitaler Hochfrequenz-Empfänger beruhen auf der Verarbeitung von komplexen bzw. analytischen Signalen in sogenannten "Quadratur-Systemen" [3, 4]. Aus dem reellen Eingangssignal $s(t)$ wird mit Hilfe eines 90° -Phasenschieber-Netzwerkes und/oder mit einer komplexen Signalmischung (Multiplikation) ein komplexes Signal $z(t)$ erzeugt:

$$z(t) = A(t) \cdot e^{j \cdot (\phi(t) + \phi_0)} \quad \text{mit} \quad \phi = \text{atan} \left(\frac{s_I}{s_R} \right)$$

das aus zwei realen, zueinander orthogonalen Signalkomponenten $s_R(t)$ und $s_I(t)$ besteht. Damit läßt sich $z(t)$ in der Real-/Imaginärteil-Schreibweise darstellen, wie folgt:

$$z(t) = A(t) \cdot (s_R(t) + j \cdot s_I(t)) \quad \text{mit} \quad A(t) = \sqrt{\text{Re}(z(t))^2 + \text{Im}(z(t))^2}$$

Zur Verarbeitung des aus $s(t)$ gewonnen komplexen Signals $z(t)$ wird in den Quadratur-Systemen der Realteil von $z(t)$ im sogenannten Inphasezweig (I-Kanal) und der Imaginärteil von $z(t)$ im sogenannten Quadraturzweig (Q-Kanal) herangezogen, wobei allgemein die Eulerschen Relationen für komplexe Zahlen gelten:

$$z = e^{j\alpha} = \cos(\alpha) + j \cdot \sin(\alpha) = \cos(\alpha) - j \cdot \sin(-\alpha)$$

oder für die konjugiert komplexe Darstellung:

$$\bar{z} = e^{-j\alpha} = \cos(\alpha) - j \cdot \sin(\alpha) = \cos(\alpha) + j \cdot \sin(-\alpha)$$

Da die Eingangssignale $s(t)$ der digitalen Empfänger, Fig. 2, grundsätzlich auf eine Bandbreite B zu begrenzen (Anti-Aliasing) sind und sie sich hierbei in einer hohen Frequenz-Lage (ZF-Lage) befinden, kann nur eine Bandpaß-abtastung (Unterabtastung) der Signale $s(t)$ am Eingang des Digitalteils bzw. am Signalausgang des vorgeschalteten, analogen RF-FRONTEND's erfolgen. Hierbei sind folgende Einschränkungen bei der reellen Bandpaßabtastung solcher Signale bezüglich der Wahl der Abtastfrequenz f_a , der Bandbreite B und der Bandmittenfrequenz f_m einzuhalten:

1. Das Abtasttheorem darf nicht verletzt werden und es muß deshalb gelten:

$$f_a > 2 \cdot B$$

2. Um Überfaltungen (Aliasing) der Abtastspektren in Teilbereichen zu vermeiden, müssen die Teilspektren bei der periodischen Wiederholung in die Mitte der Abtast-Halbbänder fallen. Das Basisband-Teilspektrum ist dann bei $f_a/4$ zentriert. Damit ist dann eine weitere Bedingung zu erfüllen:

$$f_a = \frac{4 f_m}{(2i + 1)} \quad \text{mit} \quad i = 0, 1, 2, 3, \dots, \infty$$

wobei für den in der Regel nicht ganzzahligen Unterabtastungsfaktor gilt:

$$R_F = \frac{2 f_m + B}{f_a}$$

Unter dieser Voraussetzung für die Bandpaß-Abtastung von reellen Bandpaß-Signalen können nun die Quadratur-Systeme der digitalen Empfänger prinzipiell in zwei Gruppen: Bandpaß-Systeme und Tiefpaß-Systeme [4] eingeteilt werden. Beide Systeme, die sich im wesentlichen nur im technischen Realisierungsaufwand unterscheiden, dienen zur komplexen Abwärtsmischung, Filterung und Abtastraten-Reduktion (Dezimation) der reellen Eingangssignale $s(t)$ und geben üblicherweise die Ausgangssignale $s_B(t)$ nach der Abwärtsmischung in eine Basislage (Pseudo-Tiefpaßbereich) in reeller Form aus. (Eine komplexe Signalweiterverarbeitung im Basisbandbereich ist natürlich auch möglich.)

Zum Vergleich der weiter unten beschriebenen, neuen Lösung einer digitalen ZF-Stufe (Digitaler Abwärtsmischer), die ebenfalls eine Quadratur-System-Variante darstellt, werden kurz die beiden herkömmlichen Quadratur-System-Konzepte aufgezeigt.

Nach Fig. 3 besteht das Quadratursystem der herkömmlichen, komplexen Bandpaß-Lösung im wesentlichen aus zwei eingangsseitigen 90° -Phasenschiebernetzwerke ($\pi/2$) für das abgetastete Eingangssignal $s_a(n)$ und der Mischfrequenz f_c der 1. Mischstufe, aus der komplexen 1. Mischstufe, aus einem komplexen Bandpaß-Filter bzw. aus einem komplexen Frequenzabtafilter [3] und aus einer 2. Mischstufe mit einem I-Q-Addierer zur Reellmachung des komplexen Ausgangssignals $z(n)$. Hierbei erfolgt die erforderliche Abtastraten-Reduktion des mit f_a abgetasteten Summensignals $s_s(n)$ auf die Abtastrate f_b mit dem ganzzahligen Abtastraten-Reduktionsfaktor:

$$R = \frac{f_a}{f_b} \quad \text{mit} \quad f_b < f_a$$

erst am Ausgang des Systems mit einem zweiten Signal-Abtaster der Abtast-Taktfolge $m \cdot T_b$. Damit ist die gesamte Signalverarbeitung im Quadratursystem bei einer hohen Taktrate (Taktwiederholfrequenz) $f_0 \geq f_a$ auszuführen. Dies stellt an die einzelnen Signal-Verarbeitungsbausteine des Systems erhebliche Anforderungen bezüglich Verarbeitungsbandbreiten, Taktraten und Signallaufzeiten.

Die zweistufige, digitale Abwärtsmischung und Signalfilterung erfolgt gemäß nach dem Frequenzplan von Fig. 4. Das reelle, abgetastete Eingangssignal $s_a(n)$ wird durch die 1. Signalmischung (Hilbert-Transformation, [4]) in ein analytisches Bandpaß-Signal $x(n)$ mit den Komponenten $x_R(n)$ (Realteil) und $x_I(n)$ (Imaginärteil) umgesetzt. Anschließend wird das gewünschte Nutzspectrum $Y(f)$ mit dem komplexen Bandpaß-Filter aus dem Abtastgesamt-Spectrum $S_a(f)$ bei der Bandmittenfrequenz f_m herausgefiltert und liefert somit das komplexe Filter-Ausgangssignal $y(n)$ mit den Komponenten $y_R(n)$ (Realteil) und $y_I(n)$ (Imaginärteil). Zur Reellmachung des komplexen Filtersignals $y(n)$ muß das Signalspectrum $Y(f)$ noch in die Basisband-Lage (Pseudo-Tiefpaßlage) mittels einer weiteren komplexen Mischung verschoben werden und dann der Realteil $z_R(n)$ zu dem Imaginärteil $z_I(n)$ des komplexen Mischproduktes $z(n)$ mit dem I-Q-Addierer aufsummiert werden.

Bei dieser Bandpaß-Lösung müssen sowohl die benötigten, komplexen Filterkoeffizienten des aufwendigen Bandpaß-Filters zur Frequenzband-Selektion als auch einer der beiden Mischerfrequenzen f_c bzw. f_T , die mit den "Digital Controlled Oscillator's" (DCO1 bzw. DCO2) erzeugt werden, von extern einstellbar sein. Dies erfordert in der technischen Realisierung einen sehr hohen Schaltungsaufwand sowie eine schwierige Dimensionierung der komplexen Bandpaß-Filter und Mischerstufen, so daß diese Lösung in der praktischen Anwendung nur selten aufgegriffen wird. Für mehrkanalige, digitale Empfänger, die in kleine Geräte, z. B. "Handy's", zu implementieren sind, ist diese Version einer digitalen ZF-Stufe nicht brauchbar. So ist auch heute eine solche Bandpaß-Version, die für die standardmäßigen 6 bis 12-kanaligen ZF-Stufen von GNSS-Empfängern benötigt werden und mit wenigen "ASIC's" und /oder "FPGA's" auskommen müssen, auf der Basis dieses Konzeptes nicht kostengünstig durchführbar.

Im Gegensatz dazu wird die komplexe Tiefpaß-Lösung von Quadratur-Systemen nach Fig. 5 allgemein verwandt. Sie ist in der technischen Realisierung wesentlich aufwandsgünstiger auszuführen. So reduziert sich hier unter anderem das 90° -Phasenschiebernetzwerk der Bandpaß-Version auf eine 90° -Phasenverschiebung ($\pi/2$) der Mischerfrequenz f_c für den 1. Q-Mischer bzw. der Mischerfrequenz f_T für den 2. Q-Mischer. Die wesentliche Vereinfachung erbringt hier vor allem das komplexe Tiefpaß-Filter mit festen Filterkoeffizienten an Stelle des komplexen Bandpaß-Filters der Bandpaß-Lösung mit variablen Filterkoeffizienten. Auch ist bei der Tiefpaß-Version nur der DCO1 zur Wahl der erforderlichen Mischerfrequenz $f_c =$ Mittenfrequenz f_m des selektierten Signal-Spektrums, einstellbar zu machen.

Damit erfolgt nach Frequenzplan der komplexen Tiefpaß-Lösung, Fig. 6, die Abwärtsmischung des ausgewählten Nutzspektrums $Y(f)$ in die komplexe Tiefpaß-Lage symmetrisch zur Frequenz Null. In dieser komplexen Tiefpaß-Lage wird dann das Spektrum $Y(f)$ mittels des Tiefpaß-Filters aus dem komplexen und tiefpaßversetzten Abtastspektrum $S_a(f)$ herausgefiltert. Die Abstraten-Reduktion des komplexen Tiefpaß-Ausgangssignals $y(n)$ auf die Abtastfrequenz f_b kann auch hier bereits nach dieser Tiefpaß-Filterung erfolgen. D.h. der nachfolgende Signalverarbeitungsteil mit der 2. Mischstufe und I-Q-Addierer kann mit entsprechend verringerter Taktrate betrieben werden. Eine besonders einfache Reellmachung des komplexen Tiefpaßsignals $y(n)$ ergibt sich, wenn vorher eine Abstraten-Reduktion auf $f_b = 2 \cdot \text{Tiefpaßbandbreite } B_T$ erfolgt ist. Damit kann dann die 2. Mischerstufe mit der festen Mischfrequenz $f_T = f_b/4$ betrieben werden. D.h. die 2. Mischung ist mit einer Koeffizientenfolge auszuführen, die nur die 3 Werte: 1, 0, -1 annimmt. Dies ergibt für die 2. Mischstufe, Fig. 5, eine sehr einfache Schaltungsausführung, wie noch später aufgezeigt wird.

Der nachfolgend beschriebene, neue, *digitale Abwärtsmischer* (ZF-Stufe) stellt ebenfalls eine sehr vorteilhafte Variante dieser Tiefpaß-Lösung dar und ist besonders für die Anwendung bei vielkanaligen Hochfrequenz-Empfängern von "Spread-Spectrum-Signale", wie z.B. bei kombinierten Satelliten-Navigationsempfängern, geeignet. Nach Fig. 7 ist diese ZF-Stufe vor allem durch ein *neuartiges I-Q-Kanal-Tiefpaß-Filter* gekennzeichnet. Es stellt ein Sonderfall eines reellen Frequenzabast-Filters [3] mit einer $\sin^2 x/x^2$ - Übertragungscharakteristik (Spalttiefpaß-Filter 2. Ordnung) im Frequenzbereich dar und kann dementsprechend besonders gut an das "Spread-Spectrum" der Satelliten-Navigationssignale und der MSS-Signale angepaßt werden (Optimal-Filter).

Da digitale Tiefpaß-Filter [5] für die oben erwähnten Anwendungen benötigt werden, die eine exakt linearphasige Übertragungsfunktion erfordern, kommen üblicherweise nur FIR-Filterstrukturen von einem Filtergrad $N > 40$ in Frage, die in der Regel ebenfalls N Filterkoeffizienten C_n aufweisen. Dies erfordert in der technischen Filterrealisierung einen nicht vertretbaren Aufwand von mindestens N "Hardware-Multiplizierern" pro Empfangskanal. Eine starke Verringerung des Hardware-Aufwands wird nur durch die Verwendung einer *Frequenzabast-Filterversion* erreicht, die ebenfalls linearphasiges Übertragungsverhalten aufweist. Im vorliegenden Fall des digitalen ZF-Stufen-Tiefpaßfilters sind sogar keine (echten) Koeffizienten-Multiplizierer mehr notwendig. (Die hier erforderlichen Multiplikationen mit dem Faktor 2 können durch Additionen oder Linksverschiebung entsprechender Bitstellen im Datenformat geschehen. Dies wird später noch genauer erläutert.)

Hierbei gliedert sich das Filter nach Fig. 7 und 8 in zwei Funktionsblöcke, dem Integrator 2. Ordnung (Akkumulator) und dem nachgeschalteten Differentiator 2. Ordnung (Kammfilter). Genau zwischen diesen beiden Filterblöcken, also in der Mitte der Filterstruktur, wird die Reduzierung (Dezimation) der Abtastfrequenz von f_a auf f_b um den ganzzahligen Faktor R vorgenommen.

Dies stellt einen weiteren technischen Vorteil dieses digitalen Abwärtsmischers gegenüber den herkömmlichen Verfahren dar. Damit ist die Signalverarbeitung pro Quadraturkanal nur zur Hälfte (1. I-Q-Mischung und 1. Filterhälfte) bei der hohen Abtastfrequenz f_a auszuführen. Dagegen kann die zweite Hälfte des Abwärtsmischers mit der erheblich reduzierten Abtastfrequenz f_b arbeiten. Dies stellt an einen Großteil der Verarbeitungsbausteine des Systems wesentlich geringere Anforderungen bezüglich Verarbeitungsbandbreiten, Taktraten und Signallaufzeiten. Demzufolge ist dieses Abwärtsmischer-Verfahren nach Fig. 8 besonders für die technische Realisierung von digitalen, mehrkanaligen ZF-Stufen mit preiswerten "ASIC's" und/oder "FPGA's" geeignet. Außerdem ergibt sich für den Empfänger eine einfache digitale Abstimmbarkeit der Empfangsfrequenzen bzw. der Empfangskanäle, indem der DCO für die 1. Mischfrequenz f_m digital einstellbar (Programmierung der Frequenzschritt-Weite des Phasen-Akkumulators) ausgeführt wird, das z.B. bei kombinierten Satelliten-Navigationsempfängern in der Regel notwendig ist. Ebenfalls kann die 2. Mischstufe zur Reellmachung [5] des komplexen Tiefpaß-Ausgangssignal $v(m)$, die mit der festen Mischfrequenz $f_T = f_b/4$ betrieben wird, sehr einfach realisiert werden, wie dies noch an einem späteren Beispiel aufgezeigt wird. So werden in den nachfolgenden Abschnitten die wesentlichen Komponenten des Abwärtsmischer-Systems, wie z.B. das Tiefpaßfilter und die Mischstufen, noch genauer beschrieben bzw. dazu Realisierungsvorschläge angegeben.

3. Die technische Ausführung des neuen, mehrkanaligen, digitalen Abwärtsmischers

Der digitale Abwärtsmischer besteht nach Fig. 8 aus k gleichen, unabhängigen ZF-Quadratur-Kanälen, die alle mit dem gleichen, digitalen Eingangssignal $s_a(t)$ bzw. $s_a(n)$ (Bandpaßsignal von der Wortbreite a bit mit der Abtastfrequenz f_a abgetastet) vom A/D-Umsetzer des RF-FRONTEND's versorgt werden. Jedoch liefert jeder ZF-Kanal sein digitales Ausgangssignal $s_{1b}(t)$, $s_{2b}(t)$, ..., $s_{kb}(t)$ (Wortbreite f bit) in der reellen Tiefpaßlage bei der reduzierten Abtastfrequenz $f_b = f_a/R$ getrennt und parallel an den reellen Basisband-Prozessor zur Signalweiterverarbeitung. Hierbei kann der Basisband-Prozessor mit einem standardmäßigen Signalprozessor oder Mikroprozessor (μP) realisiert werden.

Somit weist jeder der einzelnen, neuen ZF-Quadratur-Kanäle des digitalen Abwärtsmischer eine Struktur auf, die in Fig. 9 aufgezeigt wird. (Dies entspricht Fig. 7, jedoch mit mehr schaltungstechnischen Einzelheiten). Mit dem 1. I-Q-Mischer, der seine digitale Mischfrequenz f_m (Wortbreite b bit) vom einstellbaren DCO1 (Phasen-Akkumulator [6]) zur Frequenz-Selektion des Empfangskanals bezieht, wird das reelle Eingangssignal $s_a(n)$ (Wortbreite a bit) in den komplexen Tiefpaßbereich symmetrisch zur Frequenz-Nullage, Fig. 6, abgemischt. Anschließend führt das digitale I-Q-Tiefpaßfilter die erforderliche Bandbegrenzung des abgemischten Signals (Wortbreite c bit) sowie die Abstraten-Reduktion um den Faktor R aus. Um den "Prozeß-Gewinn" G des Filters zu berücksichtigen, muß das Filter für eine erweiterte Verarbeitungswortbreite von d bit ausgelegt werden. (Die Dimensionierungsvorschrift dazu folgt später noch.)

Zur Reellmachung des komplexen Filterausgangssignal (Wortbreite d bit bzw. e bit) ist dann der 2. I-Q-Mischer mit der festen Mischfrequenz $f_b/4$ (Wortbreite 2 bit) - geliefert von einem besonders einfachen DCO2 (Johnson-Zähler) - und ein nachfolgender Addierer zur Summation der entsprechenden Real- und Imaginär-Teile des Quadratur-Signals notwendig. Eine Begrenzung der Wortbreite von f bit auf g bit des reellen, digitalen ZF-Kanal-Ausgangssignals $s_b(m)$, die für die digitale Signalweiterverarbeitung mit dem Basisband-Prozessor erforderlich ist, kann mit dem sogenannten "Formatierer" am Signalausgang vorgenommen werden. Weitere Einzelheiten folgen anschließend.

3.1 Das digitale Tiefpaß-Filter des Abwärtsmischers

Das spezielle Tiefpaßfilter des digitalen Abwärtsmischer stellt ein entsprechend modifiziertes Frequenzabtast-Tiefpaßfilter [6] von 2. Ordnung mit einer $\sin^2 x/x^2$ -Übertragungscharakteristik im Frequenzbereich (Spalttiefpaß 2. Ordnung) dar und kann dementsprechend an das "Spread Spectrum" [2] der modernen Satelliten-Kommunikations-Signale (MSS, GLOBALSTAR usw.) und der Satelliten-Navigations-Signale (GPS, GALILEO usw.) optimal angepaßt werden (Optimal-Filter). Wie schon früher erwähnt, besteht das Filter nach Fig. 10 aus der Hintereinanderschaltung eines Integrators 2. Ordnung (Akkumulator) mit einem Differentiator 2. Ordnung (Kammfilter). An dem gemeinsamen Verbindungsstelle der beiden Elementarfilter wird die Abstraten-Reduktion der Integrator-Ausgangsdaten y_n um den Faktor R mit Hilfe des sogenannten "Dezimators" vorgenommen. Diese Daten u_m , die um den Faktor R in der Taktrate verringert sind, werden dann in den nachfolgenden Differentiator eingespeist. Somit kann die Abstratenreduktion technisch sehr einfach mit einem Abtast-Register an den Integrator-Ausgangsdaten y_n vorgenommen werden. Hierbei bedeutet die Größe T die Abtastperiodendauer der Abtastfrequenz f_a bzw. die Verzögerung des Datensignals durch die Datenspeicher um die Zeitdauer T , die ebenfalls mit entsprechenden Registern ausführbar ist. Dementsprechend gilt für die Größe $R \cdot T$ bzw. $4 \cdot R \cdot T$ im Differentiator teil eine Signalverzögerung von der Dauer $R \cdot T$ bzw. $4 \cdot R \cdot T$ und bezieht sich hiermit auf die reduzierte Abtastfrequenz $f_b = 1/R \cdot T$. Damit kann das gesamte Tiefpaßfilter nur mit einfachen Logikbausteinen, wie unter anderem mit Daten-Registern, Addierern und Subtrahierern, ausgeführt werden. Es sind keine Koeffizienten-Multiplizierer notwendig! So können auch die erforderlichen Multiplikationen mit dem Faktor 2 durch weitere Additionen oder Linksverschiebung entsprechender Bitstellen im Datenwort realisiert werden.

Anhand der in Fig. 9 und 10 aufgezeigten, funktionalen Filterstruktur können nun die einzelnen z-Übertragungsfunktionen für die beiden Elementarfilter: Integrator und Differentiator sowie für das gesamte Tiefpaßfilter unter Berücksichtigung des Abstraten-Reduktionsfaktors R aufgestellt werden. Daraus ergibt sich für das Integrator-Filter, das eine rein rekursive Filterstruktur (Polstellen-Filter) besitzt, eine Übertragungsfunktion:

$$H_I(z) = \frac{1}{4R^2} \cdot \frac{z}{(z-1)^2} \quad \dots\dots \text{Akkumulator 2. Ordnung}$$

Jedoch weist dieser Integrator bei seiner alleinigen Verwendung, infolge seiner rein rekursiven Filterstruktur, einen schwerwiegenden Nachteil auf. Er ist so nicht stabil, d.h. bei Eingangssignalsprüngen von x_n kann das Ausgangssignal y_n des Integrators exponentiell über alle Grenzen wachsen. *Nur in Verbindung mit dem nachgeschalteten Differentiator*, der eine rein nichtrekursive Filterstruktur (Nullstellen-Filter) besitzt und *unter Verwendung einer speziellen 2-er-Komplement-Zahlendarstellung* für die gesamte Filter-Arithmetik ist diese *Filter-Instabilität vollständig vermeidbar*. Damit gilt für den Differentiator folgende Übertragungsfunktion:

$$H_D(z) = \frac{1}{4} \cdot \frac{(z^{4R} - 1)^2}{z^{8R}} \quad \dots\dots \text{Kammfilter 2. Ordnung}$$

Auf diese Weise wird der Pol bei $1/4 \cdot R^2$ des Integrators, der im wesentlichen für Instabilität des Integrators allein verantwortlich ist, durch eine der 4 Nullstellen des Differentiators auf dem Einheitskreis kompensiert. Folglich ergibt sich dann die Übertragungsfunktion $H(z)$ für das gesamte Tiefpaßfilter zu:

$$H(z) = H_I(z) \cdot H_D(z)$$

$$H(z) = \frac{1}{(4R)^2} \cdot \frac{1}{z^{8R-1}} \cdot \frac{(z^{4R} - 1)^2}{(z-1)^2} \quad \dots\dots \text{Tiefpaßfilter des Abwärtsmischers}$$

Zur Dimensionierung der Verarbeitungswortbreite (Signal-Dynamik) der Filter-Arithmetik ist der sogenannte "Prozeßgewinn" G der Tiefpaßfilterung von entscheidender Bedeutung, der sich wie folgt darstellt:

$$G = 16R^2$$

Oder unter Einbezug des Filtergrades N eines äquivalenten FIR-Elementarfilters ergibt sich für G folgende Beziehung:

$$G = N^2 \quad \text{mit} \quad N = 4R \quad \dots\dots \text{Grad des äquivalenten FIR-Elementarfilters}$$

Dazu müssen für eine stabile Tiefpaßfilterung, Fig. 9, alle internen, arithmetischen Operationen im *2-er-Komplement-Code ohne Werteüberlauf (Modulo-2-er-Komplement-Code, Stibitz-Code)* mindestens eine ganzzahlige Wortbreite d aufweisen von:

$$d \geq (1 + c + \text{ld}(G)) \quad \dots\dots \text{Filter-Wortbreite in bit}$$

Nur so kompensiert der Differentiator, Fig. 10, exakt die nicht vermeidbaren Grenzwert-Zyklusschwingungen des Integrators. Die Filterantwort v_m ist immer richtig. In diesem Zusammenhang ist es zweckmäßig für die Zahlen im 2-er-Komplement-Code den Zahlenbereich von -1 bis +1 zu wählen, d.h. mit *Fractional-Arithmetik* [6] zu rechnen. Hierbei erfolgt eine eventuell notwendige Erweiterung des Zahlenbereichs an den Addier- und Subtrahiereingängen durch eine entsprechende Rechtsverschiebung der Binärstellen. Die binäre Zahlendarstellung wird damit LSB-bündig. Frei werdende MSB-Stellen werden dann mit dem gleichen Vorzeichenbit belegt. Die Weiterverarbeitung (2. Mischung, Addition von Realteil- und Imaginärteil usw.) des Filterausgangssignals innerhalb des digitalen ZF-Kanals, Fig. 9, kann dann in der Regel mit einer reduzierten Wortbreite auf e bit bzw. f bit erfolgen. In diesem Fall ist die entsprechende Rundung bzw. Abschneiden von Bitstellen MSB-bündig auszuführen. Zur Dimensionierung dieser Wortbegrenzung ist natürlich der noch zulässige Quantisierungsfehler für die Signalweiterverarbeitung zu berücksichtigen. In Abschnitt 4 wird dazu ein Schaltungsbeispiel eines "Gray-Code-Formatierers" angegeben.

Um den Amplitudengang sowie den exakt linearen Phasengang dieses Filters zu belegen, der zur Filterung von "Spread-Spectrum-Signalen" notwendig ist, wird anhand eines Anwendungsbeispiels für einen kombinierten GPS-GLONASS Navigationsempfänger (ASN-22), der auf $f_b = 5,71$ MHz normierte Amplitudengang (Betrag) in Fig. 11 und der lineare Phasengang (Rad) in Fig. 12 des Filters aufgezeigt. Hierbei erfolgt eine digitale Abwärtsmischung eines mit $f_a = 40$ MHz abgetasteten ca. 15 MHz breiten ZF-Bandes von der Bandmitten-Frequenz $f_m = 70$ MHz in mehrere ca. 1 MHz breite Subbänder in die Basislage bei $f_b/4$ von ca. 1,43 MHz. Ebenfalls wird damit auch aufgezeigt, daß die Übertragungsfunktion $H(z)$ des Filters jeweils Nullstellen bei dem ganzzahligen Vielfachen der Frequenz $f_0 = f_b/4$ (ein Viertel der auf f_b reduzierten Abtastfrequenz) besitzt. Dies ist besonders vorteilhaft zur Unterdrückung der Intermodulationsprodukte in den periodischen Abtastspektren der Signale, die durch die Abtastfrequenzen f_a und f_b hervorgerufen werden können. Deshalb wird in diesem Anwendungsfall ein Abtastraten-Reduktionsfaktor $R = 7$ für die 40 MHz-Abtastfrequenz f_a gewählt, der eine reduzierte 5,71 MHz-Abtastfrequenz f_b ($40/7$ MHz) ergibt. Mit dieser niedrigen Abtastfrequenz f_b erfolgt dann die gesamte Weiterverarbeitung des abgemischten und gefilterten Signals im Basisband.

Zur numerischen Berechnung des Amplitudenganges $G_T(\omega)$, der 3 dB-Bandbreite B_T und des Phasenganges $\Phi_T(\omega)$ werden somit für das Filter folgende Parameter und Beziehungen benützt:

$$1. \text{ Abtastfrequenz: } f_a = 40 \text{ MHz} \quad \text{Abtastraten-Reduktionsfaktor: } R := 7 \quad 2. \text{ Abtastfrequenz: } f_b = \frac{f_a}{R}$$

$$2. \text{ Abtastfrequenz: } f_b = 5.714 \text{ MHz} \quad \text{Elementarfiltergrad: } N := 4 R \quad N = 28$$

$$\text{Nullstellen-Abstand: } \Delta f := \frac{f_a}{N} \quad \Delta f = 1.429 \text{ MHz} \quad \text{Prozeßgewinn: } G := N^2 \quad G = 784$$

$$M = 1000 \quad n := 0.. \frac{N}{2} - 1 \quad \omega = \frac{1 \cdot \pi}{M - 1} \quad t := 1.. M \quad f_t = \frac{\omega \cdot t}{2 \cdot \pi}$$

Mit der Substitution $z = e^{-j \cdot \omega \cdot t}$ und für eine Abtastraten-Reduktion von $R = 7$ lautet damit die Übertragungsfunktion des Filters (normiert auf $f_b = 5,71$ MHz):

$$H(z) := \frac{1}{(16 \exp(-j \cdot 7 \cdot \omega \cdot t))} \cdot \frac{(\exp(-j \cdot 4 \cdot \omega \cdot t) - 1)^2}{(\exp(-j \cdot \omega \cdot t) - 1)^2}$$

$$G_T(\omega) := 20 \log(|H(\omega)| + 10^{-6}) \quad \dots \text{Amplitudengang in dB}$$

$$\Phi_T(\omega) := \arg(H(\omega) + 10^{-10}) \quad \dots \text{Phasengang in Rad}$$

Zur numerischen Berechnung der Gruppenlaufzeit des Filters in $\text{ns} := 10^{-9} \cdot \text{s}$ gilt:

$$D_T(t) = \frac{10^3}{f_b \cdot \pi} \left[\frac{d}{dt} \arg \left[\frac{1}{(16 \exp(-j \cdot 7 \cdot \omega \cdot t))} \cdot \frac{(\exp(-j \cdot 4 \cdot \omega \cdot t) - 1)^2}{(\exp(-j \cdot \omega \cdot t) - 1)^2} \right] \right]$$

Damit ergibt sich eine konstante Gruppenlaufzeit $D_T(1) = 700.701 \text{ ns}$, die identisch mit der Signallaufzeit durch das Filter ist.

3-dB Bandbreite: $B_T = 2 \cdot 0.08401 f_b$ 1. Nullstelle bei: $F_0 = 0.25 f_b$
 1-tes Nebenzipfelmaximum von -22.6 dB bei $0.3664 f_b = 2.094 \text{ MHz}$

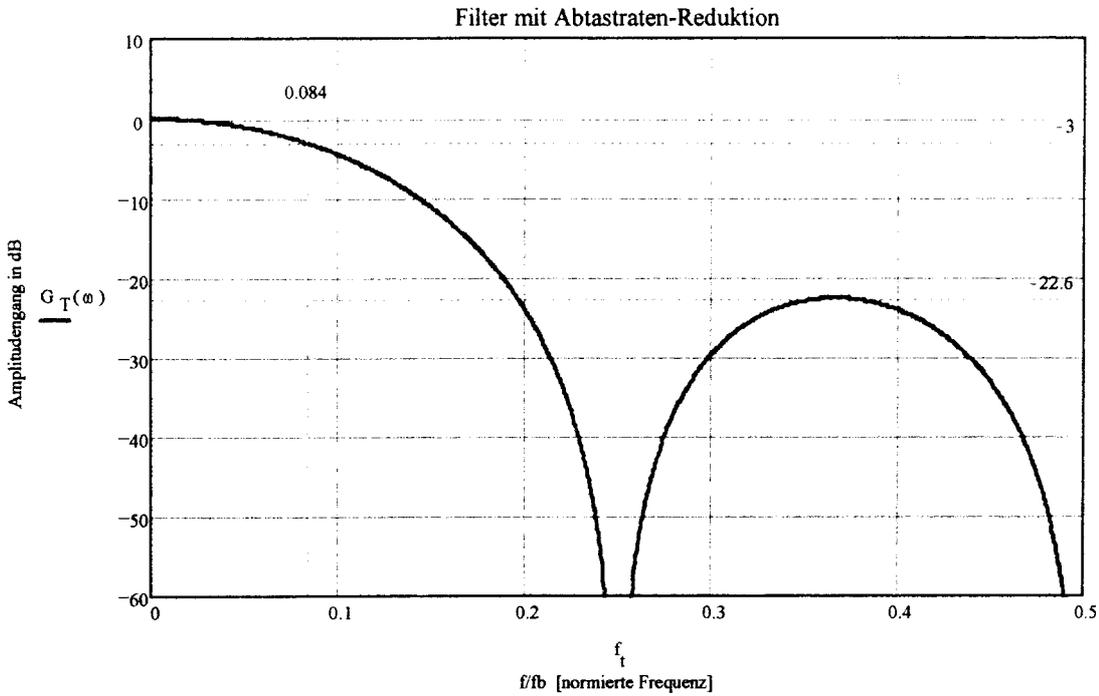


Fig. 11 Amplitudengang des Tiefpaßfilters einer Satelliten-Navigationsanwendung für GPS und GLONASS

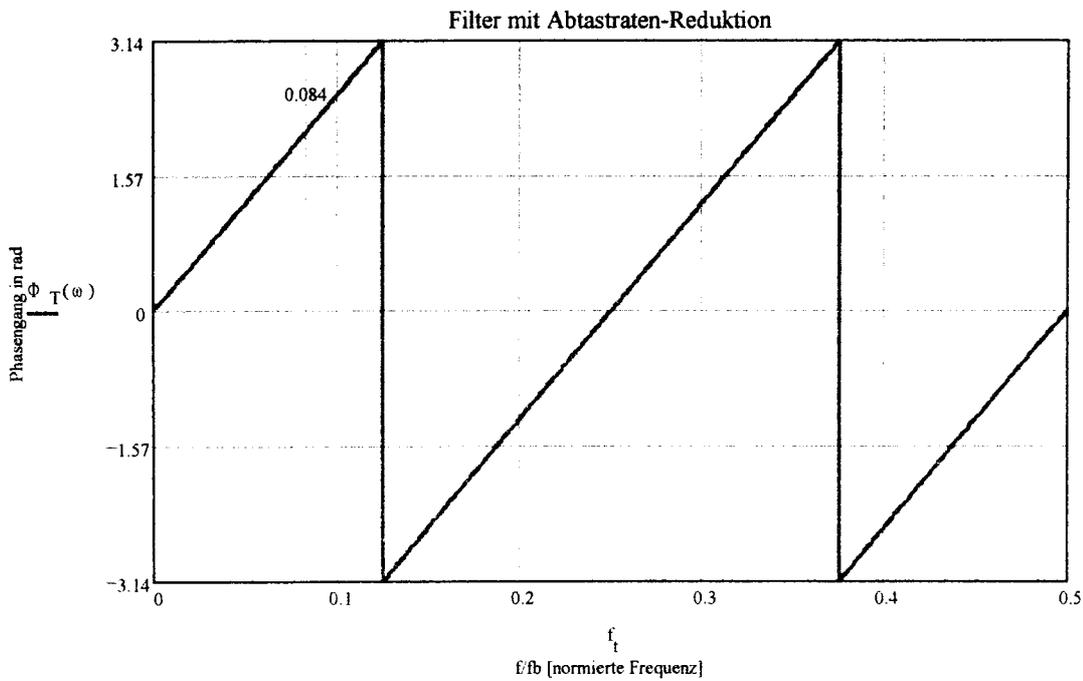


Fig. 12 Phasengang des Tiefpaßfilters einer Satelliten-Navigationsanwendung für GPS und GLONASS

3.2 Die erste, frequenzprogrammierbare Mischstufe

Der technische Schaltungsaufwand für die erste Mischstufe, Fig. 9, (I-Mischer 1, Q-Mischer 1) und für die beiden Phasen/Amplituden-Codierer (PA-Codierer für I-Kanal bzw. Q-Kanal) des DCO1 hängt sehr stark von der verwandten Wortbreite a des Eingangssignals $s_a(n)$, der Wortbreite b des Mischersignals der Frequenz f_m sowie von deren Code-Darstellungen ab. Dagegen wird die Wortbreite von L bit des Phasen-Akkumulators des DCO1 nur von der gewünschten Genauigkeit ΔF (Frequenzschritt-Weite) der Frequenzeinstellung F für die Frequenz-Kanalselektion C_n und von der Abtastfrequenz f_a bestimmt. Damit gilt allgemein für die vom DCO1 generierte Trägerfrequenz:

$$f_m = \frac{\Delta F}{2^L} \cdot f_a$$

Weitere Einzelheiten dazu sind der entsprechenden Literatur [6] usw. zu entnehmen.

Anhand eines *typischen Anwendungsfalls für einen kombinierten GPS-GLONASS-Navigationsempfänger (ASN-22)* werden nun die Vorteile dieses speziellen Mischerkonzeptes aufgezeigt. Hierbei ist üblicherweise [1, 2] eine Quantisierung des Eingangssignals $s_a(n)$ mit einer Wortbreite $a = 2$ bit für den digitalen Abwärtsmischer vollkommen ausreichend. Erfolgt die Codierung von $s_a(n)$ im 2-er-Komplement, das damit nur die folgenden 4 Werte in dezimaler Schreibweise annimmt:

$s_a(n)$ -Werte: 1, 0, -1, -2

so reduzieren sich die beiden *Multiplizierer* und *PA-Codierer*, Fig. 13, der 1. Quadratur-Mischstufe auf *einfache kombinatorische Netzwerke*. Da die Signalverarbeitung im Abwärtsmischer grundsätzlich im 2-er-Komplement-Code ohne Überlauf (Stibitz-Code) erfolgt, können die Multiplikationen des 2 bit Wortes A (vom ADU) mit einem 3 bit Mischer-Wort B (vom PA-Codierer) ausgeführt werden, das sich nur durch 4 verschiedene Pegelwerte für die 8 Cosinus-Sinus-Phasen des Mischsignals B der Frequenz f_m darstellt. Damit ergeben sich für die Wertfolge von B folgende Sequenzen in dezimaler Darstellung:

I-Kanal (Cosinus-Signal B_I):, +2, +2, +1, -1, -2, -2, -1, +1, +2, +2,

Q-Kanal (Sinus-Signal B_Q):, +1, -1, -2, -2, -1, +1, +2, +2, +1, -1,

x-----8-Werte-Zyklus-----x

Mit diesem Multiplikator-Code für f_m , dessen Signalsequenz folglich eine zur Nulllage symmetrische, trapezförmige Einhüllende besitzt, ergibt sich so nur eine Wortbreite $c = 4$ bit des Produktes C (Eingangswort zu den Tiefpaßfiltern). Deshalb kann der Produkt-Code C der Mischstufen nur die 7 Werte in dezimaler Darstellung annehmen:

Produkt-Code-Werte C: -4, -2, -1, 0, +1, +2, +4

Der symmetrische Produkt-Code C verringert somit den Quantisierungsfehler im Mischprodukt für I-Kanal und Q-Kanal des um 1 LSB unsymmetrischen Eingangs-Codes A. Ein "Offset-Fehler" von $1/4$ des Aussteuerbereichs bezüglich der 2 bit Wortbreite von A wird dadurch vermieden. Ebenfalls werden durch diese symmetrische Darstellung des Mischfrequenz-Codes die quadratischen und höherquadratischen Anteile von Störfrequenzen in den Mischsignal-spektren (Quantisierungsfehler-Spektren) erheblich verringert, d.h. das Ausgangssignal des Produktes C weist somit einen kleineren Quantisierungsfehler auf als dies seiner Wortbreite c nach der Theorie entspricht.

Zur technischen Realisierung dieser Quadratur-Mischstufe mit einfachen, kombinatorischen Netzwerken werden nun folgende Vorschläge gemacht.

Die 3 bit Phasen-Amplituden-Codierung des I-Kanal- und Q-Kanal-Mischsignals $B(b_0, b_1, b_2)$ aus dem gemeinsamen 3 bit Phasensignal $P(p_0, p_1, p_2)$, Fig. 13, des Phasen-Akkumulators des DCO1, Fig. 9, gründet sich damit auf die nachfolgende Zuordnungstabelle, wobei von dem *L bit breiten Phasensignal P_A des Akkumulators nur die 3 höchstwertigsten Bits* zu verwenden sind. So stellen die b_0 - bzw. p_0 -Stellen die MSB-Stellen der 2-er-Komplement-Codewörter B bzw. P dar. (Die Wertigkeiten der Bitstellen werden hierbei nach fallenden Potenzen geordnet. Die MSB-Stelle = Vorzeichenstelle besitzt somit die Wertigkeit -2^0).

Phase P _A [Grad]	Akku-Phase P [Binär-Code , Dez.]				I-Kanal-Ampl. B _I [2-er-Kompl. , Dez.]				Q-Kanal-Ampl. B _Q [2-er-Kompl. , Dez.]			
P _{A_i} =	p0 _i =	p1 _i =	p2 _i =	DP _i =	b0 _i =	b1 _i =	b2 _i =	DB _i =	b0 _i =	b1 _i =	b2 _i =	DB _i =
135	1	1	1	7	0	0	1	1	1	1	0	-2
90	1	1	0	6	1	1	1	1	1	1	0	-2
45	1	0	1	5	1	1	0	2	1	1	1	-1
0	1	0	0	4	1	1	0	2	0	0	1	1
-45	0	1	1	3	1	1	1	-1	0	1	0	2
-90	0	1	0	2	0	0	1	1	0	1	0	2
-135	0	0	1	1	0	1	0	2	0	0	1	1
-180	0	0	0	0	0	1	0	2	1	1	1	-1

Tabelle 1 Zuordnung der Phasen-Worte P des DCO1's zu den Amplituden-Worten B_I und B_Q für das Mischsignal der 1. Quadratur-Mischstufe mit der Mischfrequenz f_m

Anhand von Tabelle1 sind damit die Schaltungen der kombinatorischen Netzwerke der PA-Codierer, Fig. 13, für I- und Q-Kanal eindeutig festgelegt. Die minimisierten Bool'schen Gleichungen zur Erzeugung der 3 bit Amplituden-Ausgangsdaten B(b0, b1, p2) aus den 3 bit Phasen-Eingangsdaten P(p0, p1, p2) können dazu angegeben werden, wie folgt:

I-Kanal-Amplitudenwort B_I:

b0 = (p0∧p1)V(p0∧p2)V(p0∧p1∧p2)MSB (VZ)

b1 = p1V(p0∧p2)V(p0∧p2) bzw. b1 = p1V(p0"+"p2)

b2 = p1LSB

Q-Kanal-Amplitudenwort B_Q:

b0 = (p0∧p1)V(p0∧p2)V(p0∧p1∧p2)MSB (VZ)

b1 = p1V(p0∧p2)V(p0∧p2) bzw. b1 = p1V(p0"+"p2)

b2 = p1LSB

(Das Zeichen "+" steht hier für die EXKLUSIV-ODER-FUNKTION bzw. Moduladdition ohne Überlauf; Variablen-Unterstreichung für Variablen-Negation)

Dem entsprechend kann auch das kombinatorische Netzwerk der Multiplizierer, Fig. 13, der ersten Mischstufe aus der Zuordnung des 4 bit Produkt-Code-Wortes C(c0, c1, c2, c3,) zu dem 2 bit Eingangs-Wort A(a0,a1) und dem 3 bit PA-Codierere-Ausgangswort B(b0, b1, b2) hergeleitet werden. Damit gilt hier folgende Codewort-Zuordnungstabelle für die beiden identischen Multiplizierer der 1. Mischstufe:

Input A [2-er-Kompl., Dez.]			Input B [2-er-Kompl., Dez.]				Produkt C [2-er-Kompl., Dez.]				
$a_0 = a_1 = DA_1 =$			$b_0 = b_1 = b_2 = DB_1 =$				$c_0 = c_1 = c_2 = c_3 = DC_1 =$				
0	0	0	0	1	0	2	0	0	0	0	0
0	0	0	0	0	1	1	0	0	0	0	0
0	0	0	1	1	1	-1	0	0	0	0	0
0	0	0	1	1	0	-2	0	0	0	0	0
0	1	1	0	1	0	2	0	0	1	0	2
0	1	1	0	0	1	1	0	0	0	1	1
0	1	1	1	1	1	-1	1	1	1	1	-1
0	1	1	1	1	0	-2	1	1	1	0	-2
1	0	-1	0	1	0	2	1	1	1	0	-2
1	0	-1	0	0	1	1	1	1	1	1	-1
1	0	-1	1	1	1	-1	0	0	0	1	1
1	0	-1	1	1	0	-2	0	0	1	0	2
1	1	-2	0	1	0	2	1	1	0	0	-4
1	1	-2	0	0	1	1	1	1	1	0	-2
1	1	-2	1	1	1	-1	0	0	1	0	2
1	1	-2	1	1	0	-2	0	1	0	0	4

Tabelle 2 Zuordnung der Eingangsworte A und B zu den Produktworten C der identischen I- und Q-Multiplizierer der 1. Quadratur-Mischstufe

Aus der Tabelle 2 ergeben sich dann die minimierten Bool'schen Gleichungen für den 4 bit Produkt-Code C(c0, c1, c2, c3) der Multiplizierer, Fig. 13, der 1. Quadratur-Mischstufe aus dem 2 bit Eingangswort A(a0,a1) und dem 3 bit Eingangswort B(b0,b1,b2) wie folgt:

Produkt-Code C:

$c_0 = (a_1 \wedge b_2) \vee (a_1 \wedge a_0 \wedge b_2)$ MSB (VZ)
 $c_1 = (a_1 \wedge a_0 \wedge b_0) \vee (a_1 \wedge b_2) \vee (a_1 \wedge a_0 \wedge b_0)$
 $c_2 = (a_0 \wedge b_0) \vee (a_1 \wedge a_0 \wedge b_0) \vee (a_1 \wedge a_0 \wedge b_2) \vee (a_1 \wedge a_0 \wedge b_1)$
 $c_3 = (a_0 \wedge b_0)$ LSB

Dies erfordert zur technischen Realisierung der Multiplizierer einen nur sehr geringen technischen Aufwand. D.h. die Multiplizierer können mit wenigen, logischen Grundbausteinen, wie z.B. mit UND-Gatter, ODER-Gatter und Inverter, ausgeführt werden.

3.3 Die zweite Festfrequenz-Mischstufe

Zur Reellmachung des komplexen Filter-Ausgangssignals $v(m)$, Fig. 7 und 9, ist eine 2. *komplexe Mischung mit der Trägerfrequenz $f_T = f_b/4$* allgemein bei der Tiefpaß-Lösung notwendig. Dies wurde bereits unter anderem in Abschnitt 2 beschrieben. Die Erzeugung der *Mischfrequenz f_T* erfolgt hier, Fig. 14, mit einem einfachen 2 bit *Johnson-Zähler (2 bit Ringzähler)*, der direkt durch seine 1:4 Teilung der 2. Abtastfrequenz f_b die erforderlichen um 90° verschobenen Mischfrequenzen f_T und jf_T bereit stellt. Damit kann diese 2. Mischung durch eine ganz triviale Signalmultiplikation ausgeführt werden, da die Mischfrequenz-Amplitude nur aus den 3 Werten: -1, 0, +1 besteht. Wie folgt stellen sich dann die Träger-Sequenzen als 4-Werte-Zyklen dar:

I-Kanal (Cosinus-Signal) :, +1, 0, -1, 0, +1, 0, -1,

Q-Kanal (-Sinus-Signal) :, 0, -1, 0, +1, 0, -1, 0,

x--Zyklus--x

Um nicht unnötige Phasenfehler in der Modulation des Realteils und Imaginärteils des Mischer-Ausgangssignals $w(m)$ (Einseitenband-Signal) beim Loslaufen (Einschalten) der ZF-Stufe hervorzurufen, muß der Johnson-Zähler für diese Mischung noch in eine feste Phasenlage relativ zum DCO der 1. Mischstufe gebracht werden. Mit der Annahme, die willkürliche Phasenlage des DCO's sei ψ und die des Johnson-Zählers ε , muß gelten:

- a) für das obere Seitenband: $\psi = \varepsilon$, d.h. die Oszillatoren (DCO's) müssen beim Einschalten gleichzeitig mit $\psi - \varepsilon = 0$ (gleiche Phasenlage) loslaufen.
- b) für das untere Seitenband: $\psi = -\varepsilon$, d.h. beide Oszillatoren müssen beim Einschalten nur an ihrem Cosinusausgang (Realteil) gleichzeitig mit $\psi = \varepsilon$ (gleiche Phasenlage) loslaufen, am Sinusausgang (Imaginärteil) muß der Johnson-Zähler dagegen mit $\psi - \varepsilon = 180^\circ$ (180° Phaselage gegenüber dem DCO) gestartet werden.

Jedoch im Fall einer reinen BPSK-Modulation des Empfangssignals $s_a(n)$, Fig. 9, wie dies bei GPS-, GALILEO- und GLONASS-Navigationsempfängern immer gegeben ist, sind diese obigen Einschaltbedingungen der DCO's belanglos. Hiermit wird nur bestimmt, ob das Modulationsergebnis $w(m)$ am Mischerausgang mit 0° oder 180° Phasenlage zum DCO2-Signal (Anfangsphase) erscheint.

Eine besonders vorteilhafte, *allgemein gültige, technische Realisierung einer solchen 2. Festfrequenz-Mischstufe* wird in Fig. 14 angegeben. Aus den 2 binären Zustandsfolgen des Johnson-Zählers:

Q1-Zustandsfolge:, 0, 0, 1, 1, 0, 0,

Q2-Zustandsfolge:, 1, 0, 0, 1, 1, 0,

x--Zyklus--x

können mit Hilfe eines sehr einfachen, *kombinatorischen Schaltung* die erforderlichen, *binären Mischsignale (Trägersignale) INV_I, NUL_I bzw. INV_Q, NUL_Q* für die Trägerfrequenz f_T bzw. für die Trägerfrequenz jf_T (90° -phasenverschoben) gewonnen werden. Dem entsprechend stellt sich die Zuordnungstabelle der Codierschaltung für die Ausgangsvariablen INV_I, NUL_I und INV_Q, NUL_Q bezüglich der Johnson-Zähler-Zustandsfolge Q1,Q2, wie folgt dar:

Johnson- Zähler	I-Kanal-Träger [2-er-Kompl., Dez.]	Q-Kanal-Träger [2-er-Kompl., Dez.]																												
$Q1_i = Q2_i =$	$INV_{I_i} = DZ_{I_i} =$	$INV_{Q_i} = NUL_{Q_i} = DZ_{Q_i} =$																												
<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td></tr> </table>	0	0	1	0	1	1	0	1	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>1</td></tr> <tr><td>x</td><td>0</td></tr> <tr><td>1</td><td>1</td></tr> <tr><td>x</td><td>0</td></tr> </table>	0	1	x	0	1	1	x	0	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>x</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>x</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> </table>	x	0	0	1	1	1	x	0	0	0	1	1
0	0																													
1	0																													
1	1																													
0	1																													
0	1																													
x	0																													
1	1																													
x	0																													
x	0	0																												
1	1	1																												
x	0	0																												
0	1	1																												

Tabelle 3 Mischer-Code-Zuordnung der 2. Quadratur-Mischstufe (x = beliebiger Binärwert)

Damit ergeben sich dann für diese Codierschaltung der Trägeramplituden INV_I , NUL_I und INV_Q , NUL_Q bezüglich der Johnson-Zähler-Zustandsfolge $Q1, Q2$ die folgenden Bool'schen Gleichungen:

I-Kanal-Amplitudenwort:

$INV_I = Q1 \underline{V} Q2$

$NUL_I = Q1 \underline{+} Q2 = NUL_Q$

Q-Kanal-Amplitudenwort:

$INV_Q = Q1 \underline{V} Q2$

$NUL_Q = Q1 \underline{+} Q2 = NUL_I$

(Das Zeichen "+" steht hier für die EXKLUSIV-ODER-FUNKTION bzw. Moduladdition ohne Überlauf; Variablen-Unterstreichung für Variablen-Negation)

Mit jeweils diesen Bitsignalpaaren INV , NUL werden nach Fig. 14 die Multiplikation der komplexen Tiefpaßfilter-Ausgangssignale $v(m)$ in der 2. Mischstufe derart ausgeführt, daß gemäß der Multiplikatorfolge: 1, 0, -1 des Trägersignals die Signale $v_R(m)$ (Realteil) und $v_I(m)$ (Imaginärteil) entweder unverändert durchgeschaltet, auf den Wert Null gesetzt oder invertiert werden. Diese Signalmultiplikationen sind damit sehr einfach aus der Hintereinanderschaltung von EXKLUSIV-ODER-Gatter mit UND-Gatter realisierbar. Ebenfalls kann die restliche Schaltung der 2. Mischstufe aus einfachen Logikbausteinen und Speicherelementen aufgebaut werden. So ist z.B. der Johnson-Zähler mit zwei D-Flip-Flops besonders einfach zu realisieren. Dies gilt gleichermaßen für die Codierschaltung der Bit-Signale INV_I , NUL_I und INV_Q , NUL_Q , die mit EXKLUSIV-ODER-Gatter bzw. deren Negation und mit ODER-Gatter vollständig ausgeführt werden können. Die oben erwähnte, binäre Signal-Invertierung von $v(m)$ im Fall der Signalmultiplikation mit dem Faktor -1 ergibt eine 1-er-Komplement-Code-Darstellung von $v(m)$ für das Ausgangssignal $w(m)$. Diese Code-Darstellung ist hier jedoch zulässig, da für die nachfolgende digitale Weiterverarbeitung nach der Realteil-Imaginärteil-Addition eine entsprechende Wortbreiten-Reduzierung von $s_b(m)$, Fig. 9, mit Hilfe eines sogenannten "Formatierers" erfolgen muß. Hierbei sind die Fehler, die durch den um 1 LSB sich vom 2-er-Komplement-Code unterscheidenden 1-er-Komplement-Code hervorgerufen werden, entweder vernachlässigbar oder können bei der Signal-Skalierung im nachfolgenden Formatierer berücksichtigt werden.

4. Wortbreiten-Reduktion des ZF-Stufen-Ausgangssignals mit dem Gray-Code-Formatierer

Die durch die Tiefpaß-Filterungen und Signalmischungen in dem ZF-Quadratur-Kanal der digitalen ZF-Stufe, Fig. 9, bedingten Vergrößerung der Signaldynamik von a bit bis f bit, muß an die wesentlich kleinere Wortbreite von g bit der nachfolgenden Signalverarbeitung, z.B. mit entsprechenden Mikroprozessoren, angepaßt werden. Vor allem bestimmt der Prozeßgewinn $G = 784$ - siehe dazu Abschnitt 3.1 - der Tiefpaßfilterung die Wortbreite der komplexen Ausgangssignale $v(m)$ und $w(m)$. So ergibt sich z.B. bei einer Wortbreite der Tiefpaß-Eingangssignale $x(n)$ von $c = 4$ bit bereits eine erforderliche Verarbeitungs-Wortbreite $d = 15$ bit für die Tiefpaßfilterung. Etwa die gleiche Größe von 12 bit bis 15 bit müssen auch die Wortbreiten e und f der 2. Mischstufe sowie des Realteil-Imaginärteil-Addierers am Abwärtsmischer-Ausgang aufweisen. (Die genau dazu erforderlichen Wortbreiten e bzw. f sind eine Funktion der Signalstatistik des Eingangssignals $s_a(n)$ und müssen im Einzelfall berechnet werden).

Damit entspricht dieser Wortbreiten-Reduktion des Ausgangssignals $s_b(m)$ von f auf g bit eine Signal-Skalierung mit dem Faktor $\ll 1$ bzw. einer ganzzahligen Signal-Division mit dem Wert $\gg 1$. Folglich kann auch diese Wortbreiten-Reduktion als eine Art einer zweiten "Signal-Quantisierung" des Signals $s_b(m)$ betrachtet werden. Außerdem muß diese Skalierung bzw. Division gemäß der vorliegenden Signalstatistik von $s_a(n)$ variabel ausführbar sein. D.h. die Werte (LIMIT) für die Skalierung bzw. Division sind von extern einstellbar zu halten, da infolge einer in der Praxis nur gegebenen teilweisen Aussteuerung durch $s_a(n)$, die Signaldynamik der Tiefpaßfilterung von d bit nicht voll ausgenutzt wird. Demzufolge werden bei der 2-er-Komplement-Darstellung der Filterworte die oberen nicht genutzten Bitstellen durch gleiche Vorzeichen-Bits aufgefüllt. Deshalb kann eine einfache Skalierung des Ausgangssignals $s_b(m)$ nicht durch Weglassen (Abschneiden) von oberen Bitstellen realisiert werden. Im Fall einer Reduzierung der Ausgangswortbreite auf $g = 2$ bit, wie dies bei den meisten GPS- und GLONASS-Anwendungen [1] erforderlich ist, würde das in der Regel den Verlust der gesamten Signalinformation bis auf das Vorzeichen-Bit bedeuten.

Nach Fig. 15 ergibt sich mittels eines so genannten "Gray-Code-Formaterers" eine technisch besonders einfache und günstige Lösung der Wortbreiten-Reduktion für das Ausgangssignal $s_b(m)$ des Realteil-Imaginärteil-Addierers auf ein 2 bit Signal $s_f(m)$ im Gray-Code. (Falls erforderlich, ist dieses 2 bit Gray-Code-Signal mit einem einfachen EXKLUSIV-ODER-Gatter in das entsprechende 2 bit Binär-Code-Signal zu konvertieren!) Hierbei wird direkt nach der Realteil-Imaginärteil-Addition die höchstwertigste Bitstelle MSB (Vorzeichen-Bit) von den restlichen Bitstellen ($f-1$ Rumpf-Bits) des Summensignals $s_b(m)$ abgetrennt und ergibt somit über eine Zwischenspeicherung von der Taktperiodendauer $R+T$ das höchstwertigste Gray-Code-Bitsignal SIGN (MSB) des Ausgangssignals $s_f(m)$. (Es entspricht ebenfalls im Gray-Code dem Vorzeichen-Bit des 2-er-Komplement-Codes). Dagegen werden die Rumpf-Bitsignale von $s_b(m)$ der Wortbreite $f-1$ bit einer Betragsbildung unterzogen und damit ist anschließend eine $f-1$ bit breite Schwellwert-Addition mit dem einstellbaren Schwellwert LIMIT von der Wortbreite h bit auszuführen. Hierbei stellt bereits das Bereichs-Übertauf-Bitsignal ÜB dieser Addition (entspricht einer digitalen Komparatorfunktion) das 2. Bitsignal MAGN des Gray-Code-Signals $s_f(m)$ dar. Es wird ebenfalls, wie im Fall des SIGN-Signals, über eine Zwischenspeicherung der Dauer $R+T$ an den Formatiererausgang geführt. Die Zwischenspeicherungen von der Dauer $R+T$ der Bitsignale SIGN und MAGN sind funktional nicht notwendig, sind jedoch in der technischen Realisierung zur Datensynchronisation meistens von Vorteil.

Damit gelten zur Bildung der beiden Gray-Code-Bitsignale SIGN und MAGN folgende Logik-Bedingungen bezüglich des Summensignals $s_b(m)$ der Realteil-Imaginärteil-Addition:

$$s_b(m) = w_R(m) + w_I(m) \quad \dots\dots\text{im 2-er-Komplement-Code}$$

$$\text{SIGN} = \begin{cases} 1 & \text{für } s_b(m) \leq 0 \\ 0 & \text{für } s_b(m) \geq 0 \end{cases} \quad \dots\dots\text{identisch mit dem Vorzeichen VZ}$$

$$\text{MAGN} = \begin{cases} 1 & \text{für } (|s_r(m)| + \text{LIMIT}) \leq \text{Addierer - Überlauf } \ddot{U} \\ 0 & \text{für } (|s_r(m)| + \text{LIMIT}) \geq \text{Addierer - Überlauf } \ddot{U} \end{cases} \quad \dots \text{im natürlichen Binär-Code}$$

bzw.

$$\text{MAGN} = \begin{cases} 1 & \text{für } (|s_r(m)| - \text{LIMIT}) \leq 0 \\ 0 & \text{für } (|s_r(m)| - \text{LIMIT}) \geq 0 \end{cases} \quad \dots \text{im 2-er-Komplement-Code}$$

Ein Schaltungsbeispiel eines solchen 2 bit Gray-Code -Formatierers, *realisierbar mit 2-er-Komplement-Addierer*, wird in Fig. 16 aufgezeigt. Hierbei ist noch eine zusätzliche, technisch nicht unbedingt notwendige Zwischenspeicherung des Summensignals $s_b(m)$ des Realteil-Imaginärteil-Addierers mit einem D-Register vorgesehen. Damit unterscheidet sich für die technische Ausführung des 2 bit Gray-Code-Formatierers von Fig. 16 gegenüber Fig. 15 in folgenden, wesentlichen Punkten:

- Die Betragsbildung des Rumpf-Bitsignals der Breite $f-1$ aus $s_b(m)$ geschieht hier durch eine einfache *Invertierung der Rumpf-Bits in Abhängigkeit des invertierten Vorzeichenbits \underline{VZ} des Summensignals $s_b(m)$* mittels EXCLUSIV-ODER-Gatter. Diese liefern somit im Fall von $VZ = -1$ von $s_b(m)$ das Rumpf-Bitsignal im 1-er-Komplement-Code an den nachfolgenden f bit breiten Schwellwert-Addierer.
- Die 1-er-Komplement-Code-Darstellung des Rumpf-Bitsignals der Breite $f-1$ wird zur Schwellwert-Addition durch die *Belegung des "Carry In" am Überlaufeingang C_0 des Schwellwert-Addierers mit dem invertierten Vorzeichenbit \underline{VZ} von $s_b(m)$* in das notwendige 2-er-Komplement-Codesignal intern umgesetzt (korrigiert).
- Damit erfolgt die Schwellwert-Addition der Rumpf-Bitsignale mit dem Schwellwert-Signal LIMIT grundsätzlich *im f bit breiten 2-er-Komplement-Code ohne Überlauf*. Hierbei ist die *höchstwertigste Bitstelle (MSB) des Ausgangssignals $s_s(m)$* des Schwellwert-Addierers identisch mit dem *zweiten Gray-Code-Bit-Signal MAGN (2^{-1} , LSB) am Formatierer-Ausgang*. Da die *erste Gray-Code-Bitstelle SIGN (2^0 , MSB)* identisch bzw. invertiert zum Vorzeichenbit (VZ) des Formatierer-Eingangssignals $s_b(m)$ vorliegt, wird sie *unverändert über das D-Register zum Formatierer-Ausgang durchgeschaltet*.
- Der Schwellwert LIMIT mit der Wortbreite $h \leq f-1$ muß hierbei als positiver Komplementwert bezüglich der positiven Bereichsgrenze der Schwellwert-Addition vorliegen. Die Darstellung des *Schwellwertes LIMIT erfolgt somit im natürlichen Binär-Code*.
- Die Bitstellen des Ausgangssignals $s_r(m)$ bzw. $s_s(m)$ (WORD) nach dem D-Register mit der geringeren Stellenwertigkeit $< 2^0$ (unterhalb des MSB's von $s_s(m)$), die unter anderem keinen Beitrag mehr zum 2 bit Gray-Code-Signal $s_r(m)$ am Ausgang liefern, geben den *Wert bzw. dessen Komplementwert der Abweichung (Differenz) des Schwellwertes LIMIT vom Betrag des Signalwertes $s_b(m)$* an. Dieses $f-1$ bit breite Binärsignal WORD kann somit als Berechnungsgrundlage (Signal-Statistik) für eine automatische, adaptive Schwelleneinstellung mit LIMIT am Schwellwert-Addierer dienen.

Literaturverzeichnis

- [1] Elliot D. Kaplan: Understanding GPS: principles and applications, Artech House, Boston, London, 1996
- [2] W. Mansfeld: Satellitenortung und Navigation, Grundlagen und Anwendung globaler Satellitennavigationssysteme, Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig/Wiesbaden, 1998
- [3] Lawrence R. Rabiner, Bernard Gold: Theory and application of digital signal processing, AT&T, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1975
- [4] H. D. Lüke: Signalübertragung, 3. Auflage, Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 1985
- [5] H. Götz: Einführung in die digitale Signalverarbeitung, Teubner Studienskripten, B.G. Teubner-Verlag, Stuttgart, 1995
- [6] M. Zimmer: Phasen/Digital-Umsetzer: Neue Methode zur direkten digitalen Detektion von FSK- und PSK-Signalen, HF-Report 1/91, Teil1, S. 74-78 und HF-Report 2/91, Teil2, S. 62-67

A n h a n g

Bilderverzeichnis

- Fig. 1 GNSS-Frequenzband-Belegungen im oberen/unteren L-Band und C-Band
- Fig. 2 Allgemeiner Hochfrequenz-Empfänger mit digitaler ZF-Stufe und digitaler Signalauswertung (Digitaler ZF-Empfänger)
- Fig. 3 Herkömmliche, komplexe Bandpaß-Lösung
- Fig. 4 Frequenzplan der herkömmlichen, komplexen Bandpaß-Lösung
- Fig. 5 Herkömmliche, komplexe Tiefpaß-Lösung
- Fig. 6 Frequenzplan der herkömmlichen, komplexen Tiefpaß-Lösung
- Fig. 7 Die neue Tiefpaß-Variante für eine digitale ZF-Stufe
- Fig. 8 Digitale Mehrkanal-ZF-Stufe mit der neuen Tiefpaß-Variante
- Fig. 9 Ausführungsbeispiel eines ZF-Quadratur-Kanals der digitalen ZF-Stufe mit dem neuen Tiefpaßfilter
- Fig. 10 Funktionales Schaltbild des Tiefpaßfilters zur Filter-Dimensionierung
- Fig. 11 Amplitudengang des Tiefpaßfilters einer Satelliten-Navigationsanwendung bei GPS und GLONASS
- Fig. 12 Phasengang des Tiefpaßfilters einer Satelliten-Navigationsanwendung bei GPS und GLONASS
- Fig. 13 Zur Dimensionierung der ersten Mischstufe für eine GPS-GLONASS-Anwendung (Beispiel)
- Fig. 14 Vereinfachtes Schaltbild der zweiten Festfrequenz-Mischstufe (Beispiel)
- Fig. 15 Funktionales Schaltbild des 2 bit Gray-Code-Formatierers für eine GPS-GLONASS-Anwendung (Beispiel)
- Fig. 16 Vereinfachtes Schaltbild des 2 bit Gray-Code-Formatierers für eine GPS-GLONASS-Anwendung (Beispiel)

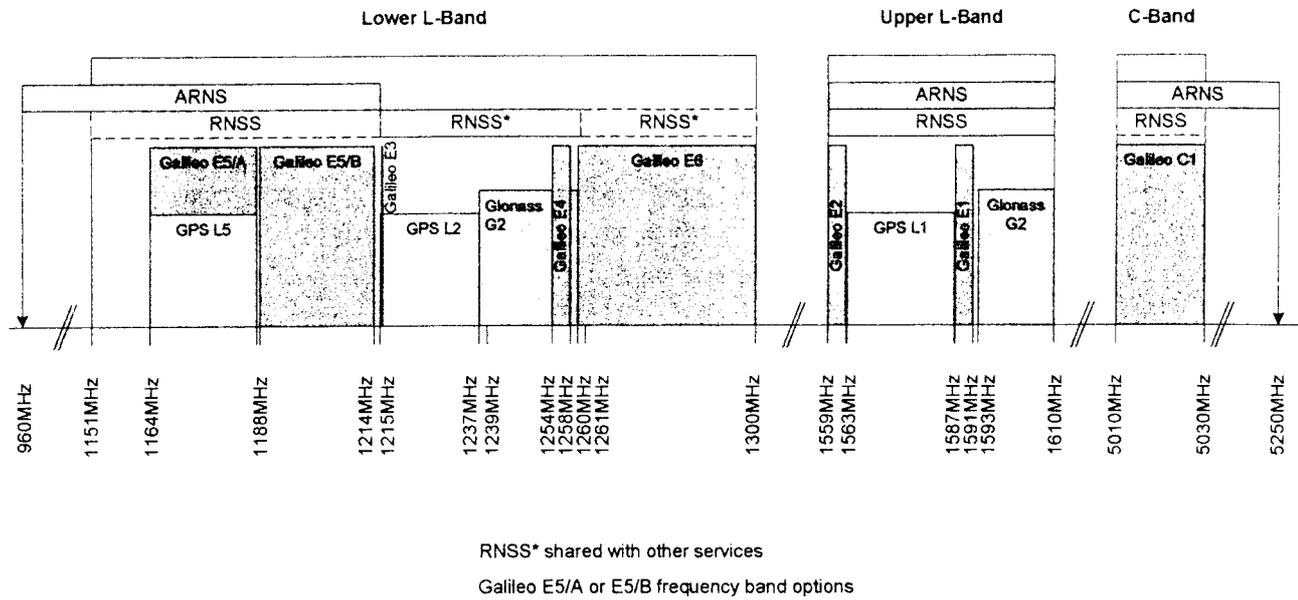


Fig.1 GNSS-Frequenzband-Belegung im oberen/unteren L-Band und C-Band

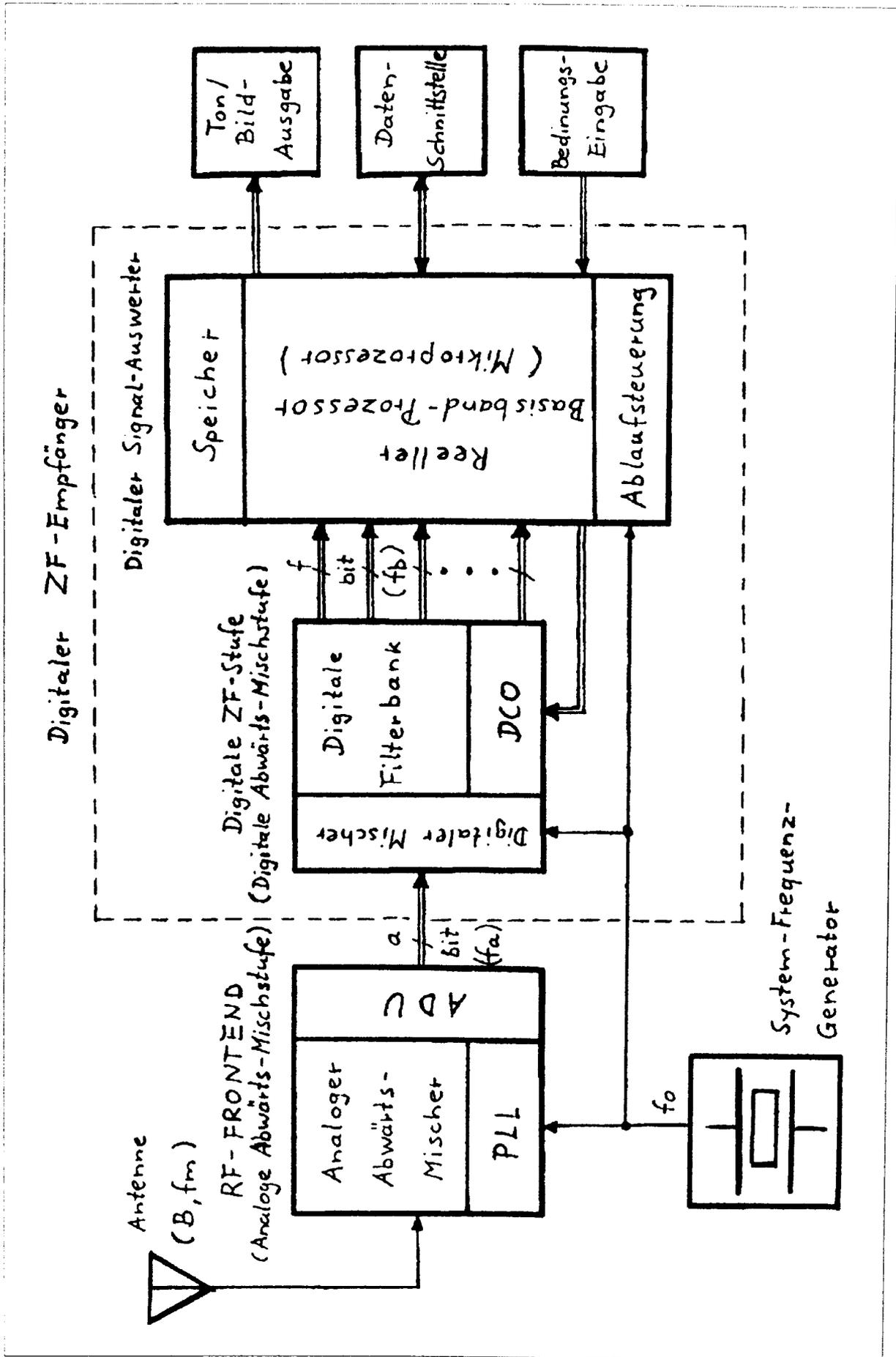


Fig.2 Allgemeiner Hochfrequenz-Empfänger mit digitaler ZF-Stufe und digitaler Signalauswertung (Digitaler ZF-Empfänger)

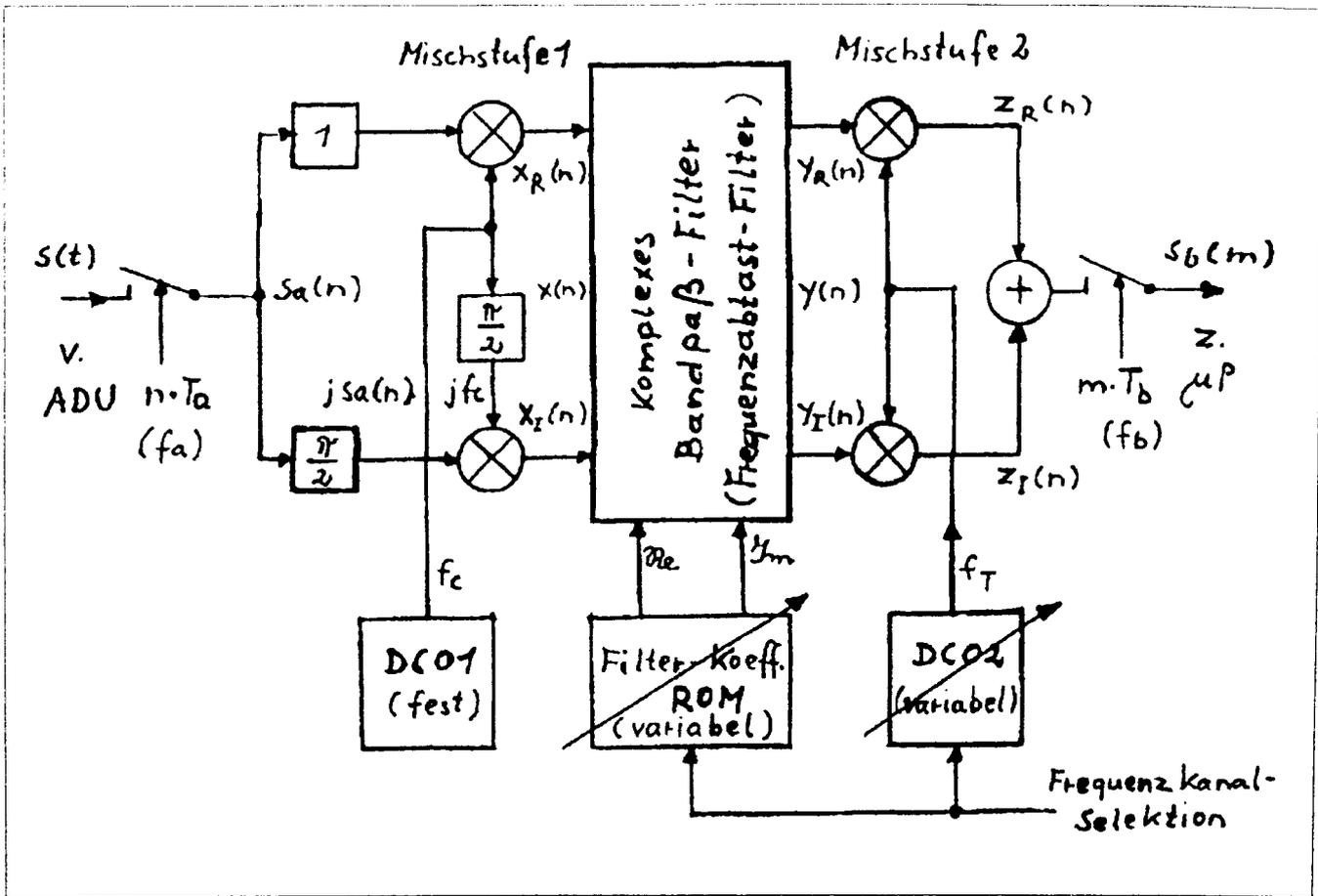


Fig.3 Herkömmliche, komplexe Bandpaß-Lösung

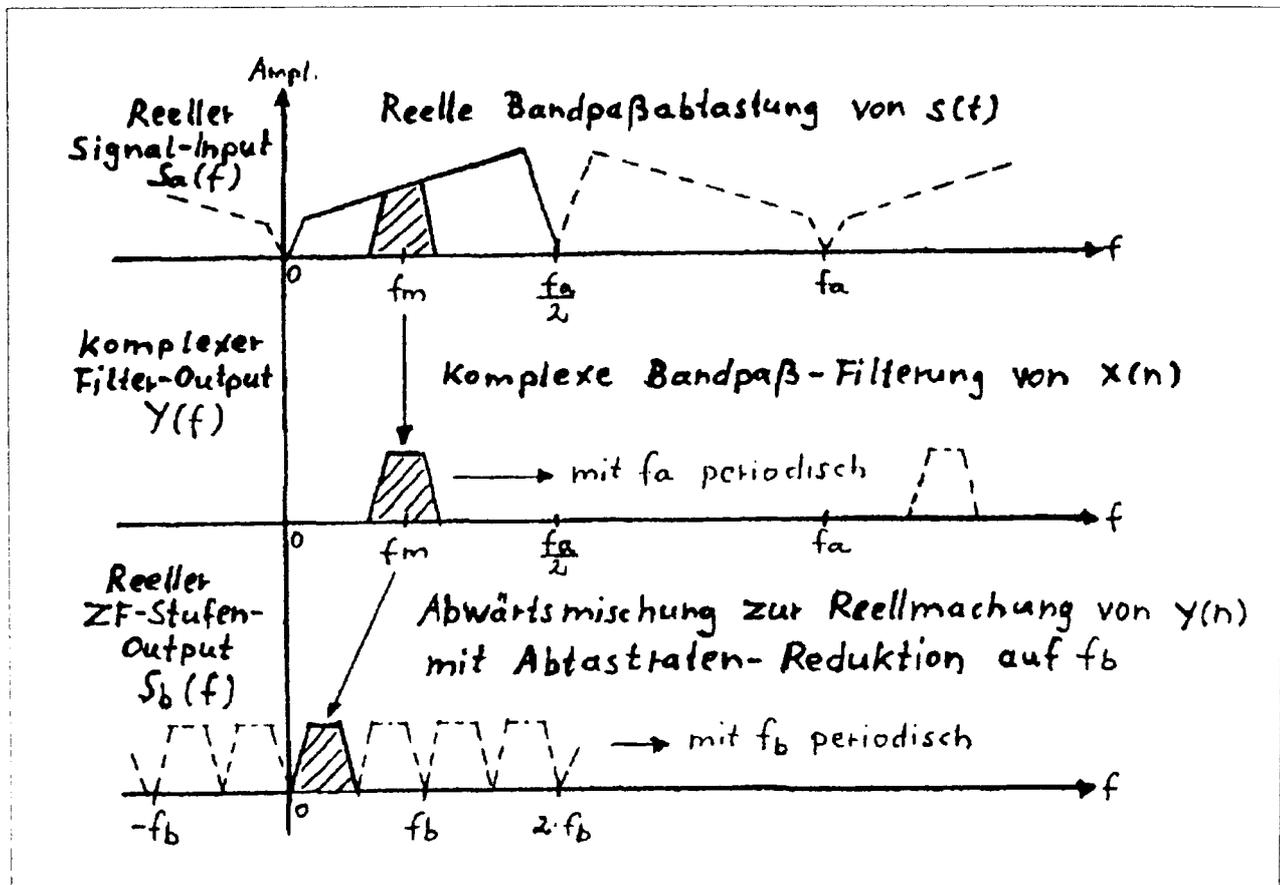


Fig.4 Frequenzplan der herkömmlichen, komplexen Bandpaß-Lösung

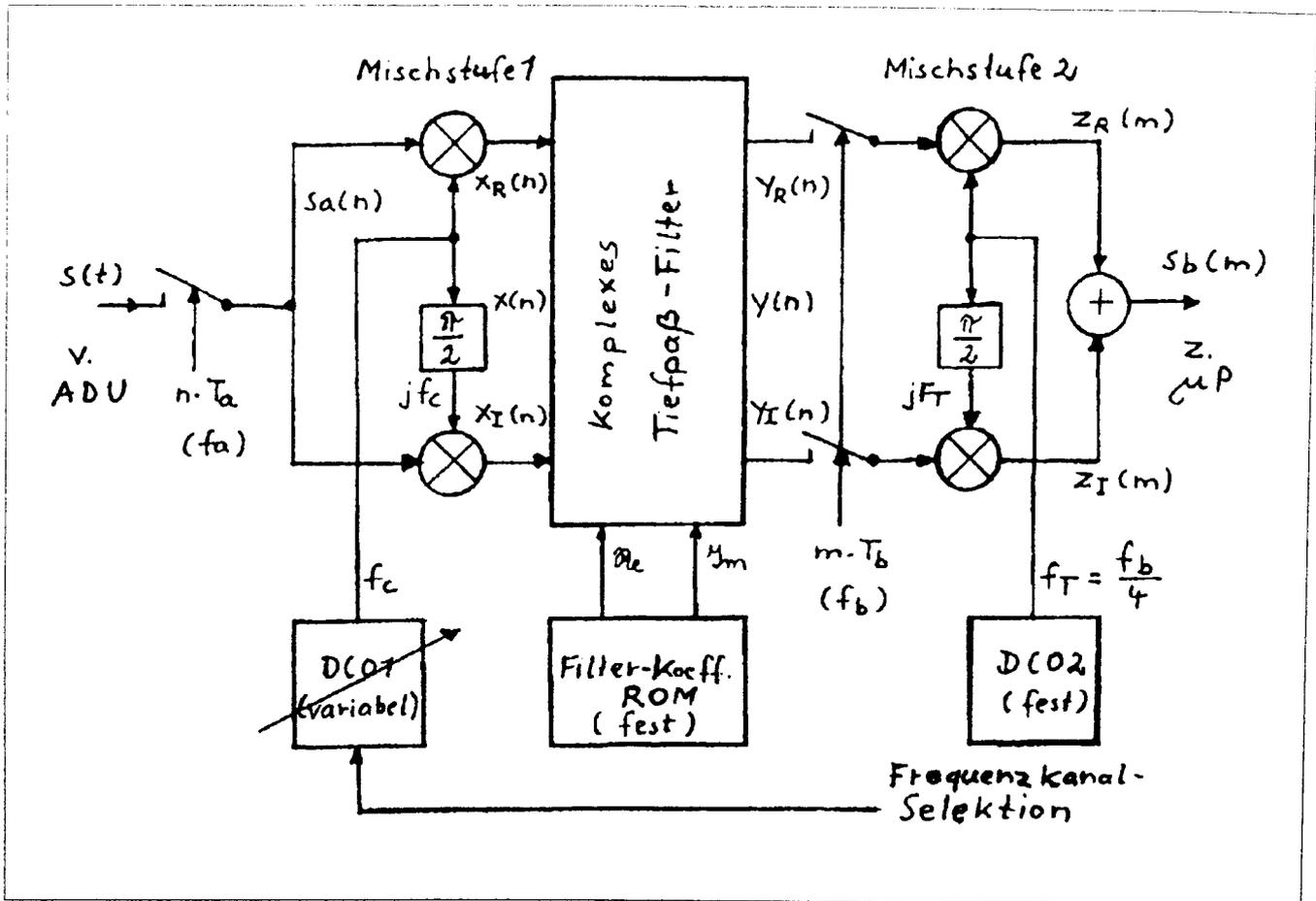


Fig.5 Herkömmliche komplexe Tiefpaß-Lösung

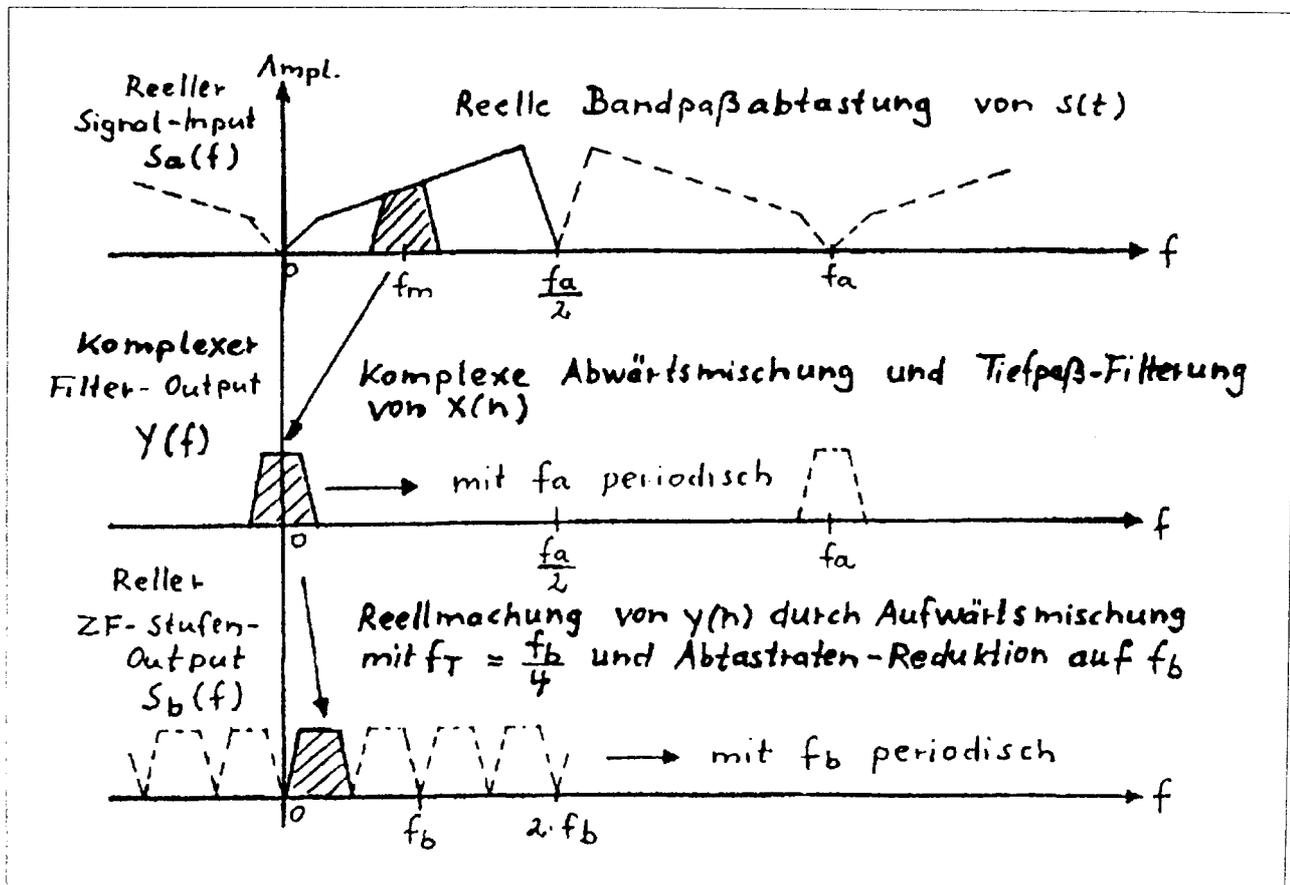


Fig.6 Frequenzplan der herkömmlichen, komplexen Tiefpaß-Lösung

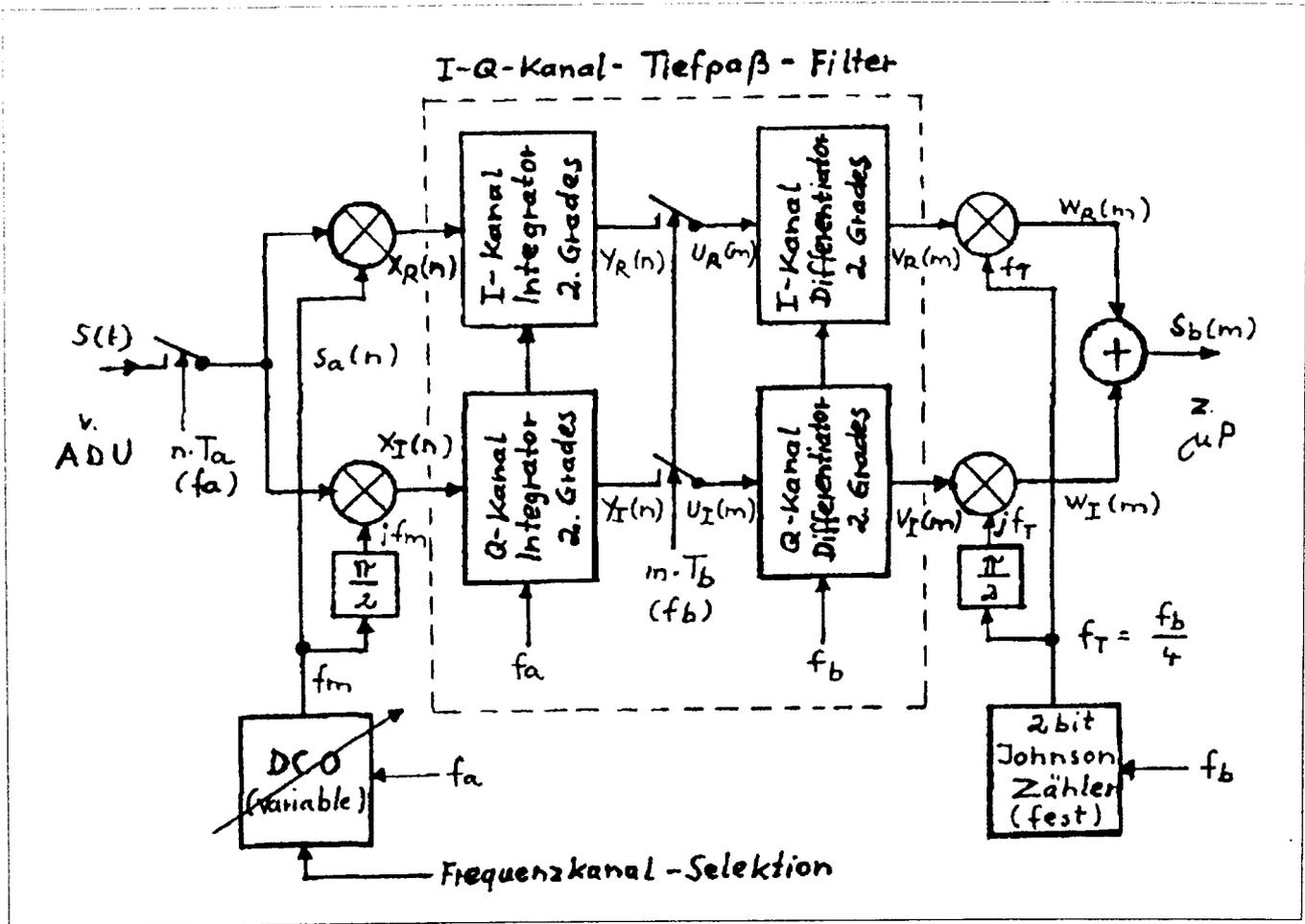


Fig.7 Die neue Tiefpaß-Variante für eine digitale ZF-Stufe (Digitale Abwärtsmischerstufe)

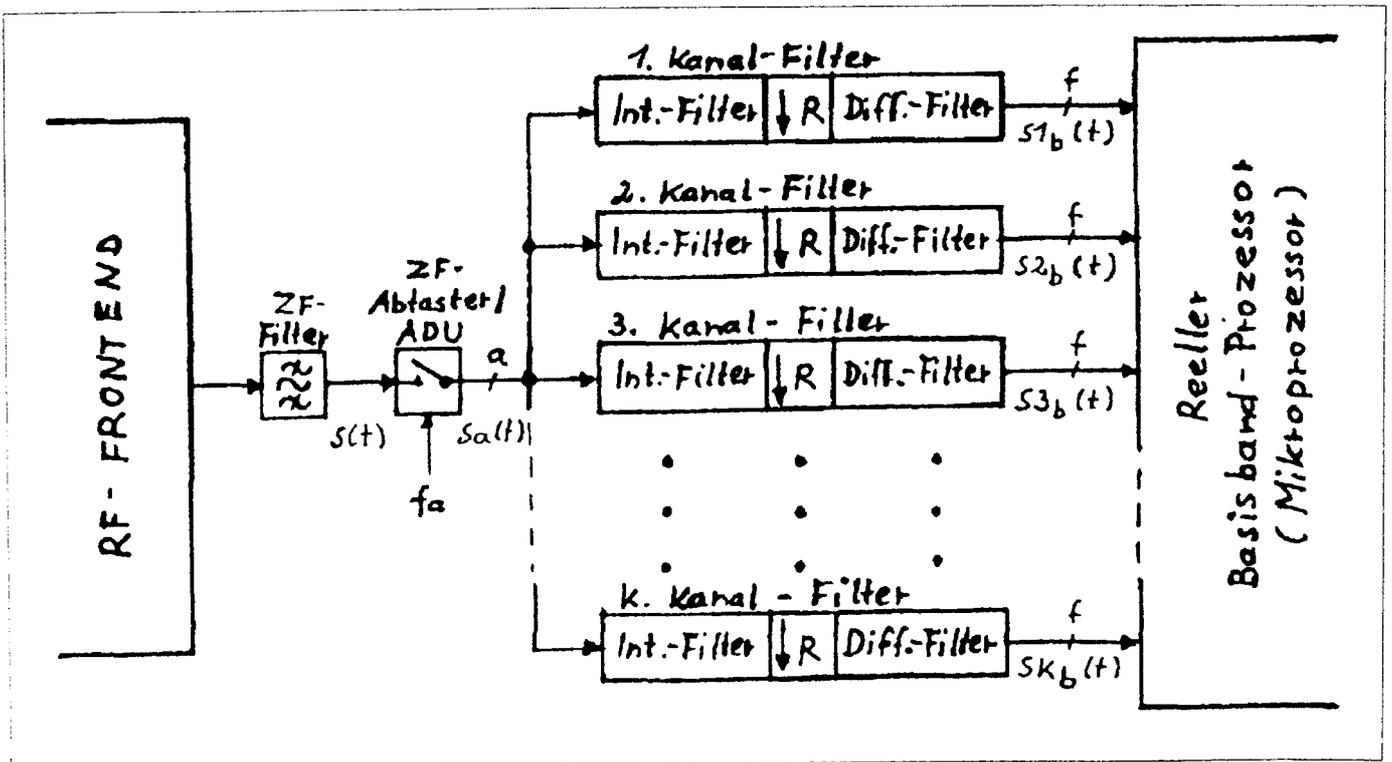


Fig.8 Digitale Mehrkanal-ZF-Stufe mit der neuen Tiefpaß-Variante

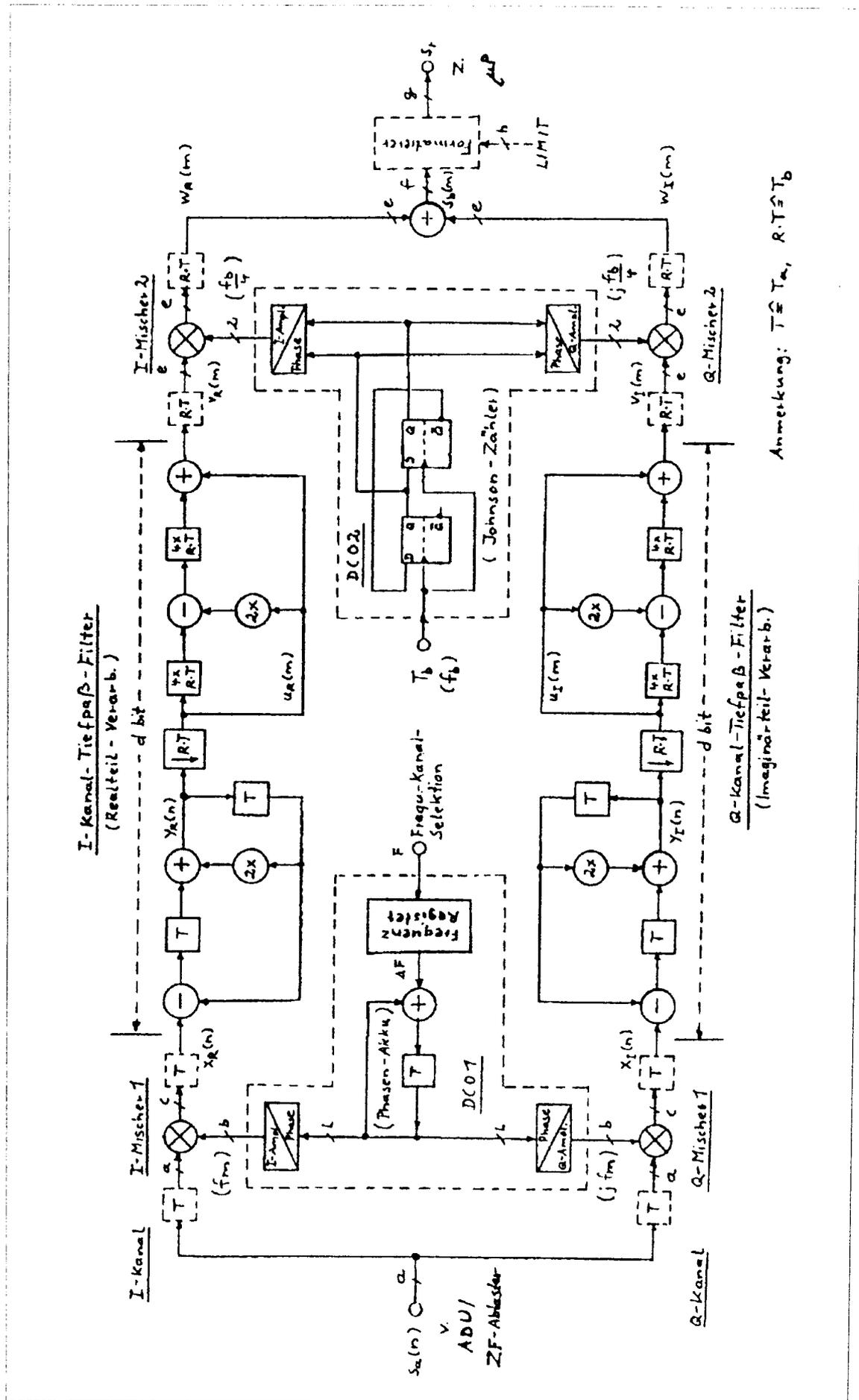


Fig.9 Ausführungsbeispiel eines ZF-Quadratur-Kanals der digitalen ZF-Stufe mit dem neuen Tiefpaßfilter

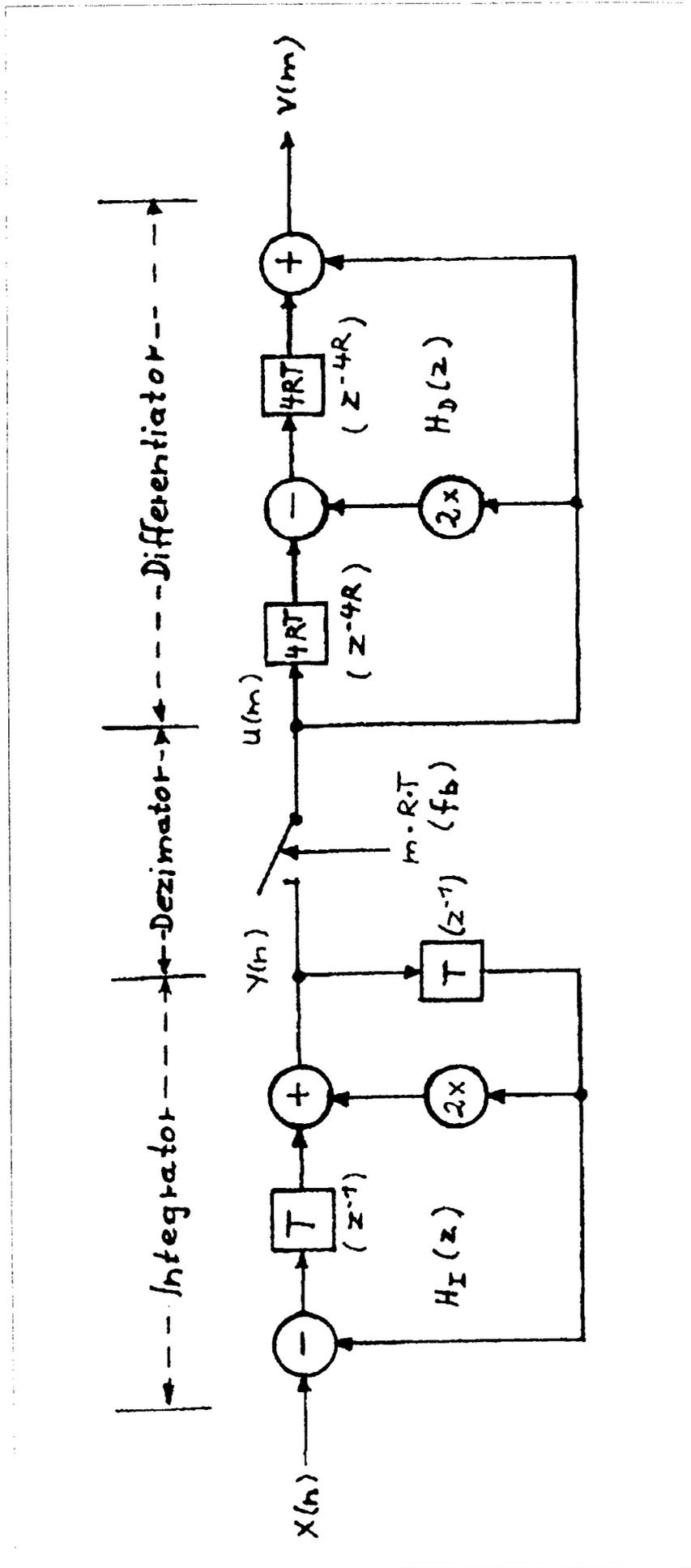


Fig.10 Funktionales Schaltbild des Tiefpaßfilters zur Filter-Dimensionierung

3-dB Bandbreite: $B_T = 2 \cdot 0.08401 f_b$

1. Nullstelle bei: $F_0 = 0.25 f_b$

1-tes Nebenzipfelmaximum von -22.6 dB bei $0.3664 f_b = 2.094 \text{ MHz}$

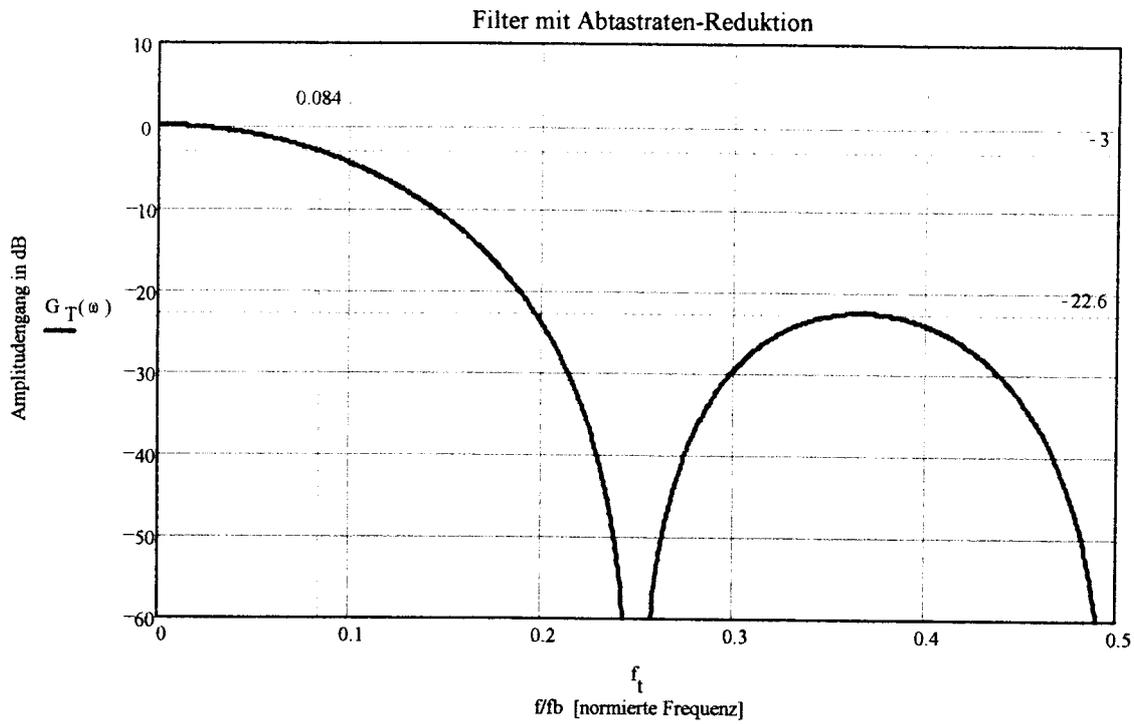


Fig. 11 Amplitudengang des Tiefpaßfilters einer Satelliten-Navigationsanwendung für GPS und GLONASS

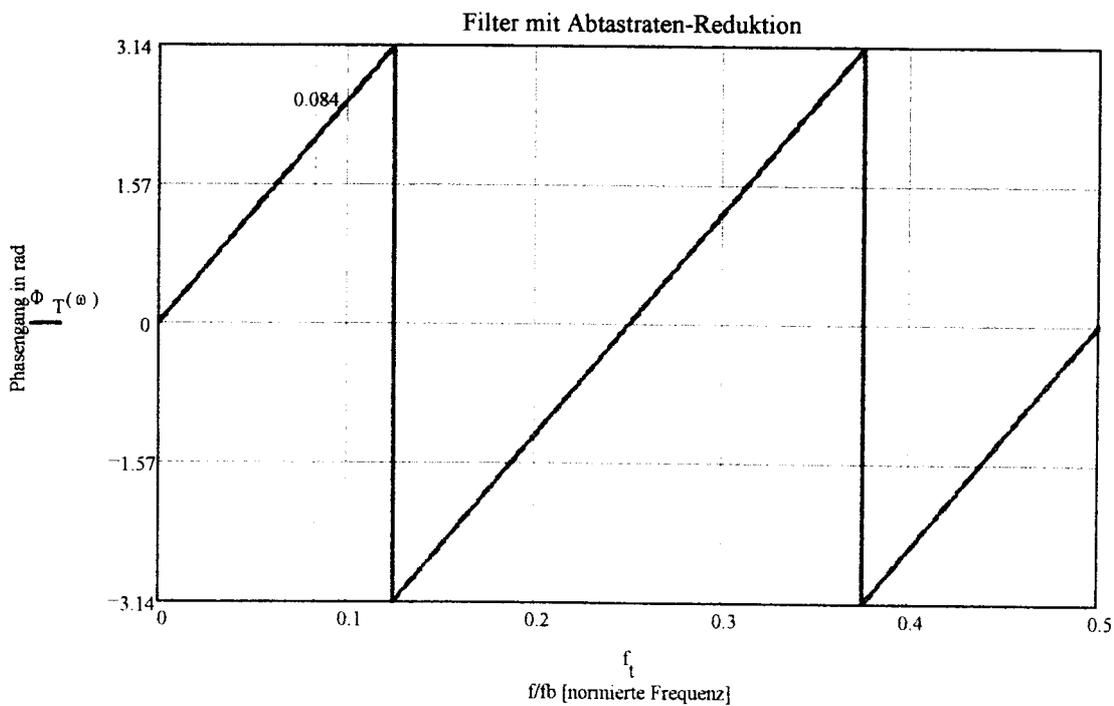


Fig. 12 Phasengang des Tiefpaßfilters einer Satelliten-Navigationsanwendung für GPS und GLONASS

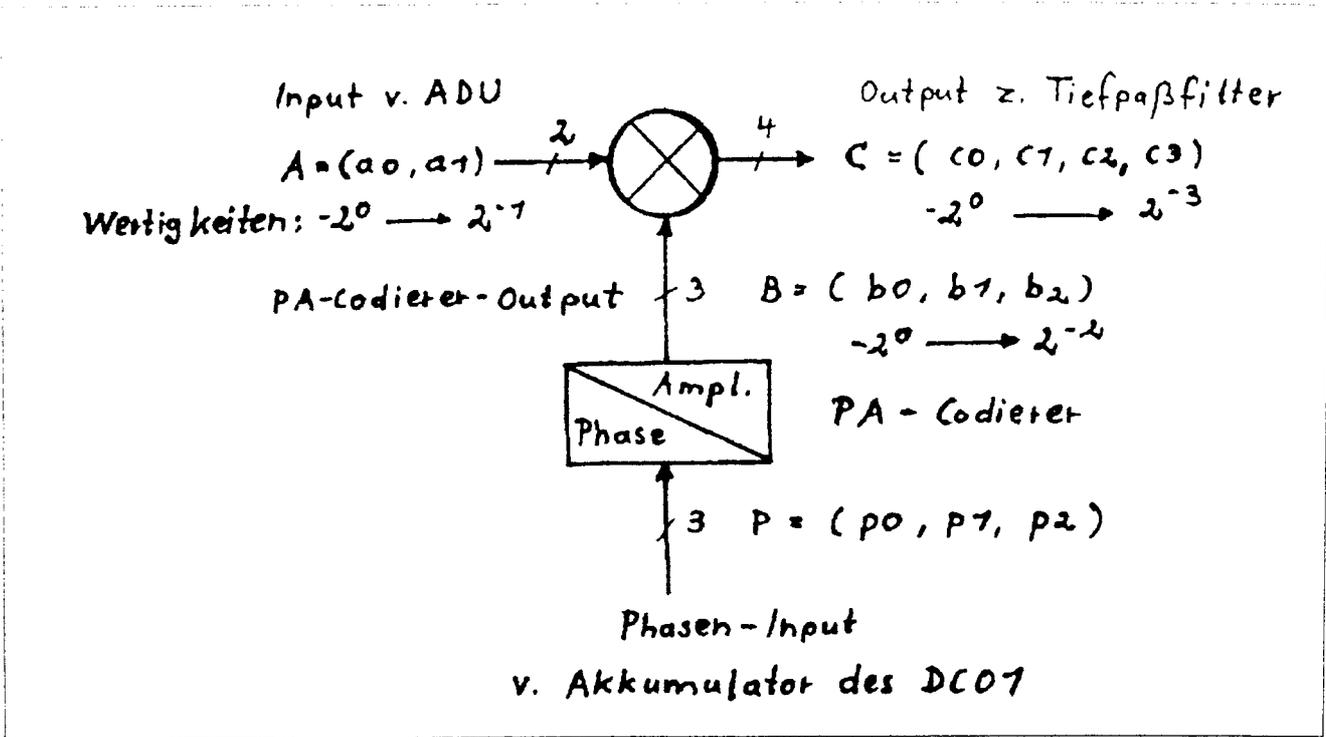


Fig.13 Zur Dimensionierung der ersten Mischstufe für eine GPS-GLONASS-Anwendung (Beispiel)

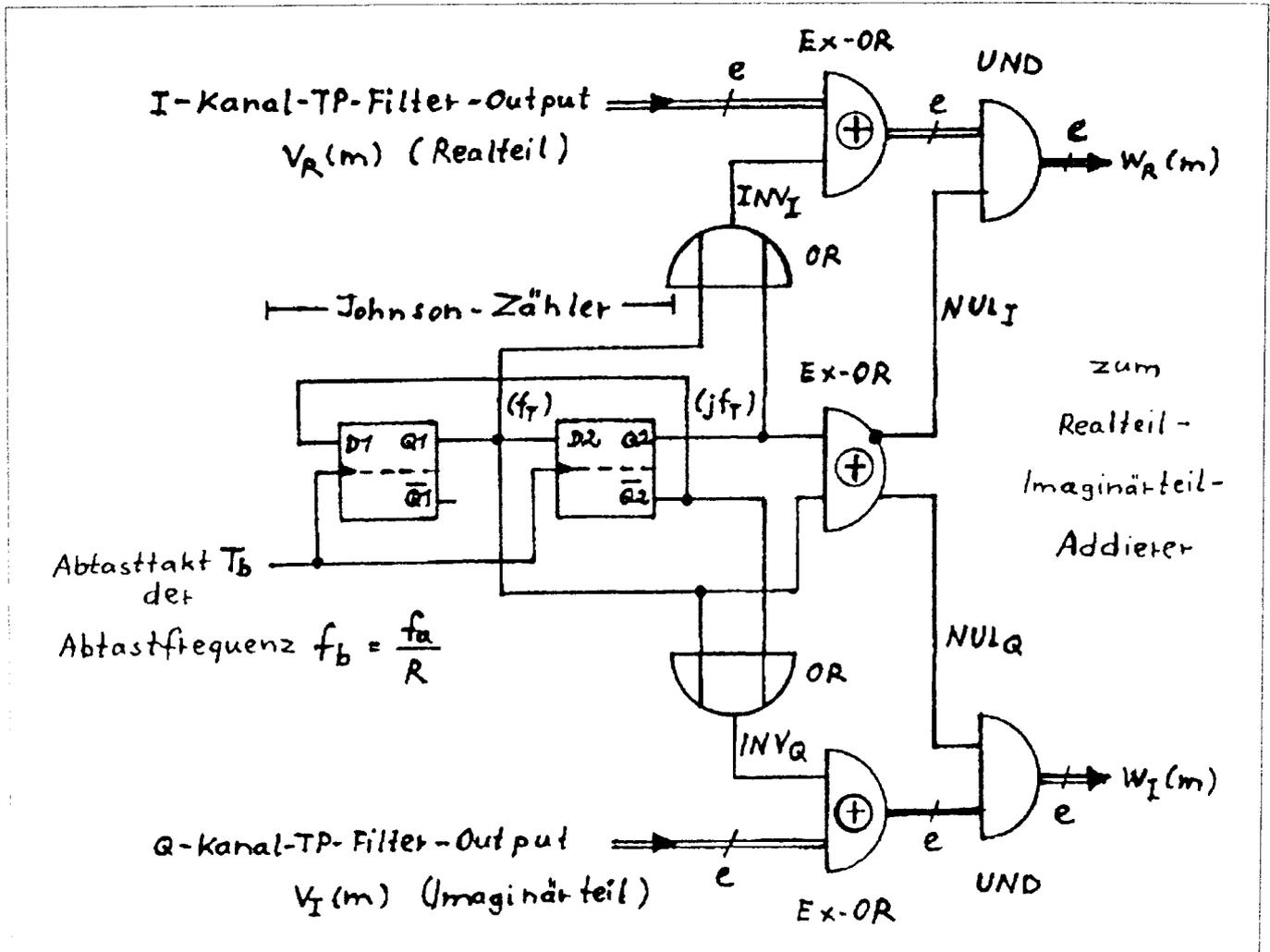


Fig.14 Vereinfachtes Schaltbild der zweiten Festfrequenz-Mischstufe

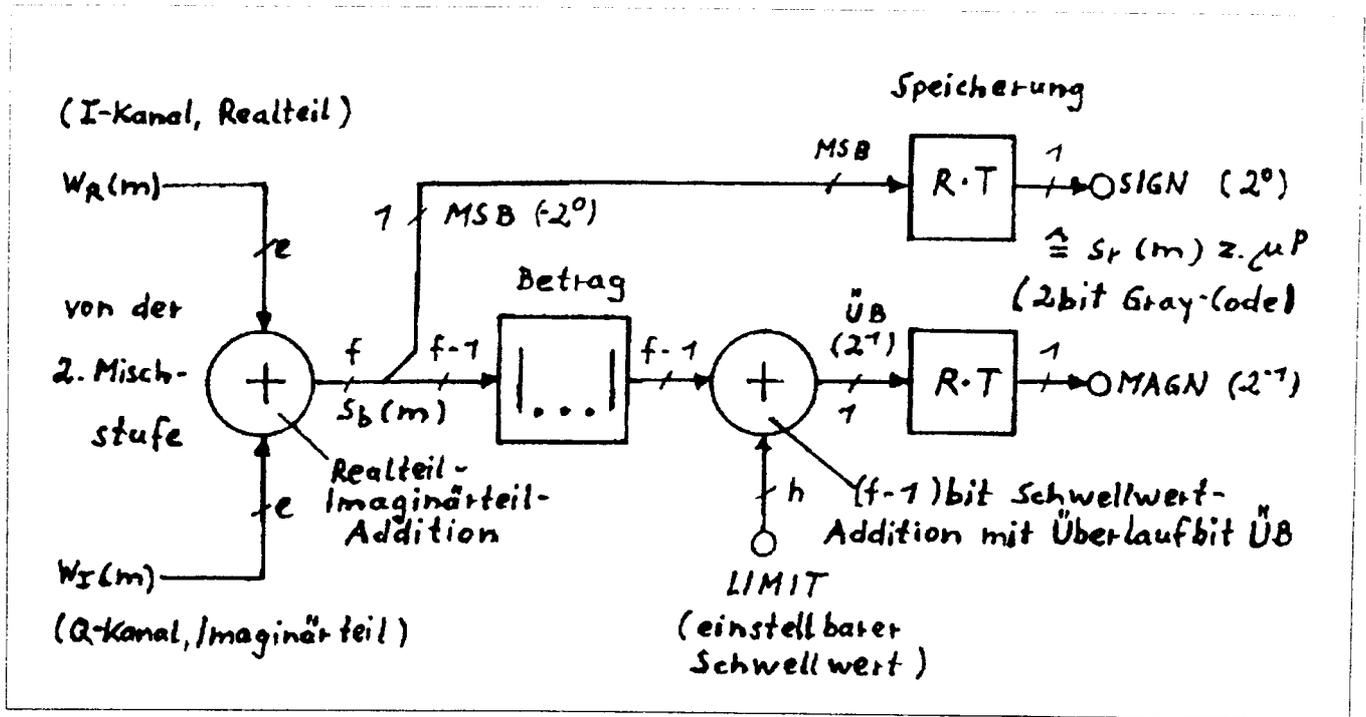


Fig.15 Funktionales Schaltbild des 2 bit Gray-Code-Formatierers für eine GPS-GLONASS-Anwendung (Beispiel)

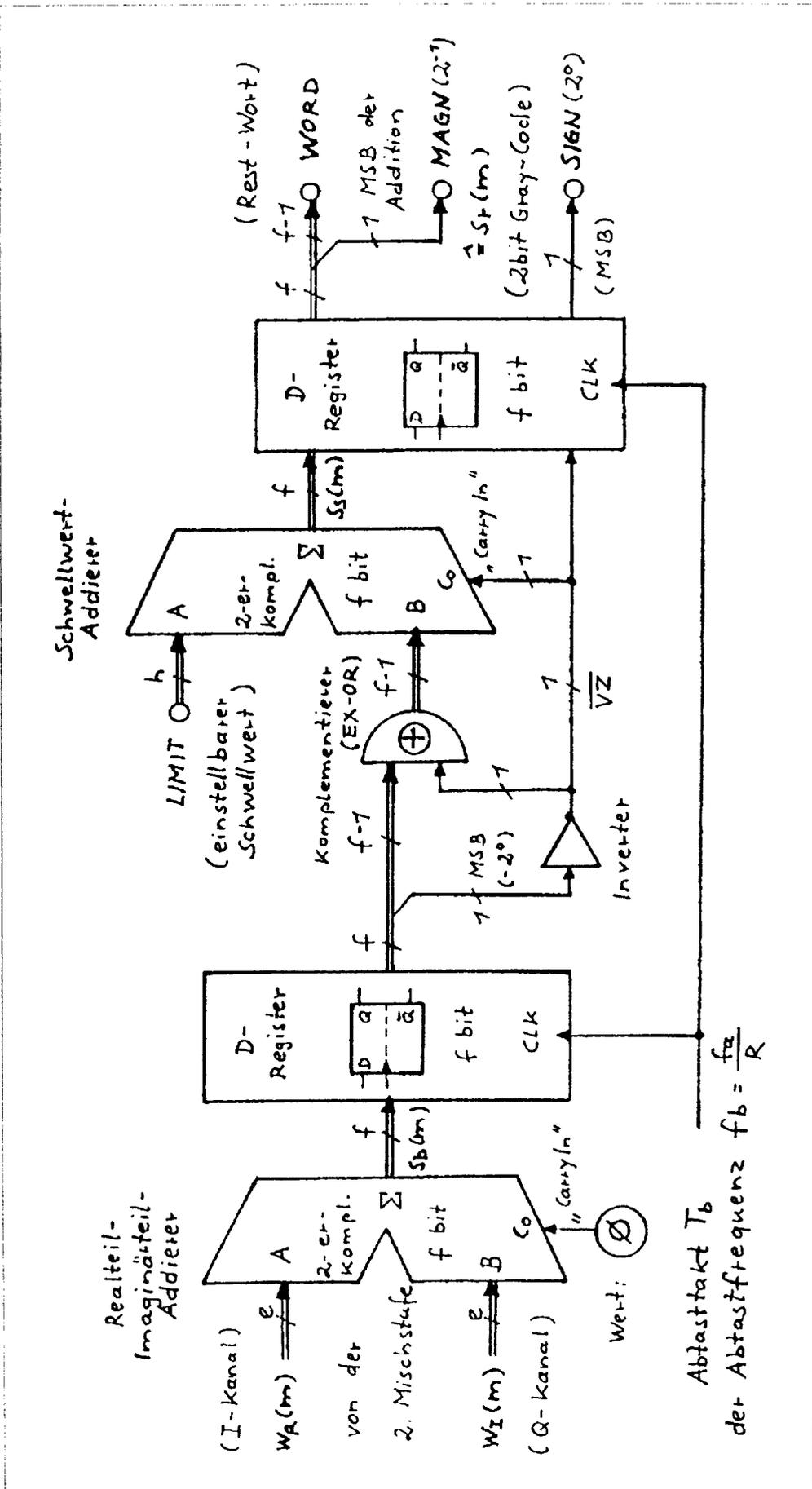


Fig.16 Vereinfachtes Schaltbild des 2 bit Gray-Code-Formaters für eine GPS-GLONASS-Anwendung (Beispiel)

Chipdesign: Mikroprozessorunterstützte Signalverarbeitung eines 512*512-Pixel-Array-CMOS-Sensors



PROF. DR.
KURT H. SCHMIDT
Prof. Schmidt lehrt Entwurf Integrierter Schaltungen im Studiengang Mikrosystemtechnik und ist Forschungsmittglied im IAF der FHF.
Tel.: 0 77 23/9 20-1 81
E-Mail:
scht@fh-furtwangen.de



DIPL.-ING.
TIBOR MILCSEVICS
Wissenschaftlicher Mitarbeiter im Studiengang Mikrosystemtechnik der FHF.
Tel.: 0 77 23/9 20-3 66
E-Mail:
mit@fh-furtwangen.de

Prof. Dr. Kurt Schmidt und Dipl.-Ing. Tibor Milcsevics

CMOS-Fotosensoren eröffnen neue Möglichkeiten in der integrierten Bildverarbeitung, z.B. [1,2]. Die verwendete CMOS-Technologie erlaubt die Integration sowohl des Sensors wie auch der umfangreichen Elektronik für die Signalverarbeitung auf dem gleichen Chip. Damit ist im Grunde ein Mikrosystem implementiert, optischer Aufnehmer und Elektronik vereint. Die Foto-CMOS-Technik wird derzeit noch weltweit in vielen Instituten verbessert.

Zunächst beginnen die Arbeiten mit einem reinen Sensorchip und einem Mikroprozessorsystem, das die Aufgabe der Elektronik übernimmt. Darüber berichtet dieser Kurzbeitrag. Später erst wird das System auf einem Chip realisiert. Eine erste Phase ist bereits begonnen worden [3] unter maßgeblicher Betreuung durch den wiss. Mitarbeiter Achim Bumüller.

Der Lichtintensitätsbereich des untersuchten Sensors Fuga 15d (512x512 pixel) von C-CAM (Belgien) ist recht groß, von 110 fW/cm² bis 8 mW/cm². Ein bei den digitalen Kameras und Camcordern bekanntes Problem („blooming“) kommt hier kaum vor. Das Spektrum reicht von 400 nm bis 1000 nm Wellenlänge. Die Empfindlichkeit der Pixel ist allerdings unterschiedlich. Das Rauschen (random noise, spatial noise) ist nicht vernachlässigbar. Außerdem kommen manchmal noch ausfallende Pixel vor, wie der Hersteller zu diesem Prototyp angibt. Mit einem Mikroprozessor kann man solche Eigenheiten individuell berücksichtigen und beispielsweise Pixel kompensieren, bzw. einzelne auch auslassen.

Anstoß der Untersuchung war eine Anwendung für einen Zeilensensor (fixed sensor with moving object albedo recognition). Im Array wird eine Zeile ausgewählt. Durch die Wahl anderer Zeilen ist sehr rasch ein Überblick über das gesamte Verhalten des Arrays möglich. Die Prozessorauf-

gaben werden mit einem schnellen 16-Bit Mikrocontroller SAB80C167 vorgenommen (Bild 1). Die Messergebnisse werden auf einem PC-System visuell dargestellt. Die Verbindung zwischen Sensor und Prozessor ist 16 Bit breit und sehr schnell (250 ns). Der PC und der Controller kommunizieren über die serielle Schnittstelle. Für die grafische Darstellung der Ergebnisse spielt die Übertragungsgeschwindigkeit keine Rolle.

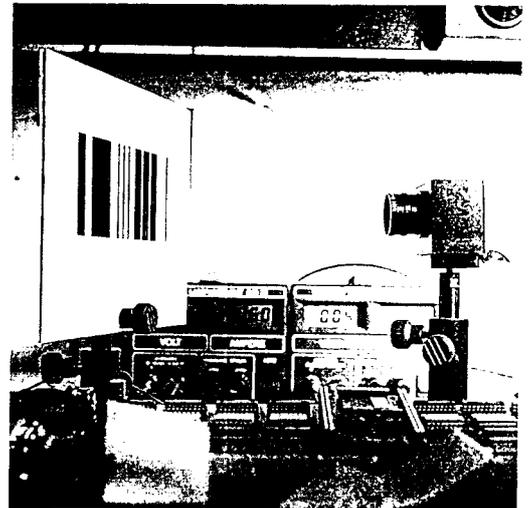


Bild 1: Messanordnung mit Sensor und Mikrocontroller

Der CMOS-Sensor besitzt random access. Die Adressierung eines Pixels braucht 90 ns, der integrierte 8-Bit A/D-Wandler braucht 160 ns. Fortlaufende Adressierung ermöglicht ein schnelleres Lesen, die Anfangsadresse der Zeile genügt. Eine Begrenzung erfolgt hier durch das Mikrocontrollersystem, weil die minimale Befehlszykluszeit wegen der langsameren Speicher nur 250 ns beträgt. Wiederholtes Pixel-Lesen benötigt eine Wartezeit. Auswertung: Eine Zeile wird 17-mal automatisch schnellstmöglich gemessen. Die erste Zeile wird nicht gewertet, es wird ein Pixel-bezogener Intensitätsmittelwert aus den nachfolgenden 16

Messungen gebildet. Das mindert den Rausch- einfluss [2]. Die Varianz betrug $\pm 2\%$ bei einer mittleren Intensität. Dann werden die Werte rechnerisch kompensiert mit einem zuvor ge- messenen, schon gespeicherten Korrekturwert des Pixels (Subtraktion). Das Ergebnis ist eine verbesserte Glättung der Pixelwerte. Aufgetra- gen ist Intensität über Pixelzeile (Bild 2).

Danksagung:

Die Verfasser danken Volker Lange für seine Un- terstützung bei der optischen Messanordnung, Andreas Gauckler für seine Hilfe bei der Reali- sierung der Konverterplatine, dem IAF der FHF, wo das Projekt angesiedelt wurde und der MPC- Gruppe und damit dem Landesministerium für erhaltene Mittel.

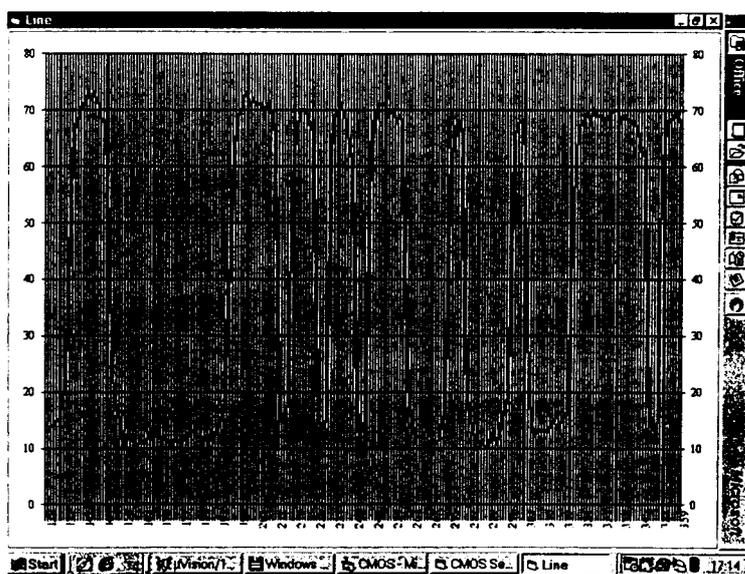


Bild 2: Abbildung des Testbildes (s. Bild 1)

Die letzte Phase der Korrektur ist das Ersetzen der ausgefallenen Pixel oder Ausreißer, die eventuell vorhanden sind. Diese bleiben un- berücksichtigt und werden durch den Interpolationswert aus den Nachbarpixeln ersetzt.

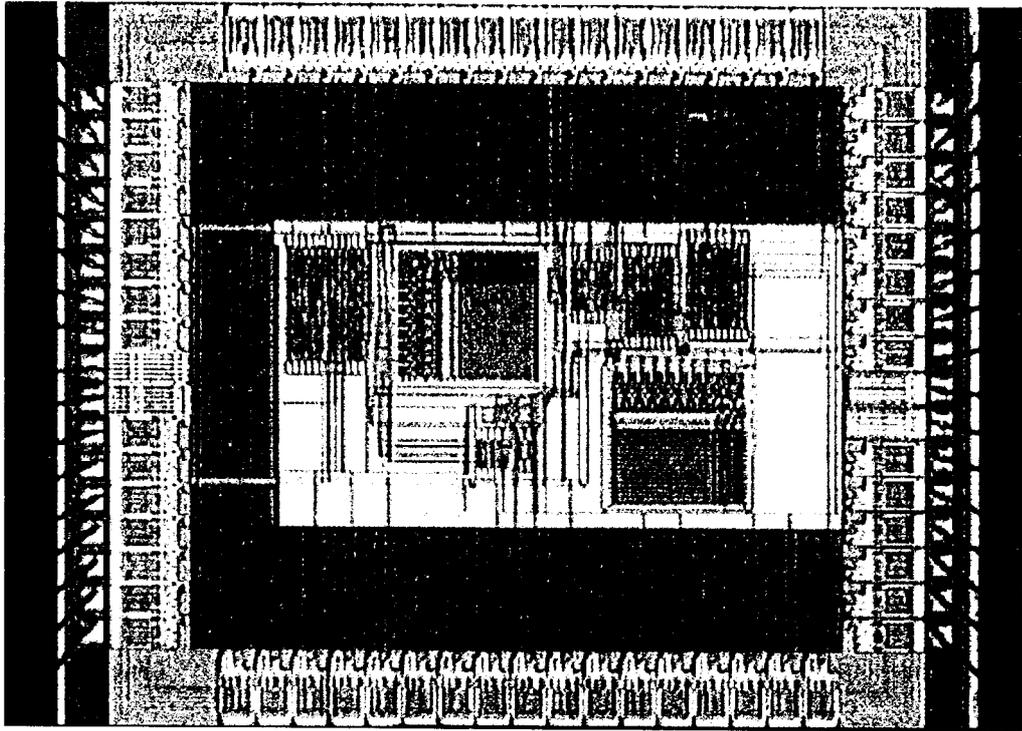
Zusammenfassung:

Das CMOS-Image-Array mit einem Prozessor- Kernel kann ein ideales Paar für die Fotosensor- auswertung bilden. Über EURORACTICE ist das Fotopixel als Zelle möglich, die zu beliebi- gen Arrays angeordnet werden können. System- on-chip Lösungen, mit flexiblen Algorithmen versehen, bieten vielseitige Anwendungsmög- lichkeiten.

Literatur:

- [1] DECKER, McGRATH, BREHMER, G. SOLDINI: A 256x256 CMOS Imaging Array with Wide Dynamic Range Pixels and Column-Parallel Output, IEEE Journal of Solid-State Cir- cuits, Vol.33, Dec.1998, pp 2081–2091
- [2] LOINAZ, SINGH, BLANKSBY, INGLIS, AZADET, ACKLAND: A 200 mW, 3.3V, CMOS Color Ca- mera IC Producing 352x288 24-b Video at 30 Frames/s, IEEE Journal of Solid-State Circuits, Vol.33, Dec.1998, pp 2092–2102
- [3] ADLER, ESCHER: Entwicklung eines Photo- zeilensensor-Kernels auf einem IC, Diplom- arbeit, FH-Furtwangen, Mikrosystem- technik, SS 1999.

Interface-Chip für Photozeilensensor (fuga_io)



- Entwurf: Fachhochschule Furtwangen
Bearbeiter: Andreas Adler, Patrick Escher
Diplomarbeit Studiengang Mikrosystemtechnik
Betreuer: Prof. Dr. Kurt Schmidt und Dipl.-Ing. Achim Bumüller
- Layouerstellung: Fachhochschule Furtwangen (Standardzellenentwurf)
- Technology: Alcatel Mietec 0.5µm CMOS C05M-D
- Chipfertigung: Alcatel Mietec über Europractice
- Herstelldatum: September 1999
- Kostenträger: MPC-Gruppe Baden-Württemberg
- Chipdaten: Chipgröße: 3.8 * 3.6 mm²
Gehäuse: JLC68
Komplexität: 850 Zellen + 2 RAMS (256 x 8)
- Funktion: Der Interface Chip mit zwei integrierten RAMs steuert den Photozeilensensor FUGA an und übernimmt in jeweils ein RAM 256 acht bit breite Daten. Wird RAM1 geschrieben, kann RAM2 ausgelesen werden. Wenn die Daten von einem auswertenden System (in diesem Fall von einem noch externen Mikrocomputersystem) ausgelesen sind, wird in RAM1 geschrieben und RAM2 wird ausgelesen. Der Interface Chip führt automatisch Selbsttests der einzelnen Baugruppen durch.
- Testergebnisse: Die RAMs arbeiten korrekt. Das ergibt auch der Built-in RAM Test. Der Sequenzer arbeitet ebenfalls korrekt. Ein Ausgangssignal hat aber ein stuck-at-1 Fehler. Von den 10 erhaltenen Chips arbeiten 9 wie beschrieben, ein Chip ist defekt.

MISP: Tastatur Schnittstellen Prozessor

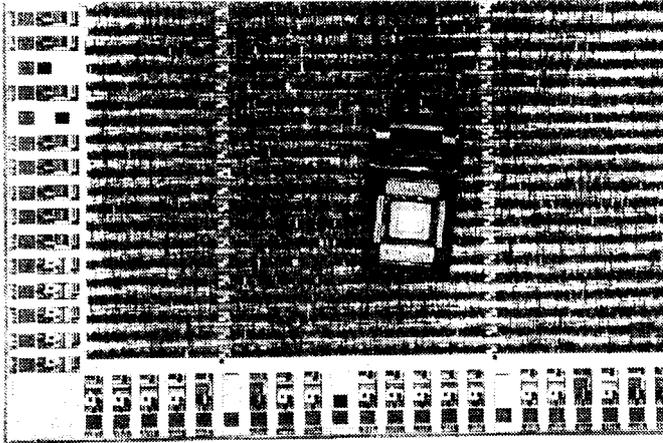


Abb. 1 Layout und Chip im Testsockel

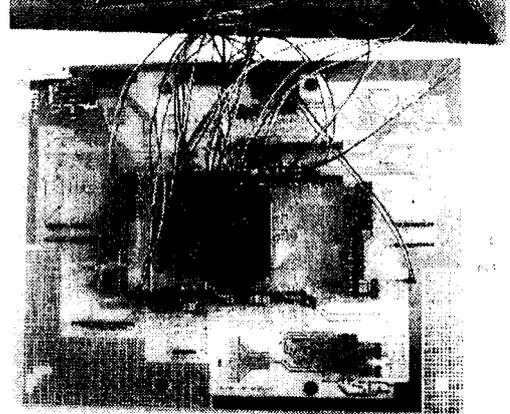
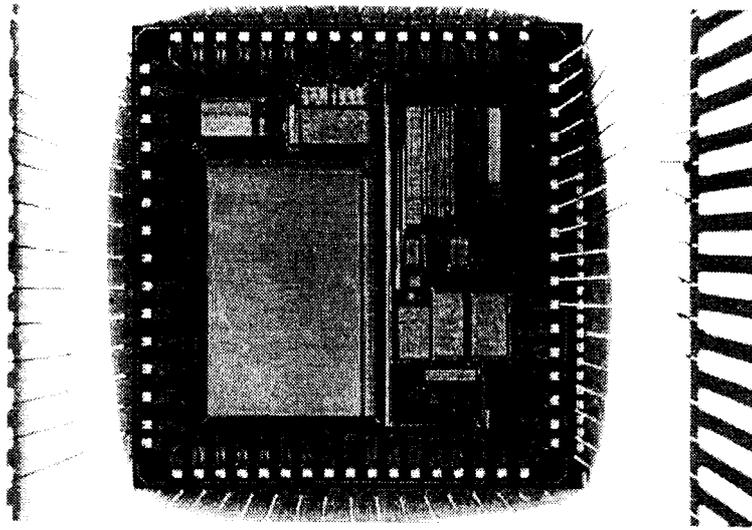


Abb. 2: Test- und Experimentierumgebung

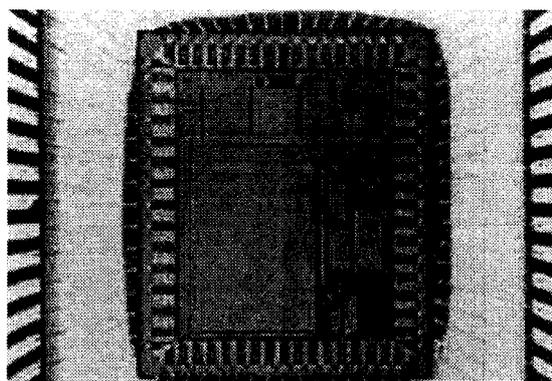
- Entwurf: Fachhochschule Furtwangen
Bearbeiter: Tobias Strauch, Andreas Gauckler
Betreuer: Dipl.Ing. Andreas Gauckler, Prof. Dr. W. Rölling
- Layout: Fachhochschule Furtwangen (Cadence)
Technologie: AMS 0.8 μ CMOS CXQ,
Chipfertigung: EuroPractice Run 402 /März 2000
- Kostenträger: MPC-Mittel FH - Verbund Baden-Württemberg
- Chipdaten: Chipdaten: 54,1 mm² (7,3mm Kantenlänge)
Cells: 15700 (INV,NANDS,FF ...)
Gehäuse: CQFP160
- Funktion: Der MISP ist ein Spezialprozessor für den Einsatz in Computertastaturen und ersetzt dort diskrete Logik, ASICs und Prozessoren. Er übernimmt die Kontrolle und Steuerung von MatrixScan, PS2-Schnittstelle, MagnetKarten, SmartCard und BarCode Einheiten. Der Chip wurde in VHDL entworfen und mit Synopsys synthetisiert, auf FPGAs (Xilinx) verifiziert, und mit Cadence wurde das Layout erstellt und überprüft. Eine Hauptschwierigkeit bestand aufgrund der Gatteranzahl darin, mit den verwendeten Tools einen ClockSkew optimierten Entwurf zu generieren, hier kam Primetime zum Einsatz.
- Testergebnisse: Die Grundfunktionen des MISP-Prozessors wurden auf dem hp82000/100MHz getestet. Das Zusammenspiel mit den externen Komponenten (keyboard, barcode, smartcard ...) wurde in einer Experimentierumgebung nachgewiesen. Aktuell wird eine bisher eingesetzte Tastaturplatine geändert um den Chip im System austesten zu können.

Thermologger_V3



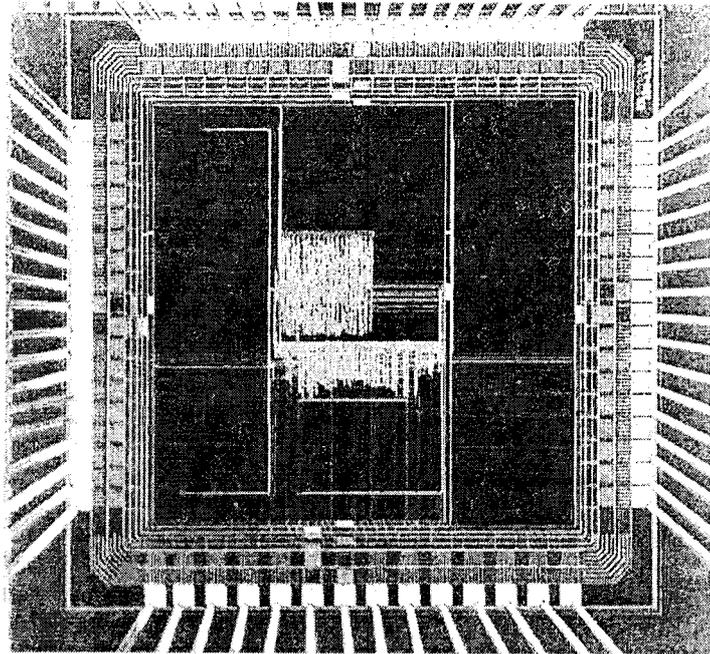
- Entwurf: Fachhochschule Offenburg
Bearbeiter: Markus Fischer
Carsten Störk
Jürgen Hauser
Betreuer: Prof. Dr. - Ing. Dirk Jansen
- Layouterstellung: Fachhochschule Offenburg (Mixed-Signal-Entwurf)
- Technologie: Alcatel Mietec 0.5 μ m CMOS-AD
- Chipfertigung: Europractice, Run 384
- Herstelldatum: Oktober 1999
- Kostenträger: MPC-Mittel FH-Verbund Baden_Württemberg
- Chipdaten: Chipgröße: 4,8 x 5,0 mm²
Chipgehäuse: JLCC 68
Komplexität: ca 40000 Transistoren
- Funktion: Der Chip enthält einen Temperatursensor, den FHOP-Prozessorkern, ein 8kB RAM, BIOS-ROM, PLL sowie Ein- und Ausgabeeinheiten. Der Chip soll die Temperatur aufzeichnen und in das RAM ablegen. Für die Kommunikation stehen 2 Parallel-Ports, ein Seriell-Port sowie die PSK-Schnittstelle zur Verfügung. Der PLL erzeugt den Takt für die PSk-Schnittstelle. Die Steuerung der Gesamtfunktion erfolgt mit dem FHOP. Das Programm hierfür ist im ROM abgelegt. Es kann aber auch noch zusätzlicher Code in das RAM abgelegt werden, sodass der Chip ein komplettes Controllersystem darstellt.
- Testergebnisse: Der Chip zeigte keine Funktion. Es wurde festgestellt, dass die Synthese mit Synopsys die Treiber des Taktsignals wegrationalisiert hat. Dadurch war die Treiberleistung für das Taktsignal zu gering. Es wurde kein gültiges Taktsignal an die Flip-Flops ausgegeben.

Thermologger_V4



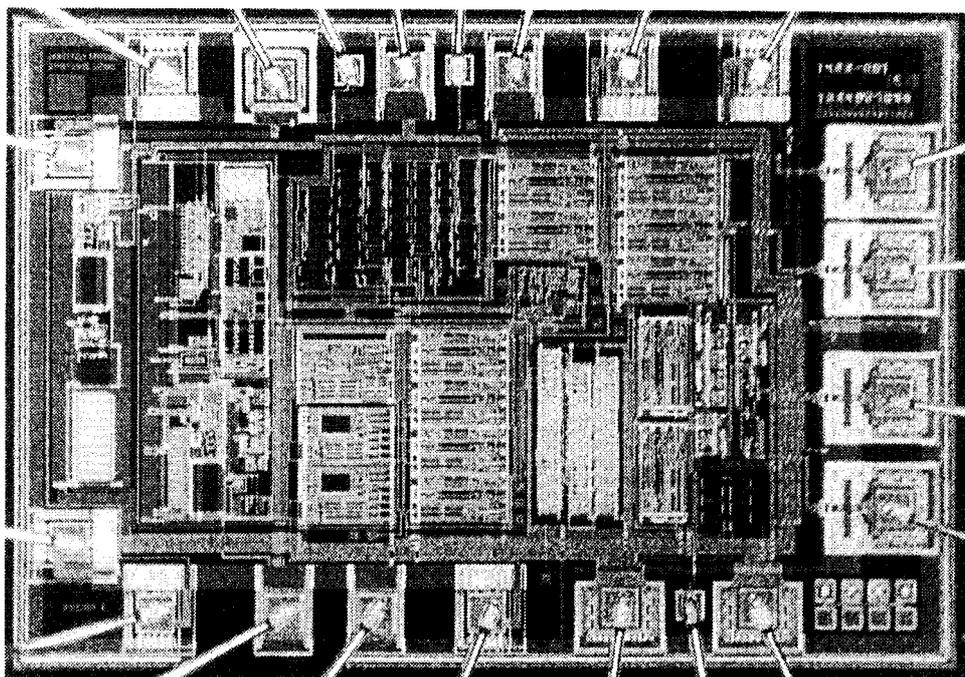
Entwurf:	Fachhochschule Offenburg Bearbeiter: Markus Fischer Carsten Störk Jürgen Hauser Betreuer: Prof. Dr. - Ing. Dirk Jansen
Layouterstellung:	Fachhochschule Offenburg (Mixed-Signal-Entwurf)
Technologie:	Alcatel Mietec 0.5µm CMOS-AD
Chipfertigung:	Europractice, Run 475
Herstelldatum:	Juni 2000
Kostenträger:	MPC-Mittel FH-Verbund Baden_Württemberg
Chipdaten:	Chipgröße: 4,5 x 5,3 mm ² Chipgehäuse: JLCC 68 Komplexität: ca 40000 Transistoren
Funktion:	Der Chip enthält einen Temperatursensor, den FHOP-Prozessorkern, ein 8kB RAM, BIOS-ROM, PLL sowie Ein- und Ausgabeeinheiten. Der Chip soll die Temperatur aufzeichnen und in das RAM ablegen. Für die Kommunikation stehen 2 Parallel-Ports, ein Seriell-Port sowie die PSK-Schnittstelle zur Verfügung. Der PLL erzeugt den Takt für die PSK-Schnittstelle. Die Steuerung der Gesamtfunktion erfolgt mit dem FHOP. Das Programm hierfür ist im ROM abgelegt. Es kann aber auch noch zusätzlicher Code in das RAM abgelegt werden, sodass der Chip ein komplettes Controllersystem darstellt. Die Taktumschaltung zwischen 32kHz und 11MHz wurde im Vergleich zur Version 3 weiterentwickelt, sodass nun jede Komponente getrennt einen Takt erhält.
Testergebnisse:	Der Chip zeigte keine Funktion. Nach langer Fehlersuche wurde festgestellt, dass beim Routing Kurzschlüsse entstanden sind, die wegen des hierarchischen Designs von keinem Test (LVS, ERC) festgestellt wurden. Es werden nun Untersuchungen angestellt, wie dieses Problem behoben werden kann.

Uhrenmakro mit Selbsttest



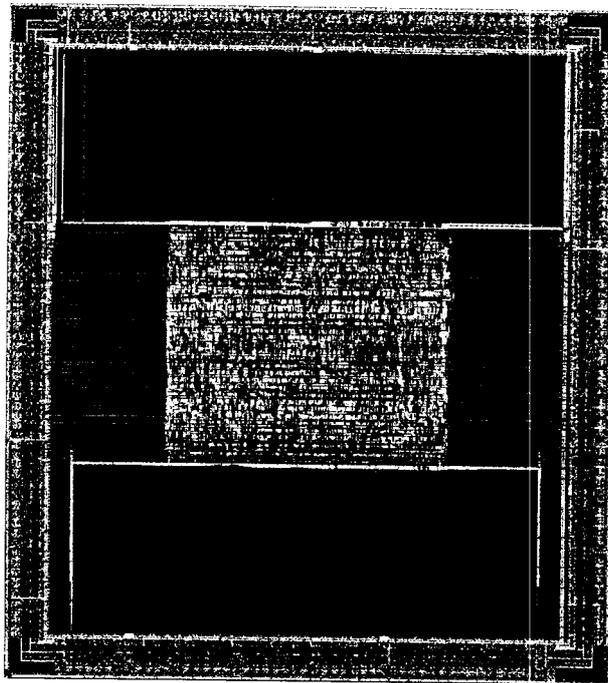
- Entwurf: Fachhochschule Ulm
Bearbeiter: Hans-Jürgen Jahn
Betreuer: Prof. Arnold Führer
- Layouterstellung: Fachhochschule Ulm
Digitaler Standardzellenentwurf
- Technologie: 0,35 μm CMOS mit 2 Lagen Poly und 3 Lagen Metal, Fa. AMS „csd“-Prozeß
- Chipfertigung: Fa. Austria Mikro Systeme (AMS), über Europractice
- Herstelldatum: 4. Quartal 2000
- Kostenträger: MPC-Gruppe Baden-Württemberg
- Chipdaten: Chipfläche: 507 x 442 μm^2 , Gatter: 840
Applikationsschaltung: 2,7 x 2,8 mm^2
- Funktion: Die abgebildete Anwenderschaltung enthält ein wiederverwendbares Uhrenmakro (IP) mit folgenden Funktionen:
Stellen der Uhr durch paralleles Laden oder durch Hochzählen der Sekunden, Minuten usw.
Parallele Ausgabe von Sekunden, Minuten, Stunden, Wochentag und Datum im BCD-Format.
Built-in-Selbsttest.
- Eine ausführliche Beschreibung ist im Workshopband Juli 2000 enthalten und in der Homepage des Labors Schaltungsintegration:
www.asic.lab.fh-ulm.de/chip_prj.htm

Laderegler für Solarsysteme



- Entwurf: Fachhochschule Ulm
Bearbeiter: Martin Barchet, Oliver Salzmann
Betreuer: Prof. Dipl.-Phys. Gerhard Forster
- Layouterstellung: Fachhochschule Ulm (Mixed Signal-Entwurf)
Analog-Teil: Standardzellen + Full Custom Design
Digital-Teil: Standardzellen
- Technologie: CXZ 0,8 μm CMOS A/D High Voltage, Fa. AMS
- Chipfertigung: Fa. AMS, Österreich, über Europractice
- Herstelldatum: III. Quartal 2000
- Kostenträger: MPC-Gruppe Baden-Württemberg
- Chipdaten: Chipfläche: 2,5 x 1,7 mm²
Gehäuse: LCC 44
Funktionsblöcke: Analogteil: Spannungsregler, 11 Komparatoren, 3 OPV, PWM
Digitalteil: ca. 150 Gatter, 4 Treiber
- Funktion: Die Energieausbeute in einem Solarsystem, bestehend aus Solarpanel, Blei-Akku und Verbraucher, wird optimiert und der Akku (12 V oder 24 V Nennspannung) in einem hinsichtlich der Lebensdauer günstigen Betriebszustand gehalten. Die Ladeendspannung ist abhängig von der Vorgeschichte und der Temperatur. Weitere Merkmale des ASICs sind u.a. der stromgeführte Tiefentladeschutz und die elektronische Sicherung.

Hough Transform ASIC



Design: Fachhochschule Mannheim
Designer: Avi Epstein
Mentor: Prof. Dr.-Ing. Gerd-Uwe Paul

Process AMS 0.6 μ CMOS (CUB)
Waferfab. FHG IIS, Run 399, Feb. 2000
Dimensions 6.8 x 7.4 mm
Frequency 100 MHz
Array size 32 Columns, 16 Rows
Package CPGA256

Function: The ASIC implementation of the Hough-Transform as a systolic array for real-time recognition of curved tracks in multi-wire drift chambers is presented. The array can handle 32 parallel input data streams. It mainly consists of 512 identical programmable processing elements. 16 histogram pixels in the feature space are produced in parallel per clock cycle. The interconnectivity pattern of the processing elements required to initialise the chip according to the pattern recognition task is computed on the host computer using the Hough-Transform equations. This pattern is then downloaded to the chip via the data input lines. The Hough-Transform ASIC is suitable for a wide range of pattern recognition applications. The integrated Hough-Transform circuit will allow the real-time execution of this transform with an increase in speed in the range of two orders of magnitude compared to previous implementations.