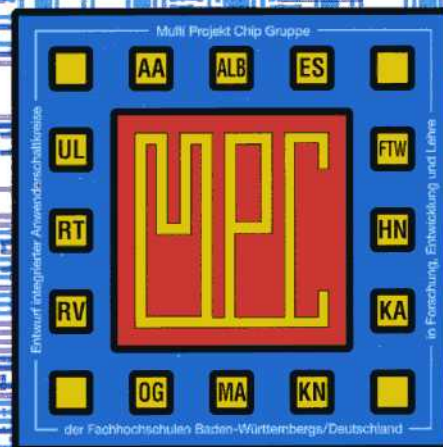


MULTIPROJEKT CHIP-GRUPPE

BADEN - WÜRTTEMBERG

Workshop Juli 1998

Konstanz



MULTIPROJEKT CHIP-GRUPPE

BADEN - WÜRTTEMBERG

Workshop Juli 1998

Konstanz

Herausgeber: Fachhochschule Ulm

© 1998 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, Postfach 100543, 78405 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 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. Mikroprozessor-Entwicklung in Deutschland: Der hyperstone 32-bit RISC/DSP Prozessor E1-32X M. Dolle, Fa. hyperstone electronics, Konstanz	5
2. Semicustom Highway Ein auf VHDL basierender Design flow bei Siemens HL A. Schütz, Fa. Siemens, München	19
3. Loop Gain mit PSPICE H. Nielinger, FH Furtwangen	27
4. Feldnumerische Simulation kreiszylindrischer Strukturen bei hohen Frequenzen F. Hamann, A. Christ, FH Offenburg	37
5. ATM Messapplikationen in programmierbarer Logik Beschreibung geeigneter Verfahren und Umsetzung in ein FPGA F. Stockmayer, FH Reutlingen	43
6. Ein schneller Weg zum IC-Layout G. Voland, FH Konstanz	55
7. Temperaturzelle in CMOS-Technologie mit Sigma-Delta-Wandler auf der Basis eines Entwurfs der Universität Delft/Ne J. Hauser, W. Vollmer, D. Jansen, FH Offenburg	61
8. Aktuelle Konzepte zum Entwurf von Temperatursensoren G. Albert, FH Mannheim	67

hyperstone electronics

Mikroprozessor-Entwicklung in Deutschland:
Der hyperstone 32-bit RISC/DSP Prozessor E1-32X



<http://www.hyperstone.com>

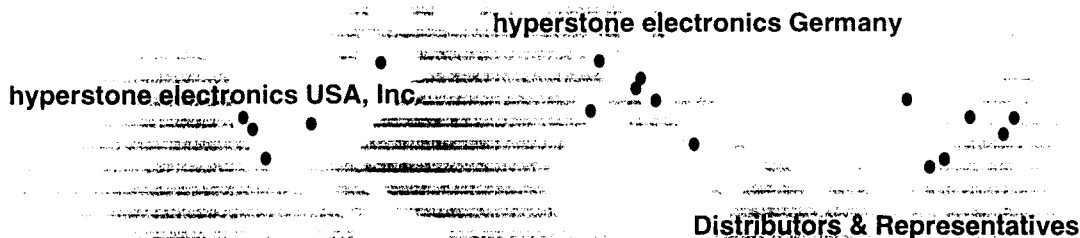
Unifying RISC and DSP



- Fabless microprocessor company
- 16 staff members
- Development of 32-bit RISC/DSP microprocessors
- Development Tools, software
- Direct chip sales
- IP Business (Licensing)
- System house, Projects
- ASICs, ASSPs
- Worldwide partnerships
 - Distributors / Reps
 - Engineering
 - 3rd Parties

07-98

Unifying RISC and DSP



07-98

hyperstone E1-32X

Architecture

07-98

Unifying RISC and DSP

RISC Microprocessors

- Load/Store architecture
- Register oriented
- On-chip caches
- Simple instructions
- Single-cycle instructions
- General purpose units
- High clock frequencies
- General purpose processing

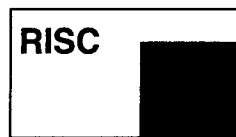
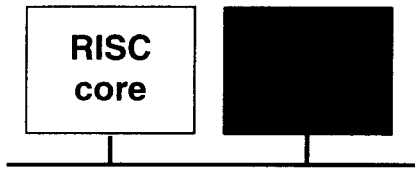
DSP Processors

- Specialized architecture
- Memory oriented
- on-chip RAM, ROM
- Complex instructions
- Multi-cycle instructions
- Dedicated units
- Limited clock frequency
- Dedicated application area

**Combine the advantages of a general purpose
32-bit microprocessor and a fast DSP unit !**

07-98

Unifying RISC and DSP



- Two-core solution**
 - Microcontroller core plus one or more DSP cores
 - Doubled resources
 - Data bottlenecks
 - TI C80, ARM7+Oak/Pine...
- Single-core solution**
 - One single core
 - Reduced cost
 - Unified instruction stream
 - Simplified programming
 - hyperstone E1-32, SH-3, NEC V830, UltraSparc ...

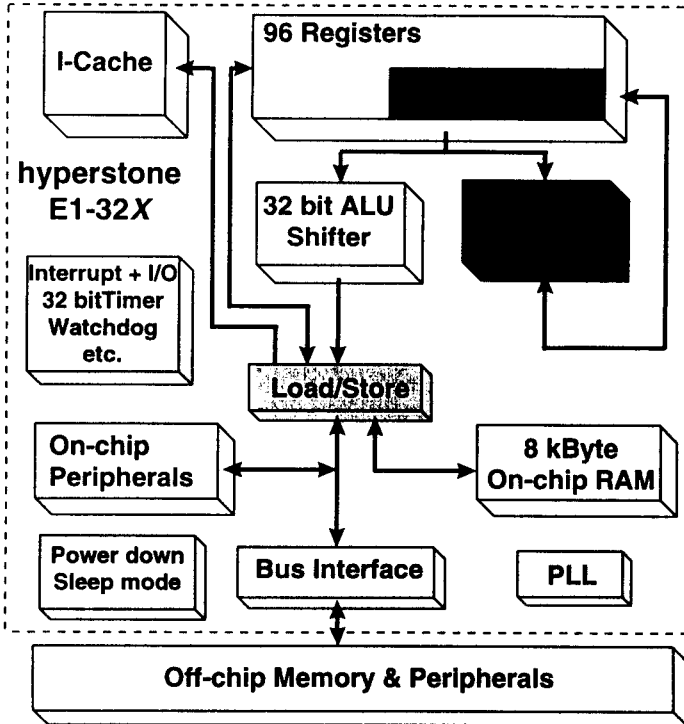
07-98

Unifying RISC and DSP

- RISC Load/Store architecture**
- Integrated DSP Unit for calculating arithmetic DSP expressions**
- Subword processing**
 - 2x16 bit data in one 32 bit registers**
- Deterministic program flow**
- Fast loop processing with simple instructions**
- Fast execution of DSP algorithms**
- High sample rate**
- Goals: Higher DSP performance & reduced cost**

07-98

Unifying RISC and DSP

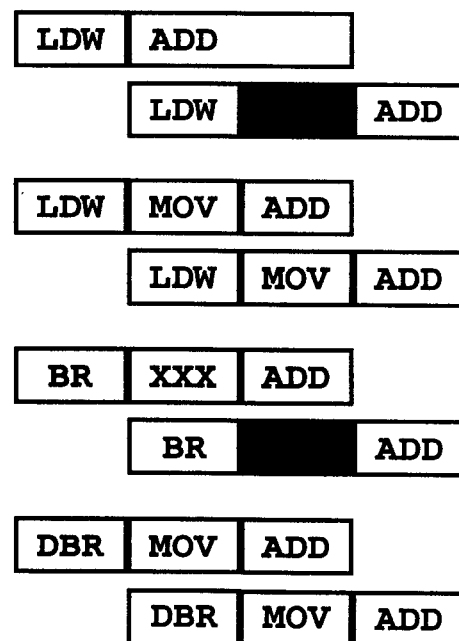


07-98

- Two stage pipeline
- Variable instruction length
- Large register file
- 32-bit single-cycle ALU
- 16 / 32-bit fixed-point DSP
- 8 KByte on-chip RAM
- 32-bit timer
- Low power consumption
- Comprehensive bus interface
- Simple system design

Unifying RISC and DSP

- Two-Stage Pipeline**
 - Decode stage**
 - ✓ Read instruction from cache
 - ✓ Decode instruction
 - Execute stage**
 - ✓ Read source register
 - ✓ Calculate result
 - ✓ Write to destination register
- Advantages:**
 - Fast pipeline refill after branches
 - Single-cycle "delayed branch"
 - Simplified hardware design
 - ✓ Simple control logic
 - ✓ No branch prediction required



07-98

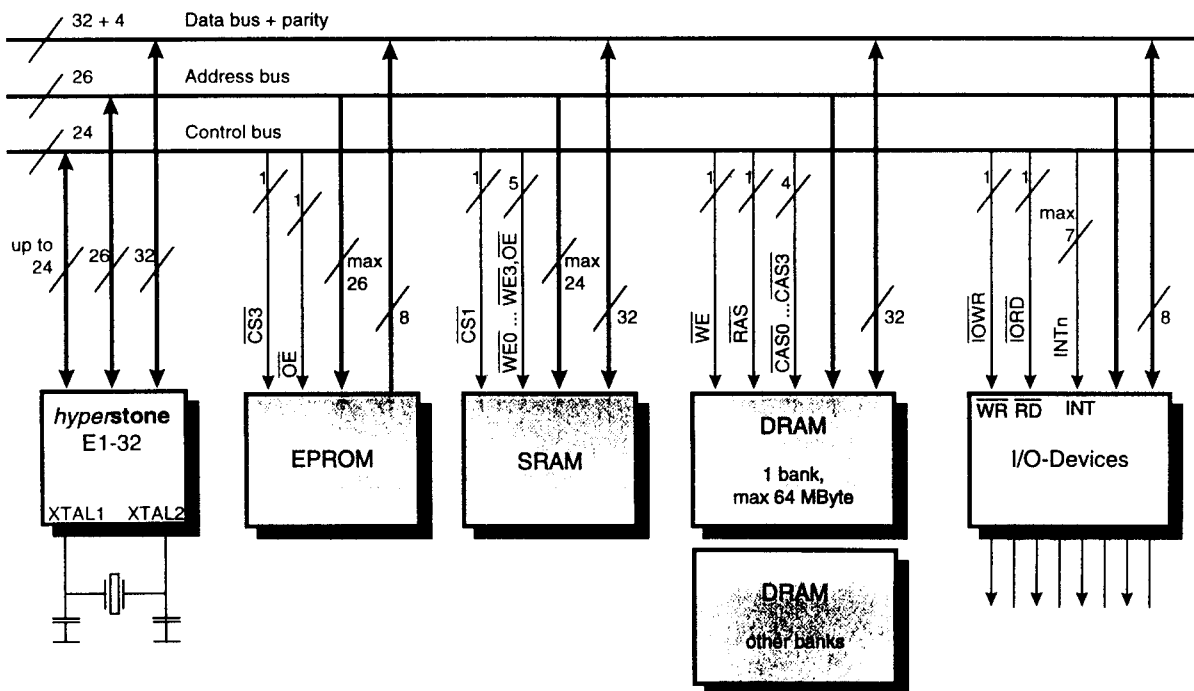
Unifying RISC and DSP

- **4 GB Address Space**
 - Divided into 4 areas of 1GB each:
 - ✓ MEM0 for DRAM (EDO or FPM)
 - ✓ MEM1 for fast SRAM
 - ✓ MEM2 for SRAM/Flash
 - ✓ MEM3 for ROM/Flash (Boot area)
 - Selectable bus width for each area
8-bit, 16-bit or 32-bit
 - Programmable bus timing for each area
- **Independent I/O address space**
 - I/O access timing encoded into I/O address
 - Wait support

⇒ **Direct connection of memories and peripherals**

07-98

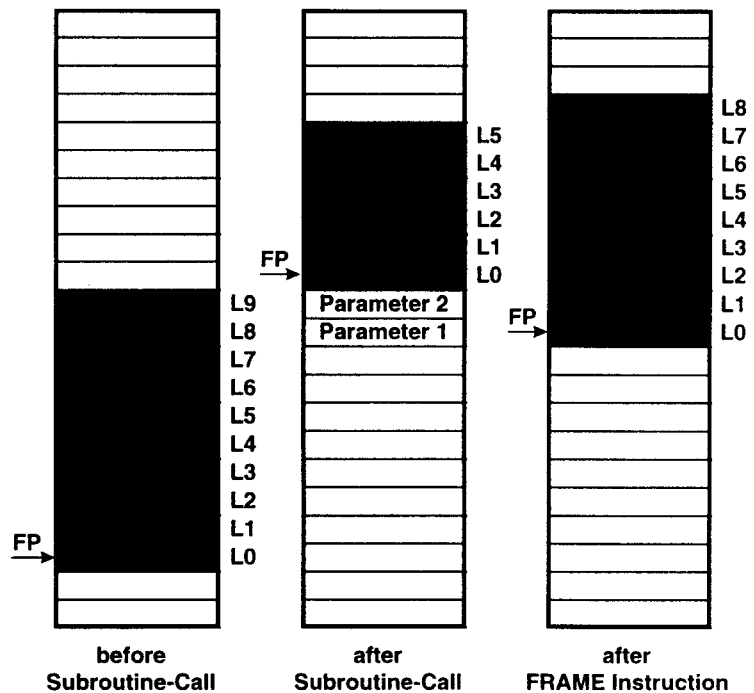
Unifying RISC and DSP



07-98

Unifying RISC and DSP

- 64 Local Registers
- Organized as register stack
- Up to 16 registers can be used by a routine
- Variable frame length
- FRAME instruction restructures current register frame
- Parameter-Passing via overlapping register frames
- Optimum use of register resources
- Automatic push and pop



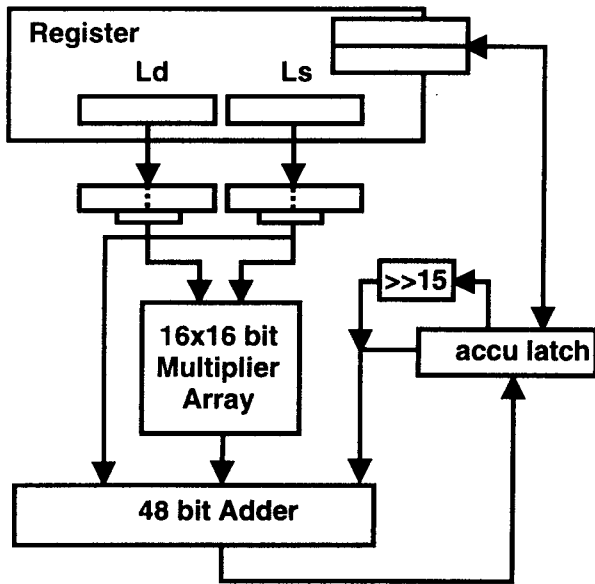
07-98

Unifying RISC and DSP

- DSP Instruction Execution Unit**
 - Instructions are strictly based on RISC principles
 - Register oriented
 - Single-Cycle instruction issue
 - Pipelined Instruction Execution
 - Parallelism between ALU, DSP and Load/Store Unit
 - Use of common resources
- DSP Algorithms:**
 - Register based implementation required
 - RISC style programming
 - Use of parallelism

07-98

Unifying RISC and DSP



- 16 bit arithmetic
 - multiply
 - multiply-add
 - multiply-accumulate
 - complex multiply
 - complex multiply-accumulate
 - add-subtract
 - fixed-point shift
 - 32 and 64 bit accumulator
- 32 bit arithmetic
 - multiply
 - multiply-accumulate
 - multiply-subtract
 - 32 and 64 bit accumulator
- Saturation via TRAP

07-98

Unifying RISC and DSP

Multiply Instructions:

		Cycles	Operations	Result
EMUL	16/32-bit multiply	1-3	1	32/64b
EHCMUL	Complex 16-bit multiply	4	6	2 x 32b

Multiply Accumulate Instructions:

EMAC	32-bit multiply accumulate	2-3	2	32/64b
EMSUB	32-bit multiply subtract	2-3	2	32/64b
EHMAL	16-bit multiply accumulate	2-4	4	32/64b
EHCMAL	Complex 16-bit multiply accumulate	4	8	2 x 32b

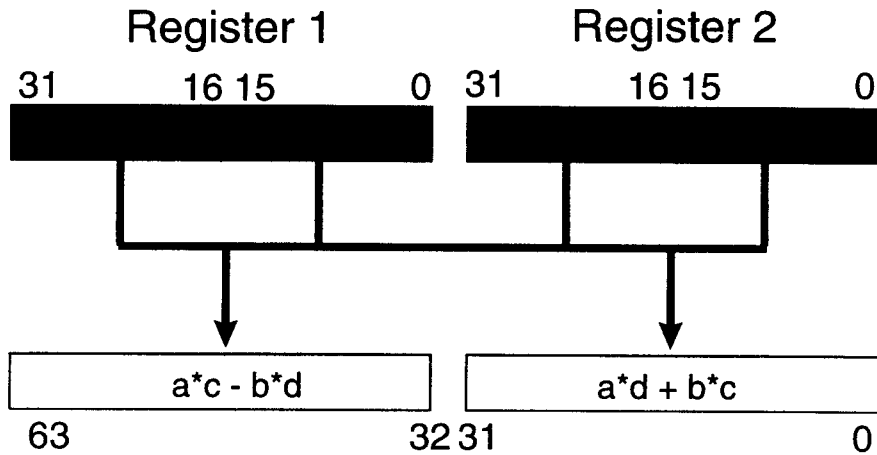
Summation Instructions:

EHCSUM	Complex 16-bit add/sub	3	4	4 x 16b
EHCFFT	Complex 16-bit add/sub for FFT	3	8	4 x 16b

07-98

Unifying RISC and DSP

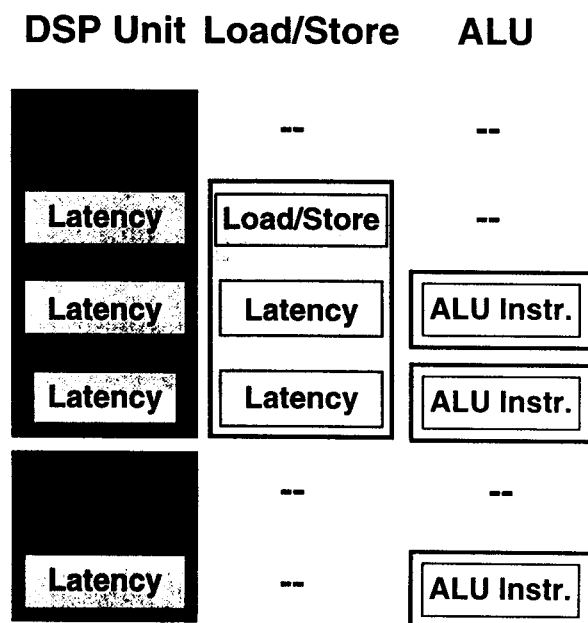
Complex Multiply: $Z = (a + bj) * (c + dj) = (ac - bd) + (ad + bc)j$



07-98

Unifying RISC and DSP

- Based on Single-Cycle Issue**
- Latency cycles are used for parallel execution**
- Load/Store during latency cycles of DSP-Execution.**
- ALU-Instructions during latency cycles of DSP-Execution and/or Load/Store-Execution**
- DSP loop control and address calculation can be hidden in the latency cycles**



07-98

Unifying RISC and DSP

- $Z = X * Y = (X_r + jX_i) * (Y_r + jY_i)$
- 5 cycles per loop

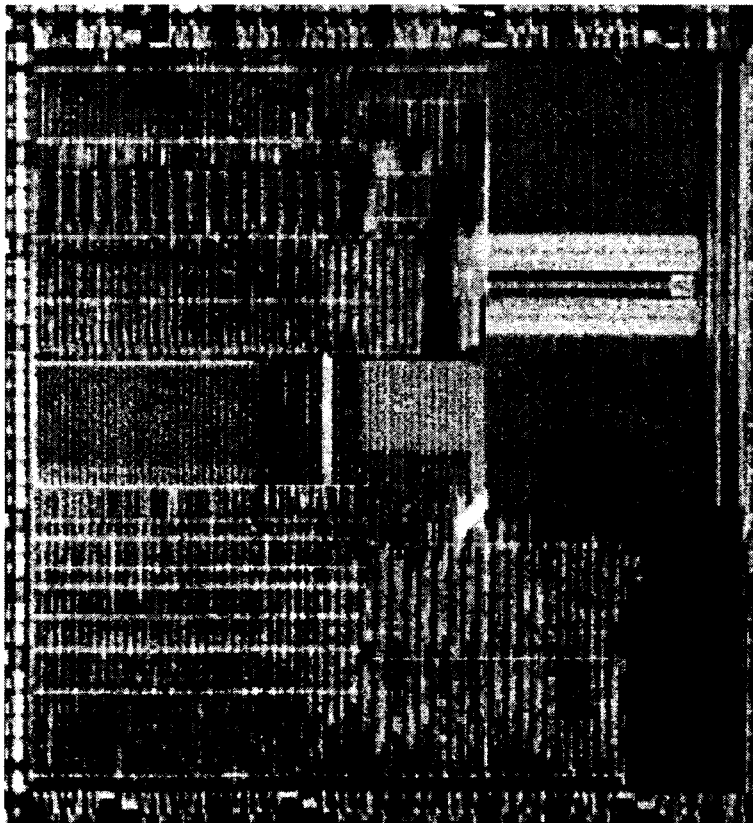
```

LDW.P      L0, L4      ; Load first x-value into L4
LDW.P      L1, L5      ; Load first y-value into L5
MOVI       L6, 100     ; Init loop counter

Loop: EHCMACD  L4, L5    ; complex multiply-add
LDW.P      L0, L4      ; load next x-value
LDW.P      L1, L5      ; load next y-value
DBNZ       Loop      ; Delayed Branch
ADDI       L6, -1      ; Decrement loop counter

STD.R      L2, G14     ; Store Result
    
```

Unifying RISC and DSP



- 0.5 μ m CMOS Process
- 26 mm² die size
- 320,000 transistors
- 80..90 MHz @ 5.0 V
- 50..60 MHz @ 3.3 V
- VDD = 2.3 ... 5.0 V
- Power-Down Current:
 - < 4.0 mA @ 5.0 V
 - < 2.5 mA @ 3.3 V
- Sleep Mode Current:
 - < 100 μ A

Unifying RISC and DSP

- Telecommunications**
 - GSM, CDMA, Cordless, WLL, PBX
- Digital Still Cameras**
- Automotive**
 - Motor control, Car comfort, ..
- Voice processing**
- Modems (CATV, ISDN, PSTN)**
- Printers, Scanners**
- Industrial Control, Robotics**
- PC Card Applications**
- Flash Memory Control**
- Navigation**
- Video Control Units**
- Internet enabling products**
- Classical embedded control**

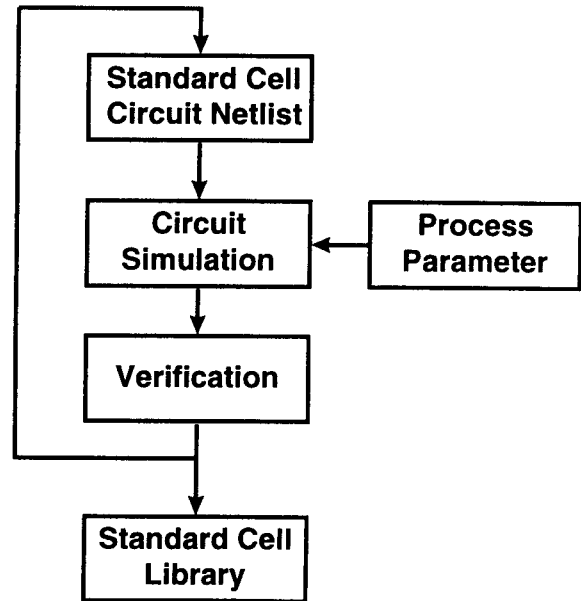
07-98

Design Flow

07-98

Unifying RISC and DSP

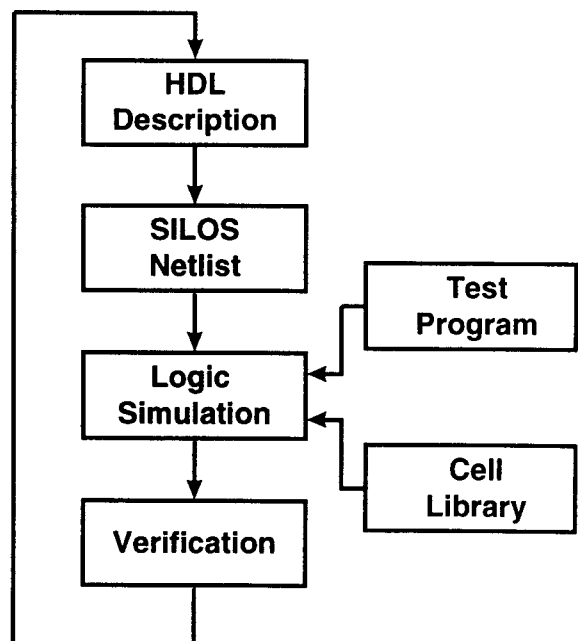
- 0.5 μm Design Rules
- Standard Cell Library
 - Basic gates, complex gates
 - Basic adder and counter cells
 - Latches, etc.
- Full Static CMOS
- Cells are optimized for fastest speed
- Equal delays for falling and rising edge
- Different output strengths



07-98

Unifying RISC and DSP

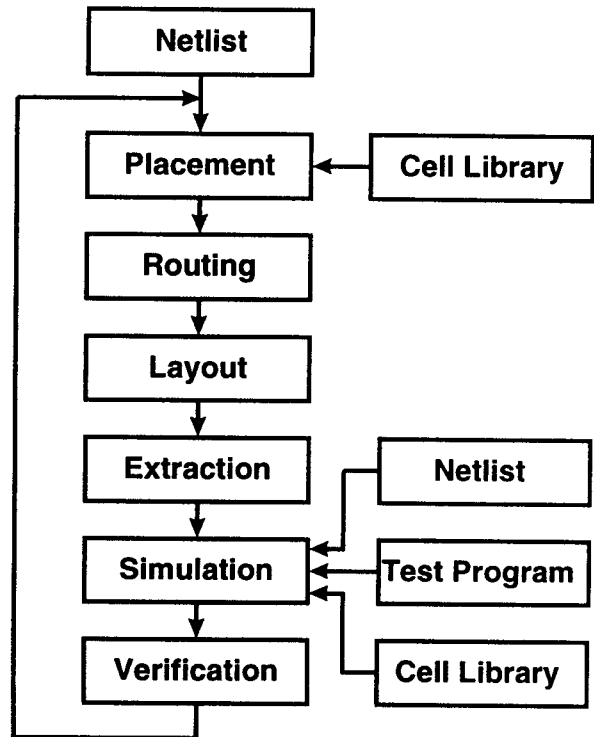
- Based on own Hardware Description Language
- No Schematics
- Description on Standard Cell Level
- No logic synthesis
- About 20,000 standard cells used



07-98

Unifying RISC and DSP

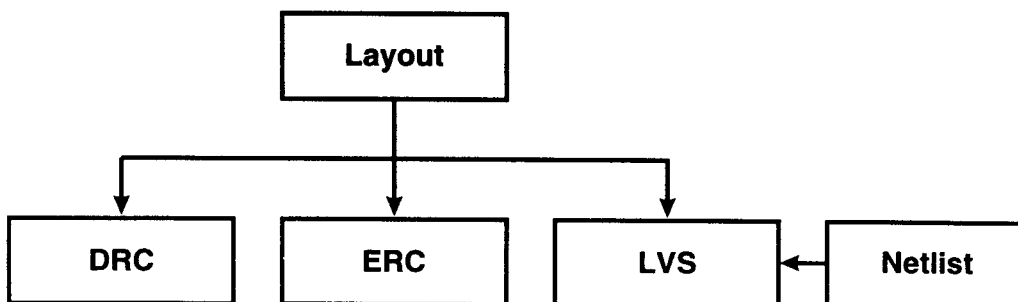
- Semi-automatic placement for dense designs**
- Automatic routing with two metal layers**
- Back-annotation for design verification**



07-98

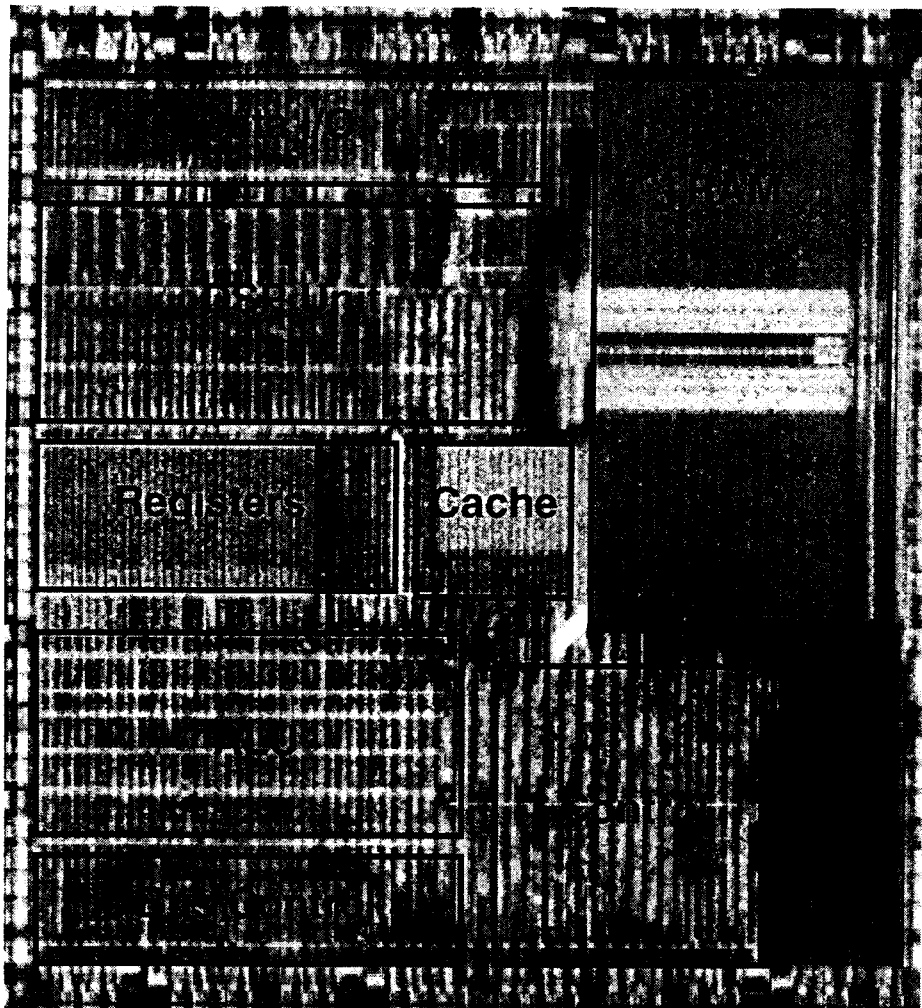
Unifying RISC and DSP

- DRC: Design Rule Check**
- ERC: Electrical Rule Check**
- LVS: Layout vs. Schematic**



07-98

Unifying RISC and DSP



SC Highway

Design Environment

Overview



SC-Highway Headlines

- Focus on Design Methodology
- Language / Synthesis based flow
- Streamlined - "Semi Custom" based
- Highly automated
- Provides "interfaces" to other flows



Methodology Aspects

- Design Guidelines
 - Multi-Clock Systems
 - Handling of Asynchronous Signals
 - Latches
- HDL Guidelines
 - Naming conventions
 - Efficient code for Simulation & Synthesis
- Test Design Rules
 - Macro Isolation
 - Handling of derived clocks



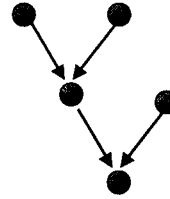
Design Flow Mission

- Iteration loops should be as short as possible
 - i.e. iterations between Synthesis and FP, not Synthesis and final layout should be the rule
 - allow iterations between Synthesis and Final Layout, but make this unnecessary
- Qualify the result of each design step
 - "sign-off" criterias for all steps in the design cycle
 - include guidelines & knowledge base to provide criterias for successful completion of each design step
- Reproducibility is a must



Make Based

- Design flow steps will only be executed, if necessary
 - If part of the source code changed
 - Due to different parameter set
- Used at subflow-level
 - Synthesis, Simulation, Floorplanning, Routing
- Advantages
 - Enormous savings of computation time
 - Automatic design data consistency



Design Data Management

- Addresses issues in parallel development
- Separates stable data from data under development
- No mutual blockages
- Data Management tool: **Clearcase**



Stability

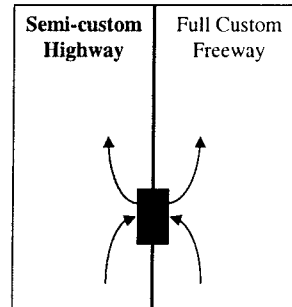
I use Fred's decoder for the receiver-test.

Flexibility

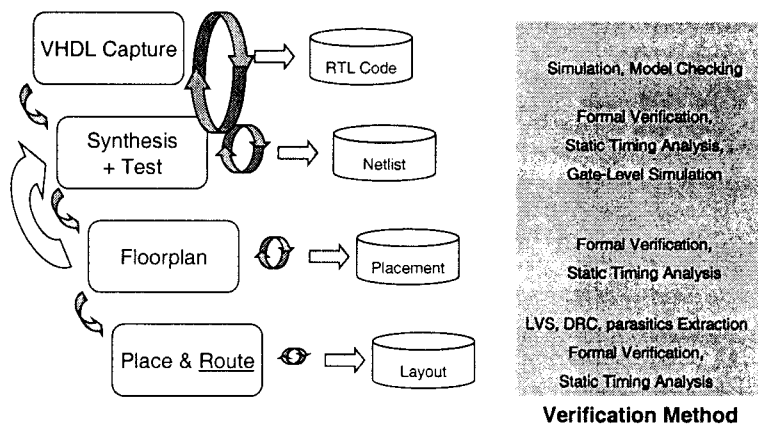
Fred can do experiments with the decoder, without disturbing me.

Macro Exchange

- "Black-Box" macro integration
- Necessary views are defined, respectively generated
- Standard file format interfaces



Flow Overview



Verification

- Two categories:
 - Verify compliance with specification
 - Verify that functionality is preserved during implementation
- Formal methods:
 - Formal Verification (CVE)
 - Model checking - RTL vs. high-level rules
 - Equivalence checking - RTL or netlist vs. netlist
 - Static Timing Analysis (Motive, PrimeTime)
 - Advantage: No pattern dependency
- Pattern based Verification
 - VHDL simulation at RT Level (Leapfrog)
 - Gate Level Simulation with VHDL testbench (Verilog, IKOS)
 - Emulation (Quickturn)



Logic Synthesis

- Automation of synthesis control
 - ssemake - make based
- Accurate, qualified wireload models
- Scan Insertion
- Efficient interface to floorplanning tool
- Efficient interface to P&R tool
- DesignCompiler (Synopsys) used

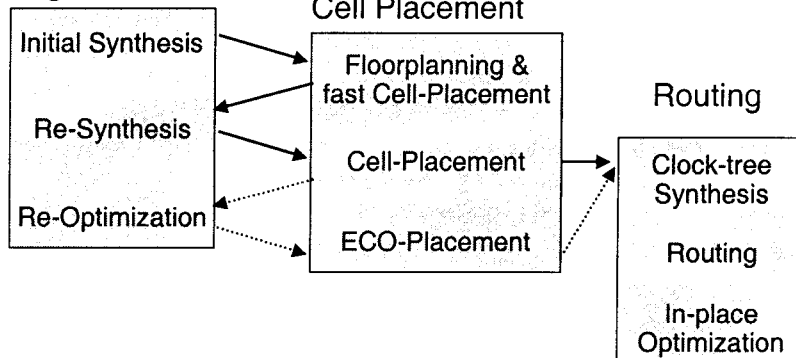
**Efficient tool usage and
automation of synthesis
is a focus**



Links to Layout

- Concurrent refinement

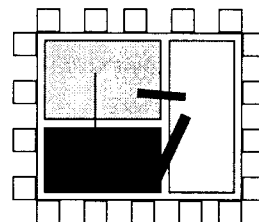
Logic



Floorplanning

- RTL floorplanning under evaluation
- Netlist level floorplanning
- Timing driven cell placement
- Tightly linked with synthesis
- Wireload model generation
- Clock and power planning / estimation
- Selected tool:
 - Planet (Avant!)

exploit physical design information early in the design cycle



Place & Route

- Timing driven place & route
- Automated and optimized interface to synthesis
 - Incremental P&R
 - Hierarchical P&R
- Advanced routing features
 - Clock tree synthesis
 - In-place optimization
 - Multi-power routing
- Selected tool
 - Aquarius (Avant!)

**guarantee
convergence and
predictability in the
physical design
domain**



Layout Verification & Extraction

- Enhanced verification features
 - Clock skew calculation
 - Power analysis
 - Electro-migration analysis
- Manage sub-micron technologies effects
 - Quasi 3-D, 3-D extraction
 - Advanced parasitic extraction (cross coupling, ...)
 - Wire resistance extraction
- Manage increasing chip complexity
 - Hierarchical extraction
- Selected Tools
 - Vampire (Cadence), PowerMill & RailMill (Synopsys)

**increased automation and
design quality**

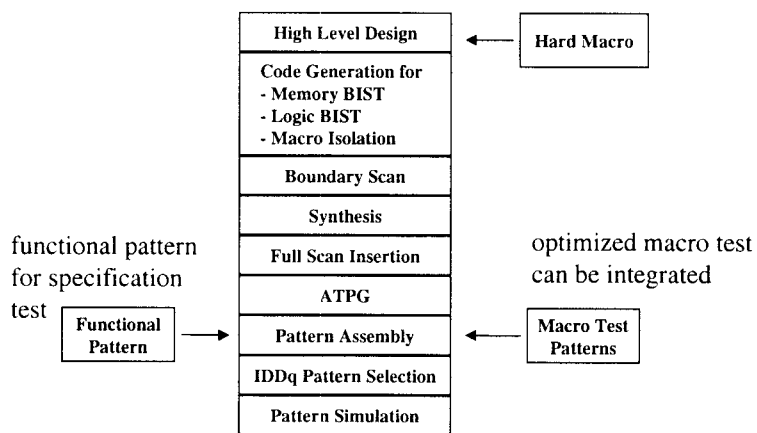


Test Features

- Automatic testpattern generation (ATPG)
 - For synthesized logic
- BIST
 - For memories
 - To observe internal macro pins
 - To stimulate internal blocks standard pattern
- Macro isolation
- I_{DDq}-pattern selection
- Pattern integration
 - Generated pattern
 - Functional pattern for parametric tests



Test Flow



Loop Gain mit PSPICE

Einführung

Die Bestimmung des loop gain mit PSPICE ist nicht ganz trivial. In der klassischen Verstärkerliteratur wird empfohlen, die Schleife aufzutrennen und die Belastungsverhältnisse an der Trennstelle genau nachzubilden. Welch ein praxisorientierter Rat: woher kennt man denn die Belastungsverhältnisse, die sich aufgrund der Nichtlinearitäten elektronischer Schaltungen iterativ einstellen? Außerdem ist durch das Auftrennen die Arbeitspunkteinstellung gestört und muß durch komplizierte Kunstschaltungen, die selbst einen Frequenzgang haben, gewährleistet werden.

In [1] ist ein Verfahren angegeben, wie man den loop gain von rückgekoppelten Schaltungen durch zwei AC-Analysen mit PSPICE bestimmt. Die Ableitung des Verfahrens ist etwas schwer verständlich, nicht zuletzt wegen der ungewohnten amerikanischen Symbole und Bepfeilung! Man kann sich nämlich bei dem Verfahren sehr leicht in den Vorzeichen irren, was prompt in [2] geschehen ist. Dort wird das Verfahren aus [1] wieder aufgegriffen, die PSPICE Analyse den fortgeschrittenen Möglichkeiten der späteren Programmversion angepaßt, aber die Erklärungen beschränken sich auf Rezeptniveau. Der erwähnte Vorzeichenfehler wirkt sich bei dem dort gewählten Beispiel nicht aus, man merkt es erst, wenn man das Verfahren auf eigene Probleme anwenden will.

Wegen der großen Bedeutung der Diskussion des loop gain bei Stabilitätsproblemen von rückgekoppelten Schaltungen in der Ingenieurausbildung wird im folgenden das in [1] vorgestellte Verfahren in der hier bei uns üblichen Weise dargestellt. Danach wird die Leistungsfähigkeit des Verfahrens an der loop gain Analyse eines Colpitts-Oszillators demonstriert.

Bestimmung des loop gain aus dem Kleinsignal-Ersatzbild einer rückgekoppelten Schaltung.

Gegeben sei das allgemeine Schaltbild eines Colpitts-Oszillators (Fig. 1) als Beispiel für eine rückgekoppelte Schaltung, die sehr bald verallgemeinert werden wird.

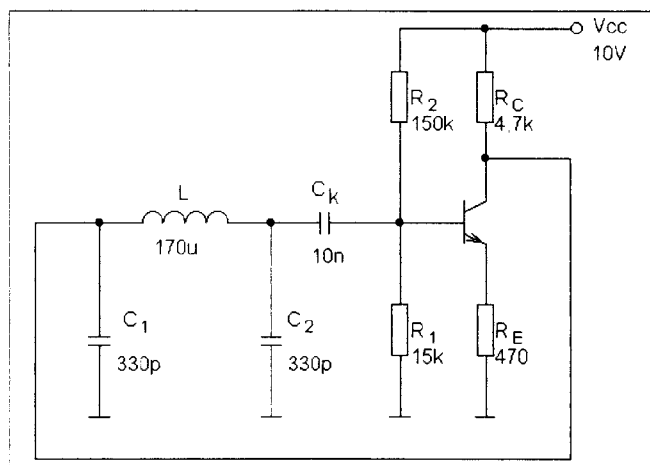


Fig. 1: Colpitts-Oszillator (Zahlenwerte aus [3])

Fig. 2 zeigt das zugehörige Kleinsignalersatzbild zu Fig. 1

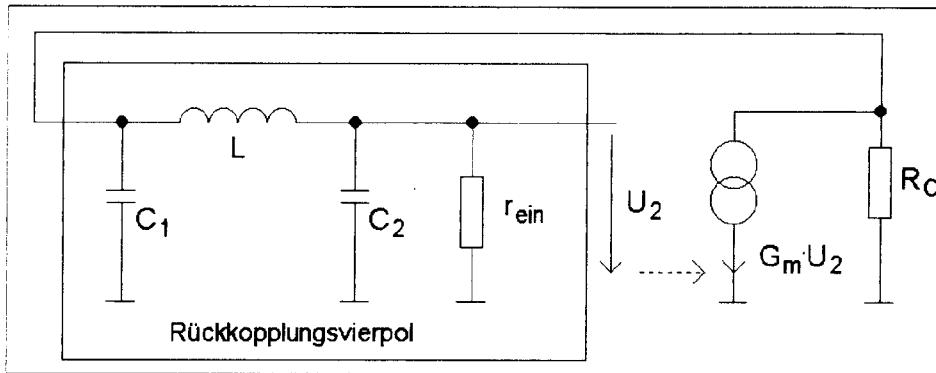


Fig. 2: Kleinsignalersatzbild zu Fig. 1 mit

$$r_{\text{ein}} = R_1 \parallel R_2 \parallel \beta \left(\frac{1}{g_m} + R_E \right), \quad G_m = \frac{1}{\frac{1}{g_m} + R_E} = \frac{g_m}{1 + g_m R_E}$$

$$(\text{= } 10,8 \text{ k}\Omega) \qquad \left(\text{= } \frac{1}{527,5 \Omega} \right)$$

Man beachte, daß der Eingangswiderstand des Verstärkers mit in den Rückkopplungsvierpol einbezogen ist. Fig. 3 zeigt das umgezeichnete Kleinsignalersatzbild, wobei jetzt zwischen der gesteuerten Stromquelle und dem Widerstand R_C die Schleife aufgetrennt ist und der Rückkopplungsvierpol verallgemeinert wurde.

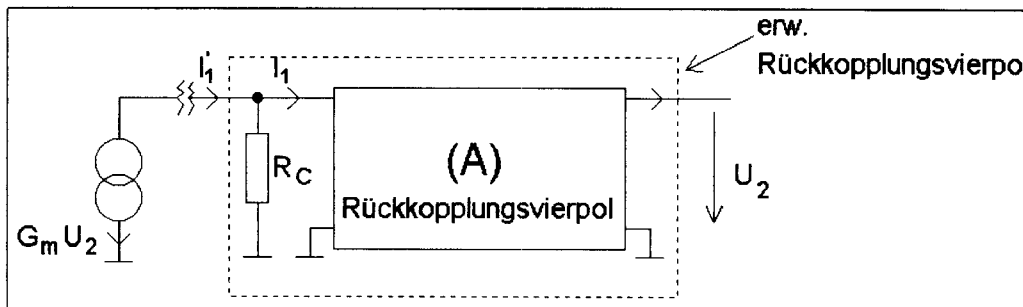


Fig. 3: Kleinsignalersatzbild zur Bestimmung des loop-gain.

Der loop gain ist definiert als die Übertragungsfunktion (Transfer-Funktion)

$$T = \frac{G_m U_2}{I_1} = \frac{\text{verstärkter Strom durch die gesteuerte Stromquelle}}{\text{Eingangsstrom des erweiterten Rückkopplungsvierpols}}$$

Beträgt die Phasendrehung zwischen U_2 und I_1' 180° , so sind die Ströme $-G_m U_2$ und I_1' in Phase, so daß bei entsprechenden Amplitudenverhältnissen eine Eigenregung möglich wird.

Mit den Regeln der Vierpoltheorie folgt für den erweiterten Rückkopplungsvierpol in Fig. 3

$$\begin{pmatrix} 1 & 0 \\ \frac{1}{R_c} & 1 \end{pmatrix} \cdot \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = \begin{pmatrix} \dots & \dots \\ \frac{A_{11}}{R_c} + A_{21} & \dots \end{pmatrix} = \begin{pmatrix} \dots & \dots \\ \frac{I_1'}{U_2} & \dots \end{pmatrix}$$

Damit ergibt sich für den loop gain

$$T = \frac{G_m U_2}{I_1'} = \frac{G_m}{I_1'/U_2} = \frac{G_m}{\frac{A_{11}}{R_c} + A_{21}} = \frac{1}{\frac{A_{11}}{G_m R_c} + \frac{A_{21}}{G_m}} = \frac{1}{y + x}$$

Man kann den loop gain auch als Spannungs-Übertragungsfunktion definieren, wenn man die Stromquelle mit Innenwiderstand R_c entsprechend umwandelt (Fig. 4).

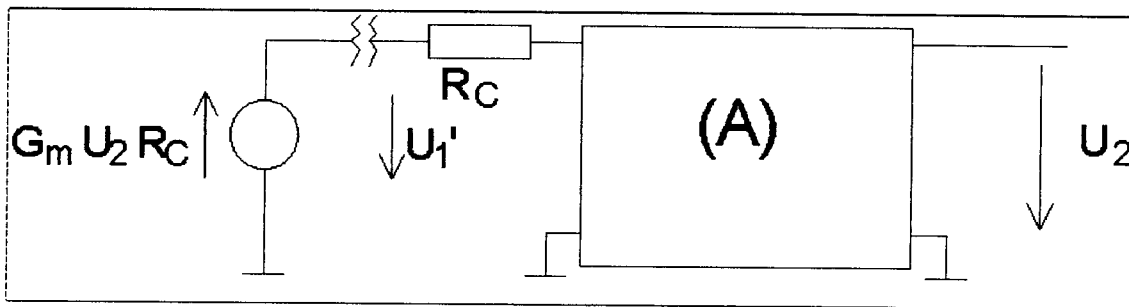


Fig. 4: Spannungs-Übertragungsfunktion des loop gain

Es gilt dann

$$T = \frac{G_m U_2 R_c}{U_1'} = \frac{\text{verstärkte Spannung an der gesteuerten Spannungsquelle}}{\text{Eingangsspannung des erweiterten Rückkopplungsvierpols}}$$

Mit den Regeln der Vierpoltheorie folgt

$$\begin{pmatrix} 1 & R_c \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = \begin{pmatrix} A_{11} + A_{21} R_c & \dots \\ \dots & \dots \end{pmatrix} = \begin{pmatrix} \frac{U_1'}{U_2} & \dots \\ \dots & \dots \end{pmatrix}$$

Damit erhält man

$$T = \frac{G_m U_2 R_c}{U_1'} = \frac{G_m R_c}{U_1'/U_2} = \frac{G_m R_c}{A_{11} + A_{21} R_c} = \frac{1}{\frac{A_{11}}{G_m R_c} + \frac{A_{21}}{G_m}} = \frac{1}{y + x}$$

also das gleiche Ergebnis wie bei der Strombetrachtung.

Bestimmung der zwei Vierpolterme des loop gain durch zwei spezielle Kleinsignalanalysen

In Fig. 5 ist das Kleinsignalersatzbild der rückgekoppelten Schaltung (closed loop!) gezeigt mit einer zusätzlichen Stromquelle I zwischen Innenwiderstand der gesteuerten Stromquelle und Rückkopplungsvierpol.

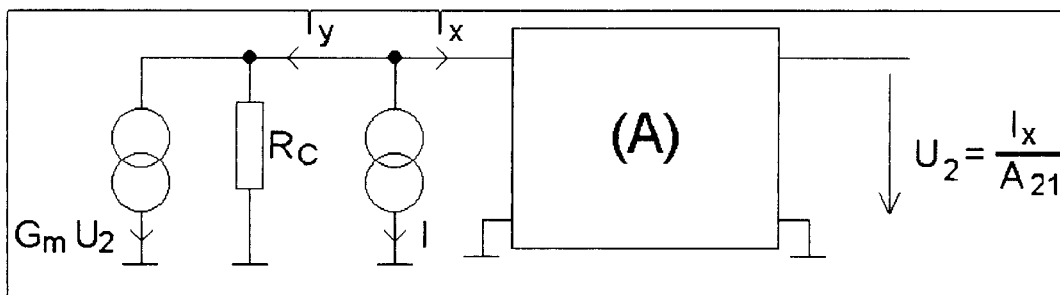


Fig. 5: Closed-loop Schaltung mit zusätzlicher Stromquelle

Der Strom I ruft die Ströme I_x und I_y hervor. Wegen der Beziehung für den Eingangswiderstand eines leerlaufenden Vierpols ($= A_{11}/A_{21}$) gilt folgender Zusammenhang:

$$(KCL) \quad I_y = G_m U_2 + I_x \cdot \frac{A_{11}}{A_{21} R_c} = I_x \left(\frac{G_m}{A_{21}} + \frac{A_{11}}{A_{21} R_c} \right)$$

Damit ergibt sich für die Stromübertragungsfunktion

$$T_i = \frac{I_y}{I_x} = \frac{A_{11}}{A_{21} R_c} + \frac{G_m}{A_{21}} = \frac{G_m}{A_{21}} \left(\frac{A_{11}}{G_m R_c} + 1 \right) = \frac{y+1}{x}$$

Die Stromübertragungsfunktion T_i ist von den gleichen Vierpolgrößen abhängig wie der loop gain!

In Fig. 6 ist das Kleinsignalersatzbild der rückgekoppelten Schaltung (closed loop!) gezeigt mit einer zusätzlichen Spannungsquelle zwischen Innenwiderstand der gesteuerten Spannungsquelle und Rückkopplungsvierpol.

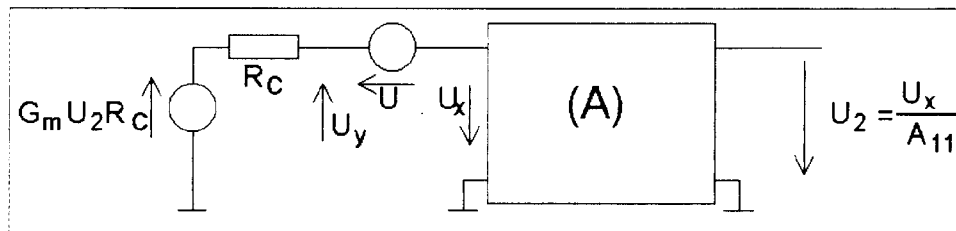


Fig. 6: Closed-loop Schaltung mit zusätzlicher Spannungsquelle

Die Spannungsquelle U ruft die Spannungen U_x und U_y hervor. Der Strom durch R_c kann durch U_x und den Eingangswiderstand des leerlaufenden Vierpols (A_{11}/A_{21}) ausgedrückt werden. Es gilt

$$(KVL) \quad U_y = G_m R_c U_2 + U_x \cdot \frac{A_{21}}{A_{11}} \cdot R_c = U_x \left(\frac{G_m R_c}{A_{11}} + \frac{A_{21} R_c}{A_{11}} \right)$$

Damit ergibt sich für die Spannungsübertragungsfunktion

$$T_v = \frac{U_y}{U_x} = \frac{G_m R_c}{A_{11}} + \frac{A_{21} R_c}{A_{11}} = \frac{G_m R_c}{A_{11}} \left(1 + \frac{A_{21}}{G_m} \right) = \frac{1+x}{y}$$

Auch hier ist die Spannungsübertragungsfunktion von den gleichen Vierpolgrößen abhängig wie der loop gain.

Durch Elimination von x und y kann man deshalb den loop gain T durch die Übertragungsfunktionen T_i und T_v ausdrücken. Nach leichter Rechnung folgt

$$y = \frac{T_i + 1}{T_i T_v - 1}$$

$$x = \frac{T_v + 1}{T_i T_v - 1}$$

und damit

$$T = \frac{1}{y+x} = \frac{T_i T_v - 1}{2 + T_i + T_v} = \frac{1 - \frac{1}{T_i T_v}}{\frac{1}{T_i T_v} + \frac{1}{T_v} + \frac{1}{T_i}}$$

Ergebnis: Der loop gain kann aus der Messung von zwei Übertragungsfunktionen an der closed-loop Schaltung gewonnen werden. Dazu wird an einer beliebigen Stelle der Schaltung eine Stromquelle (vom Bezugsknoten aus) angebracht. Der unendlich hohe Innenwiderstand stört dabei die Belastungsverhältnisse und die Arbeitspunkteinstellung nicht. Die durch die Stromquelle hervorgerufenen Ströme werden ins Verhältnis gesetzt und ergeben die Stromübertragungsfunktion T_i .

In einer zweiten Analyse wird an der Stelle, wo die Stromquelle angeschaltet war, die Schaltung aufgetrennt und die Trennstelle mit einer Spannungsquelle überbrückt. Der unendlich kleine Innenwiderstand stört dabei die Belastungsverhältnisse und die Arbeitspunkteinstellung nicht. Die durch die Spannungsquelle hervorgerufenen Spannungen werden (vorzeichenrichtig!) ins Verhältnis gesetzt und ergeben die Spannungsübertragungsfunktion T_v .

Mit einer einfachen Formel kann der open-loop gain T aus T_i und T_v berechnet werden. Sind (im gesamten Frequenzbereich!) T_i oder T_v stark unterschiedlich in ihrer Größe, so bestimmt der **kleinere** Wert den loop gain, wie man aus der zweiten Form der Verknüpfungsgleichung leicht sehen kann.

Obwohl die Ableitung des Verfahrens mit einer besonders naheliegenden und zu einfachen Verhältnissen führenden Trennstelle der Schleife durchgeführt wurde, kann man argumentieren, daß an jeder Trennstelle Ersatzquellen und Übertragungsvierpole gefunden werden, die Größen G_m und R_c werden dann lediglich komplex. Da in der vorhergehenden Diskussion nirgendwo rein reelle Werte für G_m und R_c gefordert werden mußten, gilt das Verfahren auch für beliebige Punkte der Schleife!

Berechnung von T_i , T_v und T für den Colpitts-Oszillator

Mit den Regeln der Vierpoltheorie folgt aus Fig. 2 für den Rückkopplungsvierpol

$$A_{11} = 1 + j\omega L \left(\frac{1}{r_{\text{ein}}} + j\omega C_2 \right) \approx 1 - \omega^2 LC_2 \quad \left(\text{für } \frac{1}{r_{\text{ein}}} \ll \omega C_2 \right)$$

$$A_{21} = j\omega C_1 + \frac{1}{r_{\text{ein}}} + j\omega C_2 + j\omega L \cdot j\omega C_1 \left(\frac{1}{r_{\text{ein}}} + j\omega C_2 \right)$$

$$= \frac{1}{r_{\text{ein}}} (1 - \omega^2 LC_1) + j\omega (C_1 + C_2 - \omega^2 LC_1 C_2)$$

Der Imaginärteil von A_{21} wird Null bei der Resonanzfrequenz

$$\omega_0 = \frac{1}{\sqrt{L \cdot \frac{C_1 C_2}{C_1 + C_2}}} (= 2\pi \cdot 950 \text{kHz})$$

Damit ergibt sich für $\omega = \omega_0$

$$A_{11} = -\frac{C_2}{C_1}$$

$$A_{21} = -\frac{1}{r_{\text{ein}}} \frac{C_1}{C_2}$$

Deshalb gilt für

$$T_{i|\omega_0} = \frac{G_m}{A_{21}} \left(\frac{A_{11}}{G_m R_c} + 1 \right) = \frac{\frac{A_{11}}{R_c} + G_m}{A_{21}} = \frac{-\frac{C_2}{C_1} \cdot \frac{1}{R_c} + G_m}{-\frac{C_1}{C_2} \frac{1}{r_{\text{ein}}}} = -\frac{\frac{1}{4,7 \text{ k}} + \frac{1}{527,5}}{\frac{1}{10,8 \text{ k}}} = -18,17$$

wobei die Zahlenwerte aus Fig. 1 und Fig. 2 verwendet wurden.

Entsprechend gilt für

$$T_{v|\omega_0} = \frac{G_m R_c}{A_{11}} \left(1 + \frac{A_{21}}{G_m} \right) = \frac{G_m + A_{21}}{\frac{A_{11}}{R_c}} = \frac{G_m - \frac{C_1}{C_2} \frac{1}{r_{\text{ein}}}}{-\frac{C_2}{C_1} \frac{1}{R_c}} = -\frac{\frac{1}{527,5} - \frac{1}{10,8 \text{ k}}}{\frac{1}{4,7 \text{ k}}} = -8,47$$

Für T ergibt sich dann

$$T_{|\omega_0} = \frac{1}{\frac{A_{11}}{G_m R_c} + \frac{A_{21}}{G_m}} = \frac{G_m}{\frac{A_{11}}{R_c} + A_{21}} = -\frac{G_m}{\frac{C_2}{C_1} \frac{1}{R_c} + \frac{C_1}{C_2} \frac{1}{r_{\text{ein}}}} = -\frac{\frac{1}{527,5}}{\frac{1}{4,7 \text{ k}} + \frac{1}{10,8 \text{ k}}} = -6,2$$

oder

$$T|_{\omega_0} = \frac{T_i \cdot T_v - 1}{2 + T_i + T_v} \Big|_{\omega_0} = \frac{18,17 \cdot 8,47 - 1}{2 - 18,17 - 8,47} = -6,2$$

Bestimmung von T_i , T_v und T des Colpitts-Oszillators mit PSPICE

Fig. 7 zeigt die Schematics-Eingabe des Colpitts-Oszillators in Fig. 1 für PSPICE. Das Modell Qbreak N-X definiert einen NPN Standard-Transistor mit $I_S=10fA$. Der Kollektorstrom im Arbeitspunkt ist damit 0,45mA. In dem Block lgt (loop gain transfer) verbirgt sich die Schaltung Fig. 8.

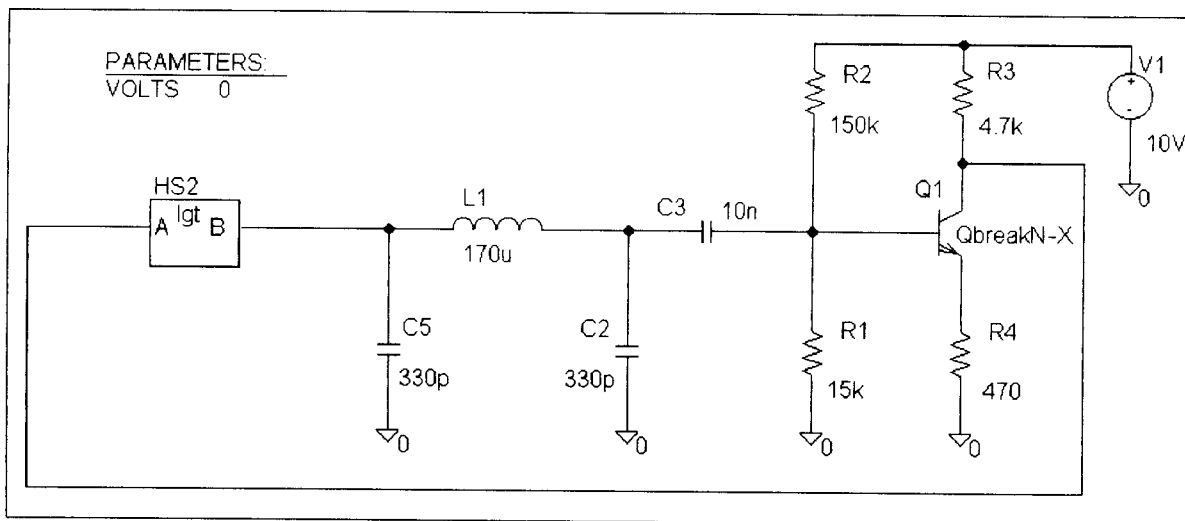


Fig. 7

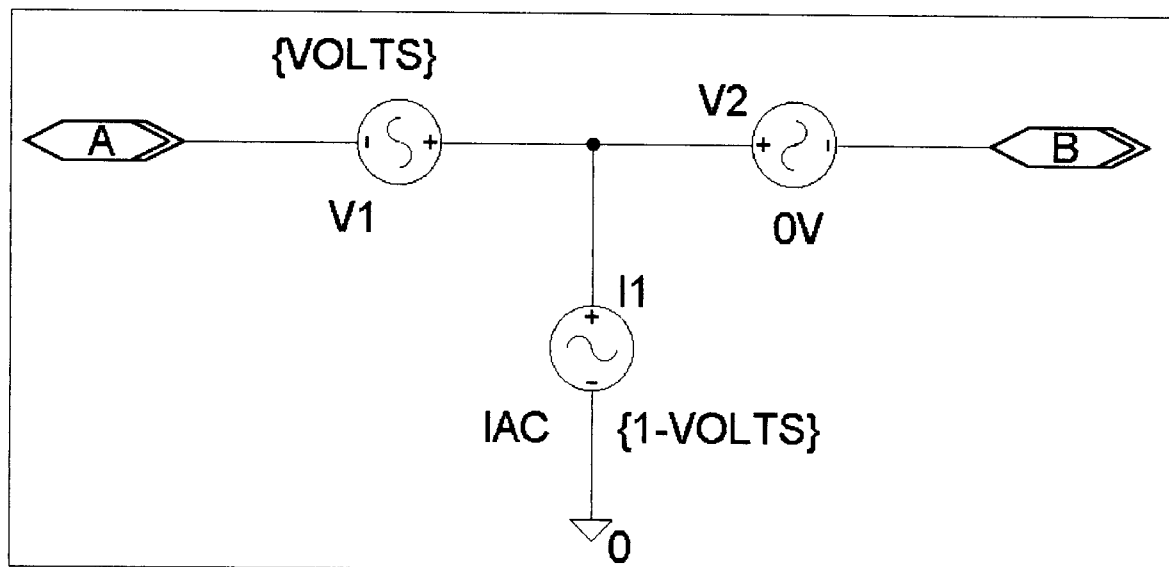


Fig. 8

Es wurde eine parametrische AC-Analyse im Frequenzbereich $900kHz \leq f \leq 1000kHz$ durchgeführt, wobei der Parameter "VOLTS" beim ersten Lauf den Wert 0, beim zweiten den Wert 1 annahm [2]. Für VOLTS = 0 ist die Stromquelle mit 1A wirksam, für VOLTS = 1 die Spannungsquelle V1 mit 1V. Die Stromquelle V2 (immer 0V!) dient zur Messung des durch sie fließenden Stroms.

Nach der AC-Analyse muß dem Auswertprogramm PROBE mitgeteilt werden, daß alle Daten aus beiden Parameter-Analysen benötigt werden. In PROBE kann man unter dem Menüpunkt TRACE ein Macro definieren, d. h. eine Verknüpfungsvorschrift für die Auswertung. Folgendes muß dort eingegeben werden:

$$Ti=(I(HS2.V1)@1)/(I(HS2.V2)@1)$$

$$Tv=- (V(HS2:A)@2)/(V(HS2:B)@2)$$

$$T=(Ti \cdot Tv - 1)/(2 + Ti + Tv)$$

Die Übertragungsfunktion T_i wird damit durch das Verhältnis der Ströme durch die Spannungsquellen V1 und V2 des Subcircuits HS2 bestimmt, die für den 1. Parameter berechnet wurden (@1!). Die Übertragungsfunktion T_v wird damit durch das **negative** Verhältnis der Knotenspannungen an den Übergabeknoten A und B des Subcircuits HS2 bestimmt, die für den 2. Parameter berechnet wurden (@2!). Die Berechnung von T geschieht dann mit der in der Ableitung angegebenen Formel.

Fig. 9 zeigt die Größen T_i , T_v und T nach Betrag und Phase. Bei einer Phase von annähernd -180° und einer Frequenz von etwa 950kHz ergeben sich in sehr guter Näherung die in der Handrechnung ermittelten Werte für $T_{i|\omega_0}$, $T_{v|\omega_0}$ und $T_{|\omega_0}$. Mit einem loop gain von 6,2 und einer Phase von -180° bei 950kHz schwingt der Colpitts-Oszillator, was durch eine Transient Analyse leicht nachgewiesen werden kann.

Zusammenfassung und Schluß

Die sehr gute Idee, den loop gain durch zwei AC-Analysen der closed-loop Schaltung zu bestimmen, wobei die Schaltung nicht gestört wird, hat sich nicht so richtig durchgesetzt, weil sie in [1] sehr spröde, in [2] fehlerhaft und rezeptmäßig dargestellt wurde. Mit der hier vorgelegten didaktischen Aufbereitung des Verfahrens wird der Hoffnung Ausdruck gegeben, daß das Verfahren in Zukunft häufiger bei der Diskussion von Stabilitätsproblemen in Elektronik und Regelungstechnik angewendet wird.

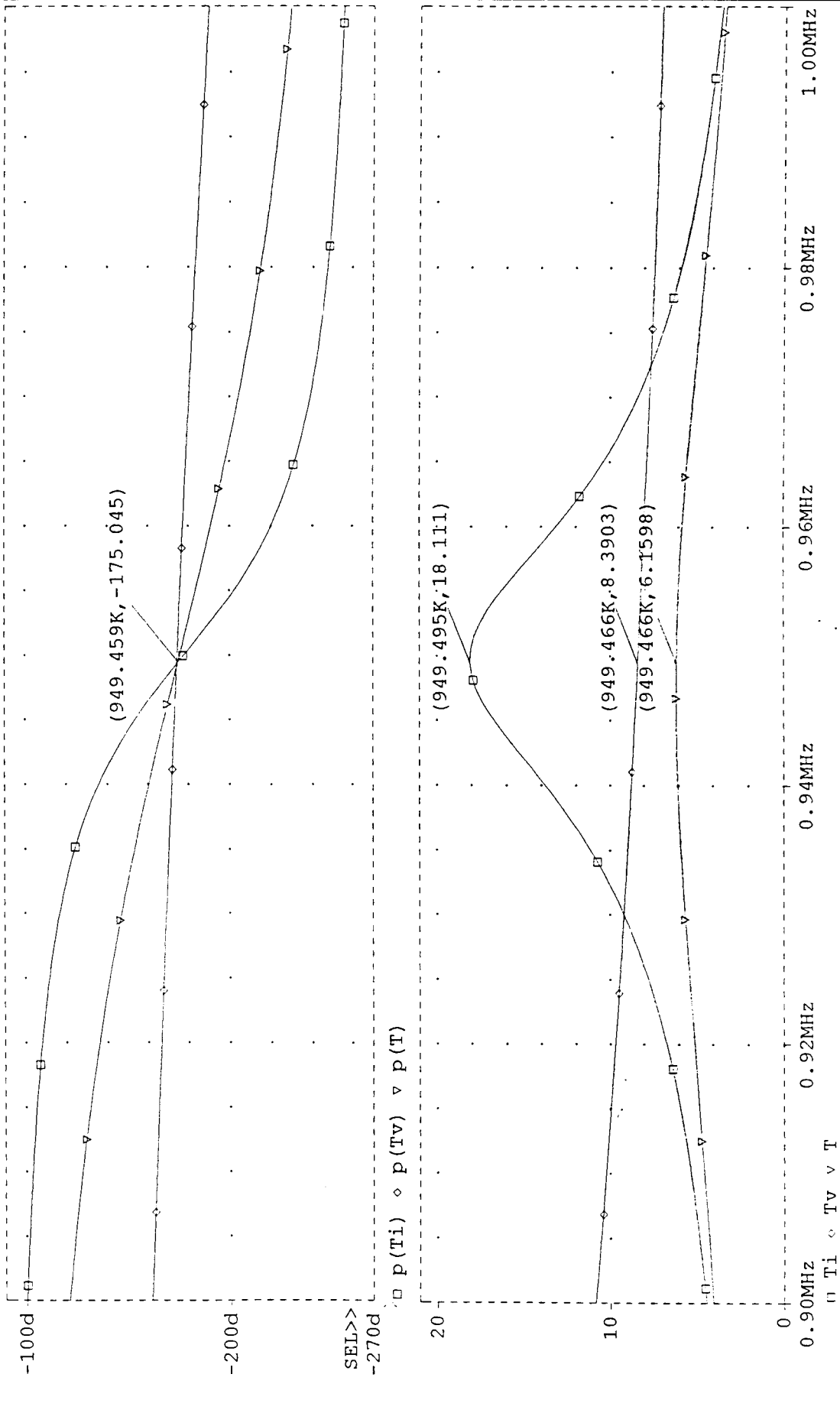
- | | |
|----------------------|--|
| [1] Paul W. Tuininga | SPICE - A guide to Circuit
Simulation & Analysis using PSPICE
Prentice Hall International Edition
Englewood Cliffs, 1988
(S. 59 - 65) |
| [2] MicroSim | Tutorial, Application Notes and Design
Ideas
MicroSim Corporation 1995
Version 6.2, April 1995
(Simulating High-QCircuits Using Open
Loop Response, p155) |
| [3] Erwin Böhmer | Elemente der angewandten Elektronik
Vieweg 1989 (S 189) |

* C:\MSIMPR63\EVAL\test1.sch

Date/Time run: 01/20/98 15:02:57

Temperature: 27.0

(A) C:\MSIMPR63\EVAL\TEST1.DAT



Time: 15:34:50

Page 1

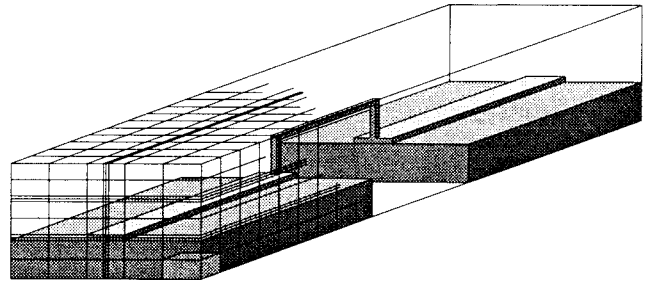
Date: January 20, 1998

F. 49 (2)

Feldnumerische Simulation kreiszylindrischer Strukturen bei hohen Frequenzen

Dipl.-Ing. (FH) Frank Hamann,
Prof. Dr.-Ing. Andreas Christ
Fachhochschule Offenburg, Badstr. 24, 77652 Offenburg
Tel. 0781/205-130, Fax 0781/205-111

Die hochfrequente, feldnumerische Analyse mit der Finite-Differenzen Methode erfordert die Diskretisierung der zu untersuchenden Struktur in einem nichtäquidistanten Gitter. Vorschriften zur Diskretisierung kreiszylindrischer Strukturen wie sie z.B. bei Durchkontaktierungen auftreten, werden untersucht und eine optimierte Lösung vorgestellt.



1. Einführung

Die quantitative Analyse der physikalischen Wechselwirkung zwischen betrachteter Struktur und elektromagnetischen Feldern innerhalb einer Strukturberandung erfordert zum einen ein die Wirklichkeit hinreichend genau beschreibendes Modell, zum anderen die mathematische Analyse dieses Modells. Die Modellbildung erfolgt bei der Finite-Differenzen Methode durch Zuordnung eines Gleichungssystems, das die elektrischen Verhältnisse hinreichend genau beschreibt, zu jeweiliger Strukturgeometrie. Die Analyse erfolgt numerisch.

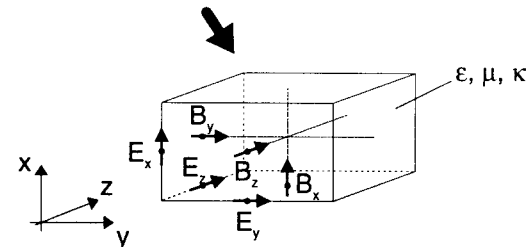


Abb. 1 Zerlegung der Struktur in Elementarquader, Verwendung eines 3-dim. nicht-äquidistanten kartesischen Gitters

1.1 Kurze Beschreibung der verwendeten feldnumerischen Simulationsmethode

Neben feldnumerischen Simulationsverfahren wie Integralgleichungsmethode oder Finite-Elemente Methode [1] auf der Grundlage tetraederförmiger Zellen erwies sich die Finite-Differenzen Methode auf der Grundlage quaderförmiger Zellen [2], [3], [4] als effektives Mittel zur Analyse passiver HF-Strukturen [5], [6], [7]. Die Gesamtstruktur wird hierbei diskretisiert, d. h. in Elementarquader i. a. nicht-äquidistanter Abmessungen und variabler Materialfüllungen eingeteilt (s. Abb. 1). Innerhalb jedes Elementarquaders wird jedoch homogene Materialfüllung vorausgesetzt.

Die komplexen Maxwell'schen Gleichungen in Integralform werden auf jeden Elementarquader angewandt. Durch Annahme einer geeigneten Anzahl linear unabhängiger Erregungen an den Toren und Berücksichtigung der metallischen Umhüllung der Struktur ergibt sich je ein Randwertproblem für die Berechnung des elektrischen Feldes innerhalb der Strukturberandung. Unter Kenntnis des elektrischen Feldes innerhalb der Strukturberandung und der für die Moden geltenden Orthogonalitätsbedingung läßt sich der auf Mode und Tor bezogene Reflexionsfaktor bestimmen. Hieraus ergeben sich schließlich die Streumatrixkoeffizienten je untersuchtem Frequenzpunkt :

$$(S) = \begin{pmatrix} S_{11} & S_{12} & \dots & S_{1n} \\ S_{21} & & & \\ \vdots & & S_{kl} & \\ \vdots & & & \\ S_{n1} & \dots & & S_{nn} \end{pmatrix}$$

2. Feldnumerische Analyse kreiszylindrischer Strukturen

Unter Verwendung der Finite-Differenzen Methode auf der Grundlage quaderförmiger Zellen ergibt sich die Schwierigkeit der Approximation kreiszylindrischer Strukturen wie sie z. B. bei Durchkontaktierungen von Multi-Layer-Platinen auftreten. Ziel aller nachfolgenden Untersuchungen ist demnach, eine Kreiszyylinder-Diskretisierungsvorschrift zu finden, die bei kleinstmöglichem Kreiszyylinder-Approximationsaufwand (= kleinstmögliche Anzahl an Elementarquadern zur Kreiszyylinderapproximation) zu wahren feldnumerischen Ergebnissen (= Streumatrizen) führt im Rahmen der Gesamtstrukturgenauigkeit. Dies entspricht der Suche nach einer idealen Kreiszyylinder-Diskretisierungsvorschrift.

2.1 Kreiszyllindrische Struktur

Die für die nachfolgenden Untersuchungen verwendete kreiszylindrische Struktur ist in Abb. 2 dargestellt. Es handelt sich hierbei um einen dielektrischen Resonator hoher Dielektrizitätszahl — $\epsilon_r = 50.0$ — in Kreiszyylinderform.

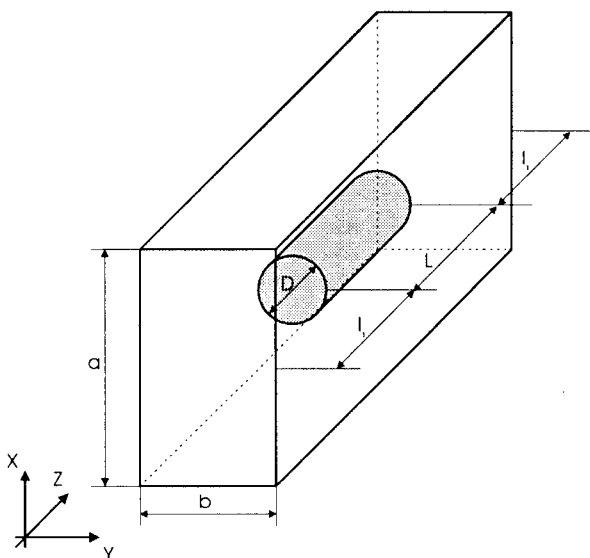
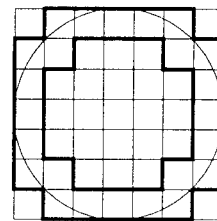


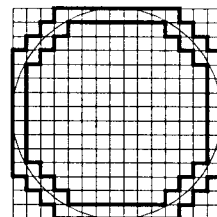
Abb. 2 Kreiszyllindrischer dielektrischer Resonator in einem Rechteckhohlleiter

2.2 Standard-Diskretisierungsvorschriften für kreiszylindrische Strukturen

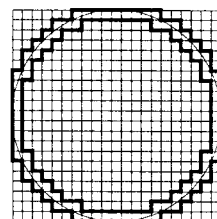
Bezüglich des in Abb. 2 dargestellten kreiszylindrischen Resonators werden zwei in ihren Eigenschaften grundsätzlich unterschiedliche Diskretisierungsarten analysiert: Relativ zum Kreis eingeschriebene und umschriebene Diskretisierungsarten mit steigender Anzahl an Unterteilungen (2D-Betrachtung, s. Abb. 3 a, b, c).



(a) Grundraaster : 7 Untert., Approximationsgitter n = 3 Untert.
 $\Delta A/A_{\text{Kreis}} = + 16.9 \%$ (umschrieben)
 $\Delta A/A_{\text{Kreis}} = - 45.4 \%$ (eingeschrieben)



(b) Grundraaster : 15 Untert., Approximationsgitter n = 7 Untert.,
 $\Delta A/A_{\text{Kreis}} = + 13.7 \%$ (umschrieben)
 $\Delta A/A_{\text{Kreis}} = - 18.0 \%$ (eingeschrieben)



(c) Grundraaster : 21 Untert., Approximationsgitter n = 11 Untert.,
 $\Delta A/A_{\text{Kreis}} = + 7.7 \%$ (umschrieben)
 $\Delta A/A_{\text{Kreis}} = - 15.4 \%$ (eingeschrieben)

Abb. 3 Darstellung der Diskretisierungsarten eingeschrieben und umschrieben mit steigender Anzahl an Unterteilungen — dargestellt ist jeweils nur der für den dielektrischen Resonator relevante Teil —, $\Delta A/A_{\text{Kreis}}$ gibt jeweils die Flächenabweichung zur Kreisfläche an

In Abb. 3 ist je Diskretisierungsart mit steigender Anzahl an Unterteilungen des Approximationsgitters die sich stetig verbessernde Annäherung an die Kreisform ersichtlich, die beim Grenzwert einer unendlichen Anzahl an Unterteilungen exakt die Kreisform annimmt.

2.3 Feldnumerische Ergebnisse der Standard-Diskretisierungsvorschriften

Die Hohlleiterabmessungen gemäß Abb. 2 werden so gewählt, daß im untersuchten Frequenzbereich nur die Grundmode ausbreitungsfähig ist. Die Struktur ist dann als elektrisches Zweitor anzusehen. Aufgrund der Symmetrie der Struktur, und der Tatsache, daß die Struktur als verlustfrei betrachtet wird, reduziert sich der Informationsgehalt der beschreibenden Streumatrix ausschließlich auf eine Größe, z. B. S_{11} . Ist weiterhin nur das Verhältnis der rück- und hinlaufenden Wirkleistungen, bezogen auf ein Tor von Interesse, so ist es ausreichend, ausschließlich z. B. die Größe $|S_{11}|$ zu betrachten.

Das qualitative Ergebnis $|S_{11}|(f)$ im Frequenzbereich 15 bis 25 GHz ist in Abb. 4 dargestellt.

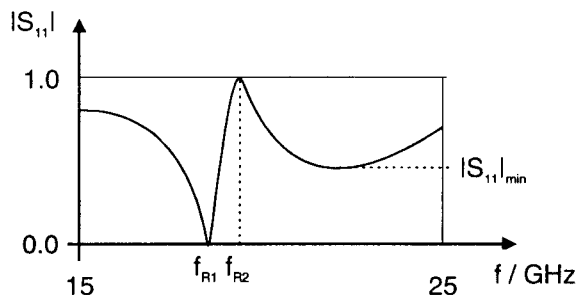


Abb. 4 Qualitative Darstellung $|S_{11}|(f)$ bezogen auf den kreiszylindrischen Resonator aus Abb. 2, charakteristische Merkmale : f_{R1} , f_{R2} , $|S_{11}|_{min}$

In Abb. 4 zeigen sich charakteristische Merkmale, die den Verlauf $|S_{11}|(f)$ maßgeblich bestimmen. Als charakteristische Merkmale sind zu nennen :

- f_{R1} : Resonanzfrequenz vom Typ $|S_{11}| = 0$
- f_{R2} : Resonanzfrequenz vom Typ $|S_{11}| = 1$
- $|S_{11}|_{min}$: Lokales Minimum von $|S_{11}|$

Wird z. B. die Diskretisierungsart umschrieben auf die in Abb. 2 dargestellte kreiszylindrische Struktur angewandt, so verschiebt sich Resonanzfrequenz f_{R1} bei steigender Anzahl von Unterteilungen zu höheren Werten unter Annäherung an einen Endwert. Die Untersuchung der Abhängigkeit des Merkmals f_{R1} bei der jeweiligen Kreiszylinderdiskretisierungsart von der Anzahl der Unterteilungen n zeigt Abb. 5. Die Verläufe $f_{R1}(n)$ und $|S_{11}|_{min}(n)$ sind qualitativ vergleichbar.

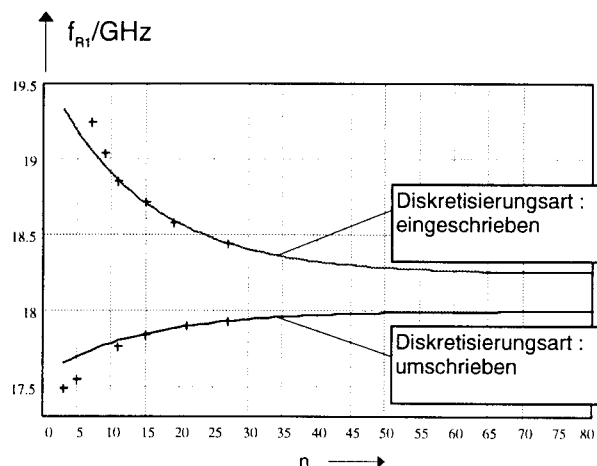


Abb.5 Darstellung $f_{R1}(n)$ mit zugehörigen Ausgleichs-e-Funktionen, Endwerte :

$$f_{R1, \text{end, eingeschrieben}} = 18.243 \text{ GHz}$$

$$f_{R1, \text{end, umschrieben}} = 17.997 \text{ GHz}$$

Aus Abb. 5 wird ersichtlich, daß Diskretisierungsart eingeschrieben grundsätzlich zu zu hohen, Diskretisierungsart umschrieben grundsätzlich zu zu niederen Resonanzfrequenzen führt. Mit steigender Anzahl von Unterteilungen konvergieren diese jedoch gegen einen jeweiligen Endwert. Die Ursache, daß diese Endwerte nicht übereinstimmen ist darin zu sehen, daß die Simulationenpunkte bereits bei $n = 27$ enden und zur Extrapolation eine Exponentialfunktion verwendet wurde. Tatsächlich liegen diese Punkte nicht auf einem derartigen Funktionsverlauf, auch wenn die Eigenschaft des Anstrebens eines Endwertes vorliegt. Aus diesem Grund wurden nur diejenigen Punkte zur Berechnung einer jeweiligen Ausgleichsexponentialfunktion verwendet, die möglichst in der Nähe des entsprechenden Endwertes liegen. Aufgrund der gegensätzlichen Eigenschaften der verwendeten Diskretisierungsarten (eingeschlossenes Volumen) läßt sich jedoch eine wichtige Schlussfolgerung gewinnen :

Der wahre unbekannte Wert $f_{R1, \text{wahr}}$ liegt zwischen den Endwerten $f_{R1, \text{end, eingeschrieben}}$ und $f_{R1, \text{end, umschrieben}}$. Diese stellen somit die äußeren Grenzen für den wahren Wert innerhalb eines gewissen Fehlerintervalls dar.

2.4 Optimierte Diskretisierungsvorschriften für kreiszylindrische Strukturen

2.4.1 Ansatz

Als Ansatz zum Auffinden optimierter Diskretisierungsvorschriften für kreiszylindrische Strukturen dient eine geometrisch variable Diskretisierungsvorschrift mit konstant 3 Unterteilungen (s. Abb. 6).

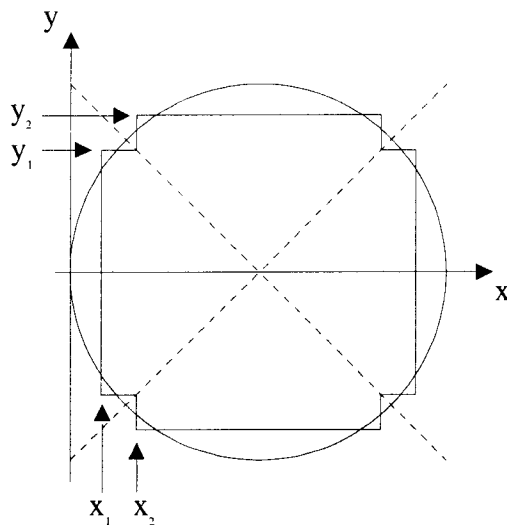


Abb. 6 Geometrisch variable Diskretisierungsvorschrift für kreiszylindrische Strukturen, 3 Unterteilungen – dargestellt ist nur der für den dielektrischen Resonator relevante Teil –

Aus Abb. 6 ergeben sich zunächst 4 Gittervariable x_1 , x_2 , y_1 , y_2 , die sich unter Berücksichtigung der $45^\circ/135^\circ$ -Achsensymmetrie auf 2 reduzieren lassen: x_1 , y_1 . Sind diese Gittervariable bekannt, so kann auch die eingeschlossene Fläche bzw. die Abweichung zur Kreisfläche berechnet werden. Aus Gründen der Anschaulichkeit werden deshalb äquivalent zu x_1 und y_1 die Größen $\Delta A/A_{\text{Kreis}}$ und y_1/R_{Kreis} variiert und die Abhängigkeit der Merkmale f_{R1} , f_{R2} und $IS_{11|\min}$ untersucht. Die Untersuchung bezieht sich auch hier auf die kreiszylindrische Struktur aus Abb. 2.

2.4.2 Feldnumerische Ergebnisse der geometrisch variablen Diskretisierungsvorschrift

Die Merkmale f_{R1} , f_{R2} und $IS_{11|\min}$ (vgl. Abb. 4) sind hier abhängig von den 2 variierten Gittergrößen $\Delta A/A_{\text{Kreis}}$ und y_1/R_{Kreis} bei konstant 3 Unterteilungen. Die sich für die

Resonanzfrequenz f_{R1} ergebende Flächenfunktion ist in Abb. 7 über dem relevanten Grundgebiet dargestellt. Die Flächenfunktion wurde durch Ausgleichung im Gaußschen Sinne, d. h. durch Minimierung der quadratischen Abweichungen von den Simulationspunkten bestimmt.

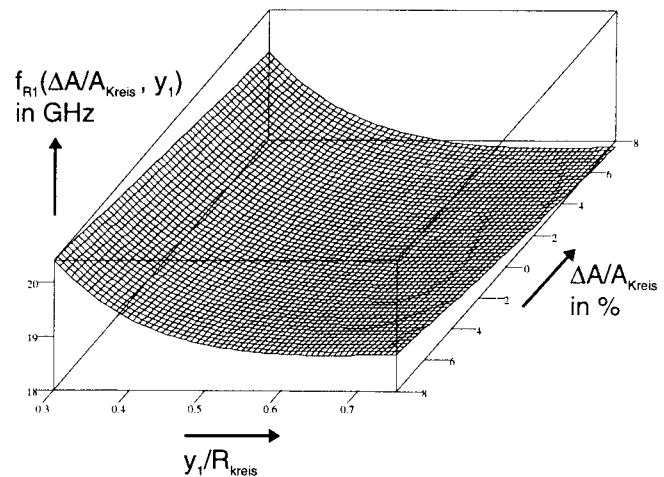


Abb. 7 Darstellung der Ausgleichsfächenfunktion $f_{R1}(\Delta A/A_{\text{Kreis}}, y_1)$

In Abb. 7 zeigt die Flächenfunktion $f_{R1}(\Delta A/A_{\text{Kreis}}, y_1)$ eine halbröhrenförmige Krümmung entlang der (y_1/R_{Kreis}) -Achse und eine näherungsweise lineare Neigung entlang der $(\Delta A/A_{\text{Kreis}})$ -Achse. Die Flächenfunktionen $f_{R2}(\Delta A/A_{\text{Kreis}}, y_1)$ und $IS_{11|\min}(\Delta A/A_{\text{Kreis}}, y_1)$ sind qualitativ mit Abb. 7 vergleichbar. Sie weisen jedoch eine veränderte Krümmung und Neigung auf. Um die notwendigen Eigenschaften zu finden, die eine optimierte Diskretisierungsvorschrift mit 3 Unterteilungen aufweisen muß, kann die funktionale Abhängigkeit $\Delta A/A_{\text{Kreis}}(y_1/R_{\text{Kreis}})$ bei den Flächenfunktionen f_{R1} , f_{R2} und $IS_{11|\min}$ unter der Bedingung, daß der jeweilige wahre Wert erreicht wird, untersucht werden. Dies sind Höhenlinien der drei Flächenfunktionen. Die wahren Werte $f_{R1, \text{wahr}}$, $f_{R2, \text{wahr}}$ und $IS_{11|\min, \text{wahr}}$ bezogen auf die kreiszylindrische Struktur aus Abb. 2 sind jedoch nicht bekannt. Andererseits sind die äußeren Grenzen gegeben innerhalb eines gewissen Fehlerintervalls durch die Diskretisierungsarten eingeschrieben und umschrieben (vgl. Abb. 5), innerhalb derer sich die wahren Werte maximal bewegen. Aus diesem Grunde werden diejenigen Höhenlinien betrachtet, die sich auf die jeweiligen Endwerte der entsprechenden Diskretisierungsart beziehen (= äußere Grenzen für wahre Werte) anstelle auf die wahren Werte selbst. Die Zusammenstellung der Ergebnisse kann Abb. 8 entnommen werden.

Aus Abb. 8 lassen sich folgende Erkenntnisse gewinnen :

- Es existiert keine ideale Diskretisierungsvorschrift für kreiszylindrische Strukturen, da sich nicht alle 3 Bereiche der Höhenlinien, innerhalb deren sich die wahren Höhenlinien bewegen, überlappen. Eine ideale Diskretisierungsvorschrift muß bei jeder Anzahl von Unterteilungen zu den auf die Struktur bezogenen wahren Werten $f_{R1, \text{wahr}}$, $f_{R2, \text{wahr}}$ und $|S_{11}|_{\text{min, wahr}}$ im Rahmen der Gesamtungenauigkeit führen.
- Es existieren unendlich viele optimale Kreiszyylinder-Diskretisierungsvorschriften, da diese je nach Gewichtung entweder den wahren Wert $f_{R1, \text{wahr}}$ oder $f_{R2, \text{wahr}}$ oder $|S_{11}|_{\text{min, wahr}}$ besser annähern.
- Die Diskretisierungsvorschriften eingeschrieben bzw. umschrieben sind nicht optimal, da sie eine Flächenabweichung zur Kreisfläche bei 3 Unterteilungen von - 45.4 % bzw. + 16.9 % aufweisen und somit außerhalb der äußersten Höhenlinien liegen.

- Eine optimale Diskretisierungsvorschrift muß alle Merkmale f_{R1} , f_{R2} , $|S_{11}|_{\text{min}}$ gleichgewichtig berücksichtigen und darf deshalb nicht außerhalb der äußeren Höhenlinien liegen. Weiterhin zeigen alle Höhenlinien über y_1 hinweg eine Annäherung, so daß sich folgende Bedingungen für eine optimierte Kreiszyylinder-Diskretisierungsvorschrift mit 3 Unterteilungen ergeben :

$$-10.0 \% < \frac{\Delta A}{A_{\text{Kreis}}} < +5.5 \% \quad (1)$$

$$60 \% < \frac{y_1}{R_{\text{Kreis}}} < 70 \% \quad (2)$$

2.4.3 Angabe einer optimierten Kreiszyylinder-Diskretisierungsvorschrift

Gewählt wurde unter Berücksichtigung der Bedingungen (1) und (2) eine Kreiszyylinder-Diskretisierungsvorschrift mit beliebiger Anzahl von Unterteilungen, die folgende Eigenschaften aufweist :

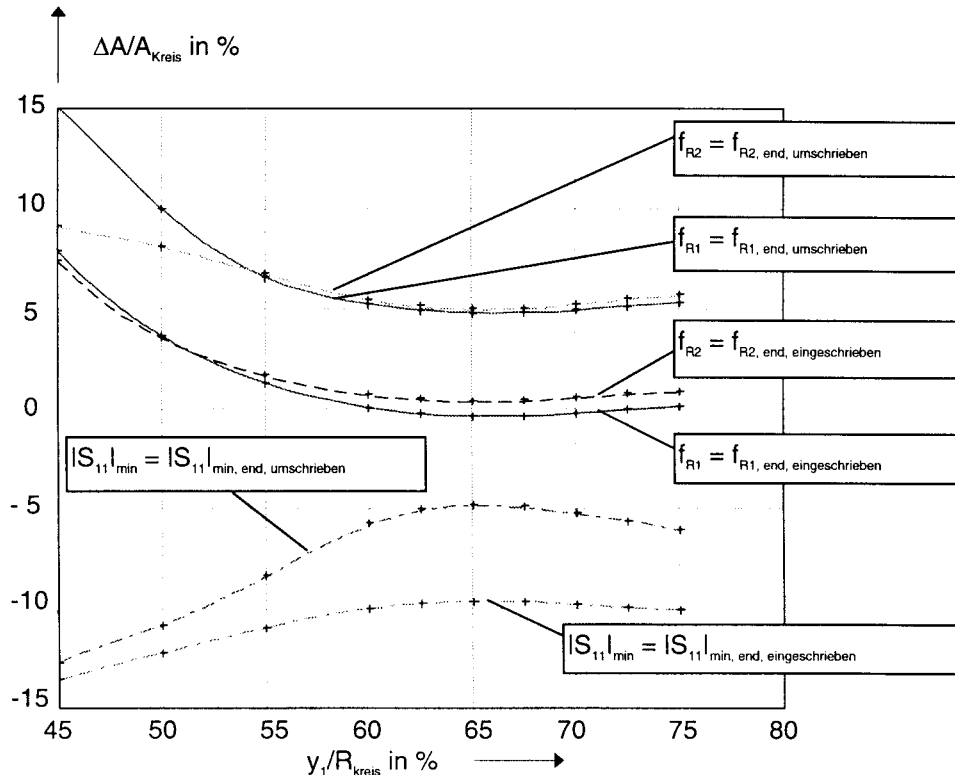
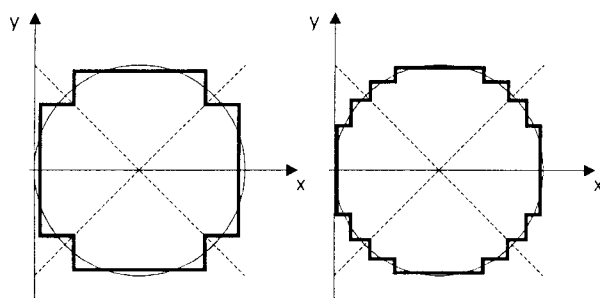


Abb. 8 Höhenlinien der Flächenfunktionen f_{R1} , f_{R2} , $|S_{11}|_{\text{min}}$

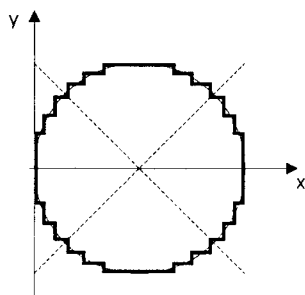
- Flächengleich mit Kreisfläche : $\Delta A/A_{\text{Kreis}} = 0 \%$, da jede Diskretisierungsvorschrift bei unendlich großer Anzahl von Unterteilungen auf Flächengleichheit mit der Kreisfläche führt.
- 45°-/135°-Achsensymmetrie aufgrund der Symmetrie des Kreises
- Mittiges Schneiden aller Treppenfunktionsabschnitte durch die Kreisfunktion, Ausnahme : Die äußeren zweifach geschnittenen Abschnitte, die zum Erreichen der Flächengleichheit entsprechend gewählt werden müssen. Bei 3 Unterteilungen ergibt sich dann :
 $y_1 = 62.3 \% R_{\text{Kreis}}$

Abb. 9 zeigt diese optimierte Diskretisierungsvorschrift mit 3, 7 und 11 Unterteilungen.



(a) 3 Unterteilungen

(b) 7 Unterteilungen



(c) 11 Unterteilungen

Abb. 9 Optimierte Kreiszyylinder-Diskretisierungsvorschrift

2.4.4 Feldnumerische Ergebnisse der optimierten Kreiszyylinder-Diskretisierungsvorschrift

Die Anwendung der unter 2.4.3 gefundenen optimierten Kreiszyylinder-Diskretisierungsvorschrift auf die in Abb. 2 dargestellte kreiszylindrische Struktur zeigt Abb. 10 im Vergleich mit den Diskretisierungsarten eingeschrieben und um-

schrieben. Die Resonanzfrequenz f_{R1} nähert sich bei steigender Anzahl von Unterteilungen rasch einem Endwert, der wie zu erwarten im Bereich zwischen den Endwerten durch die um- und eingeschriebenen Diskretisierungsarten liegt. Hierbei wird der nicht-ideale, jedoch optimierte Charakter ersichtlich. Die Diagramme für die Merkmale f_{R2} und $IS_{11|_{\min}}$ sind vergleichbar.

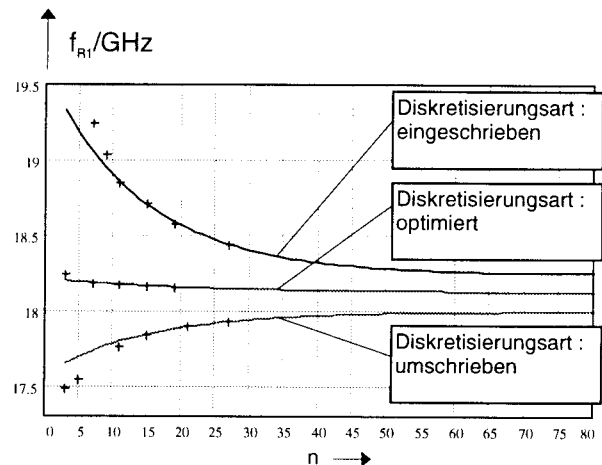


Abb.10 Darstellung $f_{R1}(n)$, n : Anzahl an Unterteilungen

Literatur

- [1] M. Buchner, "Die Methode der Finiten Elemente zur Berechnung elektromagnetischer und mechanischer Felder in Filter-Bauelementen", Reihe 21 : Elektrotechnik, Nr. 106; Düsseldorf : VDI-Verlag 1992
- [2] A. Christ and H. L. Hartnagel, "Three-Dimensional Finite-Difference Method for the Analysis of Microwave-Device Embedding", IEEE Trans. Microwave Theory Tech., vol. 35, No. 8, August 1987
- [3] A. Christ "Stromatrixberechnung mit dreidimensionalen Finite-Differenzen für Mikrowellen-Chip-Verbindungen und deren CAD-Modelle", Darmstädter Dissertation D17; Düsseldorf : VDI-Verlag 1988
- [4] K. Beilenhoff, W. Heinrich, and H.L. Hartnagel, "Improved Finite-Difference Formulation in Frequency Domain for Three-Dimensional Scattering Problems", IEEE Trans. Microwave Theory Tech., vol. 40, No. 3, March 1992
- [5] A. Christ "Reflexionsverhalten von Verbindungsstrukturen für Mikrowellenbauteile", Frequenz 42 (1988) 1, S. 7-13
- [6] A. Christ, J. M. Dieudonne, and V. Krozer, "Accurate Characterization of DR Coupling with new General-Purpose CAD Program", Electronic Letters 25 (1989) 16, S. 1026-1028
- [7] A. Christ, R. Dill, and M. Buchner, "Rigorous Analysis of Ceramic Band-Pass Filters", Archiv für Elektronik und Übertragungstechnik 46 (1992) 1, S. 58-60

ATM Messapplikationen in programmierbarer Logik

Beschreibung geeigneter Verfahren und Umsetzung in ein FPGA

Dipl. Ing. F. Stockmayer
Wandel & Goltermann GmbH
Tel.: 07121-86-1811
E-Mail: friedemann.stockmayer@wago.de

1 Grundlagen

Breitbandnetze, die auf der Übertragung von ATM-Zellen beruhen, unterliegen starken Anforderungen bezüglich der zur Verfügung gestellten Leistungsfähigkeit. Beide Seiten, sowohl die Netzbetreiber als auch der potentielle Anwender 'am anderen Ende' der Übertragungsstrecke, haben großes Interesse an einer fehlerfreien Übertragung. Mit Hilfe von sog. Netzperformance-Messungen ist eine Aussage über die Güte des angebotenen Breitbanddienstes, z.B. Videoübertragung oder File-Transfer, möglich. Dazu sind zunächst folgende, die Performance bestimmende, Kategorien zu unterscheiden:

- Innerhalb der erlaubten Maximalzeit erfolgreich übertragene Zelle (Bitfehlerfrei)
- Fehlerbehaftete Zelle (Errored Cell)
- Verlorene Zelle (Lost Cell)
- Fälschlicherweise eingefügte Zelle (Misinserted Cell)

Daraus können geeignete Parameter zur Qualitätsbeurteilung abgeleitet werden (vgl. Tab. 1).

Tabelle 1: Performance Parameter

Parameter	Beschreibung
Cell Loss Ratio	Verhältnis „verlorener Zellen“ zur Gesamtanzahl gesendeter bzw. empfangener Zellen
Cell Misinserted Rate	Anzahl falsch eingefügter Zellen pro Sekunde
Cell Error Ratio	Verhältnis fehlerbehafteter Zellen zur Gesamtanzahl übertragener Zellen
Cell Transfer Delay	Übertragungsverzögerung einer Zelle
2-Punkt Cell Delay Variation	Variation der Übertragungsverzögerung von Zellen zwischen zwei Messpunkten

Diese Qualitätsmerkmale sind vor allem bei der Netz-Inbetriebnahme, zur Systemabnahme und während der ständigen Überwachung von ATM Netzen interessant.

Bevor die Frage beantwortet werden kann, wie diese Parameter zu messen sind, sollen hier wesentliche Details einer ATM-Zelle in Erinnerung gerufen werden (s.a. Literaturangabe /1/).

1.1 Aufbau einer ATM-Zelle

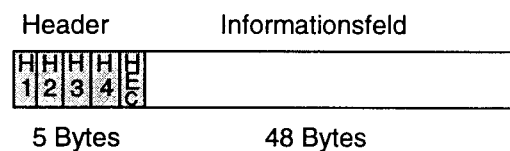


Bild 1: Aufbau einer ATM-Zelle

In den ersten vier Bytes des Headers (H1, H2, H3, H4) ist hauptsächlich die Routing Information eingetragen, also eine Angabe zur virtuellen Kanal- und Pfadzugehörigkeit dieser Zelle. Der Pfad ist hierbei ein Bündel separater Kanäle, welcher einen vorgegebenen Weg durchs Netzwerk nimmt (vgl. Bild 2). Der Begriff „virtuell“ macht deutlich, dass die Kanäle und deren Gruppierung nur über sog. Identifier (Virtual Channel - und Virtual Path Identifier, VCI, VPI) im Header der Zelle erkennbar sind, obwohl alle Zellen dieselbe physikalische Übertragungsstrecke verwenden.

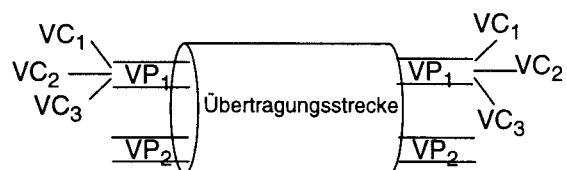


Bild 2: Virtueller Kanal und virtueller Pfad

Weil diese Angaben im Kopf der Zelle lebensnotwendig sind, sind die ersten vier Bytes durch einen CRC-8 geschützt, welcher im 5. Headerbyte (Header Error Control, HEC) übertragen wird. Die CRC-Prüfsumme dient gleichzeitig als Synchronisationskriterium auf der Empfangsseite.

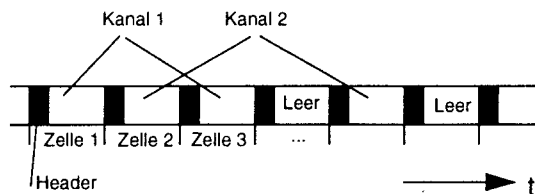


Bild 3: ATM Zellenstrom

1.2 Testzelle nach ITU-T O.191

Um die Parameter in Tabelle 1 auch während des realen Betriebes messen zu können, wird nun ein spezieller Kanal als Testkanal eingerichtet, welcher Testzellen mit besonderem Inhalt überträgt (vgl. Bild 4).

Auf der Sendeseite wird in das Informationsfeld der Testzelle eine Sequenznummer (fortlaufende Nummer, 32-Bit) eingetragen sowie eine relative Uhrzeit (Zeitstempel), welche den Absendezeitpunkt dieser Zelle festhält. Um Bitfehler festzustellen, die auf der Strecke den Inhalt beeinflussen können, ist das Informationsfeld mit einem CRC-16 gesichert.

Damit sich diese doch ziemlich statisch wirkende Zellen (viele Bytes sind unbenutzt, d.h. 0x00) nicht vom lebenden Verkehr unterscheiden, ist das Informationsfeld, mit Ausnahme des CRC-16, verwürfelt (scrambled).

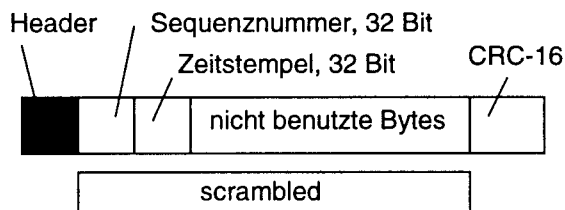


Bild 4: Testzelle nach O.191

Beobachtet man nun auf der Empfangsseite fortlaufend alle Zellen des Testkanals (identifizierbar am VCI/VPI), können anhand der Sequenznummer verlorene oder fälschlicherweise eingefügte Zellen erkannt werden.

Zellen können beispielsweise infolge FIFO-Überläufe im System verloren gehen oder aufgrund einer Prioritätsentscheidung bei der Frage, ob eine Zelle bei Systemüberlastung verworfen werden darf.

Zellen, die ursprünglich einem anderen Kanal zugeordnet waren, jetzt aber im Testkanal auftauchen (misinserted), entstehen unterwegs entweder durch falsche Headereinträge in einem Switch des Netzwerks oder infolge nicht erkannter bzw. falsch korrigierter Headerfehler.

Anhand des Zeitstempels ist eine Berechnung der

Übertragungsverzögerung möglich unter der Voraussetzung, dass die Sende- und Empfangsuhr synchron zueinander laufen. Diese Bedingung ist normalerweise am einfachsten dadurch zu erfüllen, dass Sende- und Empfangseinrichtung im gleichen Meßgerät vor Ort verwendet werden und die Messung über eine Schleife am fernen Ende durchgeführt wird.

Durch Nachrechnen des CRC-16 im Informationsfeld und Vergleich mit der ankommenden Prüfsumme ist ein Bitfehler sofort erkennbar und lässt eine Aussage über die Cell Error Ratio zu.

2 Realisierung

Die hier beschriebenen Messapplikationen sind eingebettet in eine Kette von FPGAs. Angelehnt an das OSI-Schichtenmodell hat jedes FPGA die Aufgabe die jeweilige Schicht (physikalische Schicht, ATM-Layer oder höhere Protokollebenen) zugänglich zu machen.

Für das vorliegende Konzept steht ein Baustein von Altera, FLEX10K50 zur Verfügung. Unter Ausnutzung der „In System Programmierbarkeit“ sind die Messapplikationen in jeweils einem Design realisiert und per Software entsprechend der Anwendung konfigurierbar.

Dieser Ansatz hat gegenüber einer Realisierung mit einem ASIC den entscheidenden Vorteil, dass auf eine Veränderung der Anforderung, z.B. Normänderung oder zusätzliche Funktionen sehr schnell reagiert werden kann.

Gerade dort, wo mit einem raschen Technologiewandel zu rechnen ist, empfiehlt sich eine weitgehend flexible Gestaltung der Hardware.

Somit reduziert sich auch die Bereitstellung neuer „Hardwarefunktionen“ auf ein Software-Upgrade, was die geänderten Programmierfiles der FPGAs enthält. Ein neues Board-Layout oder ASIC-Redesign entfällt.

Alle Designs sind mit der Hardwarebeschreibungssprache VHDL erstellt.

Die maximale Taktfrequenz beträgt 21 MHz.

Der nachfolgende Teil beschreibt anhand eines Blockdiagrammes die wichtigsten Bearbeitungsstufen die notwendig sind, um zum gewünschten Ergebnis zu kommen:

- Jeweils ein Zählkriterium für Lost- Misinserted- und Errored Cells im ersten Design
- Eine geeignete Darstellung der Cell Transfer Delay im zweiten Design.

2.1 Das Konzept

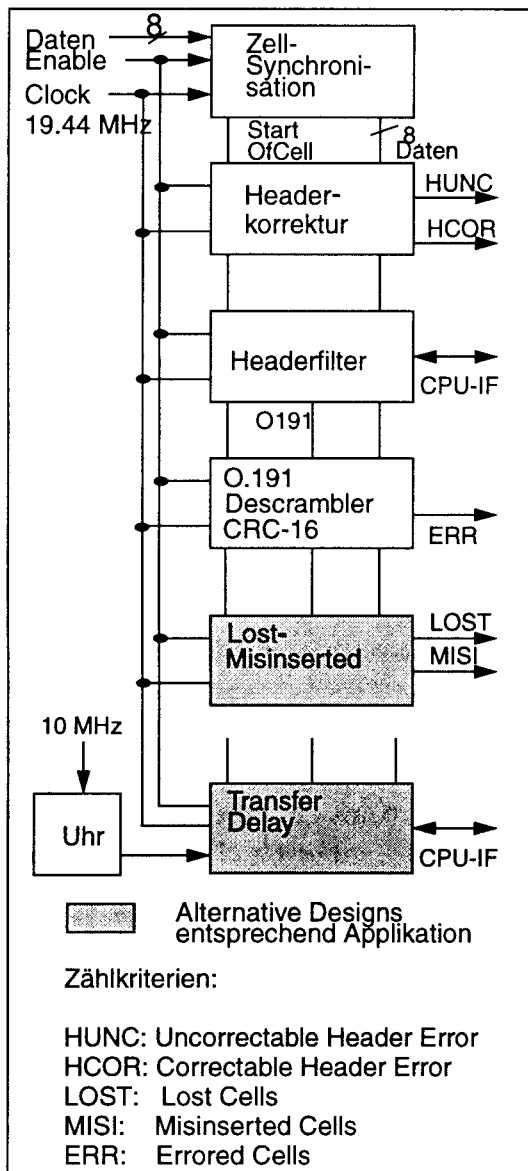


Bild 5: Blockdiagramm

2.2 Zell Synchronisation

Der ankommende Datenstrom ist bereits Byteausgerichtet. Diese Voraussetzung ist deshalb erfüllt, weil ATM-Zellen meistens in einem Synchronen Transport Modul (STM-1 oder STM-4) der synchronen digitalen Hierarchie (SDH) übertragen werden. Dabei handelt es sich um eine byteorientierte Rahmung gemäß ITU-T Empfehlung G.707. Die genannte Eigenschaft vereinfacht die Suche nach dem Zellstart (1. Headerbyte). Jedes ankommende Byte wird in ein 5-Byte tiefes Schieberegister geschoben und gleichzeitig über die letzten vier Bytes die Prüfsumme des CRC-8 berechnet und mit dem Inhalt des 5. Bytes im Schieberegister verglichen (s.a Abschnitt 6, Anhang).

Eine Synchronisation erfolgt jedoch erst dann, wenn nach einer ersten Übereinstimmung anschließend jeweils im Zellabstand (53-Bytes) sieben weitere Prüfsummen mit dem Erwartungswert übereinstimmen. Dabei werden drei Fälle unterschieden, die einerseits Auswirkung auf die Synchronisation haben und andererseits eine mögliche Korrektur eines Einzelfehlers in den Headerbytes anzeigen (HCOR oder HUNC):

- Exakte Übereinstimmung
- Abweichung in *einem* Bit des HEC Fehler ist korrigierbar (HCOR)
- Abweichung in mehreren Bits des HEC Fehler sind nicht korrigierbar (HUNC)

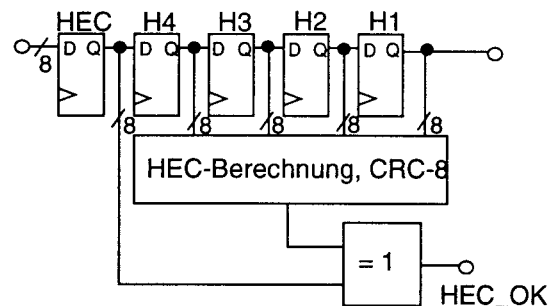


Bild 6: Schaltungsprinzip zur Synchronisation
Beispiel: Kompletter Header im Register

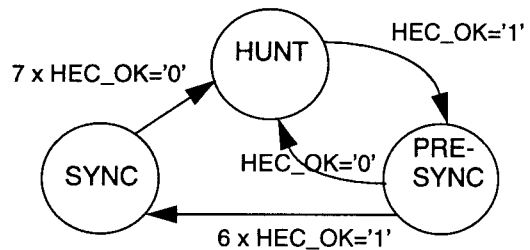


Bild 7: Zustandsautomat zur Synchronisation

In der Praxis zeigt sich häufig ein büschelartiges Vorkommen von Bitfehlern, was i.a. zu mehreren beschädigten Zellen führt. Um das Systemverhalten in diesem Fall etwas zu beruhigen, werden prinzipiell alle Zellen mit einem nicht korrigierbaren Headerfehler und alle fehlerhaften Zellen unmittelbar nach einem korrigierbaren Fehler verworfen (vgl. Automat in Bild-8).

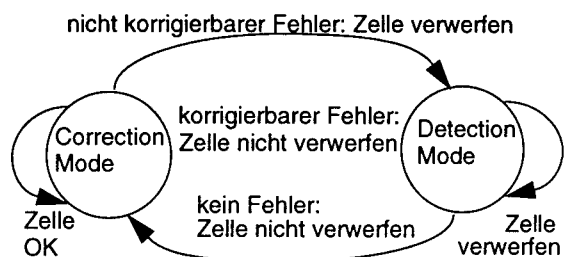


Bild 8: Zustandsautomat zur Headerkorrektur

2.3 Headerfilter

Das Headerfilter hat die Aufgabe, alle dem Testkanal zugehörigen O.191-Zellen zu identifizieren, d.h. es sucht nach einem vorgegebenen Wert (VPI, VCI) in H1 ... H4 und markiert die Zelle (Signal O191 in Bild 5).

Zwei Varianten bieten sich an:

- festverdrahteter Komparator (4 x 8 Bit)
- mit RAMs aufgebaute Look-Up-Table

Für evt. später hinzukommende Applikationen eignet sich der etwas allgemeinere Ansatz einer Look-Up-Table, unter Verwendung der bereits im Baustein vorhandenen Embedded Array Blöcke (RAM). Weil das RAM aus Softwaresicht frei programmierbar ist, erlaubt diese Lösung auch das Setzen von Don't Cares auf beliebige Bits im Suchmuster. Pro Filterkanal sind vier RAMs der Größe 256 x 1 notwendig.

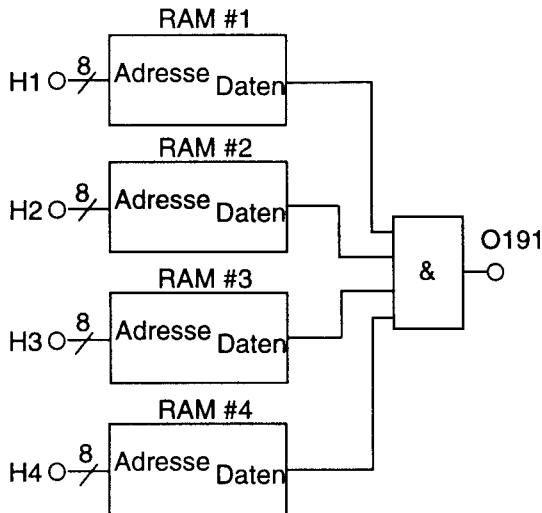


Bild 9: Prinzip der Look-Up-Table

Ist unter der Adresse, die dem Vergleichsmuster im jeweiligen Byte entspricht, eine log. '1' von der Software im RAM gespeichert, so gilt die Zelle als identifiziert, wenn alle RAM-Ausgänge gleichzeitig logisch '1'-Pegel annehmen. Alle anderen Adressen im RAM enthalten eine log. '0'.

2.4 CRC-16 und Descrambler

Nach dem Headerfilter sind nun die Zellen mit einem Start-Of-Cell Trigger (SOC) und einem Filterindikator (O191) versehen (vgl. Bild 10).

Bevor allerdings die Auswertung der Sequenznummer bzw. des Zeitstempels erfolgen kann, muß der CRC-16 berechnet und die Verwürfelung des Informationsfeldes rückgängig gemacht werden. Ein erkannter Fehler in der CRC-Prüfsumme ist als Zählkriterium für „Errored Cells“ zu werten

und der Software zugänglich zu machen.

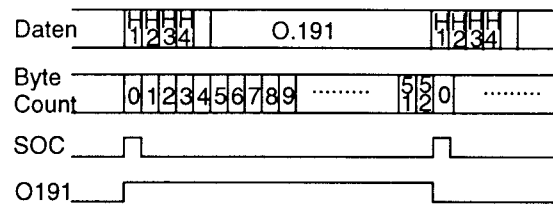


Bild 10: Impulsdiagramm nach Headerfilter

Scrambler - Polynom: $1 + x^5 + x^9$
CRC-16 Polynom: $1 + x^5 + x^{12} + x^{16}$

Im Anhang dieses Beitrages ist am Beispiel der Fehlerkorrektur im HEC ein Verfahren zur CRC-Berechnung beschrieben (s.a. 'Scrambling and CRC' im Annex-C der ITU-T Empfehlung O.191).

2.5 Zeitstempel

Sender und Empfänger haben idealerweise Zugriff auf eine gemeinsame Uhr, in diesem Fall ein 32-Bit Zähler mit einer Auflösung von 10 ns, was einer Taktfrequenz von 100 MHz entspricht. Hohe Taktfrequenzen dieser Größenordnung sind derzeit innerhalb eines FPGAs nicht ohne weiteres realisierbar. Andererseits beträgt die Systemtaktfrequenz ca. 20 MHz. Eine zeitliche Auflösung im o.g. Bereich erscheint daher ohnehin nicht sinnvoll. Ein günstiger Kompromiß bietet sich mit einer Taktfrequenz von 10 MHz an, wobei in jedem Takt der Zähler ein Inkrement von 10 ausführt.

An dieser Stelle entsteht nun allerdings ein Problem dadurch, dass zwei voneinander unabhängige Takte in einer sonst synchronen Verarbeitung aufeinanderstoßen (unsichere Abtastungen durch Verletzung der Setup- oder Hold Zeit, bzw. metastabile Zustände im Flip-Flop). Ein bidirektionaler Handshake entschärft diese Situation:

Dabei speichert die Uhr auf Anforderung die aktuelle Zeit in einem Übergaberegister und meldet die Verfügbarkeit des neuen Wertes zurück. Beide Signale werden auf den jeweils anderen Takt auf-synchronisiert (vgl. Kap. „Taktsynchrone Designs der Vorlesung CAE“, /5/).

3 Bestimmung der Cell-Transfer-Delay - Zeiten

Mit jeder ankommenden Zelle wird die aktuelle Uhrzeit festgehalten (vgl. Abschnitt 2.5). Um die Übertragungsverzögerung zu bestimmen, ist davon der Zeitstempel aus dem Informationsfeld der O.191-Zelle abzuziehen. Eindeutige Ergebnisse liefert die Differenz bis etwa 42.9s ($2^{32}-1 \times 10$ ns). Darüberhinaus werden die Messergebnisse mehrdeutig, aufgrund eines Überlaufs im Zähler der Uhr.

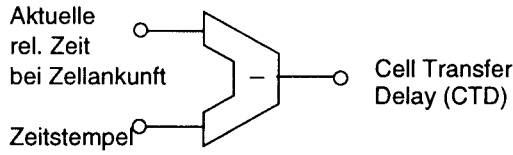


Bild 11: Übertragungsverzögerung als Differenz

3.1 Aufbereitung und Darstellung der Ergebnisse

Mit einer Datenflut an Ergebnissen, die hier entsteht, kann man allerdings zunächst nur wenig anfangen oder belastet zusätzlich kostbare Rechenzeit der CPU. Besser wäre eine Komprimierung der Ergebnisse bereits in der Hardware, z.B. aufgebaut als Histogramm in weiteren RAM-Blöcken des Bausteins.

Die CPU liest dann einfach zyklisch (100 ms) das RAM aus und bringt die Ergebnisse graphisch zur Anzeige.

Das Prinzip:

Die Übertragungsverzögerung bildet die Adresse des Histogramm-Speichers. Unter dieser Adresse wird die Anzahl Zellen abgespeichert mit genau dieser Übertragungsverzögerung.

Hinter jedem Balken des Histogramms verbirgt sich ein 16-Bit Zähler, der den Auslesezyklus (100 ms) überbrücken kann. Der Messbereich und die Auflösung (Zoom-Funktion) soll einstellbar sein.

Die Vereinfachung:

Zusammengefasst zu „Delay-Klassen“ reduziert sich die Vielzahl diskreter Werte zu einer überschaubaren Anzahl. Gewählt werden 128 Klassen, wobei jede Klasse 16-Bit Speicherplatz beansprucht, sodass in einem RAM-Block das gesamte Histogramm Platz findet.

Mit den beiden o.g. Punkten ergibt sich nun folgendes Verfahren:

Aus der berechneten Übertragungsverzögerung einer Zelle werden nach Abzug eines programmierbaren Offsets (Festlegen des Messbereichs) 7-Bit als RAM Adresse herangezogen. Die ebenfalls einstellbare Lage innerhalb des 32-Bit-Wertes bestimmt die Auflösung.

Der Inhalt unter dieser Adresse wird ausgelesen, inkrementiert und unter gleicher Adresse zurückgeschrieben.

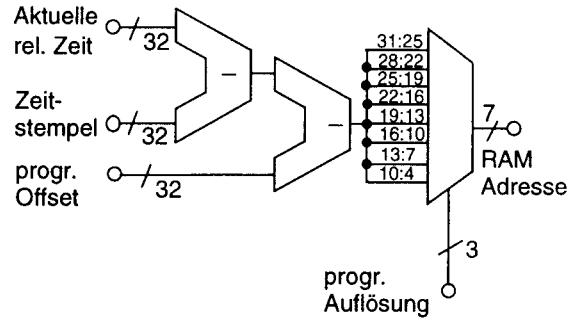


Bild 12: Berechnung der RAM-Adresse

Der Anwender kann also unter 8 Auflösungen auswählen und mit Hilfe des Offsets einen bestimmten Messbereich zwischen 0...42.9s herauszoomen.

Tabelle 2: Zoom-Funktion

RAM-Adresse	Auflösung	Messbereich (Offset = 0)
10:4	160 ns	0-20.47 µs
13:7	1.28 µs	0-163.83 µs
16:10	10.24 µs	0-1.31071 ms
19:13	81.92 µs	0-10.48575 ms
22:16	655.36 µs	0-83.88607 ms
25:19	5.24288 ms	0-0.67108863 s
28:22	41.94304 ms	0-5.36870911 s
31:25	0.33554432s	0-42.9496729 s

Beispiel:

Eine Messung mit einer Auflösung von 81.92µs (Offset=0) ergibt folgendes aktuelle Bild: Die einzige Säule im Histogramm liegt bei 4.9152 ms (Delayklasse 60) und soll mit einer Auflösung von 160 ns näher betrachtet werden. Welcher neue Offset muß programmiert werden, damit das Maximum etwa in der Mitte der Darstellung liegt ?

Die Transformationsgleichung lautet:

$$(KI \cdot Aufl + Offset)_{neu} = (KI \cdot Aufl + Offset)_{aktuell} - CTD$$

$$Offset_{neu} = (KI \cdot Aufl + Offset)_{aktuell} - (KI \cdot Aufl)_{neu}$$

$$Offset_{neu} = (60 \cdot 81.92\mu s + 0) - (64 \cdot 160 \text{ ns})$$

KI: Delayklasse; Aufl: Auflösung

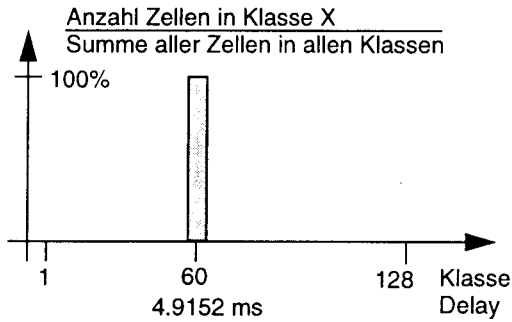


Bild 13: Darstellung mit Offset = 0

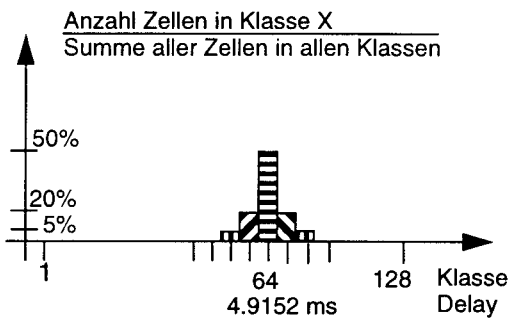


Bild 14: Darstellung mit Offset = 4.90496 ms

3.2 Schreiben und Lesen des Histogramm Speichers

Während die CPU daran interessiert ist, laufend den aktuellen Stand des Histogramms auszulesen („wachsende Säulen in der Graphik“) muß ein anstehendes Ergebnis konfliktfrei ins RAM geschrieben werden können. Diese Aufgabe übernehmen zwei voneinander abhängige Zustandsautomaten (Bild 15 u. 16). Die CPU kann damit nur indirekt über zwei Lesetore auf den RAM-Inhalt zugreifen:

- Register für Klassenindex
- Register mit Inhalt dieser Klasse

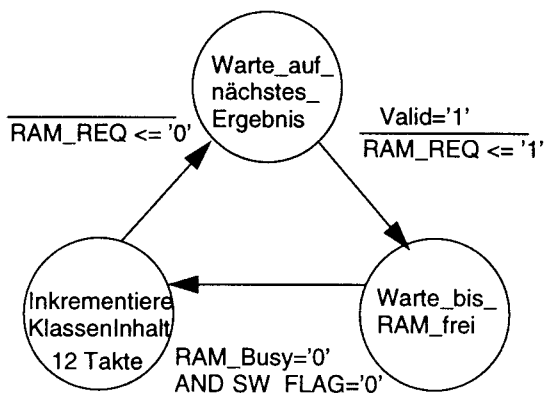


Bild 15: Übertragung Messergebnis ins RAM

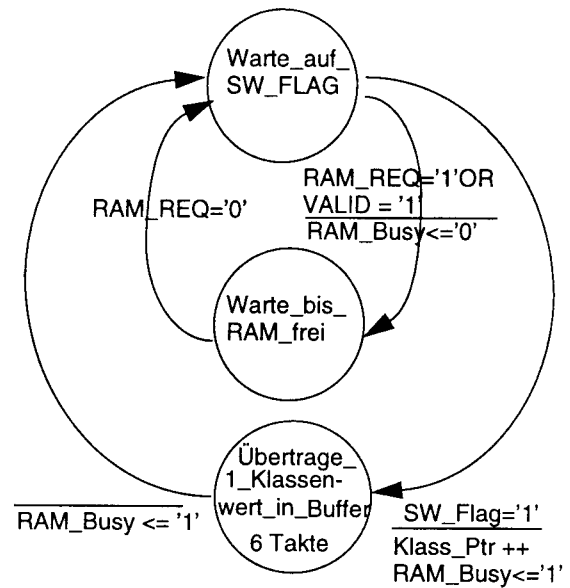


Bild 16: Auslesen des RAMs durch CPU

Über ein Schreibtor (SW_Flag) teilt die CPU dem Automaten in Bild 16 mit, dass sie beide Register ausgelesen hat und pollt anschließend das Register mit dem Klassenindex, um festzustellen, wann ein neues Wertepaar vorliegt. Der Automat seinerseits erkennt das Signal 'SW_Flag', erhöht den Klassenindex (Klass_Ptr) und nimmt mittels eines Semaphor (RAM_Busy) das RAM exklusiv für sich in Anspruch, um den nächsten Klasseninhalt in das Register zu übertragen.

Nach Ablauf der Übertragung wäre das RAM wieder frei, um z.B. den Inhalt einer Klasse aufgrund eines neuen Messergebnisses (angezeigt über Valid='1') zu inkrementieren. Der Automat in Bild 15 meldet zunächst diesen Vorgang an (RAM_REQ <= '1') und wartet bis das RAM verfügbar ist

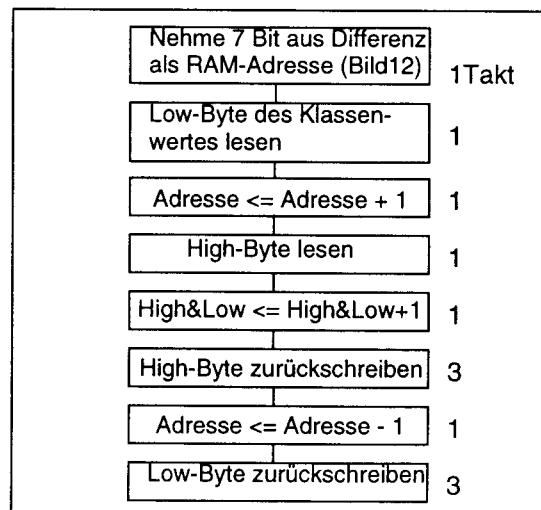


Bild 17: Aktionen im Zustand „Inkrementiere Klasseninhalt“

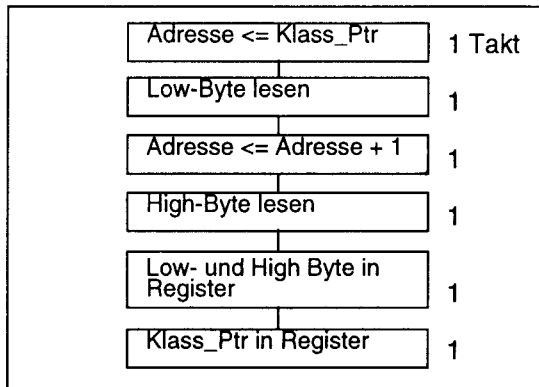


Bild 18: Aktionen im Zustand „Übertrage_1_Klassenwert_in_Buffer“

3.3 Überprüfung der Echtzeitbedingung

Das Zusammenspiel von Soft- und Hardware muss auch dann noch funktionieren, wenn jede ATM-Zelle einen Beitrag für das Histogramm liefert. Somit steht für den dargestellten Ablauf in den Bildern 15 - 18 eine Verarbeitungszeit von genau einer Zeldauer zur Verfügung ($53 \text{ Bytes} \times 50 \text{ ns} = 2.65 \mu\text{s}$). Nach Abzug der insgesamt 18 Takte, je 50 ns, aus Bild 17 und 18 sowie drei Zugriffe der CPU, je 500 ns, für Lesen der Register und Schreiben des SW_Flag bleiben noch 250 ns als Reserve übrig.

4 Bestimmung der Lost- und Misinserted Zellen

Die Messung verlorengegangener oder falsch eingefügter Zellen beruht auf der Auswertung der Sequenznummer in der O.191-Payload. Dabei gibt eine mitlaufende Referenz (SN_REF) den Erwartungswert in der Sequenznummer (SN) der nächsten Zelle vor. Im Flussdiagramm in Bild 19 bezeichnet NDIFF die aktuelle Abweichung und ODIFF die Abweichung, die sich bei der letzten Zelle ergeben hat. Erst wenn zweimal hintereinander die gleiche Differenz festgestellt wurde, ist von einer entsprechenden Anzahl 'Lost'- oder 'Misinserted'-Zellen auszugehen.

In Hardware umgesetzt, beinhaltet das Flussdiagramm etliche 32-Bit breite arithmetische Funktionen und Komparatoren. Nicht nur die zur Verfügung stehenden Ressourcen im FPGA stellen dabei eine nicht zu vernachlässigende Randbedingung dar. Um eine Verarbeitung mit der gegebenen Taktfrequenz zu erreichen, ist an einigen Stellen eine Aufteilung (Pipelining) der Logik auf mehrere Takte notwendig.

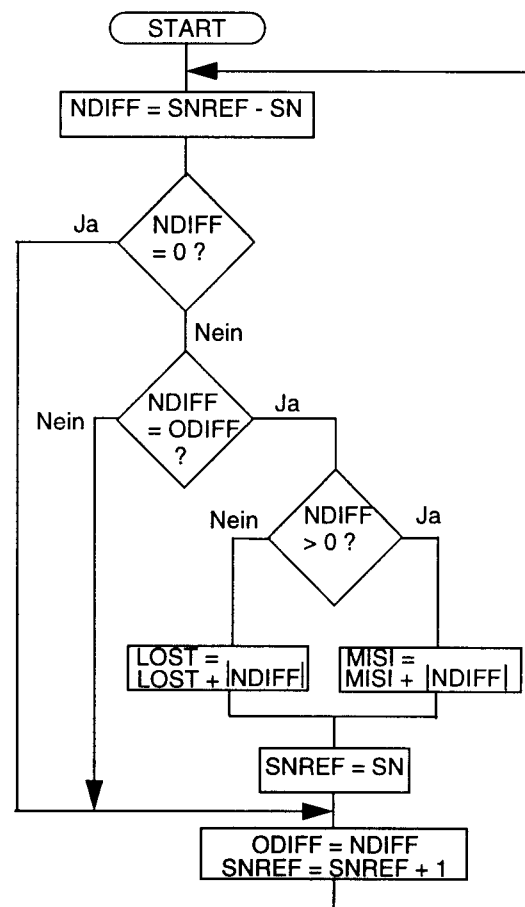


Bild 19: Vereinfachtes Flussdiagramm

5 Literatur

- /1/ „ATM Networks“
Rainer Händel, Manfred N.Huber
Stefan Schröder
ADDISON-WESLEY
- /2/ ITU-T Empfehlung O.191
- /3/ ITU-T Empfehlung G.707
- /4/ Internes Memorandum der
Fa. Wandel & Goltermann
„Spezifikation: ATM Performance
Messung“, 1997
Verf.: Dipl. Ing. F. Stockmayer
- /5/ Vorlesungsmanuskript CAE
FHTW Reutlingen, FB-EL
„Einführung in VHDL“
Prof. Dr. Kreutzer
Dipl. Ing. F. Stockmayer

6 Anhang

6.1 Berechnung des Header Error Check (HEC)

Das fünfte Byte im Kopf einer ATM-Zelle enthält eine Prüfsumme (HEC), womit Fehler in den Headerbytes H1.....H4 erkannt werden und im Fall eines Einzelfehlers eine Korrektur möglich ist. Darüberhinaus dient die Prüfsumme auf der Empfangsseite zur Zellsynchronisation, um den Start einer ATM-Zelle zu finden. Das Prinzip beruht auf einer Polynomdivision, wobei der Rest der Division der zu sichernden Nachricht angehängt und mitübertragen wird. Diese Maßnahme bewirkt, dass im fehlerfreien Fall am Empfangsort bei der Polynomdivision kein Rest übrigbleibt. Hat jedoch ein Bitfehler den Nachrichteninhalte verfälscht, lässt der berechnete Rest eine Aussage über die Fehlerposition zu.

Zur Implementierung in der Hardware stellt sich nun die Aufgabe, eine Verknüpfungsvorschrift zu finden sowie eine Tabelle abzuleiten, aus der die Zuordnung von Divisionsrest und Fehlerposition hervorgeht.

Eine allgemein durchgeführte Polynomdivision ist umständlich und wenig übersichtlich. Der hier gewählte Ansatz entwickelt die Schaltung mit Hilfe eines Gleichungssystems, ausgehend von der bitseriellen Verarbeitung mit anschließender Parallelisierung.

Die Polynomdivision kann mit Hilfe eines rückgekoppelten Schieberegisters taktsynchron mit den seriellen Binärstellen des Nachrichtenpolynoms durchgeführt werden. Beginnend mit dem höchstwertigen Bit aus H1 ($h_{1,7}$, vgl. Bild 20) enthält das Schieberegister in Bild 21 nach 40 Bit-Takten die gesuchte Prüfsumme.

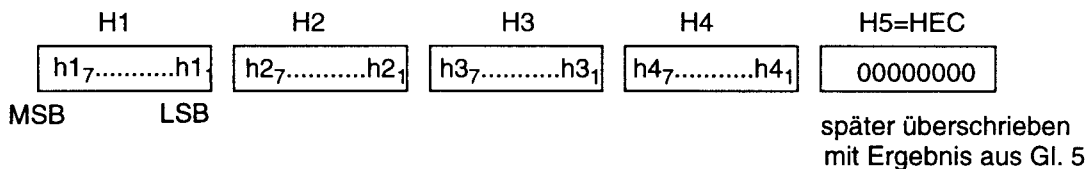


Bild 20: Headerbytes einer ATM-Zelle zur Berechnung des HEC auf der Sendeseite

Die Rückkopplungsstruktur des Schieberegisters entspricht genau den Wertigkeiten des Generatorpolynoms ($1 + x + x^2 + x^8$) mit dessen Hilfe die Prüfstellen erzeugt werden.

Es ist möglich die Rückkopplungsstruktur so zu modifizieren, dass bereits nach 32 Bittakten das Ergebnis zur Verfügung steht. Da diese Variante auf das gleiche Endergebnis führt, soll sie in dieser Betrachtung nicht weiter verfolgt werden.

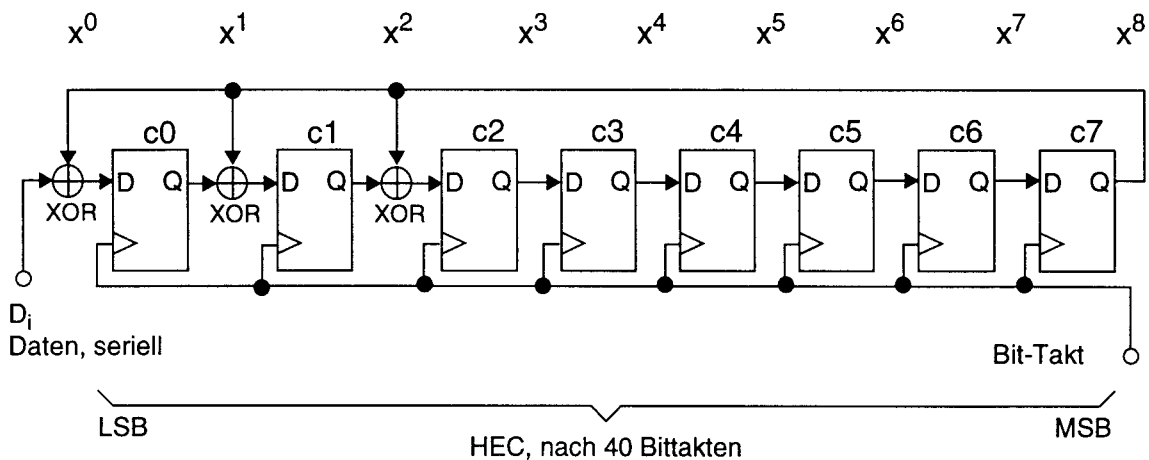


Bild 21: Rückgekoppeltes Schieberegister

6.2 Parallelisierung

In der Praxis erfolgt die Übertragung meist byteparallel. Für die Bearbeitung eines Bytes muss das serielle Schieberegister somit 8mal getaktet werden. Nach formaler Durchführung erhält man daraus ein rückgekoppeltes Netzwerk, welches byteparallel arbeitet.

Die sequentielle Schaltung in Bild 21 lässt sich dann als Gleichungssystem darstellen:

Allgemein nach n. Takt:

Wobei D_i die serielle Datenfolge darstellt mit dem höchstwertigen Datenbit (MSB) zuerst.

$$\begin{aligned} c0_n &= c7_{n-1} \text{ xor } D_i \\ c1_n &= c0_{n-1} \text{ xor } c7_{n-1} \\ c2_n &= c1_{n-1} \text{ xor } c7_{n-1} \\ c3_n &= c2_{n-1} \\ c4_n &= c3_{n-1} \\ c5_n &= c4_{n-1} \\ c6_n &= c5_{n-1} \\ c7_n &= c6_{n-1} \end{aligned}$$

Oder in Vektorschreibweise:

$$\underline{c}_n = \{A \underline{c}_{n-1} + \underline{d}_n\}_{\text{mod}2} \quad (1)$$

$$\underline{c}_n = \begin{bmatrix} c0_n \\ c1_n \\ c2_n \\ c3_n \\ c4_n \\ c5_n \\ c6_n \\ c7_n \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} c0_{n-1} \\ c1_{n-1} \\ c2_{n-1} \\ c3_{n-1} \\ c4_{n-1} \\ c5_{n-1} \\ c6_{n-1} \\ c7_{n-1} \end{bmatrix} + \begin{bmatrix} D_i \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

A
 \underline{c}_{n-1}
 \underline{d}_n

mit $\underline{c}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ Startwert, bzw. Anfangsinhalt der Flip-Flops

Beispiel:

nach 1. Bit-Takt:

$$\begin{aligned} c0_1 &= c7_0 \text{ xor } D_7 \\ c1_1 &= c0_0 \text{ xor } c7_0 \\ c2_1 &= c1_0 \text{ xor } c7_0 \\ c3_1 &= c2_0 \\ c4_1 &= c3_0 \\ c5_1 &= c4_0 \\ c6_1 &= c5_0 \\ c7_1 &= c6_0 \end{aligned}$$

nach 2. Bit-Takt: ...

$$\begin{aligned} c0_2 &= c6_0 \text{ xor } D_6 \\ c1_2 &= c7_0 \text{ xor } c6_0 \text{ xor } D_7 \\ c2_2 &= c7_0 \text{ xor } c6_0 \text{ xor } c0_0 \\ c3_2 &= c7_0 \text{ xor } c1_0 \\ c4_2 &= c2_0 \\ c5_2 &= c3_0 \\ c6_2 &= c4_0 \\ c7_2 &= c5_0 \end{aligned}$$

nach 8. Bit-Takt:

$$\begin{aligned} c0_8 &= c0_0 \text{ xor } c7_0 \text{ xor } c6_0 \text{ xor } D_0 \\ c1_8 &= c1_0 \text{ xor } c6_0 \text{ xor } c0_0 \text{ xor } c6_0 \text{ xor } D_1 \\ c2_8 &= c2_0 \text{ xor } c1_0 \text{ xor } c0_0 \text{ xor } c7_0 \text{ xor } D_2 \\ c3_8 &= c3_0 \text{ xor } c2_0 \text{ xor } c1_0 \text{ xor } D_3 \\ c4_8 &= c4_0 \text{ xor } c3_0 \text{ xor } c2_0 \text{ xor } D_4 \\ c5_8 &= c5_0 \text{ xor } c4_0 \text{ xor } c3_0 \text{ xor } D_5 \\ c6_8 &= c6_0 \text{ xor } c5_0 \text{ xor } c4_0 \text{ xor } D_6 \\ c7_8 &= c7_0 \text{ xor } c6_0 \text{ xor } c5_0 \text{ xor } D_7 \end{aligned}$$

aus Gleichung (1):

$$\underline{c}_1 = \{A \underline{c}_0 + \underline{d}_1\}_{\text{mod}2}$$

$$\underline{c}_2 = \{A \underline{c}_1 + \underline{d}_2\}_{\text{mod}2} = \{A^2 \underline{c}_0 + A \underline{d}_1 + \underline{d}_2\}_{\text{mod}2}$$

...

$$\underline{c}_8 = \{A^8 \underline{c}_0 + A^7 \underline{d}_1 + A^6 \underline{d}_2 + A^5 \underline{d}_3 + A^4 \underline{d}_4 + A^3 \underline{d}_5 + A^2 \underline{d}_6 + A^1 \underline{d}_7 + \underline{d}_8\}_{\text{mod}2} \quad (2)$$

Die neue Transformationsmatrix B für die byteparallele Verarbeitung der einzelnen Datenvektoren \underline{h} (D_7, D_6, \dots, D_0) kann entweder durch Koeffizientenvergleich mit dem Ergebnis des Beispiels nach 8. Bit-Takt gewonnen werden, oder durch direkte Auswertung der Gleichung (2).
Bei der Matrizenmultiplikation ist zu beachten, dass alle Additionen Modulo 2 durchzuführen sind.

$$\underline{c}_8 = \{B \underline{c}_0 + \underline{h}\}_{\text{mod}2} \quad (3)$$

$$\underline{c}_8 = \begin{matrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} & \begin{bmatrix} c_{0_0} \\ c_{1_0} \\ c_{2_0} \\ c_{3_0} \\ c_{4_0} \\ c_{5_0} \\ c_{6_0} \\ c_{7_0} \end{bmatrix} & + & \begin{bmatrix} D_0 \\ D_1 \\ D_2 \\ D_3 \\ D_4 \\ D_5 \\ D_6 \\ D_7 \end{bmatrix} \end{matrix}$$

$B \quad \underline{c}_0 \quad \underline{h}$

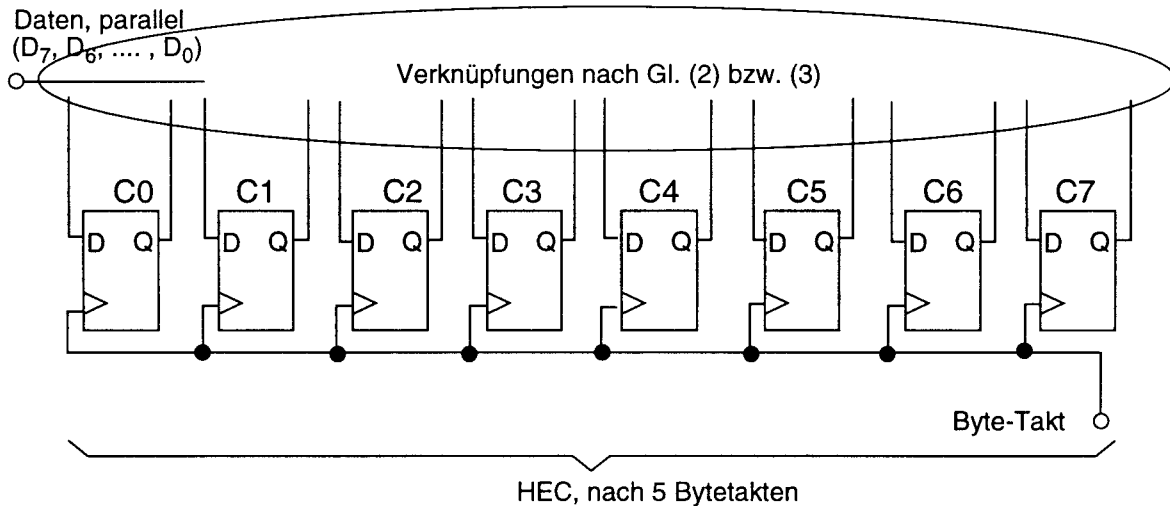


Bild 22: 8-Bit Parallele Struktur des rückgekoppelten Schieberegisters aus Bild 21

Nachdem nun eine Transformationsmatrix B für die byteparallele Berechnung gefunden wurde, kann diese Vorschrift auf die Headerbytes $H_1 \dots H_4$ und dem zunächst mit Nullen aufgefüllten HEC-Byte (Sendeseite) bzw. empfangenen HEC-Byte (Empfangsseite) angewendet werden.

$$\underline{c}_n = \{B \underline{c}_{n-1} + \underline{h}_n\}_{\text{mod}2}, \text{ mit } \underline{c}_0 = \underline{c}_0 = \underline{0} \quad (4)$$

$$\begin{aligned} \underline{c}_1 &= \{B \underline{c}_0 + \underline{h}_1\}_{\text{mod}2} = \underline{c}_8 && \text{1. Headerbyte} \\ \underline{c}_2 &= \{B^2 \underline{c}_0 + B \underline{h}_1 + \underline{h}_2\}_{\text{mod}2} = \underline{c}_{16} && \text{2. Headerbyte} \\ \underline{c}_3 &= \{B^3 \underline{c}_0 + B^2 \underline{h}_1 + B \underline{h}_2 + \underline{h}_3\}_{\text{mod}2} = \underline{c}_{24} && \text{3. Headerbyte} \\ \underline{c}_4 &= \{B^4 \underline{c}_0 + B^3 \underline{h}_1 + B^2 \underline{h}_2 + B \underline{h}_3 + \underline{h}_4\}_{\text{mod}2} = \underline{c}_{32} && \text{4. Headerbyte} \end{aligned}$$

Rest der vollständigen Polynomdivision:

$$\underline{c}_5 = \underline{h}_5 = \{B^5 \underline{c}_0 + B^4 \underline{h}_1 + B^3 \underline{h}_2 + B^2 \underline{h}_3 + B \underline{h}_4 + \underline{h}_5\}_{\text{mod}2} \quad \text{Sendeseite} \quad (5)$$

$\underline{c}_0 = \underline{0} \quad \underline{h}_5 = \underline{0}$

$$\underline{c}_5 \stackrel{!}{=} \underline{0} = \{B^4 \underline{h}_1 + B^3 \underline{h}_2 + B^2 \underline{h}_3 + B \underline{h}_4 + \underline{h}_5\}_{\text{mod}2} \quad \text{Empfangsseite} \quad (6)$$

Mit den Matrizen:

$$B^4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} \quad
 B^3 = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \quad
 B^2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \quad
 B = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

6.3 Äquivalentes Hamming Prüfschema

H1	H2	H3	H4	H5=HEC
01234567	.01234567	01234567	0123456701234567
00001011	10110101	10001010	1000001110000000
10001110	11101111	01001111	1100001001000000
11001100	01000010	10101101	1110001000100000
01100110	10100001	01010110	0111000100010000
10110011	01010000	10101011	0011100000001000
01011001	10101000	01010101	0001110000000100
00101100	11010100	00101010	0000111000000010
00010110	01101010	00010101	0000011100000001
B ⁴	B ³	B ²	B	

Prüfspalten

Die Auswertung der Gleichung (5) (oder das Hamming Prüfschema) liefert die kombinatorische Verknüpfungsvorschrift für jedes einzelne Bit im HEC-Byte.

Beispiel:

$$\text{HEC}(0) = \text{H1}(4) \text{ xor } \text{H1}(6) \text{ xor } \text{H1}(7) \text{ xor } \text{H2}(0) \text{ xor } \text{H2}(2) \text{ xor } \text{H2}(3) \text{ xor } \text{H2}(5) \text{ xor } \text{H2}(7) \text{ xor } \text{H3}(0) \text{ xor } \text{H3}(4) \text{ xor } \text{H3}(6) \text{ xor } \text{H4}(0) \text{ xor } \text{H4}(6) \text{ xor } \text{H4}(7)$$

Die Antwort auf die Frage wieviele Fehler auf der Empfangsseite korrigiert bzw. erkannt werden können ist die sog. Hamming Distanz, welche dem Prüfschema entnommen werden kann. Ändert sich ein Bit der Nachricht, so bestimmt die Anzahl Einsen in der zugehörigen Prüfspalte, wieviel Binärstellen im HEC davon betroffen sind.

Die Hamming-Distanz ergibt sich also aus dem Minimum der Anzahl Einsen pro Prüfspalte plus 1, in diesem Fall eine minimale Distanz von 4, womit 1 Fehler immer sicher erkennbar und korrigierbar ist.

6.4 Fehlerposition

Ergibt die Division am Empfangsort (s.a Gleichung (6)) ein Rest, dessen Bitmuster in einer der Spalten des Hamming Prüfschemas wiedergefunden werden kann, so zeigt diese Prüfspalte auf die Fehlerposition. Ein Rest, dessen Muster nicht in den Prüfspalten vorkommt, deutet auf einen Mehrfachfehler hin.

Beispiel:

Ein Fehler im Bit 7 von H1 führt am Empfangsort zu einem HEC, der an den Bitpositionen 0, 4 und 5 gegenüber dem empfangenen HEC invertiert ist. Ein Vergleich (xor-Verknüpfung) des empfangenen mit dem neu gebildeten HEC ergibt dann die Prüfspalte '00110001'. Damit ist die Fehlerposition identifiziert und korrigierbar.

6.5 VHDL-Beispiel

Das VHDL-Beispiel zeigt die HEC-Berechnung innerhalb zweier verschachtelter Loops. Die innere Schleife entspricht der Anwendung der Gleichung (1). Prinzipiell wäre anstelle der Loops auch die komplette Verknüpfungsvorschrift nach Gleichung (5) denkbar, doch ist ein 'Misstrauen' dem Synthesewerkzeug gegenüber unbegründet, da die Ergebnisse sowohl bzgl. der Geschwindigkeit, als auch der Fläche vergleichbar sind.

```

hec: process (CLK)
variable C, H: STD_LOGIC_VECTOR (7 downto 0);
variable CARRY: STD_LOGIC;
begin

    if CLK'EVENT and CLK='1' then
        if ENABLE ='1' then
            -- Startwert aller FF = 0
            C := "00000000";
            for i in 1 to 5 loop
                -- Headerbytes H1 ... H4, H5=0x00
                H := HEADER(i);
                -- MSB ... LSB
                for j in 7 downto 0 loop
                    -- C(7) vor ueberschreiben retten
                    CARRY := C(7);
                    C(7) := C(6);
                    C(6) := C(5);
                    C(5) := C(4);
                    C(4) := C(3);
                    C(3) := C(2);
                    C(2) := C(1) xor CARRY;
                    C(1) := C(0) xor CARRY;
                    C(0) := H(j) xor CARRY;
                end loop;
            end loop;

            -- Ergebnis abspeichern
            HEC_STORE <= C;

            -- Vergleich berechneter mit ankommendem HEC
            if (HEC_STORE xor HEADER(4)) = "00000000" then
                HEC_OK <= '1';
            else
                HEC_OK <= '0';
            end if;

            -- Syndrom ist Rest der Division
            SYNDROM <= HEC_STORE xor HEADER(4);

            HEADER(1)<=HEADER(2);
            HEADER(2)<=HEADER(3);
            HEADER(3)<=HEADER(4);
            HEADER(4)<=DIN_REG;
            HEADER(5) <= "00000000";

        end if;
    end if;
end process hec;

```

Ein schneller Weg zum IC - Layout

Prof. Dr. Gert Voland

FH Konstanz, FB Informatik
Brauneggerstr. 55, 78462 Konstanz
Tel.: 07531 / 206 644, Fax: 07531 / 206 559
Email: voland@fh-konstanz.de

Es wird eine Methode vorgestellt, mit der die Verfahren und Techniken der Layouterstellung von integrierten Schaltungen (ICs) eingeführt werden können. Das Ziel war es, einen Designablauf zu entwickeln, der auch Studenten im Selbststudium erlaubt, Schaltungen einzugeben, das logische (und eventuell zeitliche) Verhalten zu simulieren und dann das fertigungsreife Layout zu erstellen. D.h. der Design-Flow sollte auch auf kostenlosen Demoversionen lauffähig sein. Deshalb wurde MICROSIM PSPICE (Eingabe und Digital-simulator) und TANNER TOOLS (Layout Editor und P&R) gewählt.

Stichworte: Logikeingabe, Logiksimulation, Netzliste, Place & Route, Simulated Annealing, Standardzellen, Pad-Ring, Extraktion, DRC, LVS.

1. Einführung

Besonders bei der Einführung in die Techniken der IC-Erstellung ist es wichtig, daß die den Methoden zugrunde liegenden Konzepte nicht zu stark von Handhabungsschwierigkeiten der (für den professionellen Einsatz abgestimmten) Design Tools verdeckt werden.

Es wurden deshalb bei der Entwicklung dieses Ablauf folgende Ziele in den Vordergrund gestellt:

- Es soll die Designmethodik von ICs bis zur Layouterstellung eingeführt werden.
- Der Entwurf soll auf bekannten Werkzeugen aufsetzen (in diesem Fall MICROSIM PSPICE, was im 2. Semester eingeführt wird).
- Die Netzlistengenerierung und der Transfer zum Place & Route Tool muß sehr einfach sein.

- Die Layouterzeugung soll praktisch auf "Knopfdruck" geschehen. Das ist bei den TANNER TOOLS möglich.
- Der Design-Flow soll auf den (kostenlosen) Demoversionen ablaufen können damit Studenten auch z.H. damit arbeiten können.

2. Designablauf

Der Design Flow durchläuft dabei folgende Schritte (s. auch Bild 1):

Eingabe der Schaltung im Schematic-Editor von PSPICE mit Symbolen der 74xx-Bibliothek. Man muß sich hierbei auf die Symbole beschränken, die in der Layoutbibliothek vorhanden sind (s. Tabelle 3)

Es folgt eine standardmäßige Netzlistengenerierung für die Logiksimulation. Dabei werden die Timing-Daten der 74er-Bausteine benutzt. Natürlich ist hier auch ein Satz von Timing-Daten der charakterisierten Layoutzellen denkbar

Jetzt werden die Symbole auf die der Layoutzellen umgeschaltet. Dafür gibt es eine zweite Symbolbibliothek, die zwar die Namen der Logikzellen beinhaltet, ansonsten aber eine Netzliste im Format (und mit den Zellen- und Pinnamen der Layoutzellen) erzeugt. Es liegt dann also direkt eine Netzliste im TANNER-P&R-Format vor.

Für die Layouterstellung kann man den Optimierungsgrad und damit die Chipgröße einstellen. Falls die standardmäßig eingestellte etwa quadratische Form geändert werden soll, kann man die Anzahl der Reihen vorwählen. Natürlich läßt sich auch eine exakte Position der I/O-Pads vorgeben (Top, Right, Bottom, Left).

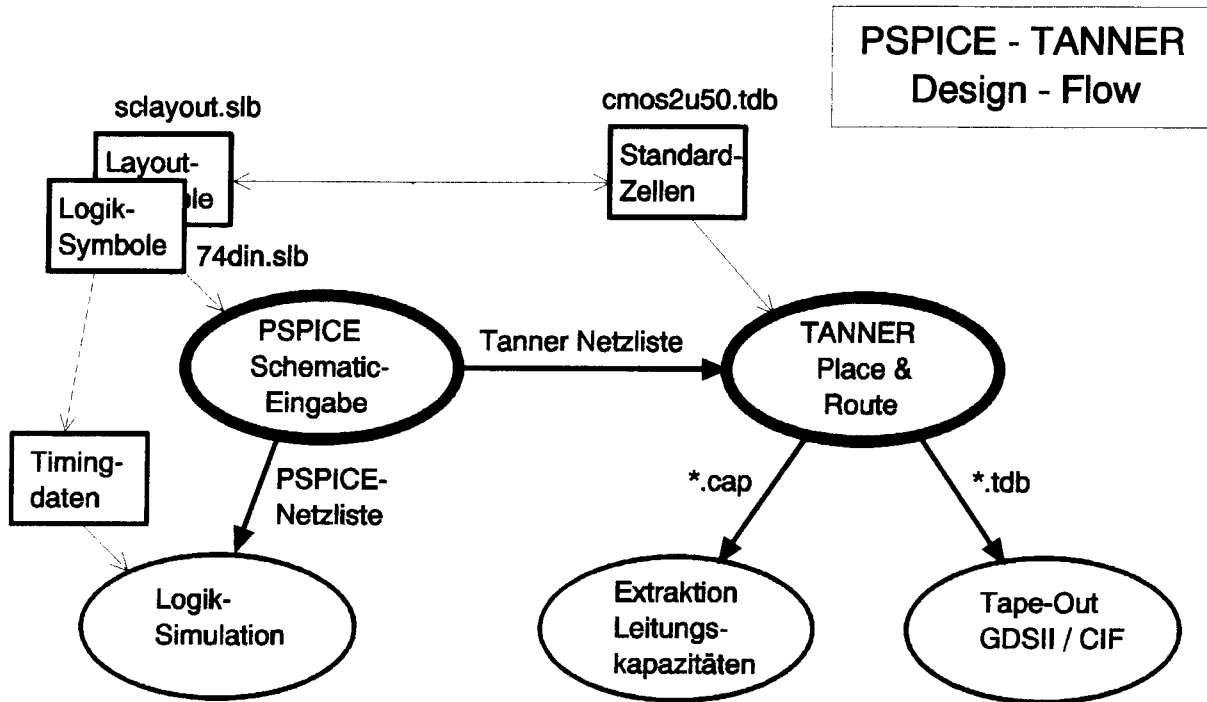


Bild 1: Übersicht des Designablaufs

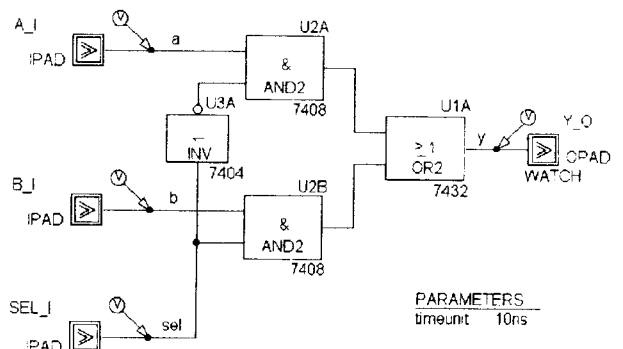
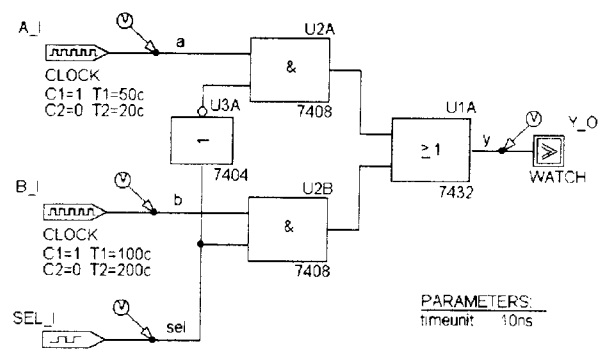
An dem fertigen Layout lassen sich dann DRC, LVS, Extraktion der Leitungskapazitäten sowie Umwandlung in GDSII oder CIF durchführen. Es ist auch ein Extraktor vorhanden, mit dem der physikalische Aufbau der IC-Schichten im Querschnitt verfolgt werden kann.

3. Beispiel

Die einzelnen Phasen werden nun an einem Beispiel erläutert. Dazu dient die Anordnung **mux2.sch**. Das Schaltbild (Bild 2) zeigt 3 Symbole für Stimuli (A_I, B_I, SEL_I), 4 Logiksymbole (1 INV, 2 AND, 1 OR) und ein Ausgangssignal (Y_O). In üblicher Weise wird eine SPICE-Netzliste erzeugt (s. Tabelle 1)

Bild 2:
Schematic der Schaltung mux2.sch mit Logiksymbolen (oben).

Die gleiche Schaltung mit Layoutsymbolen (unten).



```
*          WATCH Y_O: Will become an output pad with SCLAYOUT.SLB
X_U1A     $N_0001 $N_0002 y $G_DPWR $G_DGND 7432   PARAMS: IO_LEVEL=0 MNTYMXDLY=0
X_U2A     a $N_0003 $N_0001 $G_DPWR $G_DGND 7408   PARAMS: IO_LEVEL=0 MNTYMXDLY=0
X_U2B    b sel $N_0002 $G_DPWR $G_DGND 7408       PARAMS: IO_LEVEL=0 MNTYMXDLY=0
X_U3A     sel $N_0003 $G_DPWR $G_DGND 7404         PARAMS: IO_LEVEL=0 MNTYMXDLY=0
U_SEL_I   STIM(1,1) $G_DPWR $G_DGND sel IO_STM IO_LEVEL=0 TIMESTEP={timeunit}
+ 0s 0
+ 250c 1
...
```

Tabelle 1: Logiknetzliste von mux2.sch (PSPICE, teilweise)

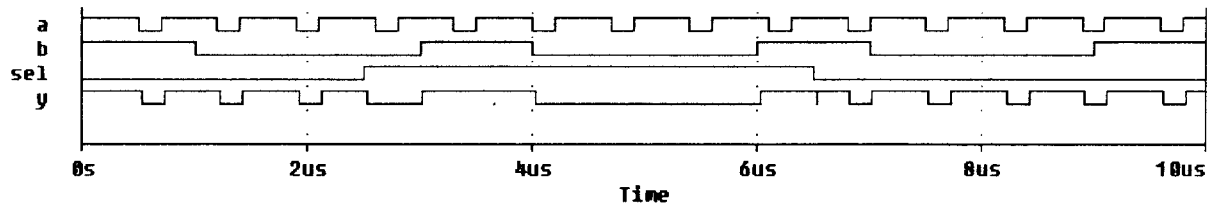


Bild 3: Logiksimulation von mux2.sch

Bei dem UND-Gatter X_U2B sind also die Eingänge mit den Signalen b und sel belegt. Der Ausgang geht auf \$N_0002. Die Versorgungsspannung ist mit \$G_DPWR und \$G_DGND angeschlossen, obwohl das im Schematic nicht sichtbar ist (hidden pins). Die in den Stimuli angegebenen Eingangsmuster führen zum Ergebnis der Logiksimulation (Bild 3).

Wenn die Symbolbibliothek von den 74er-Bauteilen auf die Layoutsymbole umgeschaltet ist, erscheint im PSPICE Schematic Editor sofort ein Bild mit den neuen Symbolen. Wichtig ist dabei nur, daß die Pins der Symbole die exakt gleiche Position haben, da sonst die Topologie der Schaltung gestört würde. D.h., unter dem Namen eines Stimulus (STIM1) liegt nun das Symbol

eines Input Pads. Für die Logiksymbole bleiben die Typen erhalten und das Ausgangssignal y muß nun ein "Watch"-Symbol erhalten, was zu einem Output Pad wird. Diese Watch-Symbol war schon in der Logikschaltung vorhanden, hat dort aber nur einen Kommentar in die Netzliste eingetragen (* WATCH ... in Tabelle 1).

Die eigentliche Umarbeitung der Netzliste erfolgt nun in den "template"-Zeilen der PSPICE Symbole. Für die Layoutnetzliste werden jetzt Einträge erzeugt, die genau den Anforderungen der TANNER TOOLS entsprechen. Dabei bleibt die Topologie der Netzliste erhalten, nur die Zellennamen und Pinnamen werden für die Standardzellen generiert (s. Tabelle 2).

```
CP          OPAD Pad DataOut ;
UY_O       Pad_y y ;
C          NOR2C Out1 Out2 A B ;
UU1A      $N0_U1A y $N_0001 $N_0002 ;
C          NAND2C Out1 Out2 A B ;
UU2A      $N0_U2A $N_0001 a $N_0003 ;
C        NAND2C Out1 Out2 A B ;
UU2B     $N0_U2B $N_0002 b sel ;
C          INV Out A ;
UU3A      $N_0003 sel ;
CP        IPADC Pad DataIn DataInB DataInUnBuf ;
USEL_I     Pad_sel sel $N0_SEL_I $N1_SEL_I ;
CP        IPADC Pad DataIn DataInB DataInUnBuf ;
UA_I      Pad_a a $N0_A_I $N1_A_I ;
CP        IPADC Pad DataIn DataInB DataInUnBuf ;
UB_I      Pad_b b $N0_B_I $N1_B_I ;
```

Tabelle 2: Layoutnetzliste von mux2.sch für die TANNER TOOLS

Aus dem AND X_U2B der Logiknetzliste ist nun der Eintrag der Zelle UU2B mit der Zelle NAND2C geworden. Die Zellen mit der Endung C haben die Eigenschaft, daß sie die invertierte und nichtinvertierte Funktion ausgeben. Da hier ein AND benötigt wird, kommt der nichtinvertierende Ausgang zum tragen. der NAND-Ausgang (\$N0_U2B) bleibt unbeschaltet. Anmerkung: Die Tatsache, daß vor der Eintrag der verschalteten Zelle noch einmal die Definition der Zelle erscheint (Zeilen, die mit C... beginnen), ist eine Eigenheit des TANNER Netzlistenformats.

Die Netzliste wird nach dem Laden der Bibliothek cmos2u50.tdb in dem TANNER P&R TOOL direkt ausgeführt. Das resultierende Layout ist in Bild 4 wiedergegeben. Hierbei wurde keine PAD-Anordnung vorgegeben. Das Layout ist komplett mit Core-, Channelrouting in Metal1 und Metal2, Versorgungskamm, PWR- und GND-Pad, Versorgungsring für die Pads, 3 Eingangs-Pads, ein Ausgangs-Pad und zwei Versorgungs-Pads versehen. In Bild 5 ist das Layout einer komplexeren Schaltung wiedergegeben. Tabelle 4 zeigt das Ergebnis verschiedener P&R-Läufe.

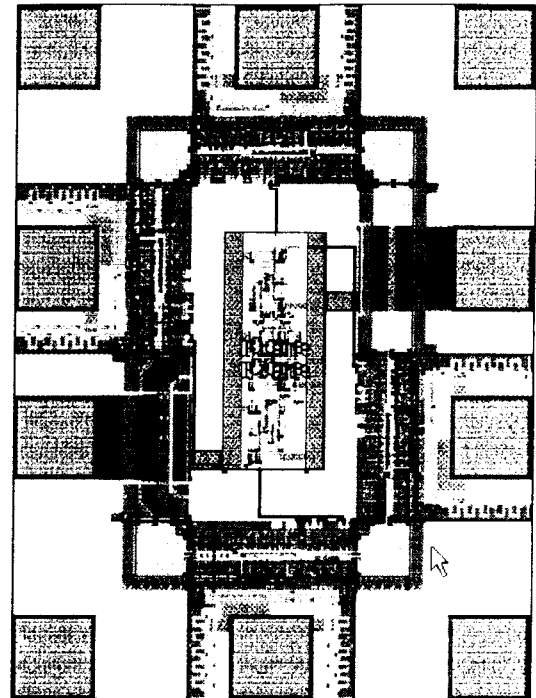


Bild 4: Layout von mux2.sch

Logic function	MicroSim eval.slb	FH Kn 74din.slb	TANNER cmos2u50.tdb	Width
7400 2-input NAND	x	x	NAND2	26
7402 2-input NOR	x	x	NOR2	26
7404 Inverter	x	x	INV	18
7408 2-input AND	x	x	NAND2C	34
7410 3-input NAND	x	x	NAND3	34
7420 4-input NAND	x	x	NAND4	42
7427 3-input NOR	x	x	NOR3	34
7430 8-input NAND	x	x	(2*NAND4, OR2)	118
7432 2-input OR	x	x	NOR2C	34
7451 And-Or-Invert	x		AOI22	42
7474 DFF, preset, clear	x		DFFPC	137
7486 2-input XOR	x	x	XOR2	46
74266 2-input XNOR	x	x	XNOR2	46
STIM1 Stimulus	x		IPADC	-
HI Digital HI port	x		Connect to PWR	-
LO Digital LO port	x		Connect to GND	-
AND3 3-input AND		x	NAND3C	46
AND4 4-input AND		x	NAND4C	54
BUF4 Buffer, strength 4		x	BUF4	50
CLOCK Clock generator		x	IPADC	-
DLATRS D-Latch, set, reset		x	LATPC	105
NAND6 6-input NAND		x	(2*NAND3, OR2)	102
NOR4 4-input NOR		x	NOR4	42
OR3 3-input OR		x	NOR3C	46
OR4 4-input OR		x	NOR4C	54
WATCH (no effect)		x	OPAD	-

Tabelle 3: Logik- und Layoutzellen

4. Programme und Bibliotheken

In Tabelle 3 sind alle Funktionen aufgestellt, die in dieser TANNER Layoutbibliothek (MOSIS / ORBIT skalierbare Standardzellenbibliothek, gezeichnet in 2 µm) zur Verfügung stehen. Einige Logikzellen sind nicht als 74er-Bausteine vorhanden. Deren Funktion wurde in der Logikbibliothek ergänzt (z.B. AND3, OR4). Die Funktionen wie NAND6 erzeugen Makroeinträge mit NAND3 und OR2.

Der Ablauf wurde mit folgenden Programmen und Bibliotheken getestet:

- PSPICE: MICROSIM DesignLab V6.x, V8.0 (Win 95) (Demo- und Vollversion)
- Logikbibliothek: 74din.slb, 4.4.98, 37 KB
- Layoutbibliothek: sclayout.slb, 17.1.97, 23 kB

- TANNER TOOLS: V5.x (DOS) und 6.04 (WIN 95) (Demo- und Vollversion)
- Zellenbibliothek: cmos2u50.tdb, 31.8.96, 93 kB.

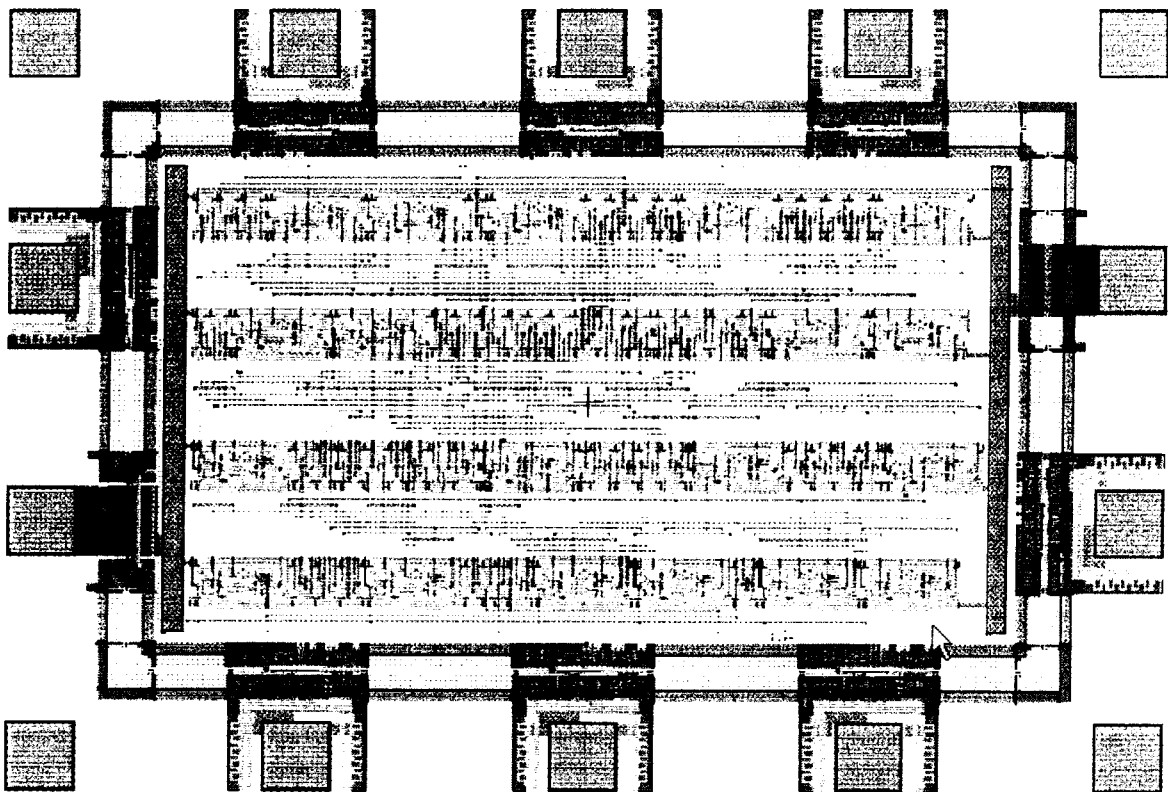


Bild 5: Fertigungsreifes Layout der Schaltung p3_arbit (4 Reihen)

Optimierung	Reihen	x	y	Fläche	Rechenzeit (Pentium 90)
keine	6	1.591	1.568	2.49	15 sec
T=0.001	6	1.327	1.339	1.78	120 sec
T=0.1	6	1.295	1.310	1.70	720 sec
T=0.1	4	1.673	1.081	1.81	120 sec (s. Bild 5)
T=0.1	8	1.184	1.595	1.89	120 sec

Tabelle 4: P&R-Läufe der Schaltung p3_arbit

Temperaturzelle in CMOS-Technologie mit Sigma-Delta-Wandler auf der Basis eines Entwurfs der Universität Delft/Ne

Jürgen Hauser

W. Vollmer, Dirk Jansen, ASIC-Design-Center
 Fachhochschule Offenburg, Badstr. 24, 77652 Offenburg
 Tel. 0781/205-267, Fax 0781/205-242,
 E-Mail: d.jansen@fh-offenburg.de

An der Fachhochschule Offenburg wurde nach einer Vorlage der Universität Delft/Niederlande eine Temperaturzelle [1] entwickelt. Diese Zelle kann in ASICs zur Temperaturmessung integriert werden.

Das System enthält einen Temperaturfühler, der laterale Bipolartransistoren in einer Bandgap-Schaltung zur Temperaturmessung verwendet. Die analoge Übergabe der temperaturabhängigen Größe an den Sigma-Delta-Wandler erfolgt durch einen temperaturabhängigen Strom. Diese Bandgap-Schaltung erzeugt auch einen Referenzstrom.

Der Analogteil ist abschaltbar. Durch diesen Power-Down-Modus wird Strom gespart und die Eigenerwärmung vermindert.

Die digitale Signalverarbeitung enthält eine Steuerung mit Timer und Dezimationsfilter zur parallelen Ausgabe des digitalen Wertes. Eine weitere Nachbearbeitung des Signals, wie z.B. eine Eichung oder Kalibrierung, kann dann in einem Mikroprozessor durchgeführt werden.

1. Einführung

Die Temperatur zu messen und für eine spätere Auswertung zu speichern, ist eine wichtige Aufgabe. Mögliche Anwendungsgebiete liegen in der Lebensmittelbranche, der Medizin, der Chemie, oder der Landwirtschaft. In diesen Bereichen muß oft die Lagerung und Transport von verderblichen Waren überwacht werden.

Zu diesem Zweck wurde an der Fachhochschule Offenburg ein IC entwickelt, das die Aufzeichnung von Temperatur-Zeit-Reihen ermöglicht (Thermologger) [3]. Abbildung 1 zeigt den gegenwärtigen

Stand [4]. Links sieht man die Vorderseite der Folienplatine. Unter der schwarzen Schutzmasse ist der Thermologger-ASIC auf die Leiterplatte gebondet, in dem ein μC und Speicher integriert sind. Dazu als externe Beschaltung ein Quarz, einige Kondensatoren, sowie den externe Temperatursensor (8-pol. IC). Auf der Rückseite (rechte Hälfte) befindet sich eine Schnittstelle zum Auslesen der Daten, die kompatibel zum Telefonkartenstandard ist.

Die Spannungsversorgung erfolgt über zwei flache Batterien. Der externe Sensor soll in der nächsten Generation des Thermologger entfallen und zusätzlich in dem ASIC integriert werden, wodurch sich die externe Beschaltung weiter verringert.

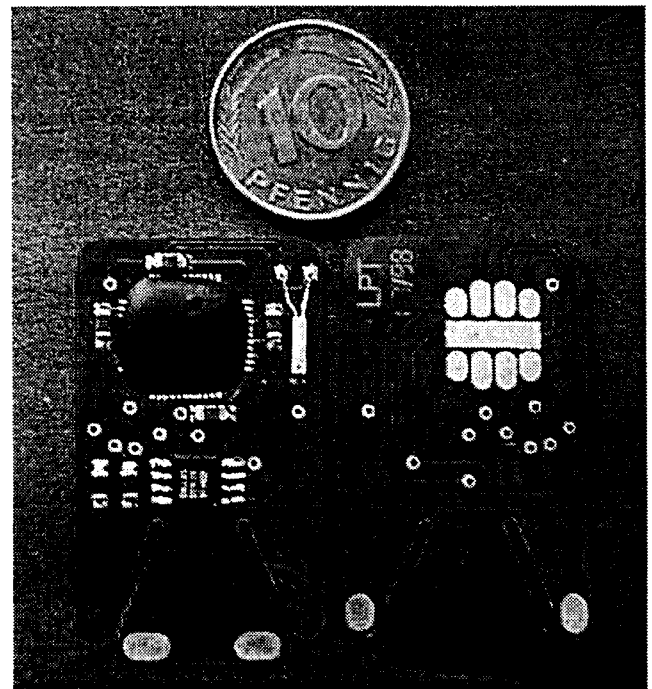


Abb.1: Gegenwärtiger Stand des Thermologger

Zu diesem Zweck muß ein geeigneter Temperatursensor entwickelt werden, der sich in einem ASIC realisieren läßt und die Daten möglichst in einem Format ausgibt, das ein Mikroprozessor verarbeiten kann.

In einer Kooperationsvereinbarung mit der Universität Delft/Ne wurde uns Schaltplan und Layout der von Dr. Bakker entwickelten Temperaturzelle [1], [2] zur Verfügung gestellt. Diese in 0.7 µm Technologie entworfene Zelle wurde von uns nachsimuliert, auf die vorliegende Aufgabe und Technologie umgestellt und im Zielprozeß Alcatel/Mietec 0.5 µm neu geroutet.

2. Aufbau der Temperaturzelle

Die Temperaturzelle besteht aus einem Analogteil und einem Digitalteil. Abb. 2 zeigt ein Blockschaltbild der Temperaturzelle.

Der Analogteil beinhaltet den eigentlichen Sensor, eine Bandgap-Referenz, sowie einen A/D-Wandler nach dem Sigma-Delta-Prinzip.

Der Digitalteil ist für die Steuerung des Analogteils zuständig. Außer dem Steuerungsschaltwerk ist noch ein Timer sowie der Codewandler vorhanden.

Als Eingangssignale der Temperaturzelle sind ein Taktsignal, der Power-on-Reset, sowie das Startsignal vorhanden. Dieses Startsignal wird von der Steuerung zum Reset und Power_on verarbeitet. Weiterhin wird ein Takt an den Analogteil übergeben.

Das Ausgangssignal des Analogteil ist ein pulsdichte-moduliertes Signal. Dieses PDM-Signal wird vom Codewandler, in unserem Fall ein Zähler, in ein 10 Bit paralleles Signal umgewandelt. Der laufende Meßzyklus wird von einem Flag angezeigt.

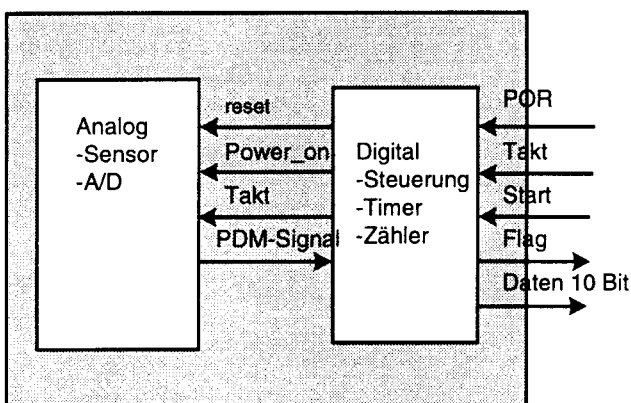


Abb.2: Übersicht Temperaturzelle

2.1. Sensor

Als Temperaturfühler werden Bipolartransistoren eingesetzt. Die Basis-Emitter-Diode dieser Transistoren ist in Vorwärtsrichtung geschaltet mit -2 mV/K temperaturabhängig. Auf dem ASIC sind diese Transistoren lateral ausgeführt. Diese Transistoren werden von Stromquellen versorgt. (Abb. 3)

Die temperaturabhängige Spannung wird einem Spannungs-Strom-Wandler zugeführt und, da der Sigma-Delta-Wandler mit Strom arbeitet, als i_{temp} ausgegeben. Der Wert von i_{temp} beträgt -100 nA bis -500 nA.

Der Sensor besteht jedoch nicht nur aus einem Bipolartransistor, sondern aus insgesamt neun Transistoren, die in zwei Pfaden im Verhältnis 1:8 angeordnet sind. Da in beiden Pfaden der gleiche Strom fließt, erreicht man unterschiedliche Stromdichten in den Transistoren und somit leicht unterschiedliche Spannungen. Der Spannungsunterschied wird dabei von genau dimensionierten Widerständen ausgeglichen.

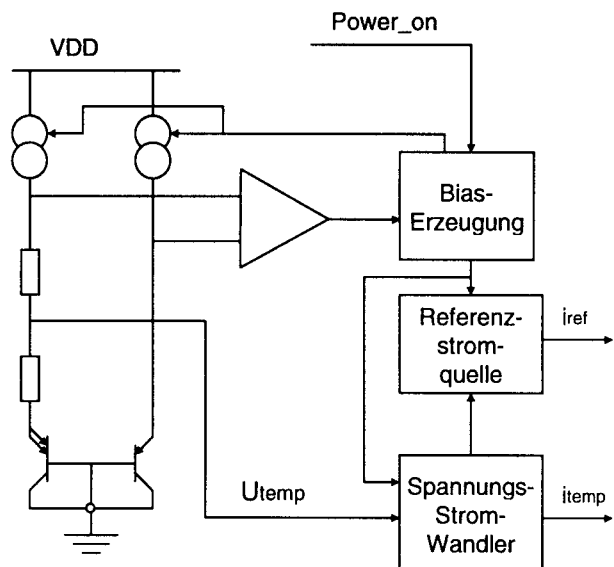


Abb.3: Übersicht Sensor

Der Strom durch die Transistoren wird dabei solange geändert, bis über dem Eingang des Differenzverstärkers keine Spannungsdifferenz mehr vorhanden ist. Gleichzeitig werden durch dieses Signal auch die Referenzstromquelle und der Spannungs-Strom-Wandler beeinflusst.

Die Referenzstromquelle besteht aus zwei über der Temperatur gegenläufigen Stromquellen. Daher benötigt zusätzlich sie noch ein zweites Signal vom Spannungs-Strom-Wandler, um die hohe Tempe-

raturunabhängigkeit zu erreichen. Als Referenzstrom fließen etwa 500 nA, wobei der Drift über den gesamten Temperaturbereich nur ca. 20 nA beträgt.

Der Sensor kann, wenn er nicht benötigt wird, in den Power-Down-Modus geschaltet werden. Dazu werden über die Biaserzeugung die Stromquellen abgeschaltet.

Weiterhin ist das System über den gesamten Betriebsbereich weitestgehend unempfindlich auf Änderungen der Versorgungsspannung. Nur bei hohen Temperaturen und oberhalb des Betriebsbereiches beginnt i_{temp} sich zu verändern.

Um den Offset, der durch Produktionstoleranzen entsteht, auszugleichen, sind Chopper integriert. Diese Chopper sind jedoch nicht in Abb. 3 enthalten. Näheres siehe [1]

Abb. 4 zeigt die Simulationsergebnisse der Ausgänge des Sensors. Darin ist in der oberen Bildhälfte die oben erwähnte Linearität über des Signals i_{temp} sowie der niedrige Drift von i_{ref} gut zu erkennen. Die untere Bildhälfte zeigt das Verhalten der Ausgänge über der Versorgungsspannung.

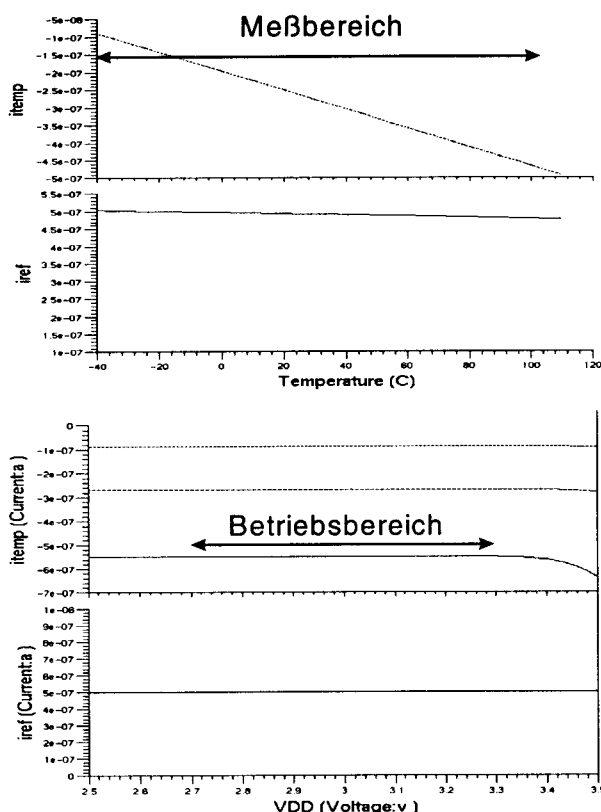


Abb.4: Simulationsergebnisse des Sensors

2.2. A/D-Wandler

Als A/D-Wandler wird ein Sigma-Delta-Wandler verwendet (Abb. 5). Dieser besteht aus den Baugruppen Schalter, Integrator, Komparator und D-Flipflop.

Als Eingangssignale dienen i_{ref} sowie i_{temp} . Am Ausgang kann man das puls-dichte-moduliertes Signal abgreifen.

Der Schalter schaltet den Referenzstrom, so daß am Integrator entweder nur i_{temp} oder das Summensignal aus $i_{ref} + i_{temp}$ anliegt. Da i_{temp} negatives Vorzeichen besitzt, wird je nach Schalterstellung auf- oder abintegriert. Der Schalter wird von dem PDM-Signal angesteuert, so daß bei einem High im Integrator auf- und bei einem Low abintegriert wird. Dadurch entsteht ein sägezahnförmiges Ausgangssignal.

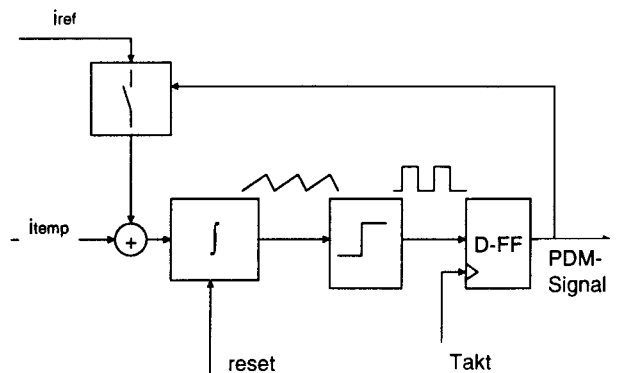


Abb.5: Übersicht Sigma-Delta-Wandler

Damit der Integrator zu Beginn der Messung auf einem definierten Niveau liegt, ist ein *reset*-Signal vorhanden.

Dieses Sägezahn-Signal wird an einen Komparator weitergeleitet. Dieser erzeugt daraus ein asynchrones Rechtecksignal. Zur Weiterverarbeitung im Digitalteil ist jedoch ein synchrones Rechtecksignal notwendig. Diese Synchronisierung erfolgt mit einem D-Flipflop.

Das Ausgangssignal ist ein Puls-Dichte-moduliertes Signal, d. h. das Eingangssignal wird durch die Dichte der Pulse dargestellt.

Abb. 6 zeigt die Simulation des Sigma-Delta-Wandlers mit einem sinusförmigen Eingangssignal. Die Abhängigkeit der Pulsdichte ist dabei gut zu erkennen.

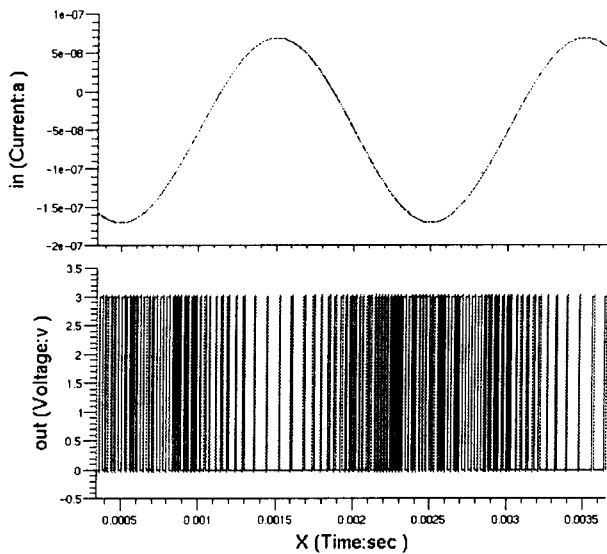


Abb.6: Simulationsergebnis mit sinusförmigem Eingangssignal

Abb. 7 stellt das PDM-Signal bei verschiedenen Temperaturen dar. Das obere Bild zeigt das Signal bei +100°C, das mittlere bei 27°C, das untere bei -40°C. Dabei erkennt man, daß bei höheren Temperaturen die Pulsdichte zunimmt. Am oberen Ende des Meßbereichs ist daher das PDM-Signal nahezu ständig auf „high“, d. h. es sind fast nur noch Pulse, aber keine Pausen mehr vorhanden.

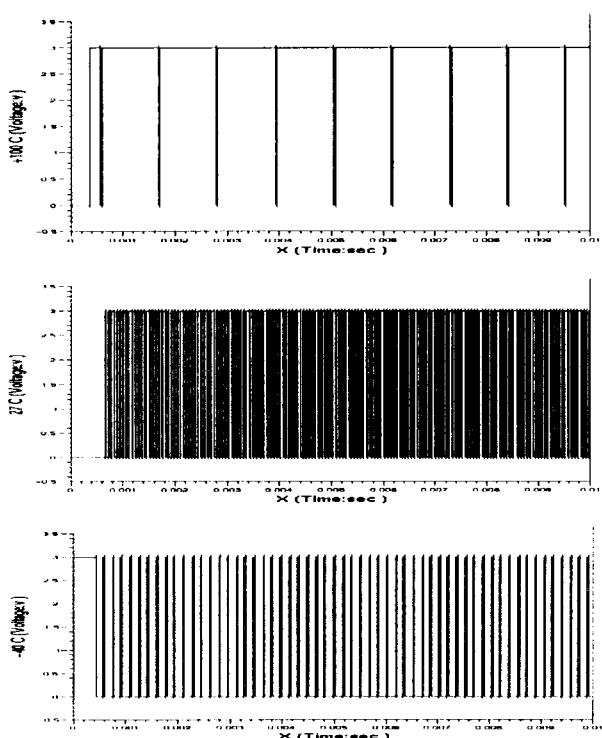


Abb.7: Simulationsergebnisse bei 3 verschiedenen Temperaturen

2.3. Digitale Signalverarbeitung

Die digitale Signalverarbeitung besteht aus drei Blöcken: der Steuerung, einem Timer, sowie einem Zähler als einfaches Dezimationsfilter (Abb. 8).

Der Zähler wandelt das PDM-Signal in ein 10 Bit paralleles Signal, indem er die Anzahl der Pulse bzw. deren Dauer zählt.

Um die Steuerung beim Einschalten in einen definierten Zustand zu bringen, wird ein Power-On-Reset benötigt.

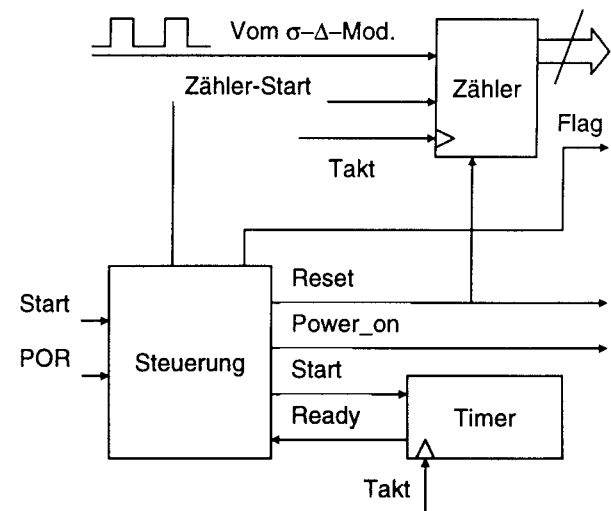


Abb.8: Übersicht Digitale Signalverarbeitung

Die Steuerung sorgt für den richtigen Ablauf des Meßzyklus. Nachdem über den Eingang *Start* eine Messung initialisiert wurde, wird der Analogteil über das Signal *Power_on* eingeschaltet. Weiterhin werden Zähler, Timer und Analogteil über den *Reset* zurückgesetzt. Gleichzeitig wird das *Flag* gesetzt und somit eine laufende Messung angezeigt. Danach werden Zähler und Timer gleichzeitig gestartet. Nach der Ablauf der Messung wird vom Timer das Signal *Ready* ausgegeben. Dies bewirkt, daß der Zähler angehalten sowie der Analogteil wieder abgeschaltet wird. Das *Flag* zeigt nun die Gültigkeit der Daten am Ausgang an.

Ein der Temperatur entsprechender Wert kann nun von einem Prozessor mit 10 Bit parallel ausgelesen werden. Danach kann nun mit Hilfe des Prozessors eine Kalibrierung durchgeführt werden.

3. Full Custom Routing

Die Thermozele soll in Mietec 0,5 μm Technologie produziert werden. Für Analogtechnik stehen jedoch keine Standardzellen zur Verfügung, wie es in der Digitaltechnik der Fall ist. Daher muß alles im Full Custom Stil geroutet werden.

Für die Unterstützung der Layout-Erstellung wurde der Mentor-Device-Generator für MOS-Transistoren so modifiziert, daß die W/L Verhältnisse aus dem Schaltplan übernommen werden. Widerstände und Kondensatoren werden von Hand ausgelegt, die Verwendung von Generatoren ist wegen der vielseitigen Formen nicht sinnvoll.

Zur Verdrahtung stehen im Mietec 0,5 μm -Prozeß drei Metal-Layer zur Verfügung. Weiterhin kann auch mit Polysilicon geroutet werden. Im Analogteil wurden nur 2 Metal-Layer verwendet, die 3. Metallisierungsebene wurde auf den Digitalteil beschränkt. Somit ist eine Verwendung der Zelle in den „einfacheren“ CMOS-Prozessen möglich. Für Kapazitäten steht noch ein zweiter Poly-Layer zur Verfügung, so daß man keine Substratkondensatoren, wie bei der 0.7 μm -Technologie, mehr routen muß.

Ein Vorteil der 0,5 μm -Technologie ist, daß N-Kanal- und P-Kanaltransistoren fast identische Thresholdspannungen haben. Die Transistoren sind somit symmetrisch. Bei der 0.7 μm -Technologie mußte dafür noch ein „Low_Voltage_Implant“ Layer eingefügt werden.

Abb. 9 zeigt ein D-FF in Full Custom Routing. Die Transistoren wurden mit einem Device-Generator erzeugt. Als Verdrahtungslayer wurden Metal 1 und Polysilicon verwendet.

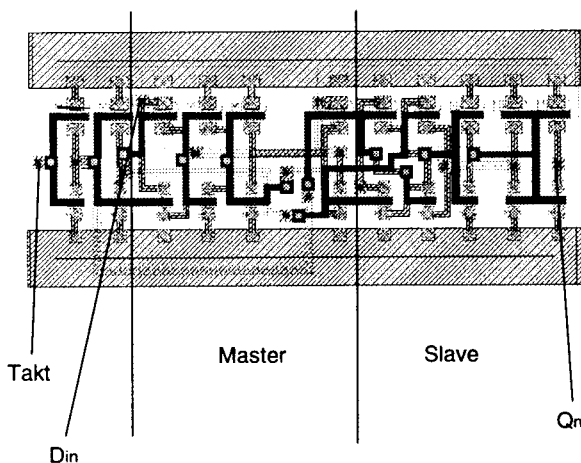


Abb.9: Full Custom Routing eines D-FF

4. Technische Daten

Temperaturbereich:	-40°C bis +100°C
Flächenbedarf:	ca. 1mm ²
Technologie:	Alcatel-Mietec 0,5 μm
Versorgungsspannung:	2,7 V ...3,3 V
Auflösung:	10 Bit \cong 0,15°C
Taktfrequenz:	32 kHz
Stromaufnahme:	ca. 10 μA bei 2 Messungen pro sec, <1 μA bei Power-down
Eichung:	Durch dig. Signalverarbeitung in μC , z.B. FHOP

Literaturverweis:

- [1] Micropower CMOS Temperatur Sensor with Digital Output
Anton Bakker, Johan H. Huijsing
IEEE JOURNAL OF SOLID-STATE CIRCUITS
VOL. 31, NO. 7, July 1996
- [2] Design and simulation of the second version of the Thermometer Building Block.
Delft University of Technology, September 1996
- [3] Applikation des Mikroprozessorkernels FHOP in einem Thermologger ASIC
T. Klumpp, D. Jansen,
MPC-Workshop, Juli 1996
- [4] Aufbau einer Chipkarte auf Basis des Thermologgerchips und Entwicklung der dazugehörigen Software
Diplomarbeit S. Schlageter, FH-Offenburg, 1998
- [5] Temperaturzelle in CMOS-Technologie mit Sigma-Delta-Wandler auf Basis eines Entwurfs der Universität Delft.
Diplomarbeit Jürgen Hauser, FH-Offenburg 1998

Aktuelle Konzepte zum Entwurf von Temperatursensoren

Gerhard Albert, HTG Mannheim
g.albert@fh-mannheim.de

Juli 1998

1 Einleitung

In steigendem Maße werden Temperatursensoren mit Batteriespeisung in der Steuer- und Regelungstechnik, in der Konsumelektronik sowie in der Überwachungstechnik z.B. für Lebensmittel eingesetzt. Es haben sich zwei Prinzipien der Realisierung herauskristallisiert, nämlich:

1. Umwandlung des analogen Sensorsignals (Spannung oder Strom als Meßwert) in eine dem analogen Signal proportionale Frequenz, welche zur Auswertung dann einem Zähler zugeführt wird.

2. Umsetzung des analogen Sensorsignals mit einem Analog-Digitalwandler, welcher jedoch als Mehraufwand eine Spannungs- oder Stromreferenz benötigt. Bei geschickter Erzeugung des Sensorsignals kann die Referenzquelle jedoch ohne großen Mehraufwand mit erstellt werden.

Das Sensorsignal kann durch die Temperaturabhängigkeiten der Bauelemente erzeugt werden. Dabei bieten sich zur Nutzung an: $R(T)$, $\mu_0(T)$, $U_{Th}(T)$ und $U_{BE}(T)$. Die klassische Methode besteht darin, die lineare Temperaturabhängigkeit der Differenz zweier Basis-Emitterspannungen zu benutzen.

Ausgeführte Schaltungskonzepte sind in [1] und [2] zu finden.

Im Weiteren soll hier das letztere Prinzip verfolgt werden, zusammen mit dem in [2] vorgeschlagenen $\Sigma\Delta$ -Wandler mit Sensor- und Referenzströmen.

2 Anforderungen

Das hier betrachtete Konzept soll die folgenden Eigenschaften aufweisen:

- Meßbereich von -50°C bis $+125^\circ\text{C}$, mit einer Genauigkeit von $\pm 0,5^\circ\text{C}$.
- Stromaufnahme, von Sensor und Referenz $< \text{ca. } 6 \mu\text{A}$, im aktiven Zustand.
- Batteriespannung von 2,7 V bis 3,3 V.
- Benutzung von Digital-CMOS-Prozessen.

D.h. die analogen Komponenten, wie Widerstände und Bipolar-Transistoren müssen durch parasitäre Elemente realisiert werden (Poly-Si-Widerstände und parasitäre Lateral- bzw. Substrat-Transistoren).

- Steuerung durch Mikroprozessor.

Diese letzte Forderung ergibt eine gewisse Entschärfung der vorstehenden Forderungen, denn durch die Prozessorsteuerung ist die Form der Sensorkennlinie unkritisch. Das bedeutet, sie kann gekrümmt sein, da der Prozessor eine Linearisierung vornehmen kann und es kann eine Eichung vorgenommen werden. Allerdings kann durch den Prozessor die Spannungsabhängigkeit des Sensor- und Referenzsignals nicht ausgeglichen werden, d.h. die Spannungsabhängigkeit muß minimiert werden. Dadurch, daß der Analog-Digital-Wandler immer den Quotienten aus Sensorsignal/Referenzsignal wandelt, kann eine gewisse Spannungsabhängigkeit der beiden Größen toleriert werden, wenn nur die Abhängigkeit bei Sensor und Referenz gleich ist.

3 Erzeugung des Sensorsignals

Die in den Bildern eingetragenen Größen n , p und q stellen die, auf die Transistoren mit dem Symbol 1 bezogenen W/L -Verhältnisse der Transistoren dar. Bei den folgenden Schaltungen ist immer angenommen, daß die MOS-Transistoren im Abschnürbereich betrieben werden und sich damit näherungsweise eine quadratische Abhängigkeit des Drainstromes von der effektiven Gate-Source-Spannung ergibt:

$$I_D = \frac{B_0}{2} \cdot \frac{W}{L} \cdot (U_{GS} - U_{Th})^2$$

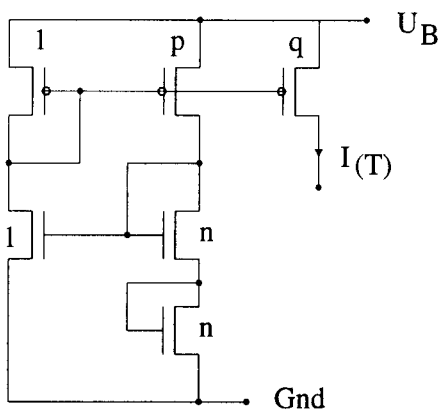


Abbildung 1: Prinzipbild der temperaturabhängigen Stromquelle nach [1]

Abbildung 1 zeigt das Prinzipbild eines Temperatursensors nach [1], der sein Ausgangssignal von der Temperaturabhängigkeit der „Steilheit“ $B_0 \sim \mu_0$ und der Abhängigkeit von U_{Th} ableitet:

$$I(T) \sim B_0(T) \cdot U_{Th}^2(T)$$

Das verwendete Prinzip in Abbildung 2 beruht darauf, daß durch den Spannungsabfall $R \cdot I(T)$ ein Offset in den Gate-Source-Spannungen der beiden N-Kanal-Transistoren erzeugt wird.

$$I(T) = \frac{1}{R^2(T) \cdot \frac{B_0(T)}{2} \cdot \frac{W}{L}} \cdot \left(1 - \sqrt{\frac{p}{n}} \right)$$

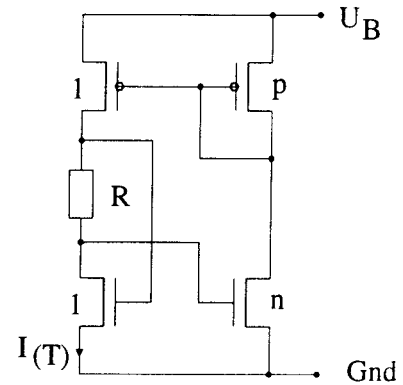


Abbildung 2:

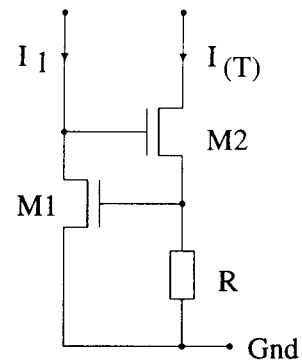


Abbildung 3:

Eine weitere Möglichkeit zur Erzeugung des Sensorsignals ist in Abbildung 3 gezeigt [3]. Es ergibt sich der temperaturabhängige Strom:

$$I(T) = \frac{1}{R(T)} \cdot \left(U_{Th}(T) + \sqrt{\frac{I_1}{\frac{B_0(T)}{2} \cdot \frac{W}{L}}} \right)$$

Hier ist im Gegensatz zu den beiden anderen Schaltungen der Strom $I(T)$ nicht über einen Stromspiegel zum Eingangsstrom I_1 zurückgeführt, da diese Schaltung auch einen nahezu vom Strom I_1 unabhängigen Strom liefern kann, wenn man das W/L -Verhältnis des Transistors M1 sehr groß wählt. Der Wurzelausdruck kann

dann gegen $U_{Th}(T)$ vernachlässigt werden.

Die vorangegangenen Stromquellen leiten ihre Temperaturabhängigkeit von prozeßabhängigen Größen ab und sind daher ziemlich abhängig vom Können des Halbleiterherstellers. Die letzte hier betrachtete Schaltung, Abbildung 4, benutzt die Temperaturabhängigkeit der Basis-Emitter-Spannung von Bipolar-Transistoren. Dieses Verhalten stützt sich hauptsächlich auf physikalische Größen, abgesehen von der schwachen Temperaturabhängigkeit der Beweglichkeit μ_0 , und ist deshalb recht prozeßunabhängig. Diese Bipolar-Transistoren sind allerdings nur als Parasitärtypen im CMOS-Prozeß vertreten, jedoch ist die Genauigkeit der Substrat-Transistoren ausreichend [2]. Laterale Transistoren müssen aber vermieden werden, denn sie besitzen immer einen parasitären Substrat-Transistor, sodaß ein unerwünschter Substratstrom fließt.

Für $n = p = 1$ erhält man folgenden Strom:

$$I(T) = \frac{U_{BE1} - U_{BE2}}{R} = \frac{k \cdot T}{e \cdot R} \cdot \ln(q)$$

Der Sensorstrom ist also proportional zur absoluten Temperatur T (ptat current), solange der Widerstand temperaturunabhängig ist.

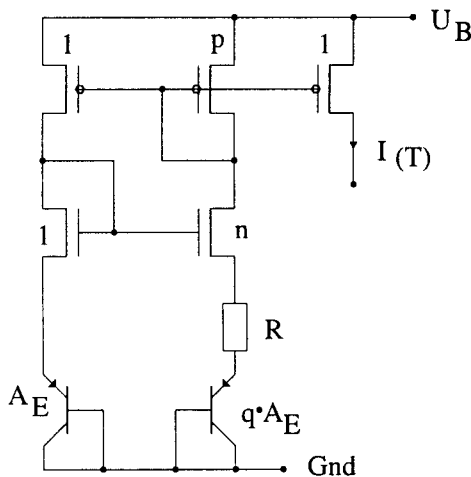


Abbildung 4: Prinzipbild einer Stromquelle mit linearer Temperaturabhängigkeit

4 Referenzquelle

Als Referenz kann im Prinzip eine Spannungs- oder Stromquelle verwendet werden. Da hier das Konzept mit Strömen verfolgt wird, sei nur der Vollständigkeit halber eine auf klassischem Prinzip beruhende Bandgap-Referenz in Abbildung 5 gezeigt. Der Strom durch die bipolaren Transistoren ist proportional zur absoluten Temperatur und damit auch die Spannung über R_1 , die Basis-Emitter-Spannung hat einen negativen Temperaturkoeffizienten, sodaß bei richtiger Dimensionierung die Spannung U_{ref} temperaturunabhängig und gleich der Bandabstandsspannung wird. Die Schaltung ist schlecht realisierbar in Digital-CMOS-Technologie, da die lateralen bipolaren Transistoren immer einen parasitären Substrat-Transistor beinhalten.

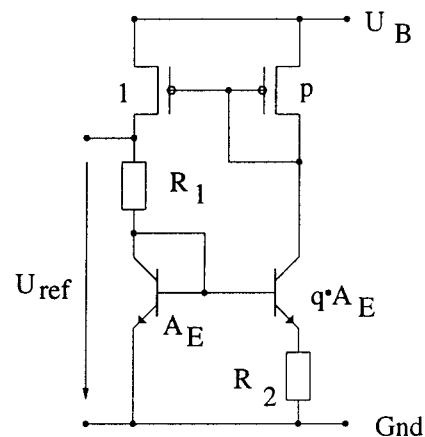


Abbildung 5: Prinzipbild einer Bandgap-Spannungsreferenz

Das Prinzip der entwickelten Stromreferenz ist in Abbildung 6 dargestellt. Oberflächlich betrachtet, ist kein großer Unterschied zu Abbildung 4 zu erkennen. Der Unterschied liegt im Widerstand R_1 . Durch ihn fließt ein Strom mit negativem Temperaturkoeffizienten, da die über ihm liegende Spannung gleich der Basis-Emitter-Spannung des linken Bipolar-Transistors ist. Bei richtiger Dimensionierung

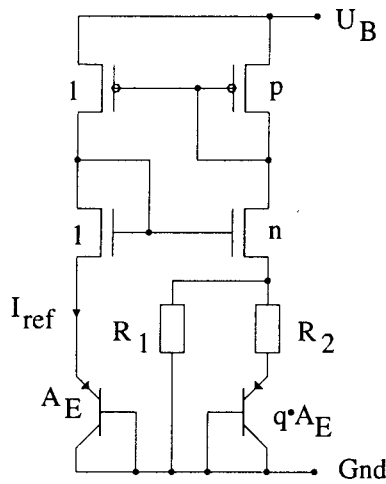


Abbildung 6: Prinzipbild der entwickelten Stromreferenz

wird der Referenzstrom bei einer vorgebbaren Temperatur T_0 unabhängig von der Temperatur. Das Funktionsprinzip zu dieser Schaltung stammt aus der I^2L -Technik [4] und ist in Abbildung 7 dokumentiert. Allerdings ist eine wesentlich andere Schaltungstechnik nötig, da die Kollektoren der im CMOS-Prozeß brauchbaren Transistoren mit Substrat verbunden sind.

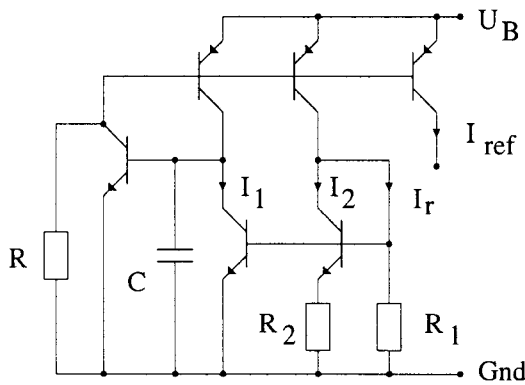


Abbildung 7: Regler für I^2L Injektorstrom nach [4].

Die einfachen Stromspiegel, der bisherigen Schaltungen weisen durch Kurzkanaleffekte eine hohe Abhängigkeit von der Betriebsspannung

auf. Deshalb wird die Schaltung durch geregelte Kaskodestufen nach [5] verbessert. Das Prinzip zeigt Abbildung 8. Über den Regeltransistor M3 wird praktisch die Drain-Source-Spannung von M1 konstant gehalten. Die im Prinzipbild angegebene ideale Stromquelle kann nur annähernd realisiert werden, sodaß die Regelschaltung nur eine begrenzte Verstärkung erreicht.

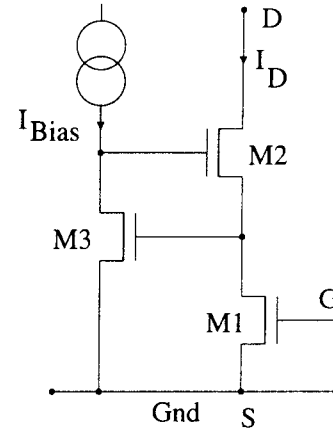


Abbildung 8: Prinzip der geregelten Kaskodestromquelle nach [5]

5 Simulierte Schaltung

Die starke Ähnlichkeit zwischen der Sensorschaltung und der Referenzschaltung ermöglicht Einsparungen. Allerdings muß das Sensorsignal noch etwas aufbereitet werden, damit der nachgeschaltete $\Sigma\Delta$ -Wandler in einem günstigen Aussteuerbereich betrieben werden kann. Und zwar muß das Sensorsignal für alle Temperaturen kleiner als das Referenzsignal bleiben, sollte aber bei der tiefsten Betriebstemperatur nicht zu klein werden. Deshalb wird zu dem temperaturproportionalen Strom ein kleiner Anteil eines Stromes mit negativem Temperaturkoeffizienten addiert. Die komplette Schaltung zeigt Abbildung 9. Die Referenzquelle ist in der rechten Hälfte der Schaltung angeordnet und besteht, abgesehen von den Stromspiegeln aus den Transistoren Q_1, Q_2 und den Widerständen R_1 und R_2 . Q_3 dient der Kaskoderegelsstufe als

temperaturabhängiger Offset. Links kann man die Aufbereitung des Sensorsignals erkennen. Der Strom I_+ ist der Strom mit positivem Temperaturkoeffizienten. Der Strom I_- ist der Strom mit negativem Temperaturkoeffizienten. Zusammen bilden die Ströme über einen Stromspiegel das Sensorsignal $I(T)$. Am rechten Rand ist der Ausgang für den Referenzstrom und ein zusätzlicher Ausgang. Die Referenzstromquelle stellt einen Regelkreis dar, deshalb muß die Stabilität gewährleistet werden. Dies wird mit den beiden Kapazitäten erreicht. Die ganze Schaltung besitzt zwei stabile Arbeitspunkte, nämlich beim Strom Null und beim gewünschten Arbeitspunkt. Deshalb werden über die Signale „Start“ und „Stop“ die beiden Arbeitspunkte durch kurze Impulse an den entsprechenden Eingängen eingestellt.

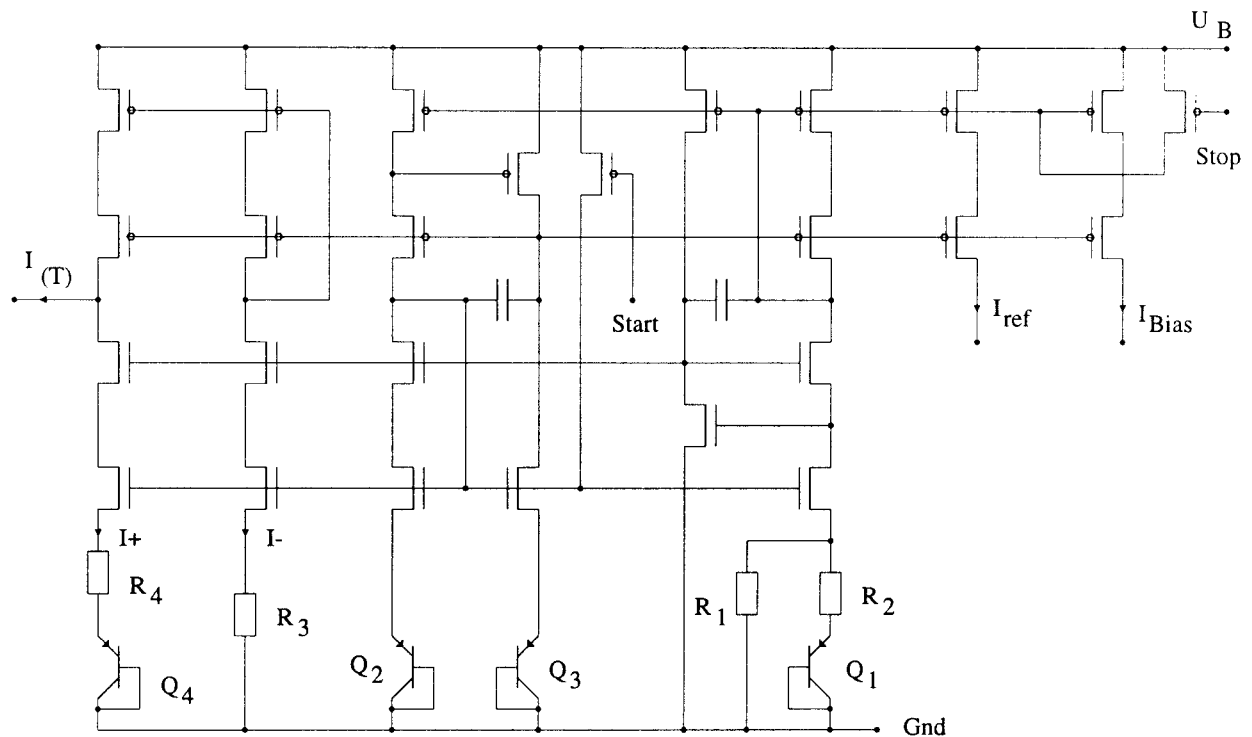


Abbildung 9: Schaltplan der kombinierten Sensor-Referenz

6 Ergebnisse der Simulation

Die Schaltung wurde für einen Referenzstrom von $1\mu A$ ausgelegt und mit AMS-1,2 μ -Parametern simuliert. Die Ergebnisse sind in den beiden folgenden Abbildungen dargestellt. In Abbildung 10 erkennt man, daß der gewünschte Referenzstrom etwa erreicht wird, allerdings ist bei hohen Temperaturen ab ca. $110^\circ C$ ein dramatischer Anstieg erkennbar. Das liegt zum Teil am sehr hohen Temperaturkoeffizienten der verwendeten Poly-Si-Widerstände. Auch die Stromspiegelschaltungen haben Anteil, denn die Temperaturkompensation ist nur für eine vorgebbare Temperatur erreichbar, sodaß bei abweichender Temperatur die Symmetrie und damit das „matching“ gestört ist. Dieses Verhalten ist bei Operationsverstärkerimplementierungen günstiger. Das bezogene Sensorsig-

gnal zeigt Abbildung 11. Die Temperaturkennlinie ist ausreichend linear und besitzt nur eine sehr geringe Abhängigkeit von der Versorgungsspannung.

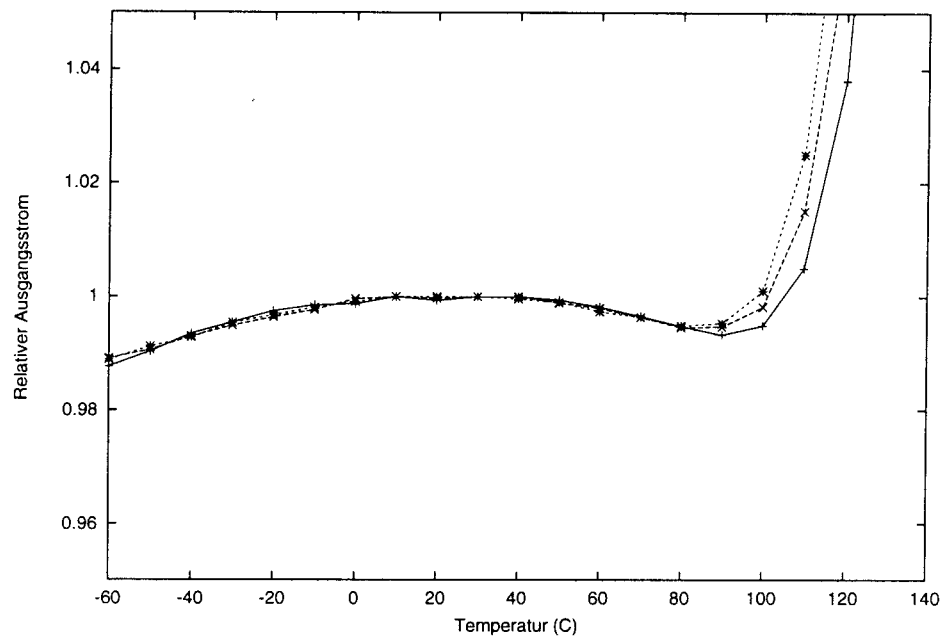


Abbildung 10: Simulierter Referenzstrom als Funktion der Temperatur; Parameter Betriebsspannung 2,7V; 3V; 3,3V

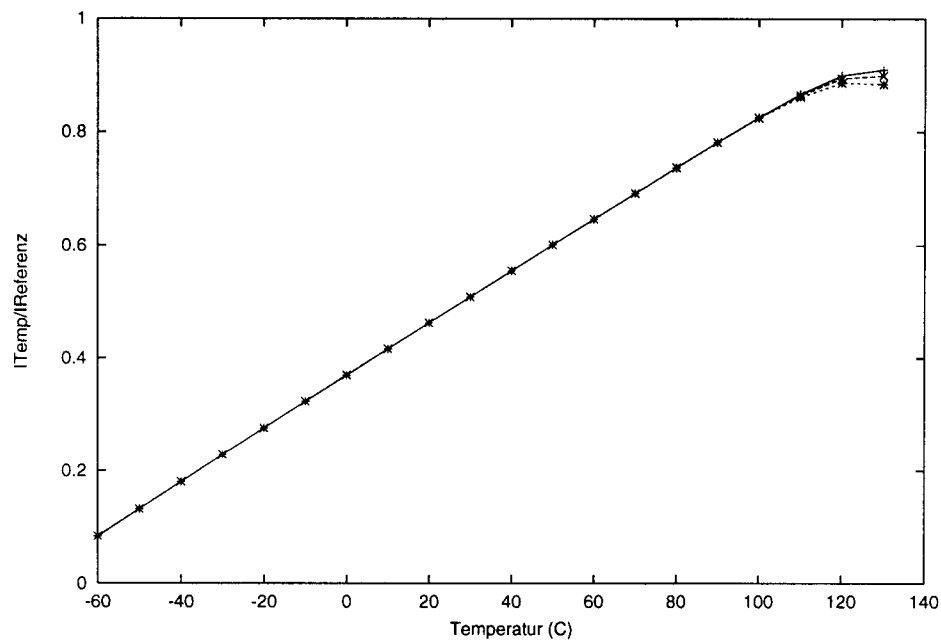


Abbildung 11: Simuliertes Sensorsignal bezogen auf den Referenzstrom als Funktion der Temperatur; Parameter Betriebsspannung 2,7V; 3V; 3,3V

Literatur

- [1] Székely, V., Márta, Cs., Kohári, Zs. und Rencz, M.: *CMOS Sensors for On-Line Thermal Monitoring of VLSI Circuits*, IEEE Transactions on Very Large Scale Integration, Vol. 5, S. 270-276, No. 3, Oct. 1997
- [2] Bakker, A. und Huijsing, J. H.: *Micropower CMOS Temperature Sensor with Digital Output*, IEEE Journal of Solid-State Circuits, Vol. SC-31, S. 289-297, No.7, July 1997
- [3] Pan, T.-W. und Abidi, A. A.: *A 50-dB Variable Gain Amplifier Using Parasitic Bipolar Transistors in CMOS*, IEEE Journal of Solid-State Circuits, Vol. SC-24, S. 951-961, No. 4, Aug. 1989
- [4] Bruun, E. und Hansen, O.: *Current Regulators for I^2L Circuits to be Operated from Low-Voltage Power Supplies*, IEEE Journal of Solid-State Circuits, Vol. SC-15, S. 796-799, No. 5, Oct. 1980
- [5] Säckinger, E. und Guggenbühl, W.: *A High-Swing, High-Impedance MOS Cascode Circuit*, IEEE Journal of Solid-State Circuits, Vol. SC-25, S. 289-297, No.1, Feb. 1990

