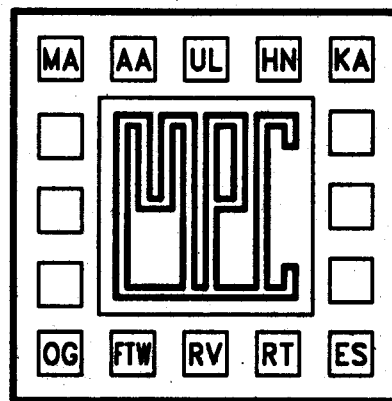


MULTIPROJEKT CHIP - GRUPPE

BADEN - WÜRTTEMBERG

WORKSHOP JANUAR 1994

ULM



HERAUSGEBER: FACHHOCHSCHULE ULM

© 1994 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,
Prittwitzstr. 10, 89075 Ulm.

Inhaltsverzeichnis Teil 1

1. SPICE 3 - Ein Überblick Seite 5
H. Gall, Fa. Bausch-Gall GmbH, München

2. Anwendung von VHDL Seite 21
Entwicklung einer Heizungsregelung mit VHDL
G. Ehmann, F. Bersdorf, B. Kohlhammer, FH Aalen

3. Zeitkontinuierliche integrierte Biquad-Filter Seite 47
M. Rieger, FH Albstadt-Sigmaringen

4. Entwicklung eines 16-Bit Mikroprozessor-Kernels Seite 65
mit Hilfe von VHDL
F. Zimpfer, T. Gieringer, D. Jansen, FH Offenburg

5. PIO-Baustein mit 48 freiprogrammierbaren Pins Seite 79
K. Ender, W. Ludescher, FH Ravensburg-Weingarten

6. Simulation und Design eines Breitbandverstärkers Seite 87
in BiCMOS-Technik
T. Ehrenbeck, M. Schmid, G. Forster, FH Ulm

Inhaltsverzeichnis Teil 2

Gefertigte Bausteine:

1. CMOS-ASIC mit niedrigem Leistungsverbrauch unter extremen Einsatzbedingungen Seite 97
K. Schmidt, B. Röllgen, FH Furtwangen
2. Parametrisierbarer kaskadierbarer digitaler Filterbaustein Seite 99
M. Sator, A. Gottscheber, B. Kohlhammer,
G. Busch, H. Graf, FH Aalen
3. Erdfreier Gyrtator auf B500A Bipolar Array Seite 101
S. Beck, R. Schrottenholzer, B. Kohlhammer,
H. Graf, FH Aalen
4. Realisierung analoger Grundsaltungen auf einem Bipolaren Transistor-Array 'FH-HN-Team1' Seite 103
M. Amann, G. Günther, H. Clauss, FH Heilbronn
5. Realisierung analoger Grundsaltungen auf einem Bipolaren Transistor-Array 'FH-HN-Team2' Seite 105
S. Eickert, M. Zimmermann, H. Clauss, FH Heilbronn
6. Baustein zur Datenverschlüsselung Seite 107
P. Jonski, A. Führer, FH Ulm

7. Baustein für eine Fussgängerampel
in Gate-Array-Ausführung Seite 109
A. Führer, M. Faulhaber, FH Ulm

8. Baustein für eine Fussgängerampel
in Standard-Zellen-Ausführung Seite 111
A. Führer, M. Faulhaber, FH Ulm

9. Automatische Messwertkorrektur
für intelligente Sensoren Seite 113
W. Rülling, FH Furtwangen

10. Kurzbeschreibung Chip "STRA" Seite 115
L. Castello, H. Töpfer, FH Esslingen

SPICE3 - Ein Überblick

Hans Gall

BAUSCH-GALL GmbH, München

Workshop

WS 93/94

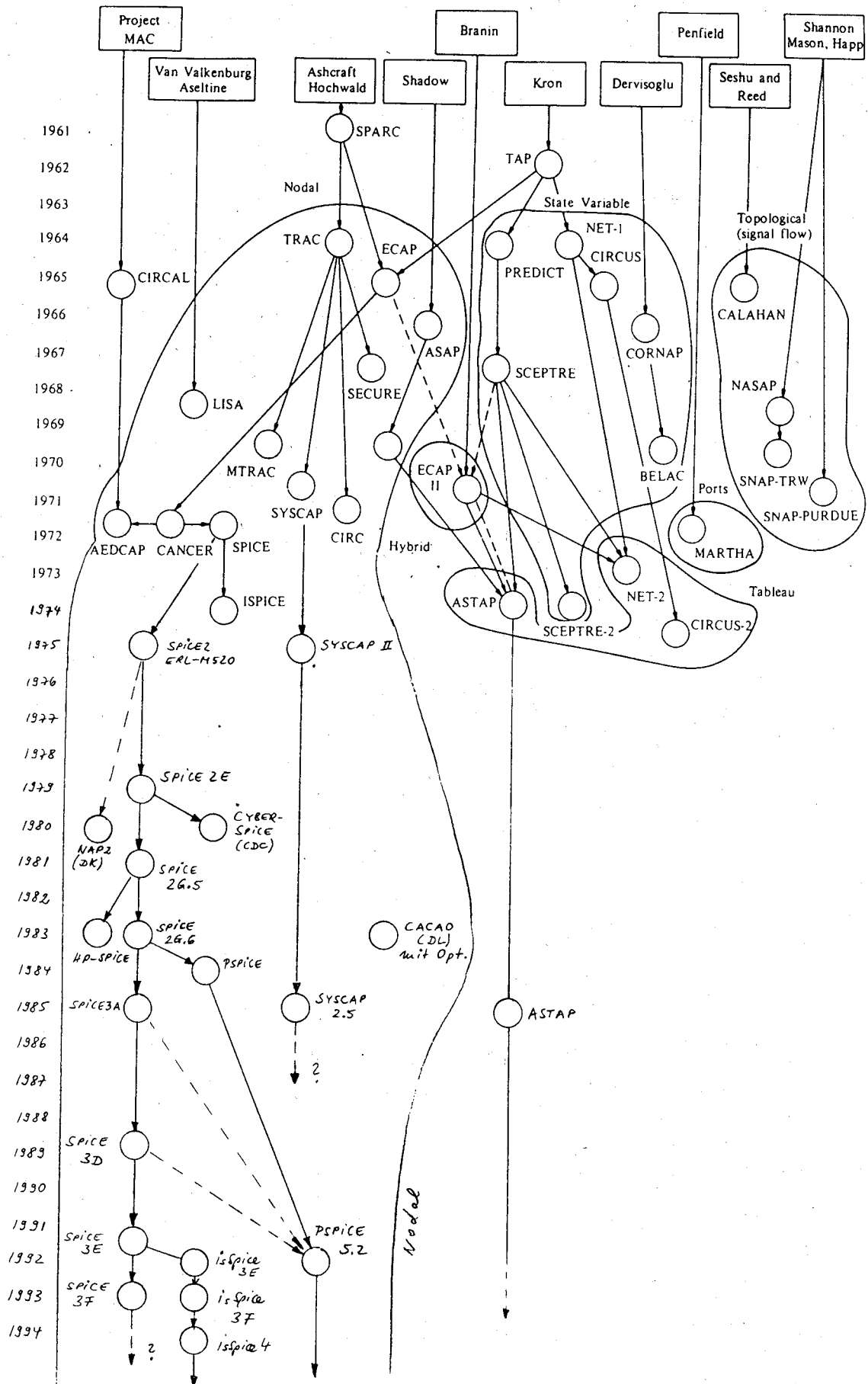
der

Multi Projekt Chip

Gruppe

28. Januar 1994

Geschichte der Netzwerkanalyseprogramme



Vergleich der Eigenschaften von SPICE 2G.6 und SPICE 3E.2 [14]

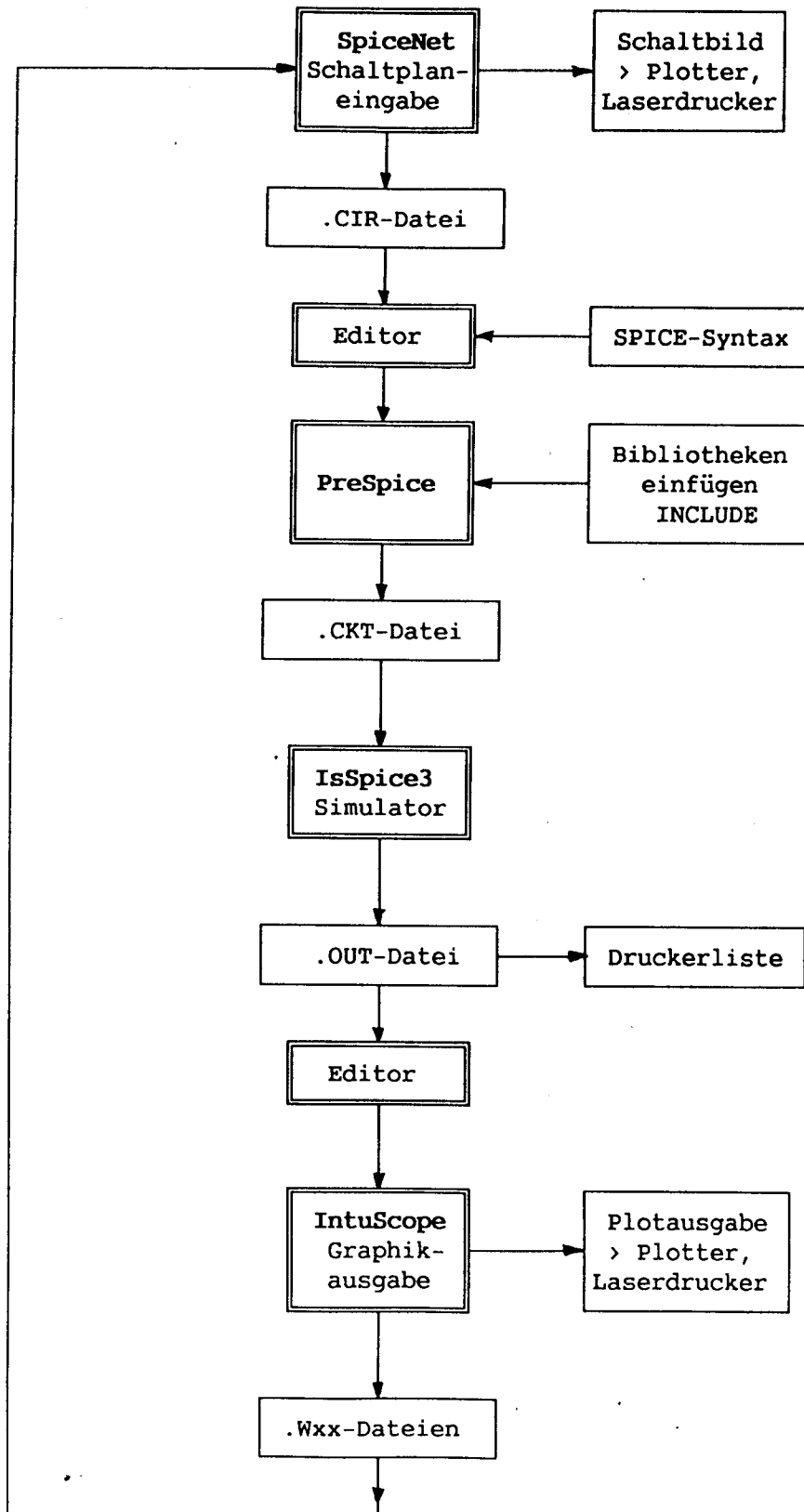
Table 1, A comparison of Berkeley SPICE features

Analysis/Control	SPICE Name	SPICE 2G.6	SPICE 3E.2
AC	.AC	Yes	Yes
Noise	.Noise	Yes	Yes *
Distortion	.Disto	Limited	Yes *
Pole-Zero	.PZ	No	Yes
DC Sweeps	.DC	Yes	Yes
Operating Point	.OP	Yes	Yes
Transfer Function	.TF	Yes	Yes
Sensitivity	.Sens	DC Only	Yes * (AC, DC, Transient)
Transient	.Tran	Yes	Yes
Fourier	.Four	Yes	Yes
Temperature	.Temp	Limited	Yes * (.OPTIONS TEMP)
Print	.Print	Yes	Yes
Plot	.Plot	Yes	Yes
Width	.Width	Yes	No

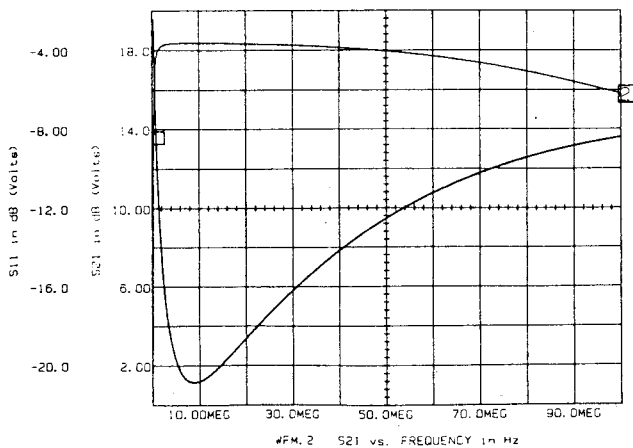
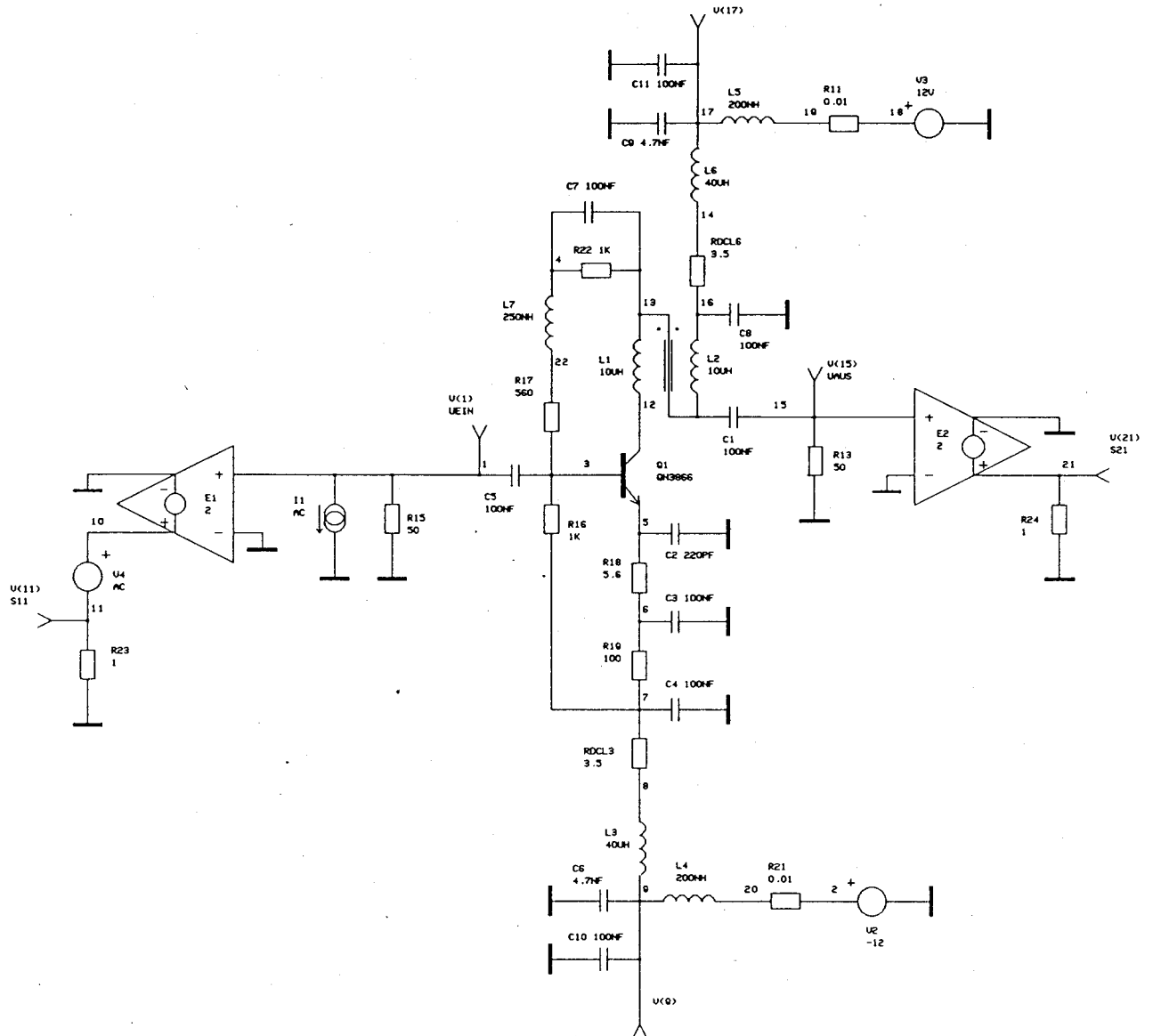
* indicates syntax change or major functional enhancement

Device Type	Key letter	SPICE 2G.6	SPICE 3E.2
Passive			
resistor	R	Yes	Yes
semiconductor resistor	R	No	Yes
capacitor	C	Yes	Yes
semiconductor capacitor	C	No	Yes
polynomial capacitor	C	Yes	No
inductor	L	Yes	Yes
polynomial inductor	L	Yes	No
coupled inductors	M	Yes	Yes
ideal transmission line	T	Yes	Yes
lossy RLCG transmission line	O	No	Yes
uniform distributed RC/RD transmission line	U	No	Yes
voltage controlled switch with hysteresis	S	No	Yes
current controlled switch with hysteresis	W	No	Yes
Sources			
independent voltage/current sources	V/I	Yes	Yes
linear dependent voltage/current sources	E/F/G/H	Yes	Yes
polynomial dependent voltage/current sources	E/F/G/H	Yes	No
arbitrary nonlinear dependent source	B	No	Yes
Semiconductors			
diodes	D	Yes	Yes
transistors	Q	Yes	Yes
JFET	J	Yes	Yes
MOSFETS (levels 1, 2, & 3)	M	Yes	Yes
MOSFETS (levels 4 (BSIM1), 5 (BSIM2), & 6 (modified level 3))	M	No	Yes
MESFET (Statz)	Z	No	Yes
Interface/Convergence			
Interactive or Batch Mode Simulation		Batch only	Yes
Command Line Control Environment		No	Yes
Integrated Graphics Post Processor		No	Yes
Convergence Algorithms		Yes (Source Stepping)	Yes (GMIN and Source Stepping)

Überblick zum Programmsystem ICAP/4 von intusoft



Anwendungsbeispiel: Breitbandverstärker 2 - 50 MHz



#PM.2 S21 vs. FREQUENCY in Hz

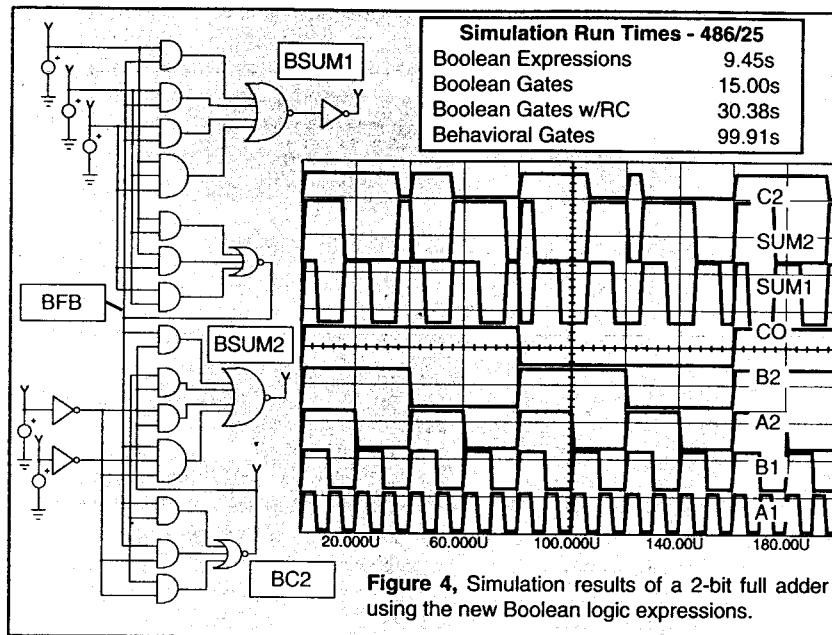
Anwendungsbeispiel: Breitbandverstärker 2 - 50 MHz (Forts.)**Von SpiceNet erzeugte Schaltungsbeschreibung (Netzliste)**

```
av24
*ALIAS V(1)=UEIN
*ALIAS V(15)=UAUS
*ALIAS VDB(21)=S21
*ALIAS VDB(11)=S11
.PRINT AC V(1) VP(1) V(15) VP(15)
.PRINT AC V(9) VP(9) V(17) VP(17)
.PRINT AC VDB(21) VP(21) VDB(11) VP(11)
.PRINT AC VR(11) VI(11)
*SPICE_NET
*INCLUDE BJTN.LIB
.AC LIN 500 0.1MEGHZ 100MEGHZ
.OPTIONS LIMPTS=10000 ITL5=0
*ALIAS V(1)=UEIN
*ALIAS V(15)=UAUS
*ALIAS V(21)=S21
*ALIAS V(11)=S11
.PRINT AC V(1) VP(1) V(15) VP(15)
.PRINT AC V(9) VP(9) V(17) VP(17)
.PRINT AC V(21) VP(21) V(11) VP(11)
L1 12 13 10UH
L2 13 16 10UH
K1 L1 L2 0.99
R22 4 13 1K
R17 22 3 560
C1 13 15 100NF
R18 5 6 5.6
R19 6 7 100
C2 5 0 220PF
C3 6 0 100NF
C4 7 0 100NF
R16 3 7 1K
C5 1 3 100NF
R15 1 0 50
RDCL3 7 8 3.5
L3 8 9 40UH
C6 0 9 4.7NF
L4 9 20 200NH
R21 20 2 0.01
V2 2 0 -12
R11 19 18 0.01
V3 18 0 12V
R13 15 0 50
L5 17 19 200NH
C7 4 13 100NF
RDCL6 14 16 3.5
L6 14 17 40UH
C8 0 16 100NF
C9 0 17 4.7NF
E1 10 0 1 0 2
I1 1 0 AC -20M
V4 10 11 AC 1
R23 11 0 1
E2 21 0 15 0 2
R24 21 0 1
C10 0 9 100NF
C11 0 17 100NF
L7 22 4 250NH
Q1 12 3 5 QN3866
.END
```

Simulation einer Digitalschaltung mit IsSpice3
(intusoft-Newsletter September 1992)

Simulating A Binary Full Adder

To illustrate the power and flexibility of Boolean logic expressions, a 2-bit binary full adder, similar to the TI SN7482, was simulated. The entire range of input states were simulated under four modeling scenarios; Boolean expressions of the entire circuit (Figure 5), simple boolean expressions for each individual gate, individual boolean gates with an RC delay added to the output of each gate, and behavioral gates using polynomial expressions. The simulations produced comparable results as shown in Figure 4, with the Boolean expressions running an order of magnitude faster than the digital gates using behavioral method.



```

Boolean expression describing 2 Bit Binary Full Adder
BSUM1 SUM1 0 V= (V(C0)&V(FB)) | (V(A1)&V(FB)) |
+ (V(B1)&V(FB)) | (V(C0)&V(A1)&V(B1))
BSUM2 SUM2 0 V= ~(V(FB)&V(C2)) | (~V(A2)&V(C2)) |
+ (~V(B2)&V(C2)) | (V(FB)&~V(A2)&~V(B2))
BC2 C2 0 V= ~(V(FB)&~V(A2)) | (V(FB)&~V(B2)) | (~V(B2)&~V(A2))
BFB FB 0 V= ~(V(C0)&V(A1)) | (V(C0)&V(B1)) | (V(B1)&V(A1))

Subcircuit using If-Then Else expression describing an A/D
x1 1 2 10 adc
.subckt adc in out bin
b1 bin 0 v= (v(in) > 1) ? 1 : 0 <-- If Then Else function
b2 out 0 v= 2*(v(in) - v(bin))
.ends

Figure 5, The boolean expressions that form the full adder and If-Then-Else show how
easy mixed mode simulations can be implemented in IsSpice3.
    
```

IsSpice3-Simulation eines Lüfterreglers mit Fuzzy-Logik
(intusoft-Newsletter Januar 1993)

A Fuzzy Fan Controller

The first processing step in a fuzzy control system is the conversion of a set of input values, called fuzzy variables, into a set of output levels, called membership values. The transfer function relationship between the fuzzy variables and the membership values is called a Membership Function; for example, the trapezoidal response between the input Humidity and the output

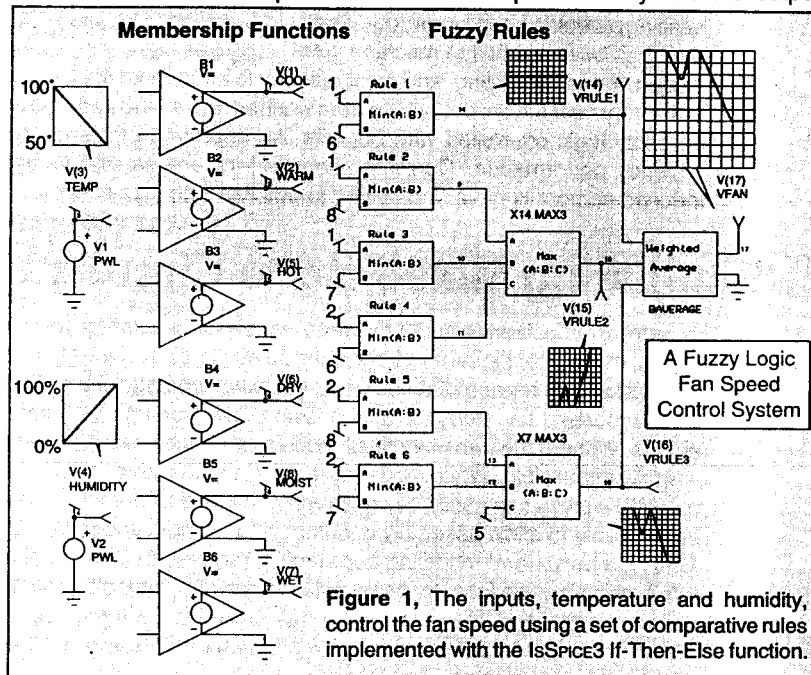


Figure 1, The inputs, temperature and humidity, control the fan speed using a set of comparative rules implemented with the IsSpice3 If-Then-Else function.

```

FUZZY LOGIC CONTROL SYSTEM
.TRAN 01 1
.PRINT TRAN V(3) V(4) V(14) V(15) V(16) V(17)
V1 3 0 PWL 0 100 1 50
V2 3 0 PWL 0 0 1 100
*COOL FUZZY INPUT MEMBERSHIP
B1 1 0 V= V(3) < 60 ? 1 : V(3) < 75 ? 5 - V(3) / 15 : 0
*WARM FUZZY INPUT MEMBERSHIP
B2 2 0 V= V(3) < 75 ? -4 + V(3) / 15 : V(3) < 90 ? 6 - V(3) / 15 : 0
BLIMIT 2 0 V= V(20) < 0 ? 0 : V(20)
*HOT FUZZY INPUT MEMBERSHIP
B3 5 0 V= V(3) < 75 ? 0 : V(3) < 90 ? -5 + V(3) / 15 : 1
*DRY FUZZY INPUT MEMBERSHIP
B4 6 0 V= V(4) < 20 ? 1 : V(4) < 40 ? 2 - V(4) / 20 : 0
*MOIST FUZZY INPUT MEMBERSHIP
B6 8 0 V= V(4) < 40 ? -1 + V(4) / 20 : V(4) > 60 ? 4 - V(4) / 20 : 1
BLIMIT2 8 0 V= V(80) < 0 ? 0 : V(80)
*WET FUZZY INPUT MEMBERSHIP
B5 7 0 V= V(4) < 60 ? 0 : V(4) < 80 ? -3 + V(4) / 20 : 1
* FUZZY RULES
XRULE1 1 6 14 MIN
XRULE2 1 8 9 MIN
XRULE3 1 7 10 MIN
XRULE4 2 6 11 MIN
XRULE5 2 8 13 MIN
XRULE6 2 7 12 MIN
X7 13 12 7 16 MAX3
X14 9 10 11 15 MAX3
* DEFUZZIFICATION
BAVERAGE 17 0 V=(V(15)*40) + (V(16)*100) / (V(14) + V(15) + V(16))
    
```

Table 1, The IsSpice3 netlist for the Fuzzy Logic Fan-Speed Control System. B elements that use other nodes as input can be placed inside subcircuits in order to avoid node renumbering problems.

```

*INCLUDE FUZZY.LIB
.SUBCKT MIN 1 2 3
*Connections In1 In2 Out
BMIN 3 0 V=V(1) < V(2) ? V(1) : V(2)
.ENDS
.SUBCKT MAX3 1 2 3 4
*Connections In1 In2 In3 Out
BMAX1 5 0 V=V(1) > V(2) ? V(1) : V(2)
BMAX2 4 0 V=V(3) > V(5) ? V(3) : V(5)
.ENDS
.END
    
```

IsSpice3-Simulation eines PID-Regelkreises
(intusoft-Newsletter Januar 1993)

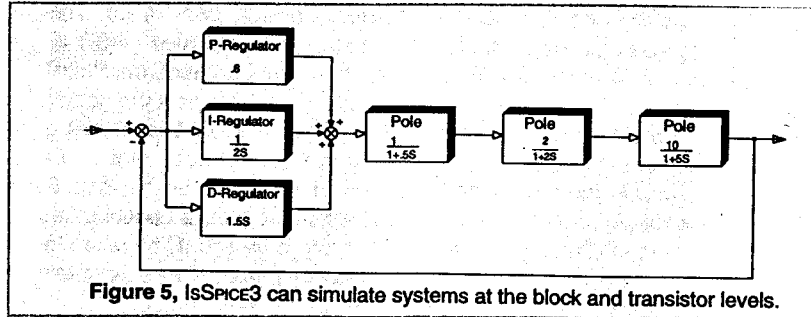


Figure 5, IsSpice3 can simulate systems at the block and transistor levels.

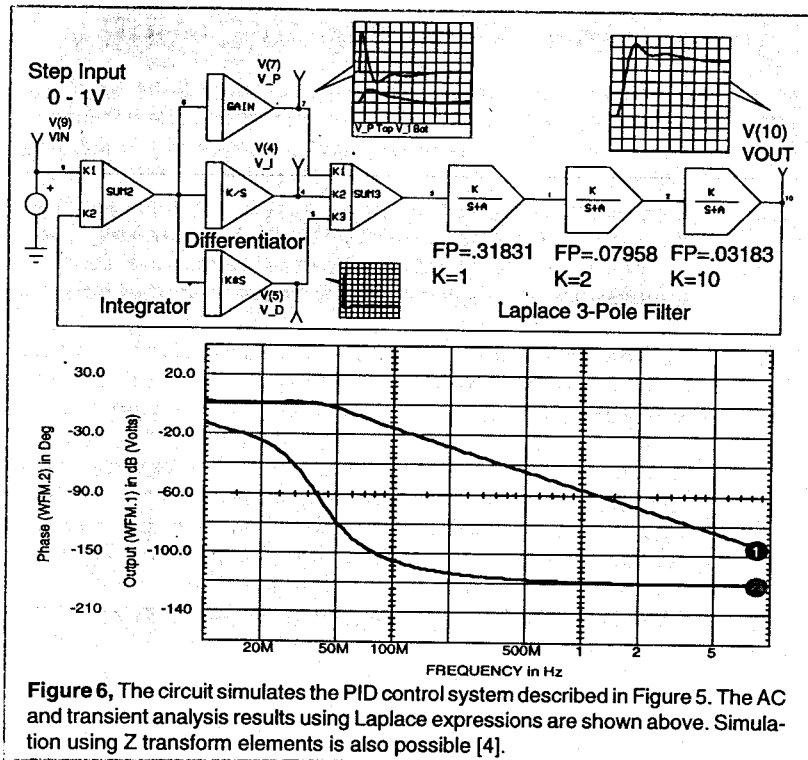


Figure 6, The circuit simulates the PID control system described in Figure 5. The AC and transient analysis results using Laplace expressions are shown above. Simulation using Z transform elements is also possible [4].

Beispiel für die Interactive Command Language bei IsSpice3
(intusoft-Newsletter September 1993)

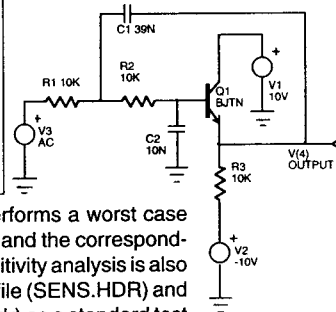
```

*Simulation control script using IsSPICE3's Interactive Command Language
.control                                     ; begin the script
op                                           ; operating point
let rtol=0.1                               ; set resistor tolerance to 10%
function tol(r,s) r + r * ((s>0)*rtol-(s<=0)*rtol) ; create tolerance function
print v(4)                                  ; print nominal output voltage
show r                                      ; show the resistor models/values
sens v(4)                                   ; run a DC sensitivity analysis
print r1 r2 r3                             ; print the resistor sensitivities
alter @r1[resistance] = tol(@r1[resistance],r1) ; change the resistor values to...
alter @r2[resistance] = tol(@r2[resistance],r2) ; ... their max/min values based...
alter @r3[resistance] = tol(@r3[resistance],r3) ; ... on the sensitivity results
op                                           ; operating point
echo worst case                             ; write into the output file
print v(4)                                  ; Worst Case Output Voltage
show r                                      ; show the resistor models/values
sens v(4)                                   ; rerun the sensitivity analysis
print r1 r2 r3                             ; show the new sensitivities
sens V(4) ac dec 10 100 100k              ; run an AC sensitivity analysis
print norm(mag(c1)) norm(mag(c2))         ; print cap. sensitivities vs. Freq.
.endc                                       ; end the script

```

Simulation Results
Initial starting v(4) = -8.40353e-01
Initial resistances r3=1e+04, r2=1e+04,
r1=1e+04
Sensitivities r1=-9.76087e-06, r2=-9.76087e-06,
r3= 2.027591e-05
Worst case v(4) = -8.03726e-01
Final resistances r3=1.1e+04, r2=9e+03,
r1=9e+03

Figure 10. The IsSPICE3 simulation script performs a worst case analysis, finding the output voltage at node 4 and the corresponding resistor values that produce it. An AC sensitivity analysis is also run. The entire script can be stored away in a file (SENS.HDR) and included in the SPICE netlist (.include sens.hdr) as a standard test procedure.



Laufzeitvergleich

für verschiedene SPICE-Versionen (*intusoft-Newsletter* November 1993)



SPICE Benchmark Runtimes

Table 1, Partial list of the benchmark circuits used to test the speed and convergence of various SPICE programs. Index is compiled from a suite of 28 circuits. Long Index is based on LONG time only. (Times are in seconds)

Software	2G.6	IsSPICE4	IsSPICE4	IsSPICE4	IsSPICE3	IsSPICE3	PSpice®/16	Number
Hardware	Vax8700	Alpha/150	MIPS/100	Pentium/60	Pentium/60	486/33	486/33	of
Op. Sys.	VMS	NT	NT	NT	DOS	DOS	DOS	Devices
Standard Benchmark Circuits								
Diff. Pair	1.64	0.42	0.58	0.47	0.77	2.65	4.12	12
RCA3040	4.66	0.68	1.10	0.85	1.10	4.02	6.76	26
UA741	11.74	0.95	1.82	1.47	1.75	5.83	13.02	41
MECL Gate	3.61	0.50	0.87	0.75	0.87	2.97	5.05	35
DC Convergence Test Circuits								
Flip-Flop	0.20	0.05	0.07	0.05	0.05	0.33	0.55	17
74STTL	3.30	0.48	0.80	0.67	0.82	2.92	4.67	18
74LTTL	3.46	0.47	0.75	1.37	0.77	2.87	5.33	16
Transient Test Circuits								
Simple RC	0.49	0.20	0.15	0.13	0.17	0.93	0.93	4
ECL Gate	1.51	0.30	0.37	0.35	0.43	1.70	2.31	14
Schmitt Trig	1.95	0.32	0.43	0.38	0.48	1.92	2.80	15
Intusoft General Test Circuits								
SAMPLE	3.83	0.60	0.95	0.77	0.98	5.00	5.60	15
MOS Chain	4.93	1.48	3.23	2.47	2.57	8.47	17.68	12
Elliptic Filter	12.21	1.07	1.62	1.38	1.58	7.03	22.14	118
DIV/4	47.03	5.70	12.08	9.38	9.77	27.80	78.22	256
LONG	2311.40	329.52	760.70	498.32	506.90	1272.80	3054.72	2967
Index	.76	5.81	3.23	3.88	3.45	1.00	0.46	
Long Index	.55	3.86	1.67	2.55	2.51	1.00	0.42	

PSpice is a registered trademark of Microsim Corp

Empfehlenswerte Literatur zur Schaltungssimulation

- [1] Donald A. Calahan,
Rechnergestützter Schaltungsentwurf,
R. Oldenbourg Verlag München Wien, 1973, ISBN 3-486-34351-3

- [2] Graeme R. Boyle, Barry M. Cohn, Donald O. Pederson, James E. Solomon,
Macromodeling of Integrated Circuit Operational Amplifiers,
IEEE Journal of Solid-State Circuits, vol, SC-9, December 1974
S. 353-363

- [3] Laurence W. Nagel,
SPICE2: A COMPUTER PROGRAM TO SIMULATE SEMICONDUCTOR CIRCUITS,
Memorandum No. ERL-M520, 9 May 1975,
ELECTRONICS RESEARCH LABORATORY, College of Engineering,
University of California, Berkeley, CA 94720

- [4] Gabor C. Temes, Jack W. LaPatra,
Introduction to Circuit Synthesis and Design,
McGraw-Hill Book Company, 1977,
ISBN 0-07-063489-0

- [5] K.C. Gupta, Ramesh Garg, Rakesh Chadha,
COMPUTER AIDED DESIGN OF MICROWAVE CIRCUITS,
ARTECH HOUSE, INC., 1981,
610 Washington Street, Dedham, Massachusetts 02026, U.S.A.,
Standard Book Number: 0-89006-106-8

- [6] E.E.E. Hofer, H. Nielinger,
SPICE, Analyseprogramm für elektronische Schaltungen,
Benutzerhandbuch mit Beispielen,
Springer-Verlag Berlin, Heidelberg, 1985, ISBN 3-540-15160-5

- [7] Lawrence G. Meares, Charles E. Hymowitz,
Simulating with Spice,
intusoft 1988,
P.O. Box 6607, San Pedro, CA 90734-6607, U.S.A.

- [8] Paolo Antognetti, Giuseppe Massobrio,
Semiconductor Device Modeling with SPICE,
McGraw-Hill Book Company, Inc., New York, 1988,
ISBN 0-07-002107-4

- [9] Peter Christiansen,
Rechnergestütztes Entwickeln integrierter Schaltungen,
CAD vom Schaltungsentwurf bis zum Chip,
VOGEL Buchverlag Würzburg, 1989, ISBN 3-8023-0256-7

Empfehlenswerte Literatur zur Schaltungssimulation (Forts.)

- [10] Hans Spiro,
Simulation integrierter Schaltungen,
Verfahren und Praxis der rechnergestützten Simulation
nichtlinearer Schaltungen,
2., verbesserte Auflage,
R. Oldenbourg Verlag München Wien, 1990, ISBN 3-486-21660-0
- [11] Karl Heinz Müller,
Elektronische Schaltungen und Systeme,
Simulieren, Analysieren, Optimieren mit SPICE,
VOGEL Buchverlag Würzburg, 1990, ISBN 3-8023-0292-3
- [12] Haybatolah Khakzar, Albert Mayer, Reinhold Oettinger
Entwurf und Simulation von Halbleiterschaltungen mit SPICE,
SPICE, Transistormodelle, Vierpol- und Signalflußmethode,
rechnergestützter Entwurf von Elektronikschaltungen mit SPICE,
expert-Verlag, Eningen, 1991, ISBN 3-8169-0650-8
- [13] Karl Heinz Müller,
A Spice Cookbook,
(englische Übersetzung von [11]),
intusoft 1991,
P.O.Box 710, San Pedro, CA 90733-0710, U.S.A.
- [14] Charles Hymowitz,
A New SPICE comes of Age,
intusoft 1992,
P.O.Box 710, San Pedro, CA 90733-0710, U.S.A.,

Anwendung von VHDL

Entwicklung einer Heizungsregelung mit VHDL

Vorstellung der VHDL-Entwicklungsumgebung von Synopsys
und Vergleich mit dem Entwicklungssystem von Mentor
Graphics

Dipl.-Ing.(FH) Gerhard Ehmman und Frank Bersdorf

Fachhochschule Aalen
Fachbereich Elektronik/Technische Informatik

Betreuer Prof. Dr. Bernd Kohlhammer

Einleitung

Gewöhnlich erfolgt bei Heizungsregelungen die Einstellung der Vorlauftemperatur über die Außentemperatur. Dazu wird ein Temperaturfühler an die Außenseite eines Gebäudes angebracht. Der Temperaturfühler wird mit der Heizungsregelung verbunden. Die Vorlauftemperatur hängt direkt von der Außentemperatur ab. Am Regler angebrachte Drehregler ermöglichen zusätzlich noch eine Anpassung an lokale Gegebenheiten. Der Nachteil einer solchen Regelung ist jedoch offensichtlich. Zum einen besteht nur ein bedingter Zusammenhang zwischen benötigter Heizleistung und Außentemperatur. So ist beispielsweise, unter der Voraussetzung, daß die Außentemperatur in beiden Fällen gleich groß ist, bei einem sonnigen, windstillen Wintertag der Bedarf an Heizenergie bedeutend geringer als bei einem bedeckten stark windigen Wintertag. Da der Regler wahrscheinlich so eingestellt ist, daß in beiden Fällen die Raumtemperatur genügend hoch ist, bedeutet dies, daß bei sonnigem Wetter die Vorlauftemperatur höher als notwendig ist, so daß auf diese Weise unnötig Heizenergie verbraucht wird. Ein weiterer Nachteil des gewählten Verfahrens ist, daß auf diese Weise ein Nachregeln der Raumtemperatur mittels an Heizkörpern angebrachter Ventile unumgänglich ist.

Die hier vorgestellte Heizungsregelung hat diese Nachteile nicht. Die nötige Heizleistung und somit auch die Vorlauftemperatur orientiert sich direkt an dem Wärmebedarf der Zimmer. Die Heizungsregelung läßt die Temperatur von bis zu vierzehn Räumen mit Hilfe eines Analog-Digital-Umsetzers ein. Für jeden Raum läßt sich die Temperatur mittels eines am Regler angebrachten Anzeigebausteins und zusätzlicher Tasten separat einstellen. Zusätzlich läßt sich für die Nachtabsenkung jeden Raumes eine extra Temperatur eingeben.

Die Regelung erfolgt so, daß zunächst der Raum bestimmt wird, der den höchsten Wärmebedarf hat. Die Temperatur dieses Raumes wird kontinuierlich geregelt. Dazu werden zwei ineinanderverschachtelte Regler verwendet. Auf diese Art ist gewährleistet, daß die Vorlauftemperatur immer den optimalen Wert hat, so daß nicht unnötig Heizenergie verschwendet wird. Die Regelung der übrigen Räume erfolgt mit Hilfe von Ventilen mittels Zweipunktregelung.

Auf Bild 1 sind die Komponenten der Heizungsregelung zu sehen.

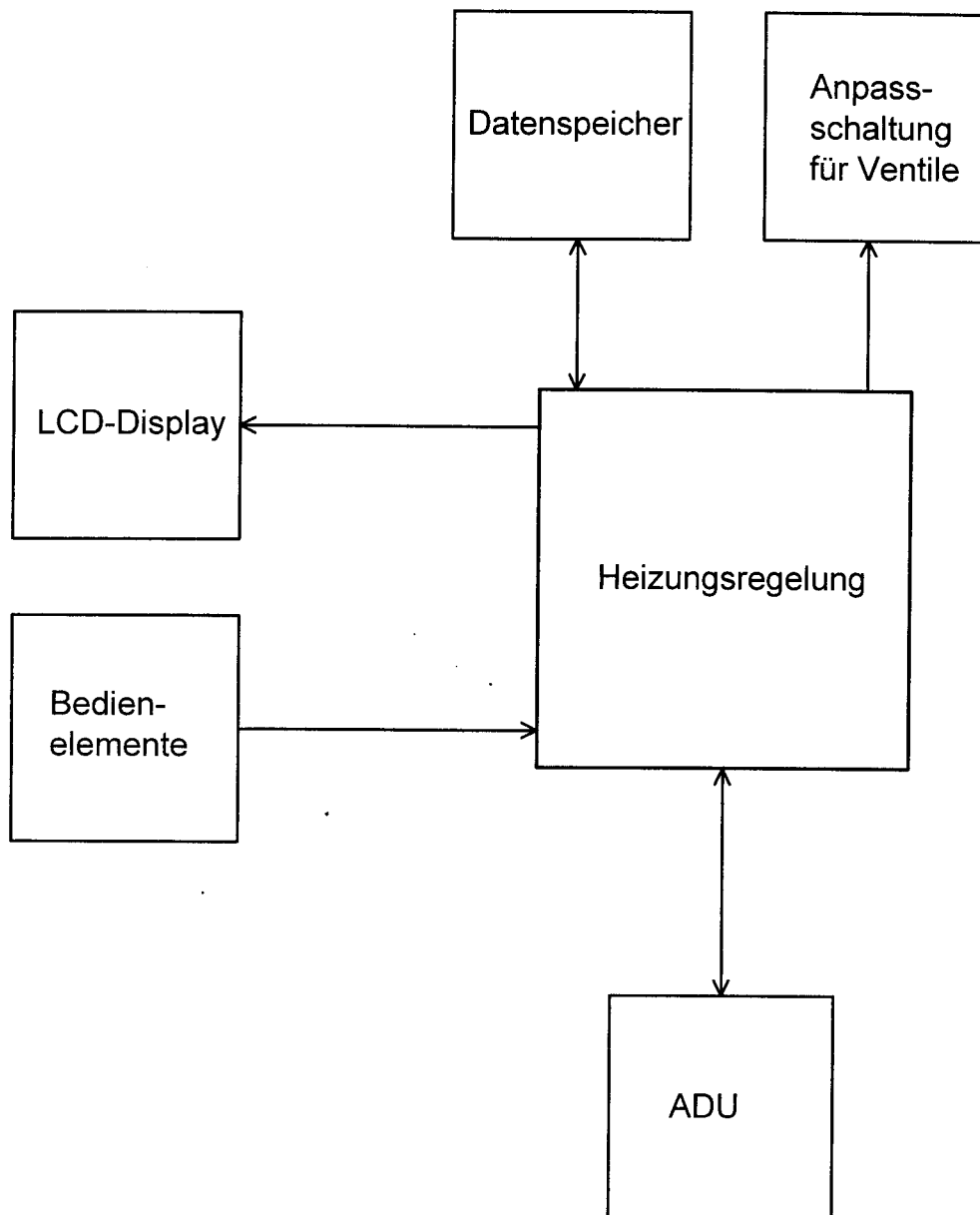


Bild 1: Gesamtaufbau Heizungsregelung

Die Heizungsregelung soll die Temperaturen von 14 Räumen regeln. Dabei wird der Raum, der den höchsten Wärmebedarf hat, durch Regelung der Vorlauftemperatur des Heizkessels, kontinuierlich geregelt, während die übrigen 13 Räume durch Ein- und Ausschalten geregelt werden (Zweipunktregelung). Bild 2 zeigt eine vereinfachte Übersicht der Heizungsregelung.

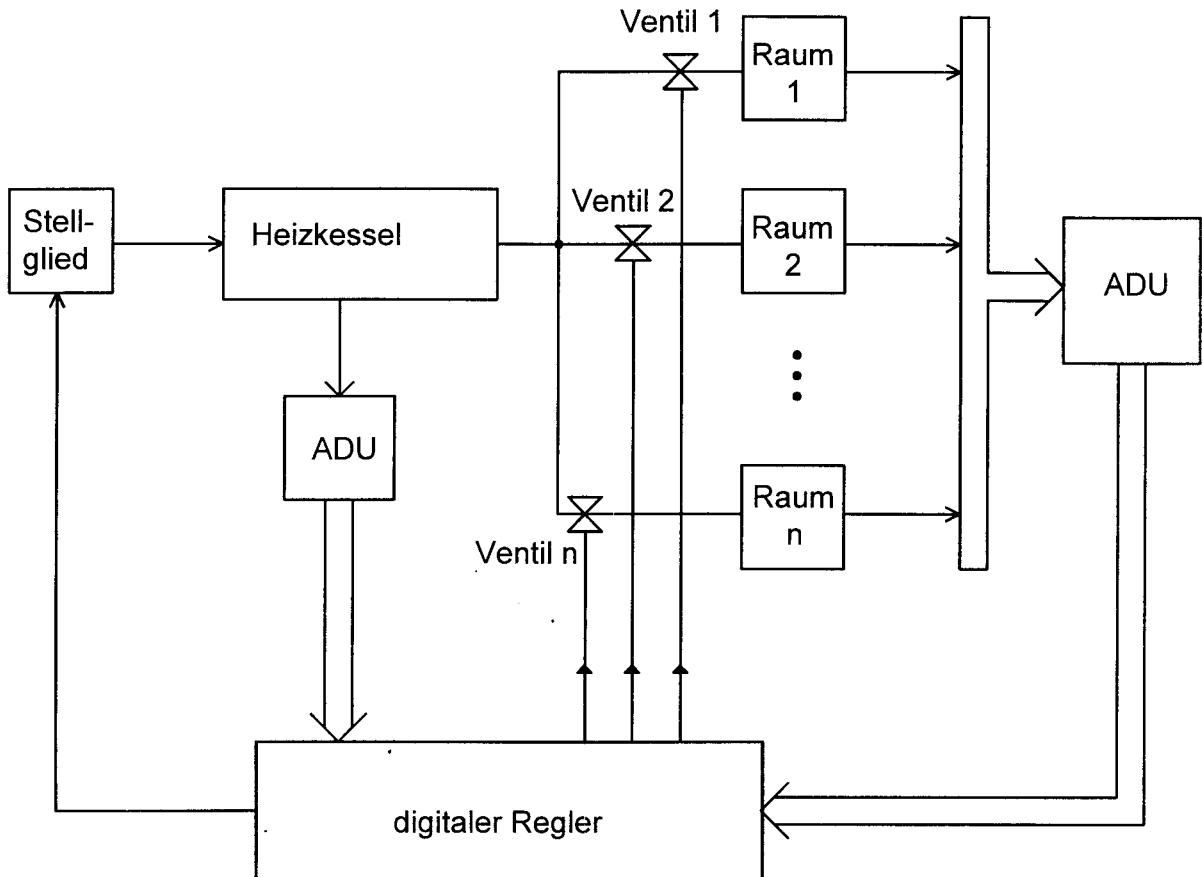


Bild 2: Übersicht über die Heizungsregelung

Die Arbeitsweise der Regelung läßt sich unter folgenden Punkten zusammenfassen:

- Jede Raumtemperatur wird durch Heizkörper beeinflusst. Die Heizkörper besitzen ein Ventil, das durch eine Steuerleitung vom digitalen Regler geschlossen oder geöffnet werden kann. Dabei bedeutet ein geöffnetes Ventil, daß ein Wärmestrom Φ vom Heizkessel ungehindert in den Heizkörper des Raumes fließen kann. Ein geschlossenes Ventil bedeutet, daß der Wärmestrom Φ nicht in den Heizkörper des Raumes fließen kann.

- Der Raum mit dem höchsten Wärmebedarf unterliegt einer kontinuierlichen Regelung. Das Ventil dieses Raumes muß natürlich immer geöffnet sein.
- Die Vorlauftemperatur des Heizkessels wird kontinuierlich geregelt. Die Regelung wird nach dem Zimmer mit dem höchsten Wärmebedarf ausgerichtet. Der Istwert der Vorlauftemperatur wird über einen Temperatursensor und AD-Umsetzer als 8-Bit Wert an den Regler übergeben. Der Regler berechnet daraus die Stellgröße zur Ansteuerung des Heizkessels
- Die einzelnen Raumtemperaturen T_1 bis T_n werden über Temperatursensoren erfaßt und digitalisiert an den Regler weitergegeben.

VHDL

In Bild 3 ist der prinzipielle Ablauf der Schaltungsentwicklung in VHDL dargestellt:

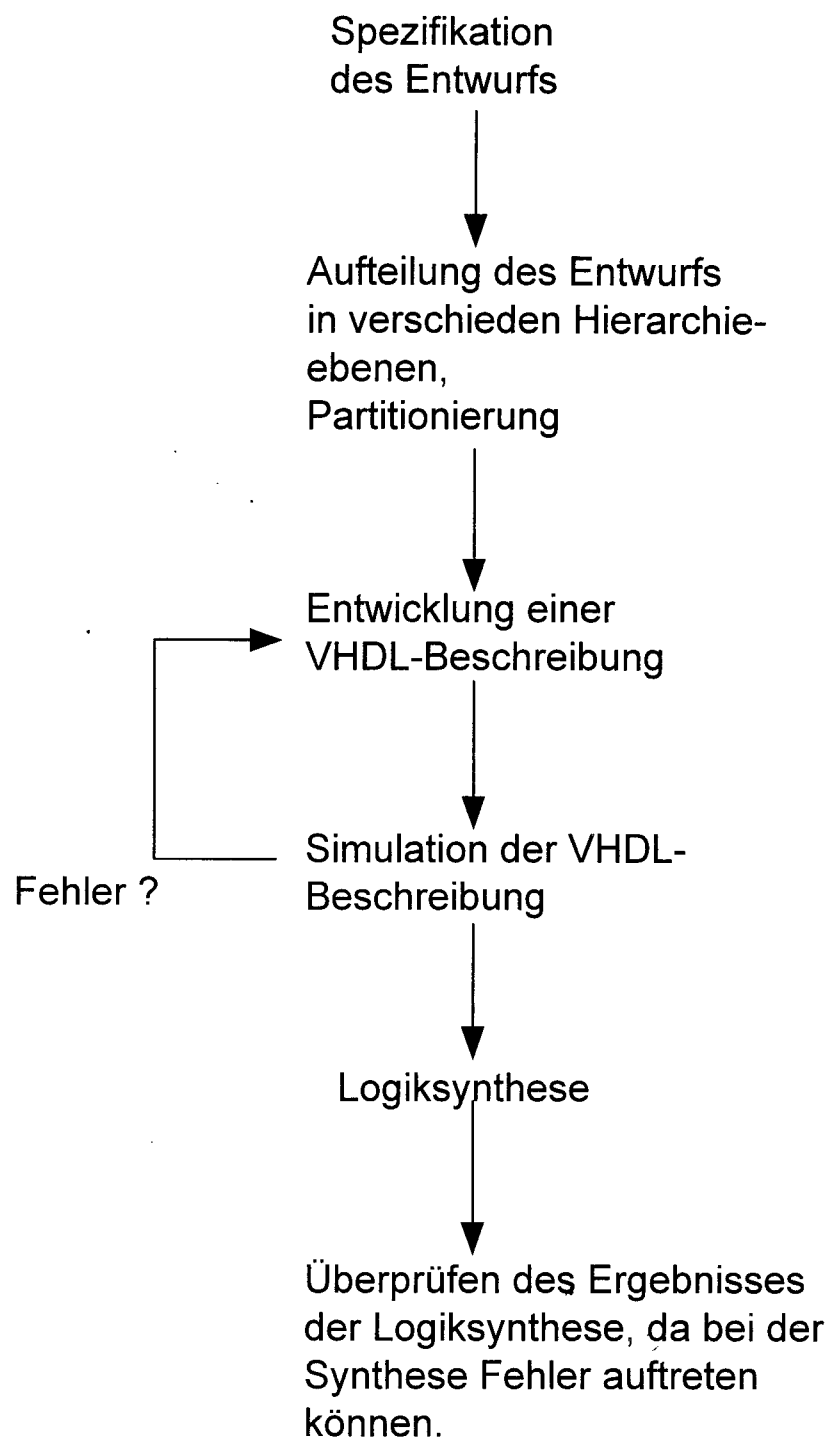


Bild 3: Schaltungsentwicklung mit VHDL

Die folgende Aufstellung zeigt Gründe, die für die Entwicklung elektrischer Schaltkreise mit VHDL sprechen:

- Die gesamte Schaltungsentwicklung ist mit VHDL möglich.
- VHDL ist unabhängig von der verwendeten Halbleitertechnologie.
- Eine Schaltungsbeschreibung kann einfach an neue Anforderungen angepaßt werden.
- Unterstützung durch verschiedene Softwarehersteller.
- Der VHDL-Sprachumfang ist durch eine Norm spezifiziert d.h VHDL kann zwischen den Entwicklungsplattformen verschiedener Softwarehersteller portiert werden.
- Die Dokumentation wird durch VHDL erleichtert.
- Durch die verschiedenen Beschreibungsformen, die VHDL ermöglicht (Verhaltensbeschreibung, Strukturbeschreibung), wird die Strukturierung einer Schaltung unterstützt.
- Die Umsetzung einer VHDL-Beschreibung in eine Netzliste ist möglich (Logiksynthese).
- Schnelle Schaltungsentwicklung durch Umsetzung einer Verhaltensbeschreibung in Netzliste.

Beim Einsatz von VHDL können allerdings auch Probleme auftreten. Um einige dieser Probleme zu zeigen, soll zunächst einmal der Aufbau des Reglers betrachtet werden.

Aufbau des Reglers

Der digitale Regler wird als Kaskadenregler realisiert, bestehend aus Führungsregler für die Regelung der Räume und Folgeregler für die Regelung der Vorlaufemperatur des Heizkessels.

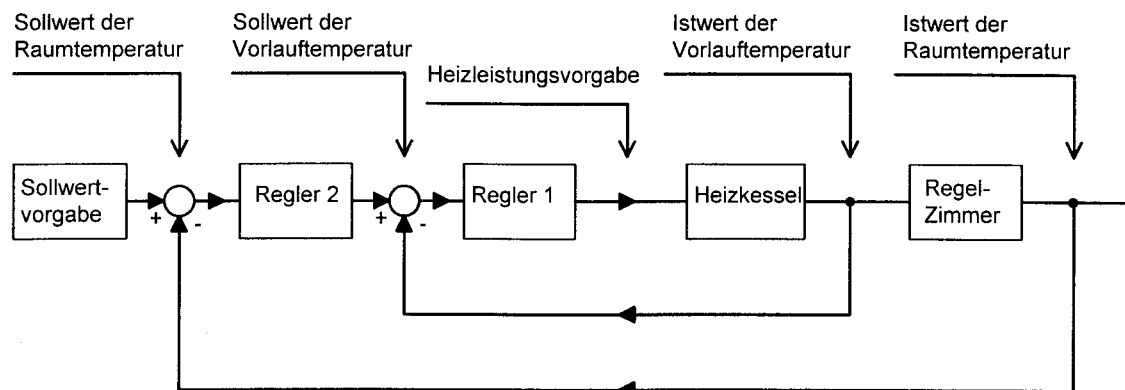


Bild 4: Prinzipieller Aufbau des Regelkreises

Da die Regler auf einem ASIC realisiert werden sollen, müssen sie so entworfen werden, daß eine Logiksynthese durchgeführt werden kann. Dabei soll zum einen darauf geachtet werden, daß die Regler ein möglichst gutes Regelverhalten aufweisen, zum anderen soll die Komplexität der Regler nicht zu groß werden. Wie es sich gezeigt hat genügt es den obigen Bedingungen, wenn für die Realisierung zwei PI-Regler verwendet werden

Die idealisierte Gleichung eines PI-Reglers lautet:

$$y(t) = k_p * \left[x_d(t) + \frac{1}{T_n} * \int_0^t x_d(\tau) d\tau \right] \quad (1)$$

wobei die einzelnen Parameter wie folgt bezeichnet werden:

$y(t)$ = Stellgröße zur Zeit t

$x(t)$ = Regelgröße zur Zeit t

$x_d(t) = w(t) - x(t)$ = Regeldifferenz zur Zeit t

k_p = Proportionalbeiwert

T_n = Nachstellzeit

Für kleine Abtastzeiten T_A kann das Integral wie folgt umgewandelt werden.

$$\frac{1}{T_n} * \int_0^t x_d(\tau) d\tau \rightarrow \frac{1}{T_n} * \sum_{i=0}^{n-1} x_d(i) * T_A = \frac{T_A}{T_n} * \sum_{i=0}^{n-1} x_d(i) \quad (2)$$

Damit ergibt sich für die zeitdiskrete Gleichung des PI-Reglers

$$y(n) = k_p * \left[x_d(n) + \frac{T_A}{T_n} * \sum_{i=0}^{n-1} x_d(i) \right] \quad (3)$$

Mit diesem Algorithmus läßt sich nach jeder Abtastung der Wert der Stellgröße y berechnen. Für die Programmierung von Regelalgorithmen ist oft ein rekursiver Algorithmus zweckmäßiger. Bei diesem Algorithmus wird die momentane Stellgröße $y(n)$ aus der vorherigen Stellgröße $y(n-1)$ berechnet. Dazu subtrahiert man Gleichung (4) von Gleichung (3):

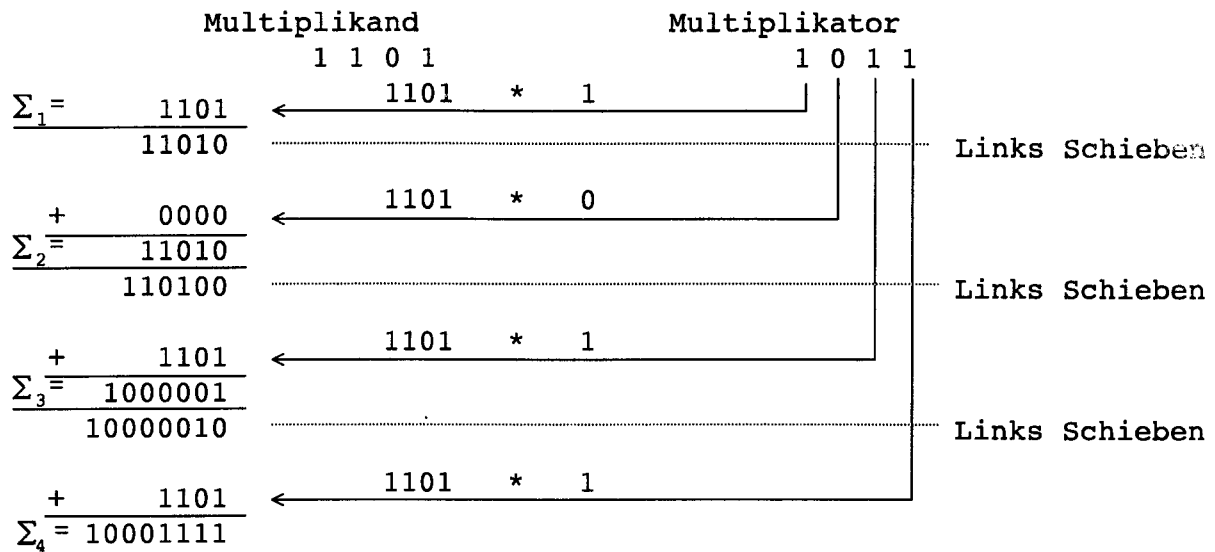
$$y(n-1) = k_p * \left[x_d(n-1) + \frac{T_A}{T_n} * \sum_{i=0}^{n-2} x_d(i) \right] \quad (4)$$

man erhält somit als rekursiven PI-Algorithmus für die Stellgröße:

$$y(n) = y(n-1) + k_p * x_d(n) + k_p * \left(1 - \frac{T_A}{T_n} \right) * x_d(n-1) \quad (5)$$

Für die Realisierung dieses Algorithmus sind auch Divisionen und Multiplikationen nötig. Da eine Division von der Logiksynthese nicht unterstützt wird, muß die Division in VHDL selbst definiert werden. Zusätzlich wird ein Multiplikationsalgorithmus ebenfalls in VHDL beschrieben, da der selbstentworfene Algorithmus weniger Platz auf dem ASIC benötigt.

Bild 5 zeigt den Ablauf einer Multiplikation in VHDL:

**Bild 5: Multiplikation**

Wie man sieht, kann die Multiplikation zweier Binär-Zahlen auf die Basisoperationen Verschieben und Addieren zurückgeführt werden. Zu beachten ist, daß das Produkt einer m Bit-Zahl mit einer n Bit-Zahl eine (m+n) Bit-Zahl ergibt. Wie man sieht sind insgesamt (n-1) Schiebeoperationen und (n-1) Additionen nötig.

Folgende Tabellen stellen den Aufwand dar, den eine Multiplikation nach obigen Algorithmus benötigt, im Vergleich zu einer Multiplikation, die automatisch von einem Logiksyntheseprogramm erzeugt wurde. In der ersten Tabelle sind die Ergebnisse der Logiksynthese des Multiplikationsalgorithmus aufgeführt, in der zweiten Tabelle das Ergebnis der automatisch erzeugten Multiplikation. Von besonderem Interesse ist hierbei die letzte Spalte, in der unmittelbar der Flächenbedarf auf einem Chip dargestellt ist. Wie man leicht erkennen kann ist der Flächenbedarf der Multiplikation in der ersten Tabelle bedeutend geringer.

Netlist Statistics for mult1_vhdl.version_1

Cell Name	Instance Count	Cost/	Cell Subtotal
sample_lib:and2p	12	2.30	27.60
sample_lib:dfc	3	9.50	28.50
sample_lib:dffp	32	8.40	268.80
sample_lib:eoi22	1	3.50	3.50
sample_lib:fa	11	7.50	82.50
sample_lib:ha	5	4.40	22.00
sample_lib:invr	1	1.20	1.20
sample_lib:nand2	2	1.30	2.60
sample_lib:nand3	2	2.40	4.80
sample_lib:nand4	3	2.50	7.50
sample_lib:nor3p	1	3.40	3.40
sample_lib:oai21	3	2.40	7.20
sample_lib:oai211	1	2.50	2.50
sample_lib:xnor2	1	3.30	3.30
sample_lib:xor2p	1	4.30	4.30
Total:	80		469.70

Netlist Statistics for mult2_vhdl.version_1

Cell Name	Instance Count	Cost/	Cell Subtotal
sample_lib:and2p	6	2.30	13.80
sample_lib:and3p	1	3.40	3.40
sample_lib:and4	1	3.50	3.50
sample_lib:aoi21p	1	3.40	3.40
sample_lib:aoi222mp	1	5.40	5.40
sample_lib:eoi22	1	3.50	3.50
sample_lib:fa	17	7.50	127.50
sample_lib:ha	30	4.40	132.00
sample_lib:invr	37	1.20	44.40
sample_lib:mux21	29	3.40	98.60
sample_lib:nand2	29	1.30	37.70
sample_lib:nand3	4	2.40	9.60
sample_lib:nand4	2	2.50	5.00
sample_lib:nor2p	3	2.30	6.90
sample_lib:oai21	4	2.40	9.60
sample_lib:oai22	1	2.50	2.50
sample_lib:oai211	13	2.50	32.50
sample_lib:or2	4	2.30	9.20
sample_lib:xnor2	31	3.30	102.30
sample_lib:xor2p	4	4.30	17.20
sample_lib:xor3p	2	7.40	14.80
Total:	221		682.80

Ähnlich wie die Multiplikation kann auch die Division auf Basisoperationen zurückgeführt werden. Bild 6 zeigt einen häufig verwendeten Divisionsalgorithmus:

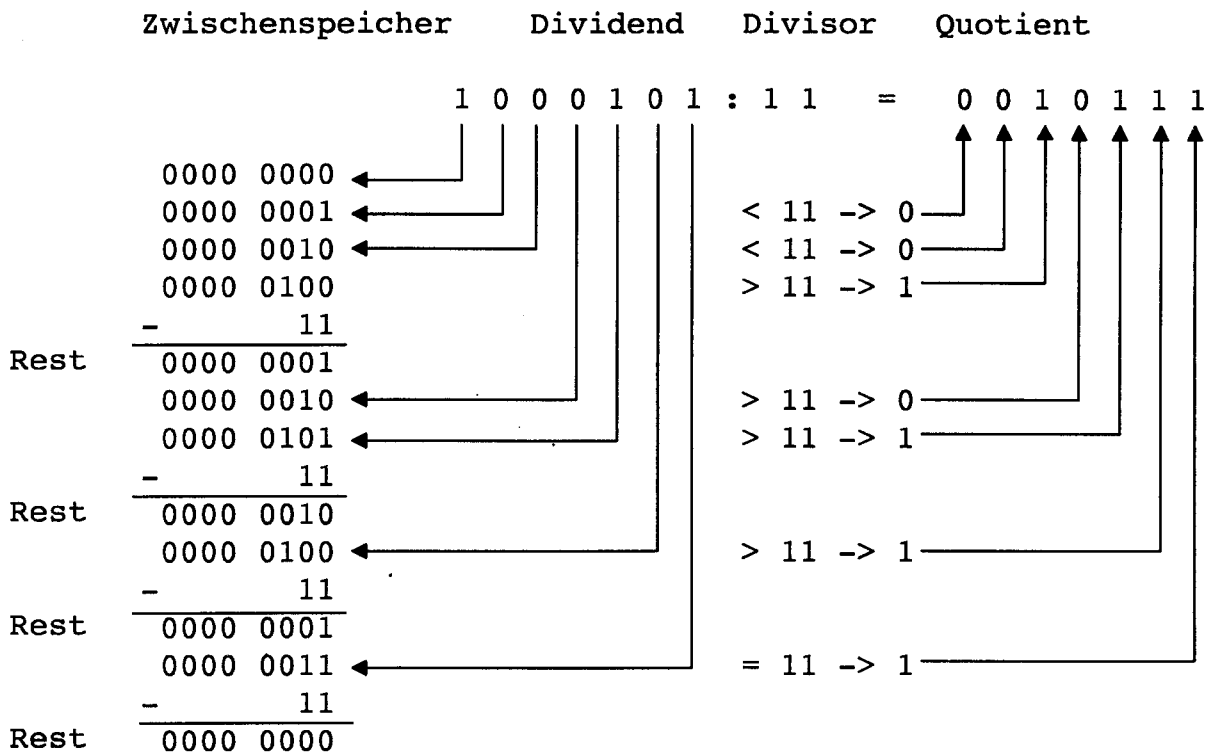


Bild 6: Division

Aus der Abbildung wird deutlich, daß der Algorithmus nur aus Schiebeoperationen, Subtraktionen und Vergleichsoperationen besteht.

Weitere Probleme, die beim Einsatz von VHDL in Verbindung mit einer Logiksynthese auftreten können:

- Ergebnisse der Logiksynthese müssen auf Fehler untersucht werden, da Fehler bei der Synthese möglich sind.
- Schon beim Entwurf sollte eine genaue Vorstellung über die Hardwarerealisierung vorhanden sein, diese Vorstellungen sollten in den Entwurf mit einfließen. Damit ist ein optimaler Einfluß auf die zu entwerfende Schaltung möglich. Dies wiederum bedeutet gute Möglichkeit zur Verifikation und Änderbarkeit.
- Nur bedingt Unterstützung von Testbarkeit (Mentor Graphics) d.h Scan-Path-Flip-Flops müssen eventuell von Hand eingefügt werden. Testbarkeit sollte von Anfang an berücksichtigt werden.
- Herstellerspezifische Erweiterungen in VHDL erschweren Portierbarkeit von VHDL auf andere Softwareplattformen
- Es kann nicht der gesamte VHDL-Sprachumfang synthetisiert werden, d.h eine spätere Logiksynthese muß von Anfang an berücksichtigt werden.
- Durch Synthese können die Schaltungen komplexer werden im Vergleich zum Entwurf von Hand.

Mit Synopsys vom digitalen Schaltungsentwurf zum integrierten Schaltkreis

Neben dem weitverbreiteten System von Mentor Graphics für den Entwurf digitaler Schaltungen mit Hilfe von Hardwarebeschreibungssprachen wie VHDL (IEEE-1076) und Verilog erschien zu Beginn des Jahres 1993 eine weitere Entwicklungsumgebung der in dieser Branche noch jungen Firma Synopsys zum Entwurf von integrierten Schaltkreisen. Im Zuge einer Diplomarbeit sollte die Installation, Konfiguration und die Beschreibung dieses Softwarepakets erfolgen, u.a. sollte ein direkter Vergleich der Softwarepakete durchgeführt werden und die Ergebnisse mit Hilfe eines noch überschaubaren sequentiellen Schaltwerks untermauert werden. Es stellte sich jedoch schon bald heraus, daß ein objektiver Vergleich nicht möglich war, da die Bewertung der Vor- und Nachteile eines jeden Systems vom jeweiligen Anwender abhängig war. Trotzdem soll kurz auf einige bemerkenswerte Einzelheiten eingegangen werden:

- Mehrere verschiedene Ein- und Ausgabedatenformate von Synopsys erlauben den vielfältigen Einsatz von Bibliotheken und HDL-Quelltextener von Synopsys. Darüberhinaus verfügt Synopsys über ein Ausgabedatenformat, das dem Datenformat entspricht, welches Mentor Graphics verwendet.
- Für die Übersetzung eines VHDL-Quelltextes benötigt der HDL-Compiler von Synopsys erheblich weniger Zeit als der VHDL-Compiler von Mentor Graphics.
Bemerkung:
Mentor Graphics hat einen schnellen *VHDL Compiler* bereits angekündigt (Quick VHDL).
- Der *HDL-Compiler* bietet eine grafische Benutzeroberfläche (gvan), die es dem Anwender während der Übersetzung ermöglicht fehlerhafte Quelltexte sofort zu korrigieren und die Übersetzung nach der Korrektur fortzusetzen.
- Verglichen mit der Oberfläche von Mentor Graphics weist die Oberfläche von Synopsys eine erheblich einfachere Menüstruktur auf → geringere Einarbeitungszeit für Anfänger und Einsteiger.
- Mit Hilfe des Synopsys Graphical Environment (SGE) lassen sich anwenderspezifische Funktionsblöcke und Schaltungen entwickeln, deren Strukturen sich mit Hilfe des VHDL oder Verilog Netlisters in HDL-Quelltexte übersetzen lassen. Dabei wird eine vollständige Schaltung komplett übersetzt (es müssen keine Ergänzungen in der Entity und Architecture vorgenommen werden), bei einem anwenderspezifischen Funktionsblock muß lediglich die logische Funktion in der Architecture ergänzt werden.
- Die Oberfläche des VHDL System Simulators ist übersichtlicher aufgebaut als die Oberfläche von QuickSim. Eine mehrfarbige Darstellung von logischen Werten, die sowohl im Zustandszeitdiagramm, als auch im Schaltplan selbst, möglich ist, vereinfacht die Verifikation der Funktion der Schaltung.

- Die Logiksynthese mit Hilfe des *DesignCompilers* von Synopsys wird erheblich schneller durchgeführt als die Logiksynthese mit Hilfe des Programms AutoLogic von Mentor Graphics.
- Die Testbarkeit von den erzeugten Strukturen ist durch den Einsatz des *Test Compilers* gegeben, der verschiedene Testmethoden zur Verfügung stellt. Die Überprüfung von integrierten Schaltkreisen mit Hilfe verschiedener Testmethoden war bisher nicht möglich.

Bemerkung:

Mentor Graphics bietet zwischenzeitlich zwei eigenständige Softwarepakete für die Erstellung von Testvektoren und zur Durchführung von Test mit integrierten Schaltkreisen an. (FlexTest und Flex Scan)

- Testvektoren können ohne allzu große Vorkenntnisse erzeugt werden. Für die Wahl der Testverfahren sind nur wenige Parameter notwendig.
- Der Wechsel innerhalb der einzelnen Designhierarchien ist unproblematisch.

Überblick über die einzelnen Elemente der Entwicklungsumgebung von Synopsys

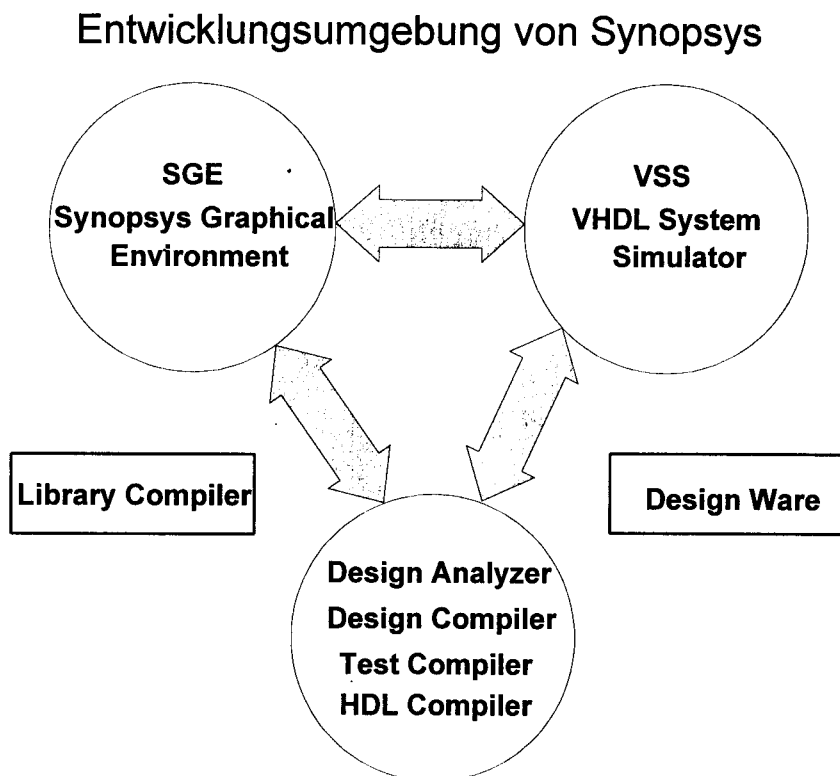


Bild 1. Entwicklungsumgebung von Synopsys

Die Entwicklungsumgebung von Synopsys besteht im wesentlichen aus drei Funktionsblöcken. Ein Bestandteil ist das Synopsys Graphical Environment (SGE= graphische Entwicklungsumgebung von Synopsys), der zweite Bestandteil ist der VHDL System Simulator (VSS) und der dritte

Bestandteil sind die für die Logiksynthese notwendigen Compiler sowie ein *Test Compiler*, der die Voraussetzungen für die Überprüfung der Schaltung mit Testvektoren schaffen soll. Weiterhin verfügt Synopsys über den *Library Compiler*, der die Schnittstelle zwischen ASIC-Bibliotheken oder vollkundenspezifischen Bibliotheken und Synopsys darstellt. Die Software unter der Bezeichnung Design Ware soll dem Anwender, neben vier Bibliotheken mit gebräuchlichen Logikelementen, eine strukturierte Grundlage für die Schaffung eines anwenderspezifischen Inventars von bereits erstellten Entwürfen zur Verfügung stellen. Ziel der Design Ware ist es, den Entwurf effizienter zu gestalten, indem auf bereits erstellte Komponenten zurückgegriffen wird.

SGE - Synopsys Graphical Environment

Bei der Erstellung von digitalen Schaltungen für integrierte Schaltkreise gibt es prinzipiell zwei unterschiedliche Vorgehensweisen. Herkömmlicherweise wurden logische Schaltungen "per Hand" entworfen, d.h. die Struktur des entstandenen Entwurfs wurde zunächst mit Hilfe anderer Methoden entwickelt. Die Umsetzung in einen integrierten Schaltkreis geschah durch das Übertragen der Strukturen in einen Stromlaufplan, der dann von Logiksyntheseprogrammen weiterbearbeitet wurde. Diese Vorgehensweise kann z.B. auch bei Synopsys eingesetzt werden. Andererseits bietet Synopsys aber durch den *HDL Compiler* die Möglichkeit logische Strukturen mit Hilfe von Hardwarebeschreibungssprachen in Stromlaufpläne umzusetzen. Welche Vorgehensweise vorteilhafter ist, kommt auf den Einzelfall an. Allerdings ist der Schaltungsentwurf mit einer Hardwarebeschreibungssprache bei größeren Veränderungen der logischen Funktion erheblich flexibler als der Schaltungsentwurf der herkömmlichen Art.

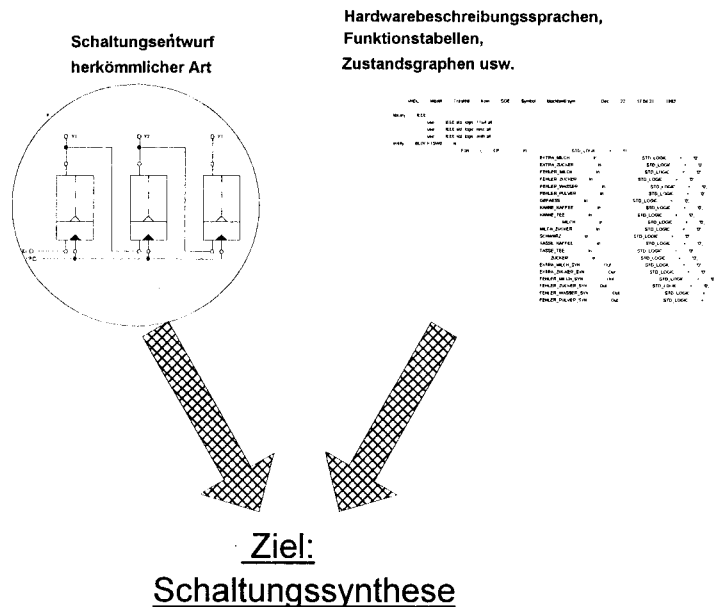


Bild . Vorgehensweise beim Schaltungsentwurf

Wie das System von Mentor Graphics mit dem Programm DesignArchitect und dem DesignManager, bietet Synopsys mit dem Programmpaket SGE (Synopsys Graphical Environment) die Möglichkeit Funktionsblöcke zu erzeugen und diese miteinander zu verknüpfen. Es stehen darüberhinaus noch einige Bibliotheken mit entsprechenden Standardfunktionsgliedern zur

Verfügung (z.B. IEEElib, MVLlib, verilib). Die Erstellung eines vom Anwender definierten Funktionsblocks in der Entwicklungsumgebung von Synopsys unterscheidet sich nur unwesentlich von der Erstellung im System von Mentor Graphics. Das Synopsys Graphical Environment enthält grundsätzlich drei große Programmblöcke, den *Navigator*, die *Symbol- u. Stromlaufplaneditoren* und ein sogenanntes *VHDL-interface*.

Mit Hilfe des *Navigator*s ist es möglich, die Simulation im Stromlaufplan und gleichzeitig im Zustandszeitdiagramm zu verfolgen. Darüberhinaus können zusätzliche Variablen und Signale, lediglich durch eine entsprechende Markierung dem Zustandszeitdiagramm hinzugefügt werden. Weiterhin besteht die Möglichkeit die Simulation schrittweise im Quelltext nachvollziehen zu können, da der Navigator mit dem VHDL Debugger verknüpft ist.

Hauptbestandteile des SGE sind allerdings die beiden Editoren, einerseits der Symbol-Editor zum Erzeugen von anwenderspezifischen Funktionsblöcken, andererseits der Schematic-Editor zum Erstellen von Stromlaufplänen.

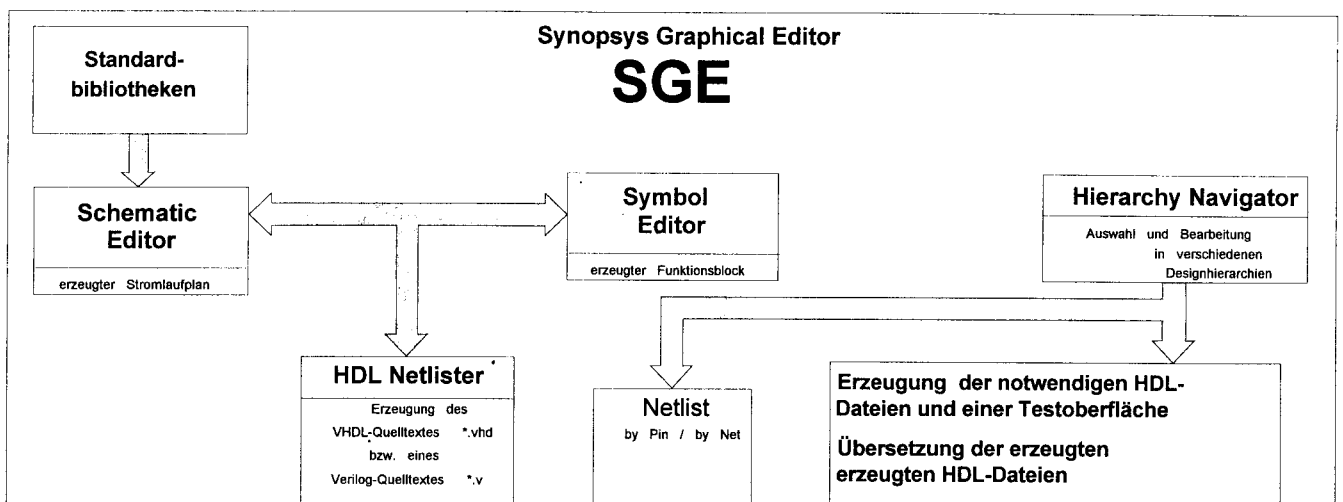


Bild 2. Struktur des Synopsys Graphical Environment (SGE)

Der Symbol-Editor

Anwenderspezifische Funktionsblöcke, die spezielle logische Schaltwerke enthalten können, können sowohl im Symbol Editor erzeugt werden als auch im Schematic Editor. Die Erstellung im Symbol Editor ist allerdings etwas umständlicher zu handhaben wie im Schematic Editor.

Nachdem ein Symbol für einen anwenderspezifischen Funktionsblock erstellt wurde (auf die einzelnen Schritte soll in diesem Rahmen verzichtet werden), besteht die Möglichkeit einen VHDL- oder Verilog-Quelltext mit Hilfe des HDL-Netlisters erstellen zu lassen. Bei der Erzeugung des Quelltextes werden die Eigenschaften, die dem Funktionsblock im Symbol Editor zugewiesen worden sind, z.B. Anstiegs- u. Abfallzeiten, im VHDL-Quelltext berücksichtigt. Die Entity des erzeugten VHDL-Quelltextes wird dabei vollständig erstellt, die Architecture muß allerdings durch die logische Funktion des Schaltwerks ergänzt werden. Nachdem dies geschehen ist, muß der nun vollständige Quelltext mit dem *HDL Compiler* übersetzt werden.

Schematic Editor

Mit dem Schematic Editor können nicht nur Funktionsblöcke erstellt werden, die Hauptaufgabe des Schematic Editor ist die Erzeugung von Stromlaufplänen, die u.a. anwenderspezifische Funktionsblöcke enthalten können. Prinzipiell ist der Aufbau des Schematic Editor dem Aufbau des Symbol Editors sehr ähnlich, so daß sehr leicht ein kompletter Stromlaufplan erzeugt werden kann. Auch der Schematic Editor ermöglicht die Erzeugung eines VHDL- bzw. Verilog-Quelltextes mit Hilfe des HDL-Netlisters. Ergänzungen oder Änderungen im HDL-Quelltext sind in diesem Fall nicht mehr notwendig, da die eingesetzten Teilschaltwerke die logische Funktion des Schaltwerks repräsentieren und lediglich die einzelnen Komponenten miteinander verknüpft werden.

Bemerkung:

Die Erstellung der Architecture wird bei dem System von Mentor Graphics nicht unterstützt, was bei kleinen Schaltwerken unproblematisch ist. Allerdings verringert das System von Synopsys den Arbeitsaufwand und schließt Verknüpfungsfehler bei der Erstellung der Strukturen von Schaltwerken aus.

Wie bei der Erzeugung eines HDL-Quelltextes mit dem Symbol Editor müssen vom Schematic Editor generierte HDL-Quelltexte mit Hilfe des *HDL Compilers* übersetzt werden.

Hierarchy Navigator

Der Hierarchy Navigator stellt das Bindeglied zwischen der Stromlaufplanebene und der Simulationsebene dar. Der Hierarchy Navigator erzeugt bei der Übersetzung der HDL-Quelltexte automatisch eine Testoberfläche. Diese Testoberfläche könnte z.B. dazu verwendet werden die Taktversorgung des später zu simulierenden Entwurfs zu übernehmen. Auch ist es denkbar, mit Hilfe der Testoberfläche eine Außenwelt für den Entwurf herzustellen, die die spätere Einsatzumgebung modelliert und nachempfunden.

Der VHDL System Simulator

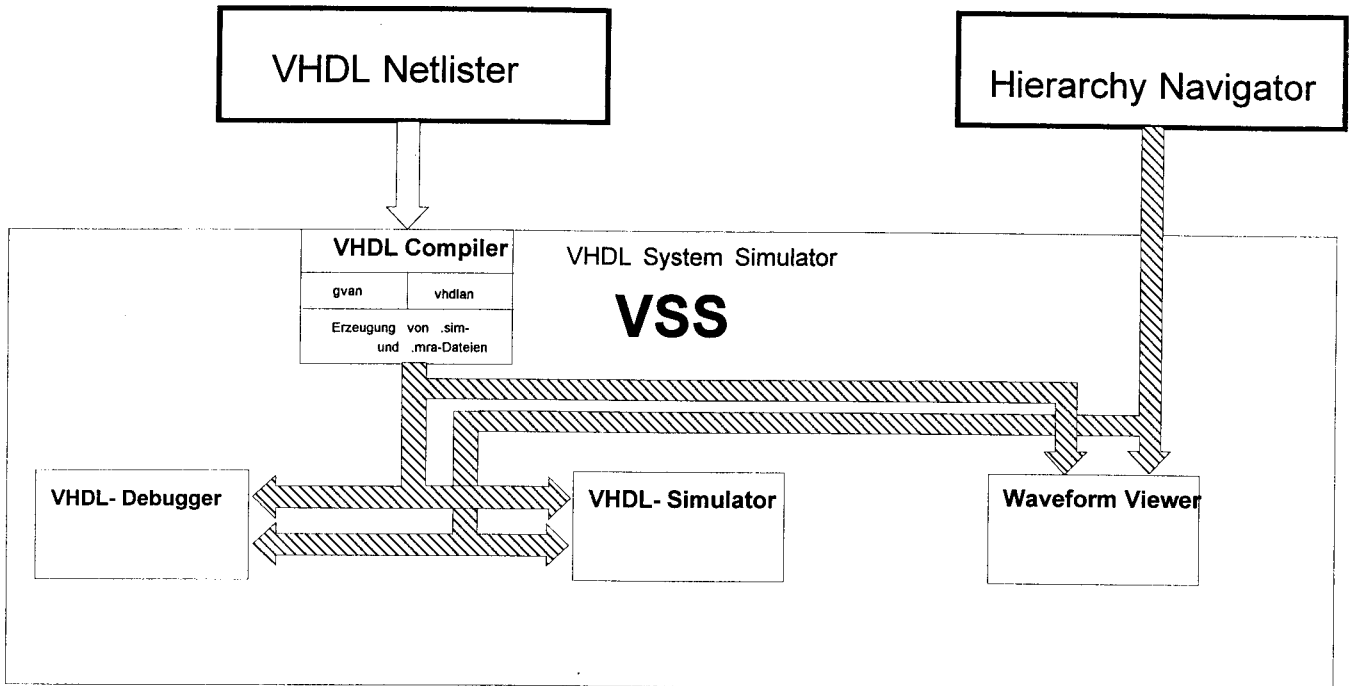


Bild 3 Struktur des VHDL System Simulator (VSS)

Der VHDL System Simulator besteht grundsätzlich aus dem VHDL Debugger, dem VHDL Simulator und dem Waveform Viewer. Gemeinsam bieten diese drei Komponenten eine mit QuickSim von Mentor Graphics vergleichbare Simulationsumgebung. Grundlage dieser Umgebung ist der VHDL Debugger, der in Zusammenarbeit mit dem VHDL Simulator die Simulation des Entwurfs durchführt. Der Waveform Viewer ermöglicht die graphische Darstellung der Simulationsergebnisse in Form von Zustandszeitdiagrammen. Wie bei der Simulation mit QuickSim von Mentor Graphics besteht die Möglichkeit durch das Einbinden von Simulationsdateien, die Simulation zu automatisieren. Eine Besonderheit von Synopsys ist die vom Hierarchy Navigator erzeugte Testoberfläche, auch Testbench genannt. Diese Testoberfläche ermöglicht es dem Anwender, eine Außenwelt des zu simulierenden Entwurfs zu modellieren und nachzuempfinden. Diese Testoberfläche kann aber auch nur zur Erzeugung von externen Taktvariablen eingesetzt werden. Ein Editor ermöglicht darüberhinaus die Veränderung des VHDL-Quelltextes während der Simulation und verhindert somit erneute Neuübersetzungen und Initialisierungen des Simulators. Die Fehlersuche im VHDL Simulator gestaltet sich einerseits durch den problemlosen Wechsel in höhere oder niedrigere Designebenen, andererseits durch die parallele Darstellung des momentanen Zustands des Schaltwerks im Zustandszeitdiagramm und im Stromlaufplan als sehr einfach. Die Übersichtlichkeit bei der Darstellung von Werten einzelner Signale oder Variablen im Stromlaufplan ist durch die verschiedenfarbige Kennzeichnung der logischen Werte "1", "0" und "X" erheblich besser realisiert als bei QuickSim von Mentor Graphics. Der *VHDL Debugger* ermöglicht die schrittweise Simulation des Entwurfs und eignet sich deshalb besonders für die Fehlersuche während der Simulation. Da der entsprechende Abschnitt des gerade zu bearbeitenden Quelltextes angezeigt wird, können Fehler in der Regel schnell lokalisiert werden.

Bemerkung:

Ein weiteres nützliches Werkzeug des VHDL Simulators ist die sogenannte Simulationsabdeckung (simulation coverage). Inwieweit das in der Dokumentation von Synopsys nur wenig beschriebene Verfahren nun wirklich die gesamte Simulation eines Schaltwerks abdeckt, konnte anhand eines praktischen Beispiels noch nicht nachgewiesen werden.

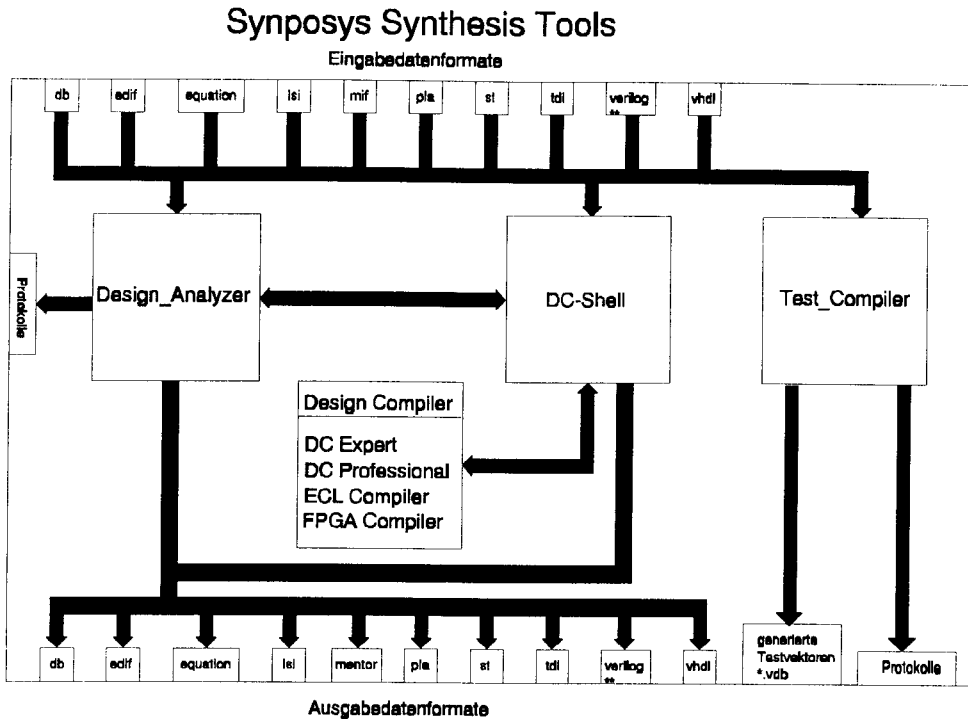
Der Waveform Viewer

Wie bereits erwähnt, verfügt der *VHDL System Simulator* (VSS) über den sogenannten *Waveform Viewer*, mit dessen Hilfe Zustandszeitdiagramme erzeugt werden können. Dieses Programm wird beim Aufruf des *VHDL Simulators* selbständig geladen. Wird der *VHDL Debugger* eingesetzt, so wird dieses Programm aktiviert, wenn die anzuzeigenden Signale und Variablen markiert werden.

Logiksynthese

Nachdem die logische Funktion der einzelnen Teilschaltwerke und des Gesamtschaltwerks mit dem *VHDL Simulator* oder dem *VHDL Debugger* überprüft wurde und die Entwürfe ordnungsgemäß arbeiten, kann mit der Logiksynthese begonnen werden. Wie das System von Mentor Graphics mit den Programmen *Design Architect* bzw. *Design Manager* und *Auto Logic* kann in der Entwicklungsumgebung von Synopsys ebenfalls eine Logiksynthese durchgeführt werden.

Zur Logiksynthese stehen die sogenannten *Synopsys Synthesis Tools* zur Verfügung. Hinter den Synopsys Synthesis Tools stehen im Grunde genommen zwei unabhängig voneinander arbeitende Strukturen, die mit Hilfe der folgenden Grafik erläutert werden sollen. Zum einen ist dies der *Design Compiler*, bestehend aus dem *Design Analyzer*, dem *DC SHELL* und diversen Compilern für die Durchführung der Logiksynthese, zum anderen der *Test Compiler*, der alle notwendigen Funktionen zur Überprüfung der erzeugten Strukturen mit Hilfe von Testvektoren zur Verfügung stellt.



** zur Zeit keine Lizenz verfügbar

Bild 4. Struktur des *Design Compilers*

Ein- und Ausgabedatenformate

Aufgrund der Vielzahl der zugelassenen Eingabedatenformaten besteht die Möglichkeit eine Logiksynthese mit den unterschiedlichsten Voraussetzungen durchzuführen. Folgende Eingabedatenformate sind erlaubt:

Datenformat	Bemerkungen
db	(nicht editierbares) synopsysspezifisches Datenformat <u>d</u> ata <u>b</u> ase
edif	EDIF= <u>E</u> lectronic <u>D</u> esign <u>I</u> nterchange <u>F</u> ormat
equation	synopsysspezifisches Gleichungsformat
lsi	Netzliste mit dem Format der LSI Logic Corporation
mif	mentorspezifische Netzliste
pla	Berkeley PLA Datenformat
State Table	editierbares Datenformat eines vollständig definierten endlichen Automaten (FSM= <u>f</u> inite <u>s</u> tate <u>m</u> achine)
tdl	Netzliste mit dem Format der Texas Design Language
verilog **	editierbares Datenformat eines Verilog-Quelltextes
vhdl	editierbares Datenformat eines VHDL-Quelltextes

Tabelle 1. Erlaubte Eingabedatenformaten des *Design Compilers*

Folgende Ausgabedatenformate sind möglich:

Datenformat	Bemerkungen
db	(nicht editierbares) synopsyspezifisches Datenformat <u>d</u> ata <u>b</u> ase
edif	EDIF= <u>E</u> lectronic <u>D</u> esign <u>I</u> nterchange <u>F</u> ormat
equation	synopsyspezifisches Gleichungsformat
lsi	Netzliste mit dem Format der LSI Logic Corporation
mentor	mentorspezifische Netzliste
pla	Berkeley PLA datenformat
State Table	editierbares Datenformat eines vollständig definierten endlichen Automaten (FSM= <u>f</u> inite <u>s</u> tate <u>m</u> achine)
tdl	Netzliste mit dem Format der Texas Design Language
verilog **	editierbares Datenformat eines Verilog-Quelltextes
vhdl	editierbares Datenformat eines VHDL-Quelltextes

Tabelle 2. Erlaubte Ausgabedatenformaten des *DesignCompilers*

Das DC-Shell und der DesignAnalyzer

Das *DC-Shell* (Design Compiler-Shell) stellt das Zwischenstück zwischen dem *Design Analyzer* und dem *Design Compiler* dar. Es besteht die Möglichkeit, eine Logiksynthese aus der Ebene des *DC-Shells* durchzuführen. Der Unterschied zwischen der Logiksynthese im *Design Analyzer* und dem *DC-Shell* besteht darin, daß im *DC-Shell* Anweisungen direkt über eine Kommandozeile eingegeben werden können, ohne eine umfangreiche Fenstertechnik, wie sie der *Design Analyzer* verwendet, in Kauf nehmen zu müssen. Der Nachteil dieses Verfahren ist allerdings, daß auf die einfache Darstellung von Entwurfshierarchien, Blockschaltbildern und Stromlaufplänen verzichtet werden muß.

Der Design Compiler

Für die Logiksynthese steht das Programm *Design Compiler* zur Verfügung. Dieses Programm stellt den Kern der Logiksynthese dar und greift auf verschiedene Compiler zurück, die im folgenden kurz beschrieben werden sollen. Der *Design Compiler* besteht prinzipiell aus vier selbständigen Compilern:

- *DC Expert*
siehe *FPGA Compiler*
- *DC Professional*
siehe *FPGA Compiler*

- *ECL Compiler*

Es besteht u.a. die Möglichkeit durch den Einsatz des *ECL Compilers* bipolare ECL-Entwürfe und Entwürfe in der GaAs-Technik durchzuführen. Der *ECL Compiler* ist dafür ausgelegt, hinsichtlich der Entwurfsbeschränkungen und -vorgaben ein Optimum aus Schaltgeschwindigkeit, minimaler Verlustleistung und benötigter Chipfläche zu erzeugen. Er gestattet eine Vielzahl von Eingabedatenformaten u.a. Datenformate der Hardwarebeschreibungssprachen VHDL und Verilog und kann Daten und Protokolle in diversen Ausgabedatenformaten erzeugen.

Bemerkung:

Im Vergleich zur CMOS-Technik stellt die ECL-Technik einen erheblich höheren Entwurfsaufwand dar, da erheblich mehr Entwurfsvorschriften beachtet werden müssen, wie z.B. zulässige Betriebsspannungsbereiche, Beschränkungen wegen des relativ hohen Leistungsverbrauchs und des geringen statischen Störabstandes usw.. Demgegenüber steht aber die Tatsache, daß ECL-Schaltkreise, verglichen mit allen anderen Schaltkreistechniken, erheblich schneller sind.

Bemerkung:

Durch den erheblich höheren Entwicklungsaufwand bei ECL-Schaltkreisen ist die Zeit, verglichen mit anderen Schaltkreistechniken, vom Entwicklungsaufwand zur Serienreife (time to market) in der Regel größer. Durch den Einsatz einer, der ECL-Technik angepaßten, Entwicklungsumgebung kann diese Zeit jedoch verringert werden.

- *FPGA Compiler*

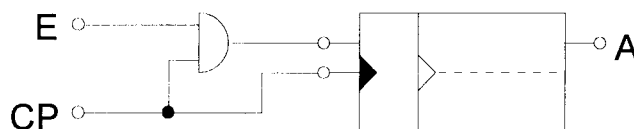
Der *FPGA Compiler* verfügt über umfangreiche Möglichkeiten zur Erzeugung und Optimierung von Schaltungen auf Basis der FPGA-Technologie (FPGA = Field Programmable Gate Array). Die Grundlage für die Umsetzung in solche Strukturen können einerseits Quelltexte in den Hardwarebeschreibungssprachen VHDL und Verilog sein, andererseits können Netzlisten, vollständige Funktionstabellen, PLA-Funktionstabellen oder bool'sche Gleichungen eingesetzt werden. Durch die Optimierung bereits erzeugter Strukturen kann die Komplexität der generierten Schaltungen auf ein Mindestmaß reduziert werden, die bei Bedarf in übliche Datenformate wie XNF- (Xilinx Netlist Format) oder EDIF- Datenformaten konvertiert werden können.

Der Entwurf von ASIC's (ASIC=application specific integrated circuit, anwender-spezifische Schaltkreise) wird durch den Einsatz der oben erwähnten Programme *DC Professional* oder *DC Expert* ermöglicht. Es sollte aber berücksichtigt werden, daß der Entwurf zunächst für einen FPGA durchgeführt werden muß. Anschließend kann mit der Umsetzung in einen ASIC begonnen werden. In Verbindung mit dem *VHDL System Simulator* kann der erzeugte Entwurf wiederum getestet werden, so daß ein kompletter, in sich geschlossener Ablauf, vom Systementwurf über die Logiksimulation und-synthese zum Schaltkreisprototypen, möglich ist.

Die Test Compiler-Entwicklungsumgebung

Dieser Block der Synopsys Synthesis Tools ermöglicht die Überprüfung der generierten Strukturen einer logischen Schaltung mit Hilfe von Testvektoren, die der *Test Compiler* selbst erzeugt. Dabei stehen dem Anwender mehrere Testverfahren zur Verfügung, wie z.B. Multiplexed Flip-Flop, LSSD oder Boundary Scan. Der *Test Compiler* unterstützt u.a. die automatische Erzeugung von Testvektoren (ATPG=automatic test pattern generation). Durch die Berücksichtigung von Testvorgaben u. -beschränkungen ermittelt der *Test Compiler* stets das optimale Testverfahren und verhindert somit zeitintensive, iterative Prozesszyklen während des Entwurfs und des Testvorgangs. Während des Tests wird gleichzeitig ein "built-in-rule-check" durchgeführt. Unter dem "built-in-rule-check" ist eine Überprüfung des Stromlaufplans auf eventuelle Verdrahtungs- und Verbindungsfehler zu verstehen. So bemängelt der *Test Compiler* z.B. folgenden Fehler mit der Fehlermeldung:

Warning: Data pin %s of cell %s is driven by a clock/enable signal (TEST-131)



Fehlerursache:

Die obige Schaltung verstößt gegen die Entwurfsvorschrift Taktvariablen nicht als Datenvariablen zu verwenden.

Solange es sich bei der Fehlermeldung um eine Warnung (warning) handelt, kann der Anwender entscheiden, ob er eine Änderung des Entwurfs vornehmen will oder den Entwurf beibehält. Tritt allerdings ein Fehler (error) auf, so muß eine Änderung des Entwurfs erfolgen. Da eine Fehlermeldung aber die Fehlerabdeckung verringert, sollte jede Fehlermeldung behoben werden.

Bemerkung:

Der Umfang des momentan installierten *Test Compiler*-Entwicklungsumgebung beschränkt sich auf den *Test Compiler*. Dadurch sind nur die Testverfahren *full scan test* und *boundary scan test* einsetzbar. Die Erweiterung mit den Programmen *Test Compiler Plus* bzw. *CTV Interface* würde zusätzlich einen *partial scan test* bzw. anwenderspezifische Testmethoden zulassen. Die Einflußmöglichkeiten des Anwenders auf den *Test Compiler* beschränkt sich daher auf die Wahl des Testverfahrens und der Testbeschränkungen.

Unter einem *full scan test* versteht man die Testmethode, bei der alle Speicher-Flip-Flops des zu testenden Schaltwerks durch testbare Speicher-Flip-Flops ersetzt werden. Dagegen werden beim *partial scan test* lediglich einzelne Speicher-Flip-Flops des zu testenden Schaltwerks durch testbare Speicher-Flip-Flops ersetzt. Die Auswahl der zu ersetzenden Speicher-Flip-Flops hängt von den Vorgaben und Beschränkungen ab, die der Anwender an das Testverfahren stellt.

Technologieunabhängige Logiksynthese

Wird die Logiksynthese zunächst ohne Technologieangabe durchgeführt, so wird die Bibliothek *gtech* (generic technology) verwendet. Diese Bibliothek enthält Standardverknüpfungsglieder und die gebräuchlichsten Speicher-Flip-Flops und soll dem Anwender eine technologieunabhängige Grundlage für den Entwurf bieten.

Design Ware Bibliotheken

Neben der *gtech*-Bibliothek gibt es noch die Bibliotheken *DW01*, *DW02* und *DW03*. Die Bibliotheken *DW01* und *DW02* enthalten bereits aufwendigere Elemente wie z.B. Addierer/Subtrahierer, Vergleicher, Aufwärtszähler (Inkrementierer)/ Abwärtszähler (Dekrementierer), Multiplizierer usw. und bieten dem Anwender ein breites Einsatzspektrum, insbesondere beim Entwurf von Schaltwerken, die arithmetische Strukturen enthalten.

Die Bibliothek *DW03* enthält mehrere spezielle asynchrone und synchrone "Silo"-Register (FIFO=first in first out, Prinzip d. Befehlswarteschlange), ein synchrones Stapel-Register (LIFO's=last in first out), ein sogenanntes Pipeline Register, asynchrone und synchrone Schreib-Lese-Speicher, Aufwärts- und Abwärtszähler und soll dem Anwender den Entwurf von z.B. I/O-, Bus-, Grafikschnittstellen erleichtern.

Technologieabhängige Logiksynthese

Wird allerdings gezielt eine Technologie angewendet, so sind die generierten Stromlaufpläne wesentlich übersichtlicher.

Arbeiten mit dem TextViewer des DesignAnalyzers

Soll die Umsetzung eines Quelltextes im VHDL- oder Verilog- Format in eine logische Schaltung erfolgen, so bietet der *DesignAnalyzer* dazu die Möglichkeit mit dem *Text Viewer*. Mit diesem Programm wird durch das Markieren einer bestimmten Stelle des Quelltextes das entsprechende Element im Stromlaufplan hervorgehoben angezeigt. Es besteht aber auch die Möglichkeit durch das Markieren eines Elements im Stromlaufplan die entsprechende Stelle im VHDL- oder Verilog- Quelltext suchen lassen. Der Einsatz des *Text Viewers* ist allerdings nur sinnvoll, wenn die Optimierung des Stromlaufplans noch nicht durchgeführt worden ist und die Strukturen des Stromlaufplanes noch unübersichtlich sind. Wird der *Text Viewer* bei bereits optimiertem Stromlaufplan eingesetzt, so ist in der Regel kein Bezug zwischen dem Quelltext und der dazugehörigen Schaltung herzustellen.

Der *Text Viewer* kann nur bei VHDL- oder Verilog-Datenformaten eingesetzt werden.

Bemerkung:

Der Einsatz des *Text Viewers* ist nur bei großen, komplexen Schaltwerken ratsam. Die Einschränkungen, die dieses Programm verursacht, müssen allerdings in Kauf genommen werden.

Zeitkontinuierliche integrierte Biquad-Filter

Martin Rieger, FH Albstadt-Sigmaringen

Abstract

Mit zeitkontinuierlichen biquadratischen Filtern in integrierter Bauweise können Aufgaben im Umfeld der analogen Signalverarbeitung in räumlich kompakter Weise und mit geringem technischem und finanziellem Aufwand gelöst werden. Parasitäre Effekte begrenzen allerdings die Einsatzmöglichkeiten dieser Filter. Zum automatischen Abgleich der Filter stehen verschiedene Verfahren zur Auswahl. Einige Anwendungen aus dem Bereich der Unterhaltungselektronik veranschaulichen die Möglichkeiten dieser Filter.

1. Einleitung

Für Filteranwendungen, bei denen ein günstiger Preis und geringer Platzbedarf eine bedeutende Rolle spielen, bieten zeitkontinuierliche biquadratische Filter, die als integrierte Schaltung realisiert sind, eine gute Lösung. Der Begriff "zeitkontinuierlich" grenzt diese analogen Filter von Switched Capacitor Filtern oder von digitalen Filtern ab.

Solche Aufgabenstellungen finden sich beispielsweise in der Unterhaltungselektronik und in der Kommunikationselektronik. Der im Vergleich zu anderen Filtern geringe Aufwand macht die hier betrachteten Filter für diese Einsatzgebiete attraktiv.

Bild 1 zeigt den prinzipiellen Aufbau einer biquadratischen Filterzelle. Das Filter besteht aus zwei Transkonduktanzverstärkern mit der Transkonduktanz g_{m1} bzw. g_{m2} , drei Kondensatoren mit den Kapazitätswerten C_0 , $m \cdot C_0$, $(m-1) \cdot C_0$ und einem Spannungsfolger mit der Verstärkung 1. Die Beschaltung der Eingänge V_a , V_b und V_c bestimmt die Filterfunktion:

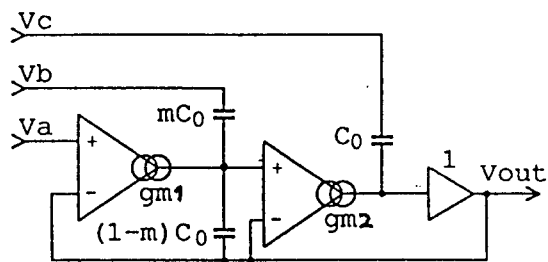


Bild 1: Prinzipschaltbild eines Biquad-Filters

Tiefpass:	V_b, V_c :	auf Masse;	V_a :	Eingangssignal
Hochpass:	V_a, V_b :	auf Masse;	V_c :	Eingangssignal
Bandpass:	V_a, V_c :	auf Masse;	V_b :	Eingangssignal
Falle:	V_b :	auf Masse;	V_a, V_c :	Eingangssignal.

Somit kann die selbe Filterschaltung durch entsprechende Umschaltung verschiedene Filterfunktionen erfüllen.

Die Ausgangsspannung des Filters nach Bild 1 ergibt sich zu:

$$V_{out} = \frac{V_a + m \cdot p \cdot \frac{C_0}{g_{m1}} \cdot V_b + p^2 \cdot \frac{C_0^2}{g_{m1} \cdot g_{m2}} \cdot V_c}{1 + m \cdot p \cdot \frac{C_0}{g_{m1}} + p^2 \cdot \frac{C_0^2}{g_{m1} \cdot g_{m2}}} \quad (\text{Gl. 1}).$$

Dabei ist p die komplexe Kreisfrequenz. Mit der Resonanzfrequenz ω_0 und der wirksamen Güte Q'

$$\omega_0^2 = \frac{g_{m1} \cdot g_{m2}}{C_0^2} \quad (\text{Gl. 2a}), \quad Q' = \frac{1}{m} \cdot \sqrt{\frac{g_{m1}}{g_{m2}}} = Q \cdot \sqrt{\frac{g_{m1}}{g_{m2}}} \quad (\text{Gl. 2b})$$

erhält man aus Gl. 1:

$$V_{out} = \frac{V_a + \frac{1}{Q'} \cdot \frac{p}{\omega_0} \cdot V_b + \frac{p^2}{\omega_0^2} \cdot V_c}{1 + \frac{1}{Q'} \cdot \frac{p}{\omega_0} + \frac{p^2}{\omega_0^2}} \quad (\text{Gl. 3}).$$

2. Elementare Filter Bausteine

Es werden nun die einzelnen Filter-Grundfunktionen sowie Varianten dazu betrachtet. Entsprechend den vorigen Ausführungen werden die Eingänge des Filters entsprechend der gewünschten Filterfunktion beschaltet.

Bild 2 zeigt den Aufbau eines Tiefpass-Filters. Die Übertragungsfunktion dieses Filters (Gl. 4a)) ergibt sich aus Gl. 1 und $V_b = V_c = 0$ und $V_a = V_{in}$. Oft wählt man $g_{m1} = g_{m2} = g_m$. In diesem Fall vereinfacht sich Gl. 4a zu Gl. 4b.

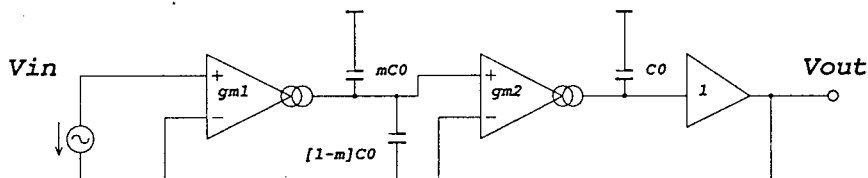


Bild 2: Tiefpass-Filter

$$\frac{V_{out}}{V_{in}} = \frac{1}{1 + m \cdot p \cdot \frac{C_0}{g_{m1}} + p^2 \cdot \frac{C_0^2}{g_{m1} \cdot g_{m2}}} \quad (\text{Gl. 4a})$$

$$\frac{V_{out}}{V_{in}} = \frac{1}{1 + m \cdot p \cdot \frac{C_0}{g_m} + p^2 \cdot \frac{C_0^2}{g_m^2}} \quad (\text{Gl. 4b}).$$

In Bild 3 ist das Prinzipschaltbild eines Hochpass-Filters dargestellt. Die Übertragungsfunktion im allgemeinen Fall und für den Spezialfall $g_{m1}=g_{m2}=g_m$ ist durch die Gleichungen 5a und 5b beschrieben.

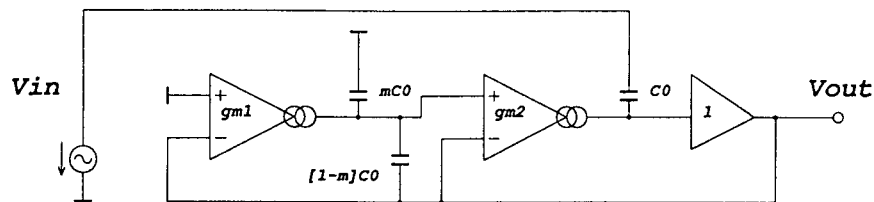


Bild 3: Hochpass-Filter

$$\frac{V_{out}}{V_{in}} = \frac{p^2 * C_0^2}{g_{m1} * g_{m2}}}{1 + m * p * \frac{C_0}{g_{m1}} + p^2 * \frac{C_0^2}{g_{m1} * g_{m2}}} \quad (\text{Gl. 5a})$$

$$\frac{V_{out}}{V_{in}} = \frac{p^2 * \frac{C_0^2}{g_m^2}}{1 + m * p * \frac{C_0}{g_m} + p^2 * \frac{C_0^2}{g_m^2}} \quad (\text{Gl. 5b})$$

Beim in Bild 4 dargestellten Bandpass, dessen Übertragungsfunktion für den allgemeinen Fall und für den Spezialfall $g_{m1}=g_{m2}=g_m$ in Gl. 6a bzw. Gl. 6b gegeben ist, kann von der Möglichkeit der Veränderung der wirksamen Güte über das Verhältnis g_{m1}/g_{m2} Gebrauch gemacht werden.

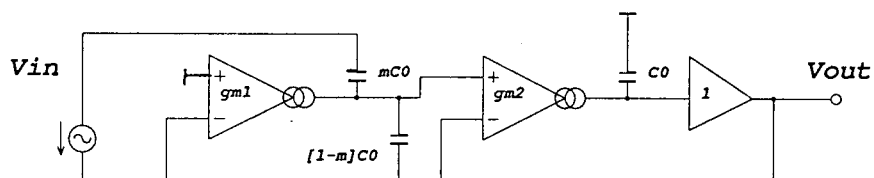


Bild 4: Bandpass-Filter

$$\frac{V_{out}}{V_{in}} = \frac{m * p * \frac{C_0}{g_{m1}}}{1 + m * p * \frac{C_0}{g_{m1}} + p^2 * \frac{C_0^2}{g_{m1} * g_{m2}}} \quad (\text{Gl. 6a})$$

$$\frac{V_{out}}{V_{in}} = \frac{m * p * \frac{C_0}{g_m}}{1 + m * p * \frac{C_0}{g_m} + p^2 * \frac{C_0^2}{g_m^2}} \quad (\text{Gl. 6b})$$

Bild 5a zeigt den auf die Resonanzfrequenz ω_0 bezogenen Frequenzgang eines Bandpasses für $m=0.5$ ($Q=Q'=2$) und $m=0.1$ ($Q=Q'=10$), wobei hier $g_{m1}=g_{m2}=g_m$ ist. Bei Bandpass-Filtern kann über ein elektronisches Steuersignal das Verhältnis g_{m1}/g_{m2} und damit die wirksame Güte eingestellt werden. Die Auswirkung einer Änderung des Verhältnisses g_{m1}/g_{m2} verdeutlicht Bild 5b für $m=0.3$: Für $g_{m1}/g_{m2}=1$ ist $Q'=3.3$, für $g_{m1}/g_{m2}=5$ ist $Q'=7.45$.

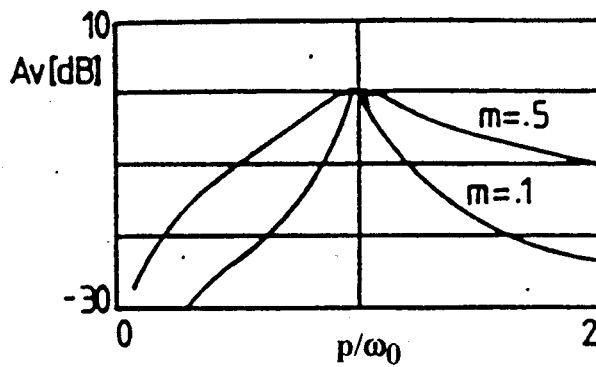


Bild 5a: Frequenzgang eines Bandpassfilters für verschiedene m bei $g_{m1}/g_{m2}=1$

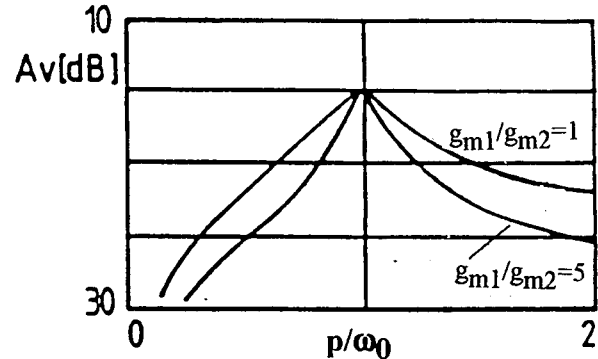


Bild 5b: Frequenzgang eines Bandpassfilters für verschiedene g_{m1}/g_{m2} bei $m=0.3$

Fallen-Filter unterdrücken Frequenzen in der Nähe ihrer Resonanzfrequenz. Bild 6 zeigt das Prinzipschaltbild eines biquadratischen Fallen-Filters. Entsprechend den Ausführungen zu Bild 1 wird die Eingangsspannung gleichzeitig an den Tiefpass-Eingang (V_a) und an den Hochpass-Eingang (V_c) angelegt, der Bandpass-Eingang (V_b) wird auf Masse gelegt. Die Übertragungsfunktion im allgemeinen Fall bzw. für $g_{m1}=g_{m2}=g_m$ ist durch Gl. 7a bzw. 7b gegeben.

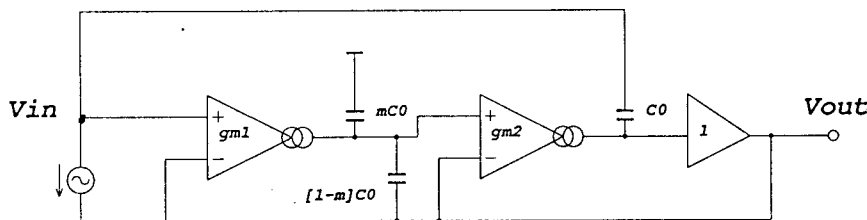


Bild 6: Fallen-Filter

$$\frac{V_{out}}{V_{in}} = \frac{1 + p^2 * \frac{C0^2}{g_{m1} * g_{m2}}}{1 + m * p * \frac{C0}{g_{m1}} + p^2 * \frac{C0^2}{g_{m1} * g_{m2}}} \quad (\text{Gl. 7a})$$

$$\frac{V_{out}}{V_{in}} = \frac{1 + p^2 * \frac{C0^2}{g_m^2}}{1 + m * p * \frac{C0}{g_m} + p^2 * \frac{C0^2}{g_m^2}} \quad (\text{Gl. 7b})$$

In Bild 7a ist der Frequenzgang eines Fallen-Filters für $m=0.1$ ($Q=10$) und $m=0.5$ ($Q=2$) bei $g_{m1}/g_{m2}=1$ dargestellt. Je höher die Güte ist, desto schmaler ist der vom Filter unterdrückte Frequenzbereich.

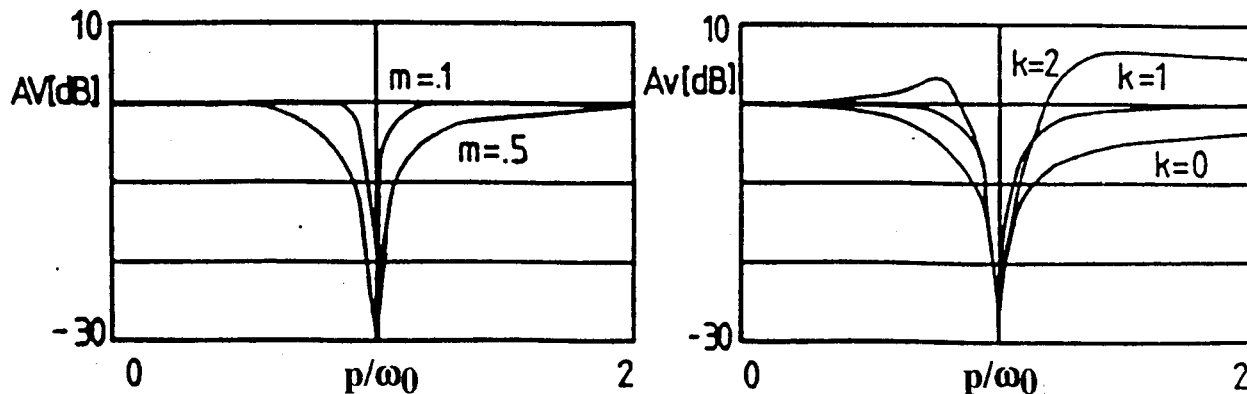


Bild 7a: Frequenzgang eines Fallen-Filters für verschiedene m bei $g_{m1}/g_{m2}=1$

Bild 7b: Frequenzgang eines Fallen-Filters für verschiedene k bei $m=0.3$ und $l=0.4$

Wird das Fallen-Filter nach Bild 6 um einen zusätzlichen Verstärker mit der Spannungsverstärkung k und einem zusätzlichen Kondensator mit der Kapazität $l \cdot C_0$, wie in Bild 8 dargestellt, ergänzt, so ergeben sich Variationsmöglichkeiten im Frequenzgang: abhängig von den gewählten Werten der Verstärkung k und Kapazität $l \cdot C_0$ kann die Charakteristik des Fallen-Filters in Richtung Tiefpass oder in Richtung Hochpass verändert werden. Die Übertragungsfunktion ist für beliebige Transkonduktanzverhältnisse mit Gl. 8a gegeben, für den Spezialfall $g_{m1}=g_{m2}=g_m$ gilt Gl. 8b. In Bild 7b ist der Frequenzgang eines Fallen-Filters $l=0.4$ und $m=0.3$ für verschiedene Verstärkungsfaktoren k wiedergegeben. Für $k=1$ sind die zusätzlichen Bauteile nicht wirksam und man erhält einen um die Resonanzfrequenz symmetrischen Frequenzgang, wie dies auch schon in Bild 7a zu sehen war. Wird $k=0$ gewählt, d. h. die Kapazität $l \cdot C_0$ wird mit einem Ende auf Masse gelegt, so erhält der Frequenzgang unterhalb der Resonanzfrequenz eine Überhöhung und oberhalb der Resonanzfrequenz tritt eine Absenkung auf. Wenn der Verstärkungsfaktor $k=2$ ist, erhält die unter der Resonanzfrequenz liegende Bandkante eine Verrundung und oberhalb der Resonanzfrequenz tritt eine Überhöhung auf.

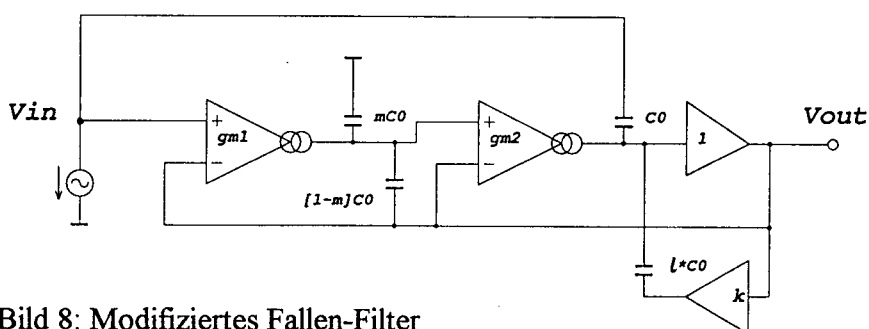


Bild 8: Modifiziertes Fallen-Filter

$$\frac{V_{out}}{V_{in}} = \frac{1 + p^2 \cdot \frac{C_0^2}{g_{m1} \cdot g_{m2}}}{1 + m \cdot p \cdot \frac{C_0}{g_{m1}} + p^2 \cdot \frac{C_0^2}{g_{m1} \cdot g_{m2}} \cdot (1 + l - l \cdot k)} \quad (\text{Gl. 7a})$$

$$\frac{V_{out}}{V_{in}} = \frac{1 + p^2 \cdot \frac{C_0^2}{g_m^2}}{1 + m \cdot p \cdot \frac{C_0}{g_m} + p^2 \cdot \frac{C_0^2}{g_m^2} \cdot (1 + l - l \cdot k)} \quad (\text{Gl. 7b})$$

Als letzte Grundsaltung wird ein als Allpass wirkendes Biquad-Filter betrachtet. Wie aus Bild 9 zu ersehen ist, ähnelt das Prinzipschaltbild des Allpasses demjenigen einer Falle. Allerdings benötigt hier der erste Transkonduktanz-Verstärker zusätzlich einen invertierten Ausgang. Die Übertragungsfunktion der Anordnung nach Bild 9 ist durch Gl. 8a gegeben. Die Schaltung arbeitet nur dann als Allpass, wenn die Bedingung $g_{m1}=0.5 \cdot g_{m2}$ eingehalten wird. In diesem Fall ist das Übertragungsverhalten durch Gl. 8b beschrieben.

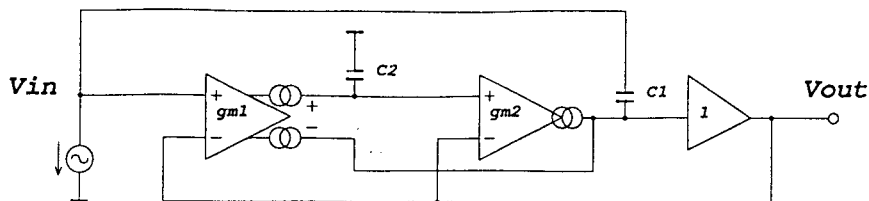


Bild 9: Allpass-Filter

$$\frac{V_{out}}{V_{in}} = \frac{1 - p \cdot \frac{C2}{g_{m2}} + p^2 \cdot \frac{C1 \cdot C2}{g_{m1} \cdot g_{m2}}}{1 - p \cdot \left(\frac{C2}{g_{m2}} - \frac{C2}{g_{m1}} \right) + p^2 \cdot \frac{C1 \cdot C2}{g_{m1} \cdot g_{m2}}} \quad (\text{Gl. 8a})$$

$$\frac{V_{out}}{V_{in}} = \frac{1 - p \cdot \frac{C2}{g_{m2}} + p^2 \cdot \frac{C1 \cdot C2}{g_{m1} \cdot g_{m2}}}{1 + p \cdot \frac{C2}{g_{m2}} + p^2 \cdot \frac{C1 \cdot C2}{g_{m1} \cdot g_{m2}}} \quad (\text{Gl. 8b})$$

3. Wirkung einer Phasenverzögerung im Transkonduktanz-Verstärker

Wesentliche Eigenschaften integrierter Biquadfilter, wie z. B. Frequenzbereich, maximale Güte oder Einfügedämpfung, sind von parasitären Eigenschaften begrenzt. Dominante parasitäre Effekte sind die Phasenverzögerung in den Transkonduktanzverstärkern und im Spannungsfolger. Im folgenden wird die Auswirkung einer Phasenverzögerung der Transkonduktanz betrachtet, die Erkenntnisse können leicht auf den Spannungsfolger übertragen werden.

Wird als Ursache der Phasenverzögerung des integrierten Transkonduktanzverstärkers ein als Tiefpass erster Ordnung modellierter Einfluß der Technologie-Grenzfrequenz f_c angenommen, so berechnet man die Phasenverzögerung $d\Phi$ bei der Frequenz f nach

$$d\Phi = \text{atan}(-f/f_c) \quad (\text{Gl. 9}).$$

Eine Phasenverzögerung von -1° erhält man bei $f=0.017 \cdot f_c$, bei $f=0.035 \cdot f_c$ beträgt die Phasenverzögerung -2° .

Die Phasenverzögerung der Transkonduktanz wird im folgenden frequenzunabhängig modelliert:

$$g_m^* = g_m \cdot e^{-d\Phi} \quad (\text{Gl. 10}),$$

wobei $d\Phi$ ein konstanter Phasenwinkel ist. Die reale, mit einer Phasenverzögerung behaftete Transkonduktanz g_m^* wird nun anstelle der idealen, verzögerungsfreien Transkonduktanz g_m in

die Transferfunktionen eingesetzt. Für ein Bandpass-Filter erhält man damit anstelle der Gl. 6b nun

$$\frac{V_{out}}{V_{in}} = \frac{m \cdot p \cdot \frac{C_0}{g_m^*}}{1 + m \cdot p \cdot \frac{C_0}{g_m^*} + p^2 \cdot \frac{C_0^2}{g_m^{*2}}} \quad (11)$$

Die Phasenverzögerung hat beim Bandpass-Filter Auswirkungen auf den Übertragungsfaktor a_0 bei der Resonanzfrequenz und auf die effektive Güte Q_{eff} . Im Idealfall ($d\Phi=0$) ist a_0 bei der Resonanzfrequenz gleich eins. Aus Gl. 11 kann man a_0 bei der Resonanzfrequenz unter Berücksichtigung von $d\Phi$ zu

$$a_0 = \frac{m}{m + 2 \cdot d\Phi} \quad (12)$$

berechnen. Für Q_{eff} , das im Idealfall ($d\Phi=0$) gleich der wirksamen Güte (Gl. 2b) ist, ergibt sich unter Einbeziehung der Phasenverzögerung aus Gl. 11:

$$Q_{eff} = \frac{Q}{1 + 2 \cdot Q \cdot d\Phi} \quad (13)$$

In den Bildern 10a bzw. 10b ist der Frequenzgang eines Bandpass-Filters mit $Q'=2$ bzw. $Q'=10$ für verschiedene $d\Phi$ dargestellt. Die oberste Kurve ist jeweils der Idealfall ($d\Phi=0$), die mittlere Kurve gilt für $d\Phi=-1^\circ$, die untere Kurve für $d\Phi=-2^\circ$. Man erkennt aus den Gl. 12 und 13 und aus den Bildern 10a und 10b, daß der schädliche Einfluß der Phasenverzögerung mit wachsender Güte zunimmt. Es wird deutlich, daß schon eine Verzögerung von -1° schädlich ist.

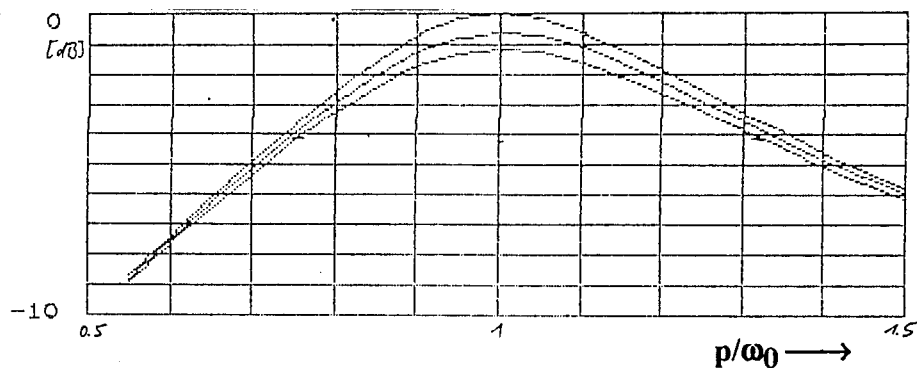


Bild 10a: Frequenzgang in [dB] eines Bandpass-Filters mit $Q'=2$ für $d\Phi=0^\circ$, -1° und -2°

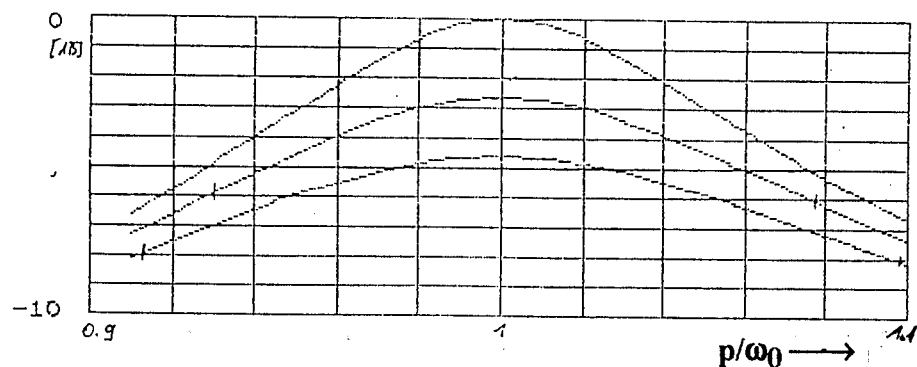


Bild 10b: Frequenzgang in [dB] eines Bandpass-Filters mit $Q'=10$ für $d\Phi=0^\circ$, -1° und -2°

Beim Fallen-Filter beeinflusst die Phasenverzögerung den Übertragungsfaktor a_0 bei der Resonanzfrequenz und die effektive Güte Q_{eff} . Im Idealfall ($d\Phi=0$) ist a_0 bei der Resonanzfrequenz gleich Null. Man kann aus Gl. 7b unter Einbeziehung der Phasenverzögerung den Übertragungsfaktor bei der Resonanzfrequenz zu

$$a_0 = \frac{2 \cdot d\Phi}{m + 2 \cdot d\Phi} \quad (\text{Gl. 14})$$

bestimmen. Für Q_{eff} , das auch hier im Idealfall ($d\Phi=0$) gleich der wirksamen Güte (Gl. 2b) ist, gilt auch hier Gl. 13.

Der Frequenzgang eines Fallen-Filters mit $Q'=3$ bzw. $Q'=10$ ist in den Bildern 11a bzw. 11b für verschiedene $d\Phi$ dargestellt. Die Kurve mit perfekter Unterdrückung der Resonanzfrequenz gehört zu $d\Phi=0^\circ$, die geringste Unterdrückung wird bei $d\Phi=-2^\circ$ erreicht. Wie aus Bild 11b zu ersehen ist, begrenzt die Phasenverschiebung die Unterdrückung bei der Resonanzfrequenz. So wird beispielsweise bei $Q'=10$ und $d\Phi=-1^\circ$ nur noch eine Unterdrückung von -12dB erreicht.

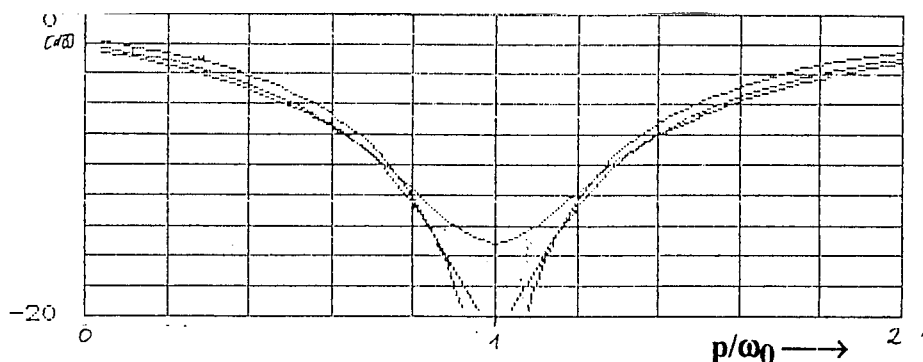


Bild 11a: Frequenzgang in [dB] eines Fallen-Filters mit $Q'=3$ für $d\Phi=0^\circ$, -1° und -2°

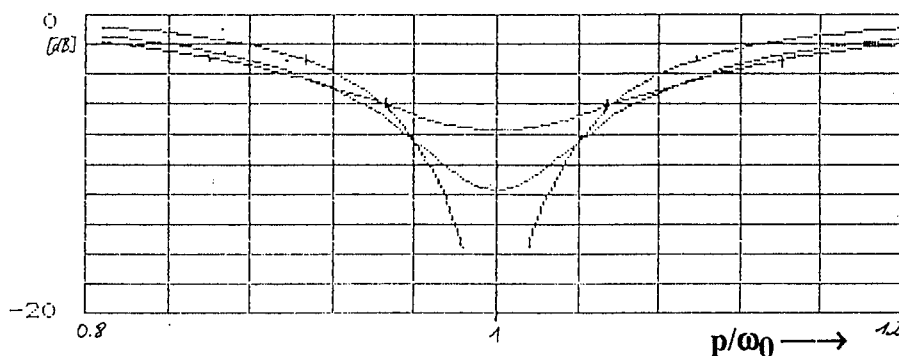


Bild 11b: Frequenzgang in [dB] eines Fallen-Filters mit $Q'=10$ für $d\Phi=0^\circ$, -1° und -2°

In Allpass-Filtern macht sich eine Phasenverzögerung als unerwünschter Amplitudengang bemerkbar. Der Übertragungsfaktor a_0 weicht dann von eins ab. Bei der Grenzfrequenz des Allpasses berechnet sich a_0 nach

$$a_0 = \frac{1 - 2 \cdot Q' \cdot d\Phi}{1 + 2 \cdot Q' \cdot d\Phi} \quad (\text{Gl. 15})$$

4. Untersuchungen verschiedener Transkonduktanz-Verstärker

Die wichtigsten Eigenschaften von Filtern sind die Einhaltung des gewünschten Frequenzganges, der Dynamikbereich sowie Leistungs- und Platzbedarf. Das Kerstück von zeitkontinuierlichen Biquadfiltern bilden die Transkonduktanzverstärker, sie bestimmen weitgehend die Leistungsfähigkeit der Biquadfilter.

Es werden verschiedene Transkonduktanzverstärker untersucht. Als Leistungsmerkmale dienen

- f_{1° : Frequenz, bei der eine Phasenverzögerung von -1° auftritt,
- V_{inmax} : maximal sinnvolle Eingangsspannung und
- V_{inoise} : auf den Eingang bezogene Rauschspannungsdichte.

Zunächst wird ein einfacher Differenzverstärker ohne Gegenkopplung nach Bild 12 betrachtet. Die Transkonduktanz ist durch

$$g_m = \frac{I_2}{2 \cdot V_T} \quad (\text{Gl. 16})$$

gegeben, wobei V_T die Temperaturspannung ist, die bei 27°C etwa 26mV beträgt. Die Transkonduktanz ist in einem weiten Bereich über I_2 einstellbar. Die Frequenz f_{1° ist technologieabhängig, bei einer Technologie mit einer Grenzfrequenz von 2.5GHz beträgt $f_{1^\circ} \approx 25\text{MHz}$. Die für einen Klirrfaktor von 1% maximal zulässige Eingangsspannung V_{inmax} beträgt hier etwa 20mV . Die Eingangsrauschspannungsdichte V_{inoise} ist abhängig von den verwendeten Transistoren, liegt jedoch üblicherweise in der Größenordnung von $5\text{ nV}/\sqrt{\text{Hz}}$. Wenn an den Stellen A und B in Bild 12 Dioden eingefügt werden, verdoppelt sich V_{inmax} , während V_{inoise} sich nur unwesentlich erhöht. Allerdings muß nun I_2 verdoppelt werden, wenn g_m gleich bleiben soll.

Als weiteres Beispiel für einen Transkonduktanzverstärker wird der Multiplizierer nach Bild 13 betrachtet. Hier berechnet sich die Transkonduktanz g_m nach

$$g_m = \frac{\frac{I_4}{I_1}}{R_3 + \frac{2 \cdot V_T}{I_1}} \approx \frac{I_4}{I_1 \cdot R_3} \quad (\text{Gl. 17}).$$

Über das Verhältnis I_4/I_1 ist die Transkonduktanz in weiten Grenzen einstellbar. Die Frequenz f_{1° beträgt hier etwa 8MHz , wenn wieder eine Technologie mit einer Grenzfrequenz von 2.5GHz zugrunde gelegt wird. Hier beträgt die maximal zulässige Spannung V_{inmax} etwa $I_2 \cdot R_3$, was in der Größenordnung von 100mV bis 1V liegt. Die Eingangsrauschspannungsdichte V_{inoise} hängt von den verwendeten Transistoren, von R_3 , I_1 und I_4 ab. Für V_{inoise} sind Werte um $20\text{ nV}/\sqrt{\text{Hz}}$ möglich. Die Transistoren Q5 und Q6 tragen hier wesentlich zum Rauschen bei, daher verringern an A, B, C und D eingefügte Dioden die Eingangsrauschspannungsdichte.

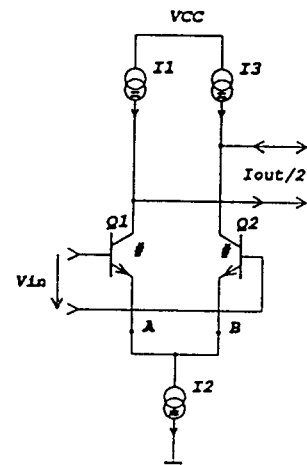


Bild 12: Differenzverstärker

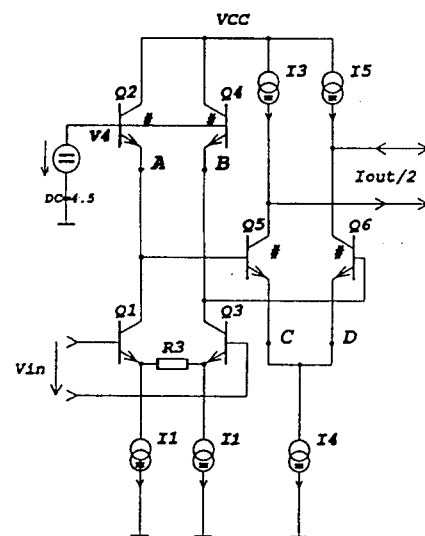


Bild 13: Multiplizierer

Vergleicht man die Ergebnisse der Schaltungen nach Bild 12 und Bild 13, so erkennt man, daß die Schaltung nach Bild 12 den besseren Frequenzgang und die Schaltung nach Bild 13 den größeren Dynamikbereich hat.

Für manche Anwendungen sind Transkonduktanzverstärker aus MOS-Transistoren interessant. Ein Beispiel zeigt Bild 14. Hier wird die Drain-Source-Strecke zur Festlegung der Transkonduktanz verwendet. Für die Transkonduktanz g_m gilt in einem gewissen Betriebsbereich

$$g_m = k \cdot \frac{W}{L} \cdot V_{DS} \tag{Gl. 18},$$

wobei k eine von der Technologie abhängige Konstante W die Gate-Breite, L die Gate-Länge und V_{DS} die Drain Source-Spannung sind. Über den Strom I_2 kann V_{DS} und damit g_m kontrolliert werden. Die Schaltung nach Bild 14 weist im Vergleich zu den Schaltungen nach Bild 12 und 13 den Vorteil des größeren Dynamikbereiches auf, doch stehen dem die Nachteile des schlechteren Frequenzganges und eines kleinen Durchstimmbereiches gegenüber.

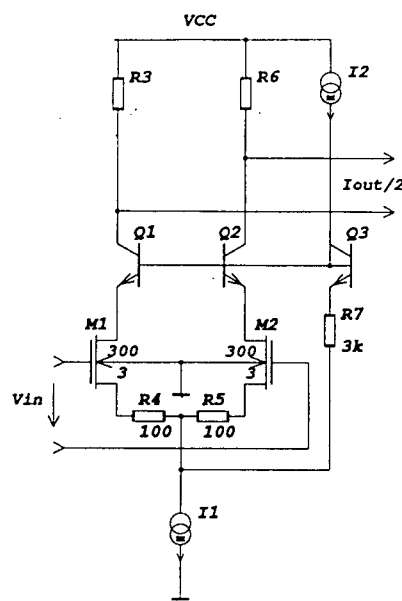


Bild 14: MOS-Transkonduktanz-Verstärker

5. Abgleich integrierter Biquadfilter

Die Resonanzfrequenz ω_0 integrierter Biquadfilter ist entsprechend Gl. 2a proportional zur Transkonduktanz und umgekehrt proportional zum Kapazitätswert C_0 . Die Transkonduktanz kann, je nach verwendetem Transkonduktanzverstärker, von Strömen, einem Widerstandwert oder von der absoluten Temperatur abhängen. Somit kann ω_0 von den Absolutwerten der integrierten Kapazitäten und Widerständen und von der Temperatur abhängen.

Der Absolutwert von integrierten Kondensatoren und Widerständen hat eine Unsicherheit in der Größenordnung von $\pm 20\%$, bezogen auf einen Wertebereich von 6 mal der Standardabweichung (6σ). Die Absolutwerte ändern sich von einem Wafer zum nächsten. Auf einem IC sind die Abweichungen von Kapazitäts- und Widerstandswerten gleicher Bauteile zueinander viel geringer als zum Absolutwert, d.h. die Relativtoleranz ist viel geringer als die Absoluttoleranz. Die Relativtoleranzen von integrierten Kondensatoren und Widerständen liegen etwa bei $\pm 1\%$.

Die Resonanzfrequenz ω_0 hängt also systematisch von den Absolutwerten der integrierten Kondensatoren und Widerständen sowie evtl. der Temperatur ab. Zudem besteht eine stochastische Abhängigkeit der Resonanzfrequenz ω_0 von den Relativtoleranzen der Kondensatoren und Widerstände und von den Transistorgrößen.

Es ist Aufgabe einer Abgleichschaltung, die Resonanzfrequenz ω_0 , unabhängig von Kapazitäts- und Widerstandswerten oder sonstigen Einflüssen, auf den gewünschten Wert zu bringen. Abhängig von der Art der verwendeten Abgleichschaltung werden mehrere oder alle Absolut- und Relativtoleranzen sowie Temperaturabhängigkeiten abgeglichen. Der Abgleichvorgang läuft, je nach Art der Abgleichschaltung, autonom oder unter der Kontrolle des Anwenders ab.

Das einfachste Verfahren besteht in einem einmaligen Abgleichvorgang mittels eines externen Widerstandes oder einer programmierbaren Stromquelle (Bild 15).

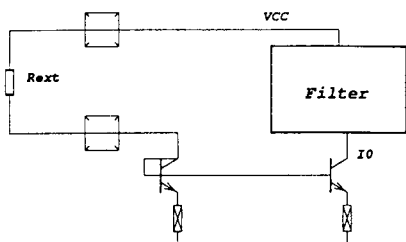


Bild 15a: Abgleich über externen Widerstand

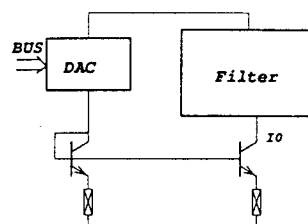


Bild 15b: Abgleich über programmierbare Stromquelle

In Bild 15a wird über den Widerstandswert von R_{ext} der Strom I_0 und damit, über eine Beeinflussung von g_m , die Resonanzfrequenz ω_0 verändert. In der Schaltung nach Bild 15b beeinflusst der Mikroprozessor über eine Busverbindung einen Digital-Analog-Converter und verändert über I_0 die Transkonduktanz und damit ω_0 .

Zum Abgleich wird R_{ext} bzw. der Einstellwert des DAC solange verändert, bis die gewünschte Resonanzfrequenz ω_0 auftritt. Während des Abgleichvorganges muß selbstverständlich ω_0 laufend gemessen werden. Der Vorteil des Verfahrens liegt neben seiner Einfachheit im direkten Verfahren, d.h. das zu beeinflussende Filter wird auch direkt gemessen. dadurch spielen Relativtoleranzen nur eine untergeordnete Rolle. Nachteilig an diesem Verfahren ist, daß nur zu einem Zeitpunkt abgeglichen wird, sodaß z. B. Temperaturänderungen oder Alterungsvorgänge ω_0 verändern. Zudem ist der Abgleichvorgang aufwendig, weil zusätzliche Meßgeräte zur Überwachung von ω_0 notwendig sind.

Bild 16 zeigt ein Verfahren, das für Anwendungen einsetzbar ist, bei denen ein Signal bei einer festen Frequenz dominant ist oder bei denen ein frequenzmoduliertes Signale verarbeitet wird.

Das Eingangssignal V_{in} durchläuft das abzugleichende Filter und gelangt dann in einen Frequenzdiskriminator (FM-Detector). Falls die Frequenz von V_{in} nicht mit der Resonanzfrequenz ω_0 des FM-Detectors übereinstimmt, tritt ein DC-Offset auf, der über eine Regelschaltung die Resonanzfrequenz des FM-Detectors auf die Frequenz von V_{in} abgleicht. Man nennt dieses Verfahren auch Automatic Frequency Control (AFC). Für das

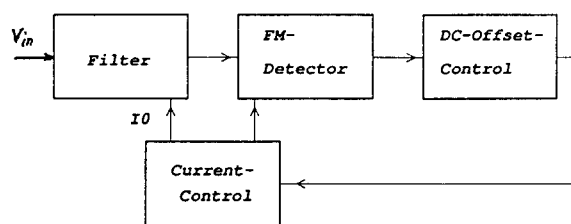


Bild 16: Abgleichschaltung mit AFC

Filter und den FM-Detector werden Schaltungen mit ähnlichen Strukturen verwendet, sodaß die Resonanzfrequenzen im Filter und im FM-Detector ein konstantes Verhältnis zueinander haben. Dadurch ist es möglich, daß das Abgleichsignal des FM-Detectors das Filter mit abgleicht.

Diese Abgleichschaltung arbeitet autonom, d. h. es werden keine zusätzlichen Meßgeräte benötigt. Temperatureinflüsse werden mit dieser Schaltung ausgeregelt. Als Nachteil dieser Schaltung ist die Empfindlichkeit auf Relativtoleranzen zwischen Filter und FM-Detector zu nennen, da die überwachte Schaltung (FM-Detector) und die zu überwachende Schaltung (Filter) nicht identisch sind.

Eine weitere Möglichkeit zum Filter-Abgleich besteht, darin in der Abgleichschaltung ein Filter der gleichen Struktur zu verwenden, wie die abzugleichenden Filter. Es wird also ein zusätzliches Filter zur Gewinnung des Abgleichsignals benötigt. Die Bilder 17a und 17b zeigen zwei Varianten, die nach diesem Prinzip arbeiten.

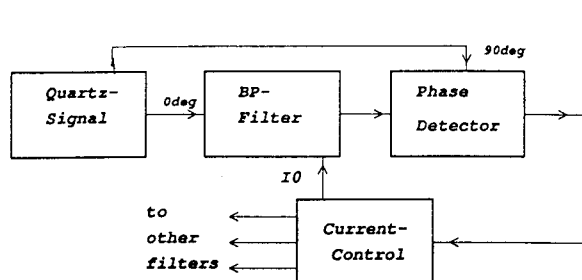


Bild 17a: Abgleichschaltung mit Bandpass-Filter als Phasenschieber

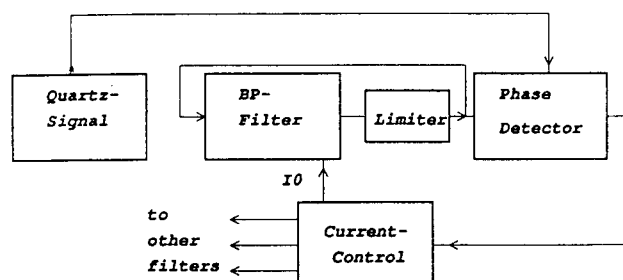


Bild 17b: Abgleichschaltung mit Bandpass-Filter als Oszillator

In Bild 17a wird ausgenutzt, daß bei einem Bandpass-Filter, das bei seiner Resonanzfrequenz ω_0 betrieben wird, zwischen Ein- und Ausgangssignal 0° Phasenverschiebung vorliegen. Sobald ein Frequenzunterschied zwischen ω_0 und dem am Eingang des Filters anliegenden Quarzsignal auftritt, weicht die Phase des Filter-Ausgangssignals von 0° ab. Das Ausgangssignal des Filters wird im Phasendetektor mit einem um 90° phasenverschobenen Quarzsignal verglichen. Eine Phasenabweichung des Filter-Ausgangssignales von 0° führt zu einer Änderung des Stromes I_0 , damit zu einer Änderung von g_m , und somit zu einer Korrektur der Resonanzfrequenz ω_0 . Das Abgleichsignal in Form eines Stromes kann dann die zur Signalverarbeitung verwendeten Filter abgleichen.

Die Schaltung nach Bild 17b arbeitet nach einem ähnlichen Prinzip. Hier wird ausgenutzt, daß ein rückgekoppeltes Bandpass-Filter auf seiner Resonanzfrequenz ω_0 schwingt. Wenn zwischen der Schwingfrequenz des Bandpass-Filters und der Frequenz des Quarzsignales eine Abweichung besteht, so erzeugt der Phasendetektor über eine Regelschaltung eine Änderung von I_0 und gleicht so ω_0 auf die Quarzfrequenz ab. Dieses Schaltungsprinzip wird als Phasenregelschleife (PLL) bezeichnet. Der Limiter wird zur Verstärkung und Amplitudenbegrenzung benötigt.

Beide hier betrachteten Schaltungen arbeiten autonom, benötigen jedoch ein Referenzsignal geeigneter Frequenz. Alle Einflüsse, die auf das Bandpass-Filter und die abzugleichenden Filter gleichermaßen wirken, werden ausgeglichen. Dagegen führen Relativtoleranzen zwischen dem Bandpass-Filter und den übrigen Filtern zu einem Abgleichfehler.

Zum Abschluß wird ein in Bild 18 dargestelltes Verfahren betrachtet, bei dem zum Abgleich benötigte Signale im IC erzeugt werden.

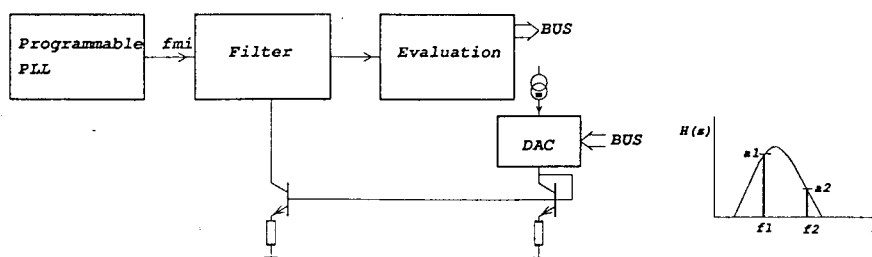


Bild 18: Abgleichschaltung mit intern erzeugten Kalibriersignalen

Eine programmierbare PLL erzeugt die zum Abgleich benötigten Signale mit den Frequenzen f_{mi} . Für den Abgleich eines Bandpass-Filters auf seine Mittenfrequenz werden symmetrisch zur gewünschten Resonanzfrequenz Signale mit den Frequenzen f_1 und f_2 benötigt. Die programmierbare PLL legt die Signale mit diesen Frequenzen nacheinander an das Filter an und eine Evaluationschaltung wertet die zugehörigen Amplituden a_1 und a_2 aus. Die Resultate werden über eine Busverbindung an den Mikroprozessor übertragen. Dort wird geprüft, ob die Amplituden a_1 und a_2 gleich groß sind. Falls dies nicht der Fall ist, wird die notwendige Stromänderung berechnet. Der neue Wert des Stromes wird über den Bus an das Filter-IC übertragen und mittels eines DAC an das Filter angelegt. Es können nun noch weiter iterative Annäherungen an die gewünschte Resonanzfrequenz erfolgen.

Dieses Verfahren gleicht die von Absoluttoleranzen verursachte Fehler aus, Relativtoleranzen spielen hier keine Rolle, da ja das abzugleichende Filter direkt gemessen wird. Diesen Vorteilen steht als Nachteil der beträchtliche Aufwand der programmierbaren PLL gegenüber.

6. Anwendungen zeitkontinuierlicher integrierter Biquadfilter

Es werden mehrere Anwendungen zeitkontinuierlicher Biquadfilter aus dem Bereich der Unterhaltungselektronik vorgestellt: Filter für Signale bei 500kHz und 260kHz, Tonfalle zur Unterdrückung von Tonträgern im Videosignal und Basisbandfilter für digitalen Rundfunk. Alle diese Anwendungen wurden in einer BICMOS-Technologie mit einer Grenzfrequenz von 2.5GHz realisiert.

Bild 19 zeigt das Blockschaltbild eines Filters mit zwei Zweigen.

Der linke Zweig filtert und demoduliert ein frequenzmoduliertes Signal, das eine Bandbreite von etwa 150kHz hat, bei 500kHz. Jeder der Blöcke besteht aus einem Biquadfilter mit Transkonduktanzverstärkern nach Bild 13. Das Bandpass-Filter, das eine Güte von 3 aufweist, übernimmt die Grobselektion. Das Fallen-Filter bei 1.1MHz, das Reste des Bildsignals unterdrücken soll, hat ebenfalls eine Güte von 3. Die Fallen-Filter bei 285kHz und 770kHz bestimmen die Bandbreite des Durchlaßbereiches. Das Fallen-Filter bei 285kHz wurde mit einer zusätzlichen Hochpaßcharakteristik versehen, wobei die Schaltung nach Bild 8 mit den Parametern $m=1$, $l=2$, $k=1.75$ verwendet wurde. Das Fallen-Filter bei 770kHz weist eine zusätzliche Tiefpaßcharakteristik auf, was mit den Parametern $m=0.7$, $l=0.5$ und $k=0$ erreicht wurde.

Der rechte Filterzweig in Bild 19 filtert und demoduliert ein frequenzmoduliertes Signal bei 260kHz. Aufbau und Funktionsweise ähneln dem Zweig bei 500kHz. Als Besonderheit ist zu erwähnen, daß die beiden Fallen bei 460kHz und 510kHz eine Unterdrückung des in diesem Zweig unerwünschten 500kHz-Signales von mehr als 55dB erreichen.

Für diese Anwendung wurde die in Bild 16 beschriebene Abgleichschaltung verwendet.

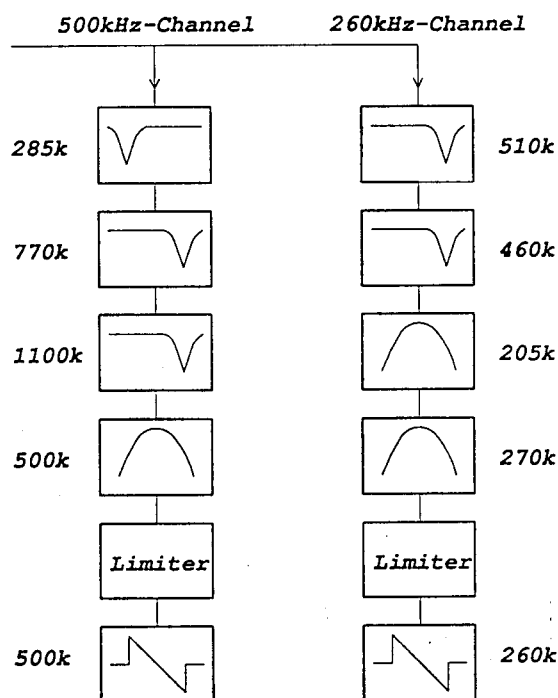


Bild 19: Blockschaltbild der Filter für 500kHz und 260kHz

Bild 20 zeigt den gemessenen Frequenzgang der beiden Filterzweige. Die Übereinstimmung zwischen dem gemessenen und dem simulierten Frequenzgang ist ausgezeichnet.

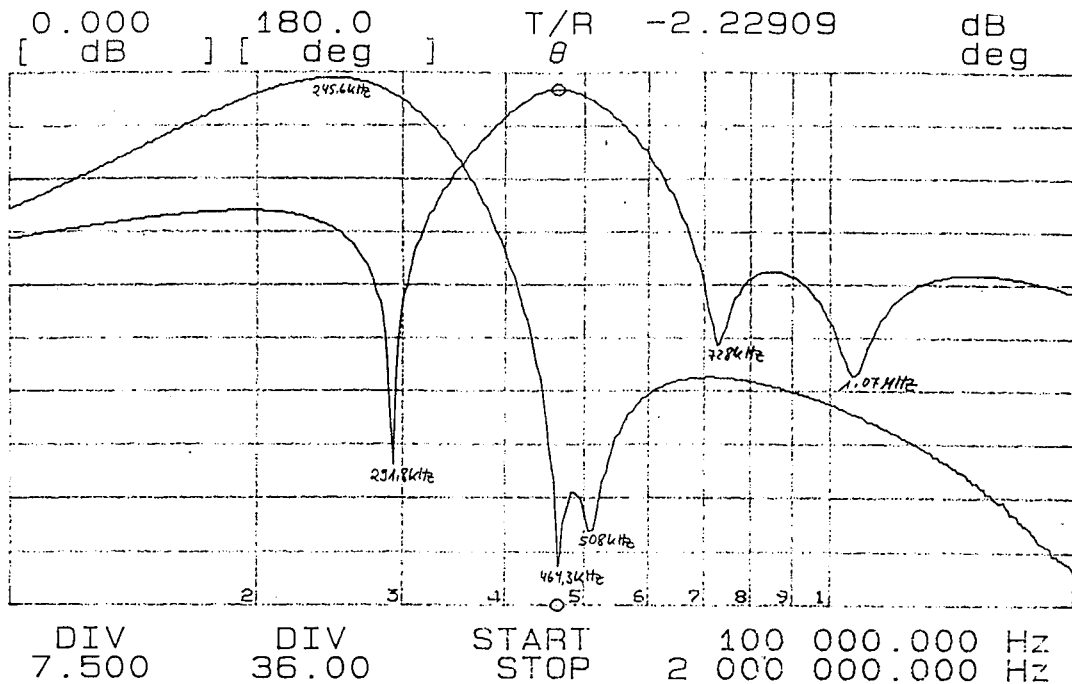


Bild 20: Frequenzgang des Filters aus Bild 19.

Das vom Fernseh-Tuner abgegebene Videosignal enthält noch Reste des Tonträgers. Da diese Tonträger-Reste den Bildeindruck stören würden, muß das Videosignal davon befreit werden. Dazu genügt in der Regel ein einfaches Fallen-Filter, wie es in Bild 6 dargestellt ist. Als Transkonduktanzverstärker wurde der in Bild 12 dargestellte Typ verwendet, wobei noch an den Stellen A und B jeweils zwei Dioden in die Emitterzweige eingefügt wurden. Das Filter war so in der Lage, ein Videosignal mit einem Pegel von 200mVpp ohne merkliche Verzerrungen zu verarbeiten. Das Filter wurde mit einer Abgleichschaltung nach Bild 17a abgeglichen. Bild 21 zeigt den gemessenen und simulierten Frequenzgang des Filters. Die Falle hat eine Unterdrückung von mehr als 30 dB bei einer Güte von 5, was durch eine teilweise Kompensation der Phasenverzögerung der Transkonduktanzverstärker erreicht wurde.

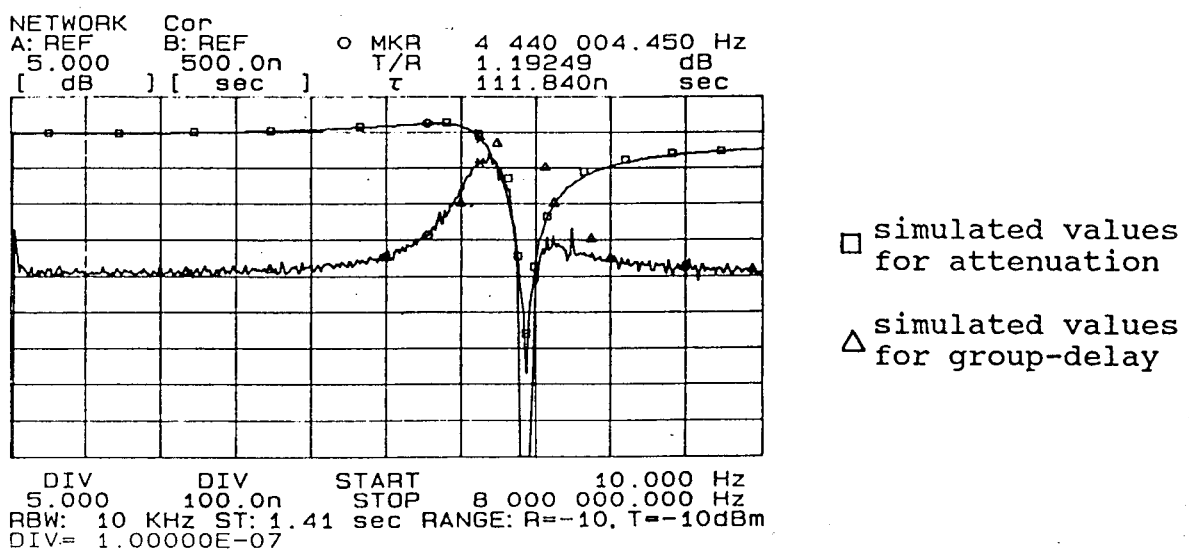


Bild 21: Frequenzgang einer Tonfalle

Als abschließendes Beispiel wird ein Basisbandfilter für digitalen Rundfunk (DAB) mit zwei unabhängigen Zweigen betrachtet. Diese Filter soll Frequenzen bis 750kHz ohne nennenswerte Dämpfung durchlassen und bei 950kHz schon eine Dämpfung von mehr als 40 dB aufweisen.

An die Ähnlichkeit der beiden Filterzweige werden hohe Anforderungen gestellt: im Durchlaßbereich dürfen die Amplitudenunterschiede höchstens ± 1.5 dB betragen, die Phasenunterschiede sollen kleiner als $\pm 1^\circ$ sein. Diese Ziel ließ sich mit der von der Technologie gegebenen vorgegebenen Relativtoleranzen, die für die Widerstände im günstigsten Fall 0.5% und für die Kondensatoren 0.35% betragen, nicht als Kette von Tiefpass-Filtern realisieren. Statt dessen wurde das Filtern aus Frequenzfallen zusammengesetzt, die weniger Probleme hinsichtlich der Amplituden- und Phasenunterschiede bereiten.

In Bild 21 ist das Blockschaltbild eines Filterzweiges dargestellt. Das Filter besteht auf 11 Fallen-Filtern nach Bild 6, wobei Transkonduktanzverstärker vom in Bild 13 dargestellten Typ verwendet werden. An die ersten 4 Filter, die am meisten auf das obere Frequenzende des Durchlaßbereiche wirken, werden besonders hohe Genauigkeitsanforderungen gestellt.

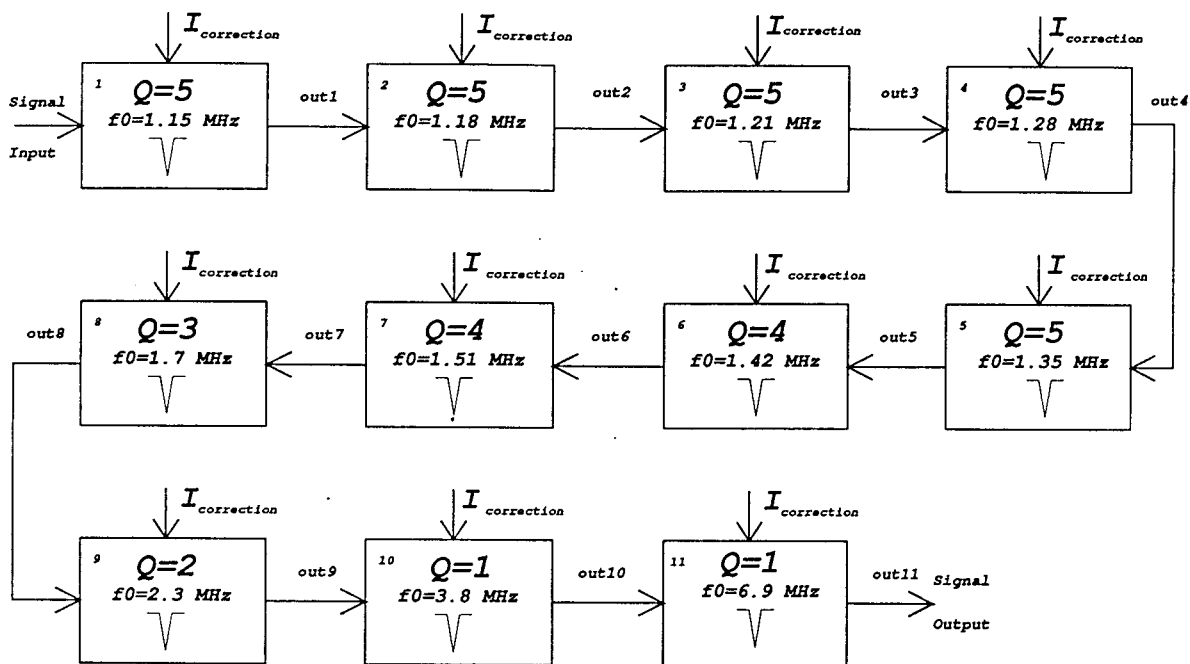


Bild 21: Blockschaltbild eines Zweiges des DAB-Basisbandfilters

Selbstverständlich läßt sich die geforderte Genauigkeit nur erreichen, wenn die Absoluttoleranzen mittels einer Abgleichschaltung eliminiert werden. Dabei fand das in Bild 17b dargestellte Schaltungsprinzip Verwendung. Es wird also ein Referenzfilter abgeglichen und das Korrektursignal auf die in den Signalzweigen befindlichen Filter als $I_{\text{correction}}$ angewendet. Somit sind die Relativtoleranzen der Filter zum Referenzfilter immer noch wirksam.

In der Simulation wurde geprüft, inwieweit die Genauigkeitsanforderungen an den Frequenzgang der Filter unter Berücksichtigung der Toleranzen eingehalten werden. Dabei wurden Monte Carlo Analysen durchgeführt, bei denen alle wesentlichen Relativtoleranzen im zu erwartenden Umfang variiert werden. Von den Simulationen wurde prognostiziert, daß die Genauigkeitsanforderungen mit der vorgegebenen Technologie gerade erfüllt werden können, wenn Bauteile mit der geringstmöglichen Toleranz verwendet werden.

Bild 22 zeigt das Simulationsergebnis einer Monte Carlo Analyse für das in Bild 21 dargestellte Filter. Die bislang vorliegenden experimentellen Ergebnisse zeigen gute Übereinstimmung mit der Simulation. Dies gilt insbesondere für die kritischen ersten 4 Filterblöcke in Bild 21.

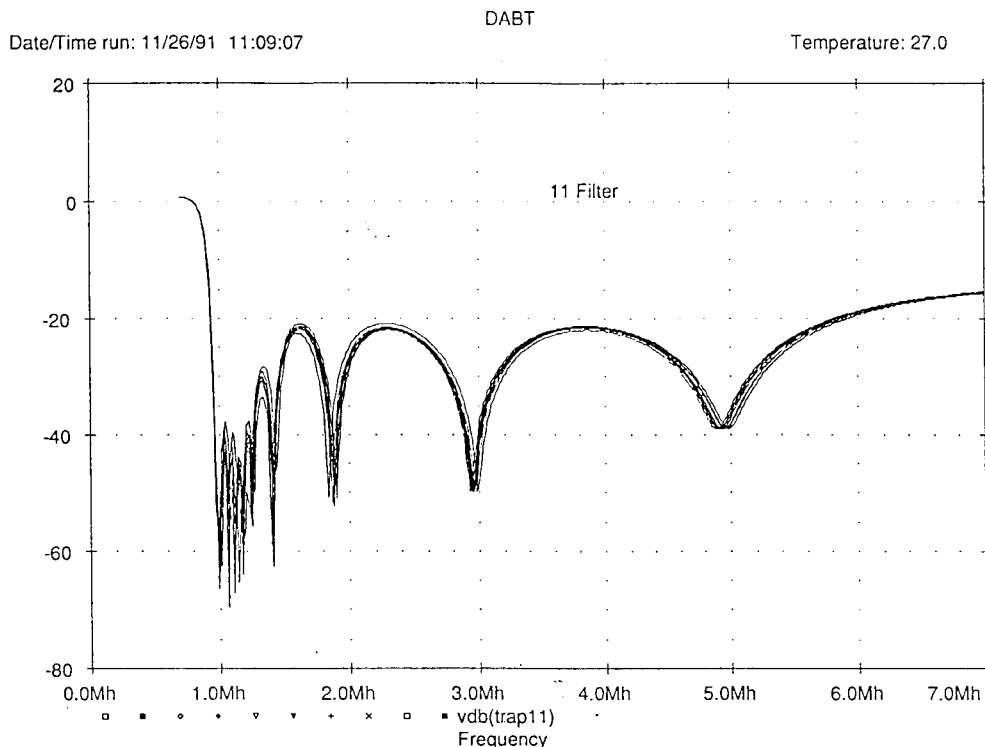


Bild 22: Ergebnis der Monte Carlo Simulation des Frequenzganges

7. Zusammenfassung

Die Möglichkeiten und Grenzen der zeitkontinuierlichen integrierten Biquadfilter wurden aufgezeigt. Die bisher gemachten praktischen Erfahrungen sind ermutigend.

Zusammenfassend seien die wichtigsten Kenngrößen der Filter genannt. Die Filter können bis etwa zu $1/100$ der Grenzfrequenz der Technologie ohne große Beschränkungen betrieben werden. Der erreichbare Dynamikbereich liegt in der Größenordnung von 75 dB, er hängt jedoch stark von der akzeptierten Verlustleistung und vom Frequenzbereich ab.

Die Güte der Filter sollte nicht größer als 10 gewählt werden, eine größere Güte wäre nur bei besonders geringen Relativtoleranzen und weit unterhalb der Technologie-Grenzfrequenz sinnvoll. Die Verlustleistung pro Biquadfilter liegt je nach Anwendung im Bereich zwischen 0.1 mW und 10 mW.

Für ein Biquadfilter ohne Abgleichschaltung und Kondensatoren wird etwa eine Fläche von 0.1 Quadratmillimeter benötigt.

Als weiterführende Literatur sei das von Y. P. Tsividis und J. O. Voorman herausgegebene Buch *Integrated Continuous-Time Filters* empfohlen, das in IEEE Press im Jahre 1993 erschien.

Entwicklung eines 16-Bit Mikroprozessor-Kernels mit Hilfe von VHDL

Fritz Zimpfer, Thomas Gieringer

Prof. Dr.-Ing Dirk Jansen

Mitteilung aus dem ASIC-Design-Center der FH Offenburg

Für die Implementation in ASIC's wurde ein kompakter Mikroprozessor-Kernel als Standardzellen-Makro entworfen. Durch konsequenten Einsatz von Hochsprachen und CAE-Werkzeugen (VHDL, Synthese) konnte ein vollständiges Design in nur vier Monaten durchgeführt werden. Der Prozessor wird in einem Testchip erprobt.

Einführung

Mit der Verfügbarkeit moderner CAE-Werkzeuge können heute komplexe Entwürfe angegangen werden. Um diese Leistungsfähigkeit zu demonstrieren und gleichzeitig den inneren Aufbau moderner Prozessoren für die Lehre zu erschließen, wurde am ASIC-Design-Center der FH Offenburg das Studentenprojekt FHOP ins Leben gerufen.

FHOP steht für First Homemade Operational Processor. Im Rahmen dieses Projekts wird ein Mikroprozessor-Kernel entwickelt, der zur Implementation als Standardzellen-Makro in verschiedenen Steuer- und Kontrollanwendungs-ASIC's geeignet ist. Durch Plazieren des Kernels kann Prozessorleistung auf einem ASIC zur Verfügung gestellt werden. Der Anwender fügt dann die für seine Anwendung spezifischen Komponenten (z.B. I/O-Ports, Interruptcontroller, Timer usw.) hinzu. Den typischen Aufbau eines solchen ASIC's mit Prozessor-Kernel zeigt Bild 1.

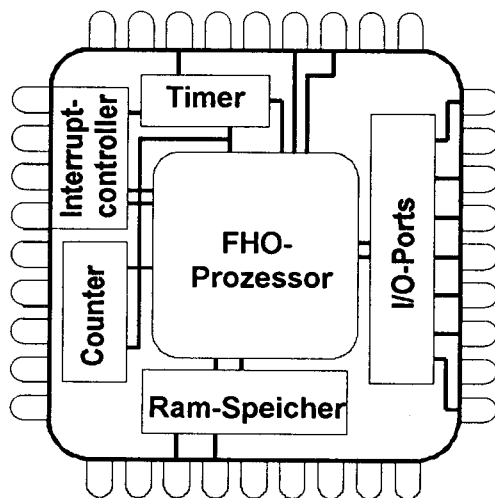


Bild 1: Einsatz des Mikroprozessor-Kernels

Entwicklungsumgebung

Im folgenden sollen die wichtigsten Werkzeuge der Entwicklungsumgebung kurz erläutert werden.

Die Entwicklungsumgebung besteht aus drei HP-Workstations mit MENTOR GRAPHICS CAE-Tools in der Version 8.2. Der Chip wird auf Grundlage der ES 2 - Bibliothek in 1,0 µm-Technologie über EUROCHIP gefertigt.

VHDL

Die Entwicklungsumgebung erlaubt Beschreibungsmodelle eines Designs bzw. Chips zu entwerfen und zu simulieren. VHDL ist eine Beschreibungssprache für digitale Hardware, die anderen Programmiersprachen ähnlich ist. Sie enthält Komponenten der Portbeschreibung (engl. entity) und der Verhaltensbeschreibung (engl. architecture). Somit kann auf abstrakter Ebene ein Teilmodul oder die komplette Architektur einer Schaltung untersucht werden. Aus der Beschreibung kann anschließend schrittweise die Gatterlogik abgeleitet, zum Teil automatisch synthetisiert werden. Damit kann die Chipentwicklung digitaler Schaltkreise wie hier beim FHOP wesentlich beschleunigt werden.

Design Architect

Mit Hilfe des DESIGN ARCHITECT können Symbole entworfen und mit VHDL-Modellen hinterlegt werden. Zusammen mit Bibliothekselementen werden die so kreierte Symbole auf einem gemeinsamen Schaltplan plziert. Der Gesamtentwurf präsentiert sich als hierarchisch gegliederte, übersichtliche Struktur.

Quicksim II

Mit dem Digitalsimulator läßt sich diese heterogene Architektur aus realen Gattern und VHDL-Modellen simulieren. Designfehler können damit früh erkannt und behoben werden.

AutoLogic

Dieses Software-Paket beinhaltet ein Syntheseprogramm, das ein korrektes VHDL-Modell automatisch in eine Gatterlogik umsetzt. Weiterhin ist eine zielgerichtete Flächen- und Geschwindigkeitsoptimierung der Schaltung auf die Zieltechnologie - hier ES2 1,0 μm Technologie - möglich.

IC - Station

Mit Hilfe dieses CAE-Werkzeugs werden die Bibliothekselemente und Makros auf dem Silizium plaziert und geroutet. Hieraus werden die Layoutdaten für die Chipfertigung generiert.

Entwicklungsschritte

Der Entwicklungsprozeß wird konsequent nach dem Top-Down-Prinzip (vgl. Bild 2) durchgeführt. Zuerst wird die Gesamtstruktur des Prozessors und dessen Befehlssatz auf dem Papier festgelegt. Dann wird der FHO-Prozessor in logische Funktionsblöcke partitioniert, um das Design übersichtlich zu halten. Diese einzelnen Komponenten werden dann in VHDL-Modellen beschrieben und mit Simulation auf ihre Funktion hin überprüft.

Im nächsten Schritt wird der gesamte Kernel aus den einzelnen Modulen zusammengesetzt und im Simulator getestet. Alle Befehle und Funktionen des Kernels werden im Simulator überprüft. Damit steht ein komplettes in VHDL beschriebenes Modell des Prozessor-Kernels zur Verfügung. Dieses ist auch für einen späteren Einsatz des Kernels in Applikationen wichtig, da hiermit eine effiziente Simulation auch komplexer Strukturen einschließlich der für den Kernel geschriebenen Software ermöglicht wird. Eine solche Simulation auf Gatterebene würde bald den Rahmen erträglicher Simulationszeiten sprengen.

Nach der Beseitigung von Entwurfsfehlern werden die VHDL-Verhaltensmodelle mit AUTOLOGIC synthetisiert. Die Synthese erlaubt durch steuernde Parameter eine zielgerichtete Optimierung auf gewünschte Eigenschaften. So konnte im vorliegenden Fall die Durchlaufzeit der ALU von zunächst über 70 ns auf unter 20 ns gedrückt werden, ohne daß die benötigte Fläche und Gatterzahl wesentlich anstieg. Hierfür war allerdings Rechenzeit von über 30 Stunden erforderlich.

Wegen Fehlens der Geometriedaten konnte das Layout bisher nicht generiert werden. Dies erfolgt, wenn von EUROCHIP das "Back - End" der ES2 1,0 μm Bibliothek vorliegt.

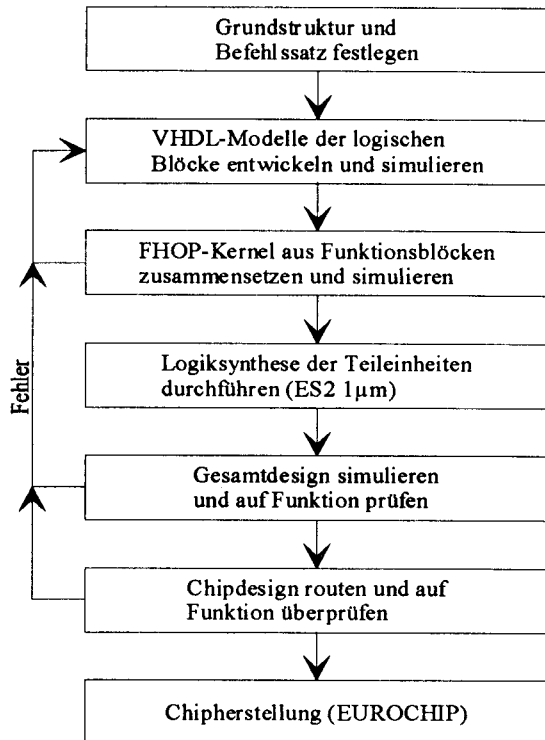


Bild 2: Top-Down-Design (verkürzt)

Der FHOP-Kernel

Für die vorgesehene Anwendung mußte eine Architektur gewählt werden, die sich durch folgende Eigenschaften auszeichnet:

- Regularität im Aufbau des Datenpfads,
- 16-Bit ALU mit integrierter Shiftoperation,
- byteorientierte Bussysteme,
- umfangreicher Befehlssatz (CISC-Konzept)
- einfaches Taktschema, kein Pipelining
- externe Schnittstelle mit Hold, Waitecycle-Steuerung und Interrupt,
- großer Adreßraum von 64 kByte,
- separater I/O-Raum.

Nicht zuletzt wurde die Architektur so gewählt, daß sie sich für eine Erläuterung der Prinzipien eines Mikroprozessors in der Lehre eignet. Das erzielte Resultat ist in der Leistungsfähigkeit am ehesten zwischen einem 8051-Controller und einem 8088-Mikroprozessor anzusiedeln. Auf eine Kompatibilität im Objektcode wie im Befehlssatz wurde bewußt verzichtet. Eine kommerzielle Auswertung wird nicht angestrebt, jedoch soll der Kernel der akademischen Welt lizenzfrei zugänglich gemacht werden.

Das Blockschaltbild (Bild 3) zeigt die Struktur des Mikroprozessors. Sämtliche Operationen sind intern auf ein 16-Bit Datenformat ausgelegt. Der externe Busanschluß ist byteorientiert, um den externen Speicher mit minimalem Aufwand adaptieren zu können. Es steht ein Adreßbereich von insgesamt

64 kByte und ein getrennter I/O-Bereich von 256 Byte zur Verfügung.

Der Prozessor besitzt eine 16-Bit ALU. Die ALU verfügt über insgesamt 16 Funktionen, wie sie in Tabelle 1 dargestellt sind. Neben den üblichen arithmetischen und logischen Funktionen ist auch eine Shiftoperation implementiert. Die Ergebnisse stehen nach 20 ns maximaler Laufzeit zur Verfügung. Die ALU wurde mit etwa 1000 Gattern realisiert. Neben den Rechenergebnissen werden die in Tabelle 2 dargestellten Flags generiert.

Tabelle 1: Funktionen der ALU

Steuerleitungen	Logische Funktionen (mc = 0)	Arithmetische Operationen (mc = 1)
000	$Y = 0$	$Y = A$
001	$Y = A \text{ and } B$	$Y = A + 1$
010	$Y = A \text{ or } B$	$Y = A - 1$
011	$Y = A \text{ exor } B$	$Y = A + B$
100	$Y = A \text{ shift left}$	$Y = A + B + CY$
101	$Y = A \text{ shift left } CY$	$Y = A - B$
110	$Y = A \text{ shift right}$	$Y = A - B - CY$
111	$Y = A \text{ shift right } CY$	$Y = (-A) - 1$

Der Registersatz besteht aus sechs 16-Bit Registern. Vier dieser Register werden als Universalregister zur Verfügung gestellt (A ... D), eines der Register wird als Stackpointer, ein Register als Programmzähler genutzt. Weiter besitzt der Registersatz ein 8-Bit breites Flagregister und ein 8-Bit breites Zwischenregister.

Tabelle 2: Flags der ALU

M	Minus
AC	Auxiliary Carry
HC	Half Carry
CY	Carry
Z	Zero
OV	Overflow
P	Parity
HP	Half Parity

Um in der Adressierung möglichst flexibel zu sein, wird die ALU zur Adreßberechnung wie auch zum Inkrementieren des Programm-Pointer-Registers verwendet. Ein eigentlicher Programm - Zähler existiert damit nicht.

Es sind insgesamt vier unterschiedliche Adressierungsarten realisiert:

- Registeradressierung (z.B. MOV r1,r2) die Adresse ist implizit im OP-Code enthalten
- direkte Adressierung (z.B. LDA adr) die Speicheradresse steht im zweiten und dritten Befehlsbyte
- indirekte Adressierung (z.B. LDA M) die 16-Bit Speicheradresse wird aus Register B entnommen
- Direktoperand-Adressierung (z.B. LDI konst) der Operand steht im zweiten und dritten Befehlsbyte

Bei dem Entwurf der Busschnittstelle des FHOP-Kernels zur externen Welt müssen die Gegebenheiten des ASIC-Entwurfs berücksichtigt

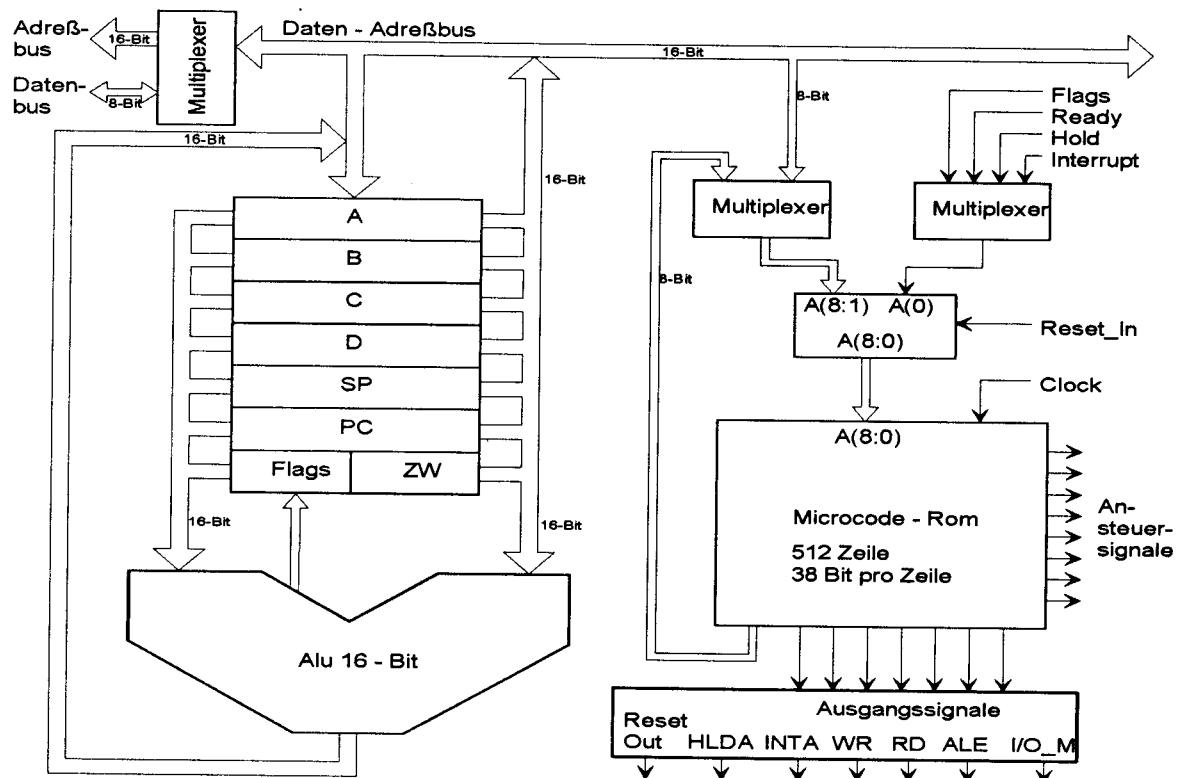


Bild 3: Blockschaltbild FHOP

werden. Da Speicher und Peripherieeinheiten sehr unterschiedlicher Zugriffszeit angeschlossen werden sollen, sind übliche Maßnahmen wie READY - Handshake zur Einfügung von Waitecyclen und ein hochohmig schalten der Prozessorsignale (HOLD) zu implementieren.

Die READY-Funktion wird im Prozessorzyklus berücksichtigt. Im Fetchzyklus sowie in allen Befehlen mit Peripheriezugriff ist dazu eine Abfrage der READY-Leitung integriert (Bild 4). Der Mikrocode wurde entsprechend gestaltet.

Ähnlich wie bei kommerziellen Prozessoren kann der Kernel über ein HOLD-Signal an der Schnittstelle hochohmig geschaltet werden. Auch dieser Zugriff erfolgt über einen Handshake und wurde durch die entsprechende Programmierung des Mikrocodes realisiert. Damit ist sowohl eine Multiprozessorfähigkeit als auch die Realisierung von selbständigen DMA-Controllern im Bussystem offen gehalten.

Die Steuerung besitzt eine Sammelinterruptleitung mit welcher, bei entsprechender Verwaltung durch einen externen Interruptcontroller, insgesamt 85 verschiedenen Interruptquellen unterschieden werden können.

Das Steuerwerk ist als klassische Mikroprogramm-Steuerung realisiert. Der Mikrocode, der speziell für den FHOP-Kernel entwickelt wurde, ist in einem auf dem Chip befindlichen ROM abgelegt. Gemäß der Intentionen bei der Entwicklung der Architektur wurde möglichst viel Steueraufwand in den Mikrocode verlagert, um ein kompaktes Design zu erreichen.

Der Mikrocode-ROM hat eine Größe von 512 Zeilen und 38 Bit pro Zeile. Er benötigt eine Chipfläche von etwa 1,4 mm² und wurde mit einem für die ES2 1,0µm - Bibliothek zur Verfügung stehenden ROM-Generator erzeugt.

Der Befehlssatz (vgl. Tabelle 3) umfaßt insgesamt 115 verschiedenen OP-Codes. Diese teilen sich in:

- 88 Ein-Byte Befehle,
- 3 Zwei-Byte Befehle und
- 24 Drei-Byte Befehle auf.

Faßt man verwandte Befehle zusammen (z.B. alle MOV-Befehle), bleiben noch 64 unterschiedliche Befehle übrig. Der Befehlssatz ist weitgehend verfügbaren Standardprozessoren ähnlich.

Das Zustandsdiagramm in Bild 4 zeigt den Fetchzyklus, wie er im Prozessor abgearbeitet wird. Dabei wird zu Beginn immer auf die Eingangssignale HOLD und INTERRUPT abgefragt. Dann beginnt der eigentliche Programmspeicherzugriff. Dieser Ablauf ist bei jedem Befehl gleich.

Der Fetchzyklus beginnt mit der Ausgabe des Program-Counters auf den Datenbus. Die

Peripherie, i.a. der Speicher wird adressiert und legt das erste Byte des Objektcodes auf den Bus. Das Byte wird eingelesen, dekodiert und gleichzeitig der Program-Counter inkrementiert. Bei Mehrbytebefehlen wird dieser Vorgang wiederholt. Anschließend wird der Befehl ausgeführt. Die Abfrage auf READY erfolgt beim Einlesen des OP-Codes. Der Zustand wird erst verlassen, wenn die READY-Bedingung logisch "1" ist.

Die HOLD-Bedingung wird in dem Zustand abgefragt, in dem der Program-Counter ausgegeben wird, ebenso der INTERRUPT. Erst anschließend werden beide Bearbeitungswege getrennt. Der HOLD-Zustand wird erst nach einem Wartetakt, der ein Umschalten der Treiber in dem hochohmigen Zustand ermöglicht, quittiert. Nach Beendigung der HOLD-Bedingung wird der Fetchzyklus neu begonnen.

Auf die INTERRUPT-Bearbeitung soll hier nicht weiter eingegangen werden.

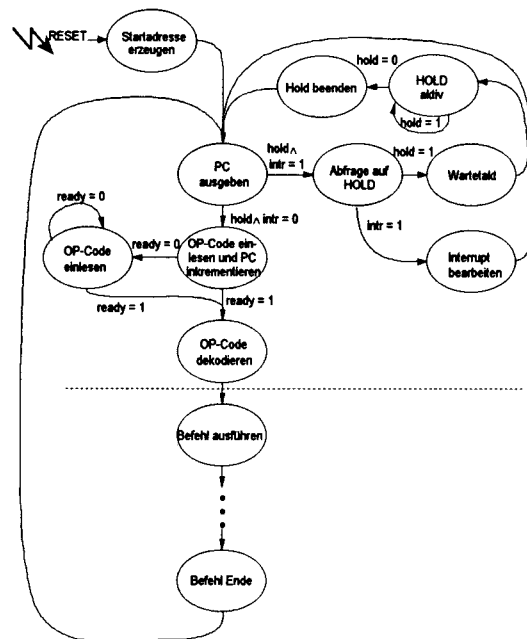


Bild 4: Zustandsdiagramm Fetchzyklus

Plazierung des Kernels auf einem Testchip

Um den Prozessor-Kernel auf seine Funktion hin überprüfen zu können, wird ein Testchip entwickelt und gefertigt. Er enthält außer dem FHOP-Kernel noch weitere Komponenten. Dazu zählen eine interne und externe Busschnittstelle, eine RAM-Makrozelle und ein Pad-Interface für die Ansteuerung der Anschlüsse des ASIC's.

An externen Komponenten werden nur noch ein EPROM oder sonstiger Speicher benötigt, ferner ein GAL oder PLD zur READY-Steuerung. Damit ist der Test des Kernels in allen Funktionen

durchführbar. Gleichzeitig liegt damit für den späteren Anwender ein übersichtliches Beispiel für die Integration des Kernels in eigene Anwendungen vor.

Bei der Entwicklung von ASICs mit Prozessorfähigkeiten platziert man den FHOP - Kernel als fertige Makrozelle auf dem zu entwickelnden Chip und baut die jeweils spezielle Anwendungsschaltung daneben auf. Hinzu kommen Speicher (RAM, ROM) nach Bedarf, sowie Schaltungen für Ports und Schnittstellen. Alle Einheiten werden miteinander verbunden. Damit ist eine anwendungsspezifische Systemintegration auf einem Chip möglich geworden (Application Specific System on Chip, ASSOC).

Der Prozessor-Kernel steht als getestetes VHDL - Modell zur Verfügung. Er kann als Symbol in Schaltpläne eingefügt und effizient auf Systemebene simuliert werden. Eine Abbildung auf Grundelemente einer Bibliothek (Netzliste) ist ebenfalls verfügbar, wodurch ein Softmakro mit definierten Eigenschaften gegeben ist. Mit relativ geringem Aufwand kann von diesen Grundelementen auf verfügbare Bibliotheken abgebildet werden (Hardmacro ES2 - 1,0 μm). Durch die Top - Down - Vorgehensweise wird damit eine relative Unabhängigkeit von der verwendeten Technologie sichergestellt und damit eine längere Lebensdauer des Designs erwartet.

Tabelle 3: Befehlssatz

Transferbefehle	Arithm. Befehle	Logische Befehle	Shift - Befehle
MOV r1,r2	INC r1	ANA r1	RAL
MOV F,A	DEC r1	ANI konst	RAR
SWP	ADD r1	ANA M	RLC
LDI konst	ADC r1	ANA adr	RRC
LDA M	ADI konst	ORA r1	
LDA adr	ACI konst	ORI konst	
STA M	ADD M	ORA M	Sprung - Befehle
STA adr	ADC M	ORA adr	JMP adr
PSH r1	ADD adr	XRA r1	JMC adr
PSH F	ADC adr	XRI konst	JMZ adr
POP r1	SUB r1	XRA M	JMO adr
POP F	SUC r1	XRA adr	
	SUI konst	INV	Unterp.- Befehle
Sonst. Befehle	SCI konst	CMP r1	CAL adr
NOP	SUB M	CMI konst	RET
DIS	SUC M	CMP M	
ENI	SUB adr	CMP adr	I/O Befehle
HLT	SUC adr		PIN port
SWI n			POT port

Zusammenfassung

Es wurde der Kernel eines 16-Bit Mikroprozessors als VHDL-Modell entwickelt, simuliert und auf eine ES2 1,0 μm -Technologie synthetisiert. Der Prozessor-Kernel ist durch hohe Regularität und insgesamt einfachem Aufbau gekennzeichnet.

Der Datenpfad enthält die Busstruktur, den Registersatz und die ALU. Die Steuerbaugruppe schließt das Mikrocode-ROM und den Befehlssatz ein. Die ALU wurde spezifisch entworfen und aus der VHDL-Verhaltensbeschreibung effektiv synthetisiert. Das Verhalten wurde auf VHDL wie auf Gatterebene überprüft. Eine Taktfrequenz von 33 MHz wird erwartet. Damit steht nun neben dem VHDL-Modell die simulierte Netzliste des gesamten Prozessor-Kernels zur Verfügung. Die Leistungsdaten des FHOP - Kernels sind in Tabelle 4 zusammengestellt.

Dem späteren Einsatz des Mikroprozessor-Kernels als Standardzellen-Makro steht damit nichts mehr im Wege. Dies kann entweder als VHDL-Modell, als Gatter-Modell (Softmakro) oder als fertig platzierte Standardzellen (Hardmakro) erfolgen.

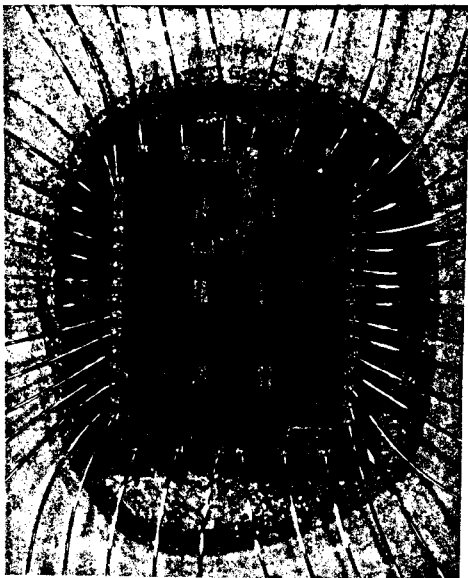
Der Prozessor-Kernel wird auf einem Testchip mit notwendigen Peripherieblöcken wie einem RAM und einer Busschnittstelle derzeit erprobt. Das Ziel dieser Entwicklung ist es, den Prozessorkernel als Standardzellen-Makro für weitere Forschungs- und Entwicklungsarbeiten an der FH Offenburg nutzbar zu machen. Damit können im ASIC-Design-Center der FH zukünftig sehr komplexe Logikschaltungen in einem Halbleiterchip integriert werden.

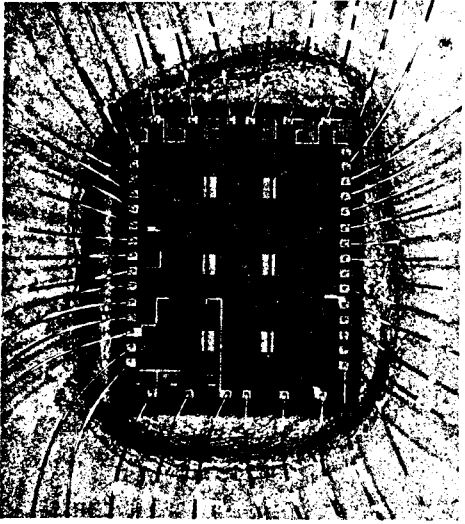
Tabelle 4: Leistungsmerkmale

Busstruktur	- Daten/Adreßbus intern 16-Bit breit - Adreßbus extern 16-Bit breit - Datenbus extern 8-Bit breit
Adreßraum	- Memorybereich 64kByte - I/O-Bereich 256 Byte
ALU	- 16-Bit Operationen - 8 Flags - Durchlaufzeit ≈ 20 ns
Registersatz	- 6 Register 16-Bit breit - 1 Flagregister 8-Bit breit - 1 Zwischenregister 8-Bit breit
geschätzte Chipfläche	4 ... 5 mm ²
Geschwindigkeit	- Taktfrequenz 33 MHz - minimale Befehlszykluszeit ≈ 8 MIPS
Peripherie-steuer-signale	- Unterbrechungs-Anforderung - READY-Eingang - HOLD-Anforderung
Befehlssatz	- 88-Einbyte Befehle - 3-Zweibyte Befehle - 24-Dreibyte Befehle

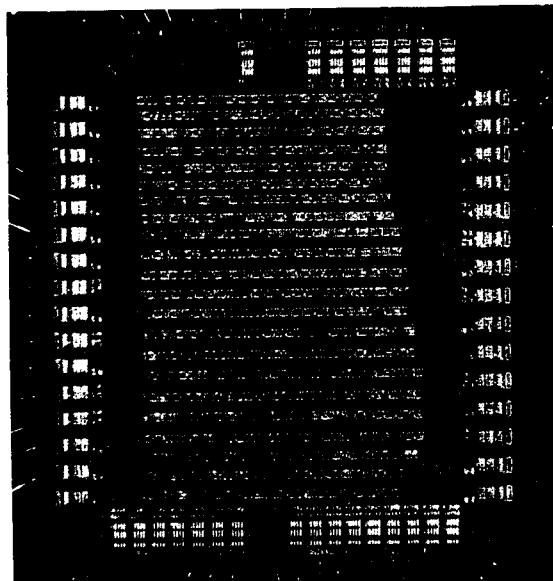
Literaturverzeichnis

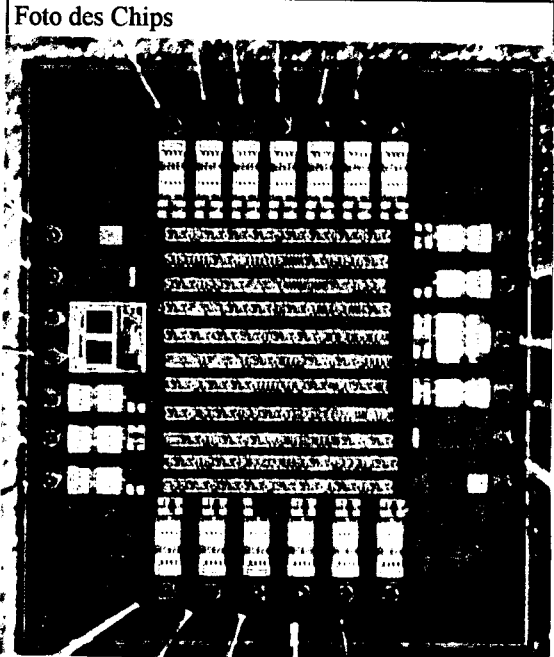
- [1] Zimpfer, Fritz: *Entwicklung des Datenpfads eines 16-Bit Mikroprozessor-Chips mit VHDL*, Diplomarbeit an der Fachhochschule Offenburg, 1994
- [2] Gieringer, Thomas: *Entwicklung der Steuerstruktur eines 16-Bit Mikroprozessor-Chips mit VHDL*, Diplomarbeit an Fachhochschule Offenburg, 1994
- [3] Jansen, Dirk: *Vorlesungsskript Schaltungstechnik*, Vorlesung an der Fachhochschule Offenburg, 1992
- [4] Mentor Graphics Corporation: *Dokumentation der CAE-Tools*, 1992
- [5] D.D. Gajski, et.al: *Computer Architecture, Tutorial*. IEEE Catalog Nr. EH0294 - 3, Computer Society Press, IEEE Washington D.C., 1991
- [6] S.B. Furber: *VLSI RISC Architecture and Organization*, Marcel Dekker, Inc New York, 1989
- [7] Spruth, William G.: *The Design of a Microprocessor*, Springer Verlag, Berlin, 1989

Bezeichnung		
<h1>ZF-Verstärker/Costas Loop</h1>		
Funktion: Der Bipolar - Array - Chip enthält einen mehrstufigen regelbaren Zwischenfrequenzverstärker für ZF bis etwa 20 MHz sowie 3 Multiplizierer zu Aufbau eines synchronen Costas - Loop Phasendemodulators. Die Schaltung ist vorgesehen zu Einbau in den FHO - GPS - Empfänger und enthält praktisch den gesamten analogen Schaltungsaufwand zur Demodulation des ZF - Signals.		
Technologie: B 500 A Bipolar - Array (AEG), 1 Metallisierungsebene zu spezifizieren		
Entwicklungsbeginn: Juli 1990	Lieferung: November 1990	
Anzahl: 30	gut /schlecht: 24/6	geprüft: Kern 12/90, Mackensen 1992
Entwurfshistorie: <ol style="list-style-type: none"> 1. Schaltungsentwurf auf Spice - Basis (Michael Kern) Juli 1990 2. Platzierung und Routen von Hand mit CHIPGRAPH gemäß AEG Designrules (Aufwand ca 2 Wochen) 3. Prüfen des Layouts durch FH - Ulm 4. Designabgabe September 1990 (Reinke, Kern) 5. Lieferung November 1990 6. Erste Tests Dez 90 (Kern) : erfolgreich 7. Komplette Vermessung: im Rahmen der Integration in den GPS-Analogteil (Mackensen 1992/93) 		
Bemerkungen: Sehr komplexer Chip mit mehreren OPs, 3 Multiplizierern und einem mehrstufigen regelbaren ZF - Verstärker. Spezifizierte Daten wurden erreicht. Übereinstimmung mit Spice - Simulation ist gut.		Foto des Chips 
Dimensionen: ca 3,5 x 4,5 mm ²	Komplexität: hoch, 90 % voll	
Package: CLCC 44		
Anschlußzahl: 44 Pin		

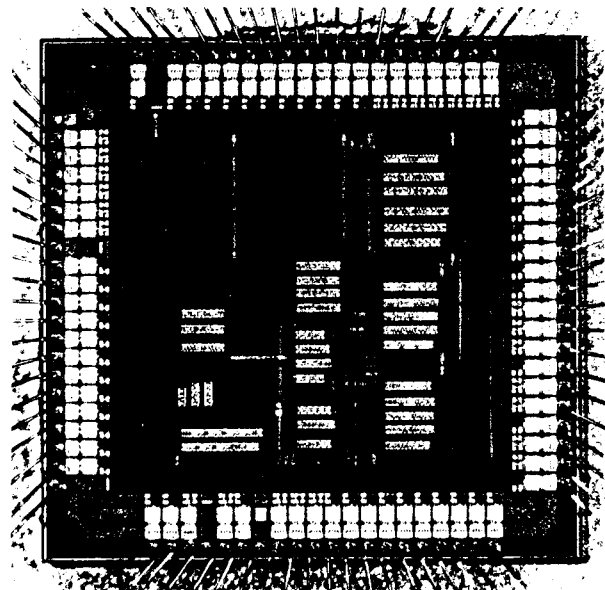
Bezeichnung		
<h1>Synchrondemodulator</h1>		
Funktion: Der Chip enthält alle Funktionen, die zum Betrieb eines induktiven Weg - Aufnehmers erforderlich sind, u.a. einen Sinus - Oszillator, einen Treiberverstärker für die Erregerspule (12 V), einen empfindlichen Vorverstärker sowie einen 4 Quadranten - Multiplizierer und einen Tiefpaß. Zum Aufbau eines Wegaufnehmers sind außer der Spule nur noch wenige externe Elemente wie Widerstände und Kondensatoren erforderlich. Die Schaltung arbeitet im NF Bereich bis etwa 40 kHz.		
Technologie: B 500 a Analog Bipolar Array (AEG), entworfen mit SPICE und CHIPGRAPH auf MENTOR - CAE		
Entwicklungsbeginn: Juli 1990	Lieferung: November 1990	
Anzahl: 30	gut /schlecht: 25/5	geprüft: Keßner 12/1990
Entwurfshistorie: <ol style="list-style-type: none"> 1. Klassischer analoger Entwurf unter Verwendung von SPICE (Keßner) 2. Platzierung und Routen von Hand in einer Ebene mit CHIPGRAPH (Keßner) 3. DRC bei FH - Ulm 4. Designabgabe September 1990 (Keßner, Reinke) 5. Lieferung November 1990 6. Test Dez 1990 (Keßner, Reinke) 		
Bemerkungen: Chip arbeitete auf Anrieb mit guten elektrischen Eigenschaften. Übereinstimmung mit SPICE - Simulation gut. Chip auf SMD - Platine montiert und in Aufnehmer integriert . Veröffentlicht in der ELEKTRONIK 14/92 und ELEKTRONIK - CAE Sonderheft 94. Erster Analog - Chip an der FHO !		
Dimensionen: 3.5 x 4.5 mm ²	Komplexität: hoch, 90 % voll	
Package: CLCC 44		
Anschlußzahl: 44 Pin		
Foto des Chips		
		

Bezeichnung		
<h1>GPS - Kodegenerator</h1>		
Funktion: Der Chip vereinigt alle notwendigen Schaltkreise zur Generierung der im GPS - System verwendeten Gold - Codes . Die Codes können selektiert, zurückgesetzt und gestartet werden. Der Kode kann einem Synchronsignal in der Phase nachgeführt werden, wobei die zeitliche Verschiebung bestimmt werden kann. Der Kode kann ferner gedithert, d.h. um +/- 1/2 Chip - Länge relativ zu einem vorgegebenen Phasenwert verschoben werden. Der Kodegenerator enthält mehrere größere Zählerketten und rückgekoppelte Schieberegister sowie umfangreiche Random - Logik. Die Schaltungen sind nur in geringem Maße regulär. Der Chip ersetzt 3 FPGA - Bausteine Xilinx 3200 /3400 und weist eine Gatterkomplexität von ca 7000 Gattern auf.		
Technologie: ES 2 1.5 um CMOS - Prozeß via EUROCHIP auf MENTOR 7.05 CAE		
Entwicklungsbeginn: 1991	Lieferung: 1. EUROCHIP # 33 August 92 2. EUROCHIP #48 Febr. 93	
Anzahl: 10	gut /schlecht: 9/1	geprüft: Wagner Juli 93
Entwurfshistorie:		
1. Schaltungsentwurf	Fiesel, H.	2 x Xilinx 3200 1 x Xilinx 3400
2. Überarbeitung zum ASIC	Behrens, H.P	IMS - Lib, später ES 2 1.5 um
3. Erstellung Fertigungsfiles	Feißt/Mackensen	EUROCHIP Run #33
4. Überarbeitung Vss - Netz	Mackensen	EUROCHIP Run #48
5. Test der Prototypen	M. Wagner	Juli 1993
Bemerkungen: Der erste Fertigungslauf war wegen eines Fehlers im ES 2 - Entwurfskit, durch den die Spannungsversorgung nicht geroutet wurde, nicht erfolgreich. Der Fehler wurde für den 2. Lauf umgangen, sodaß jetzt ein voll funktionsfähiger Chip vorliegt.		
Dimensionen: 4,1 x 4,7 mm ²	Komplexität: ca 7000 Gater	
Package: CLCC 68		
Anschlußzahl: 68 Pin		

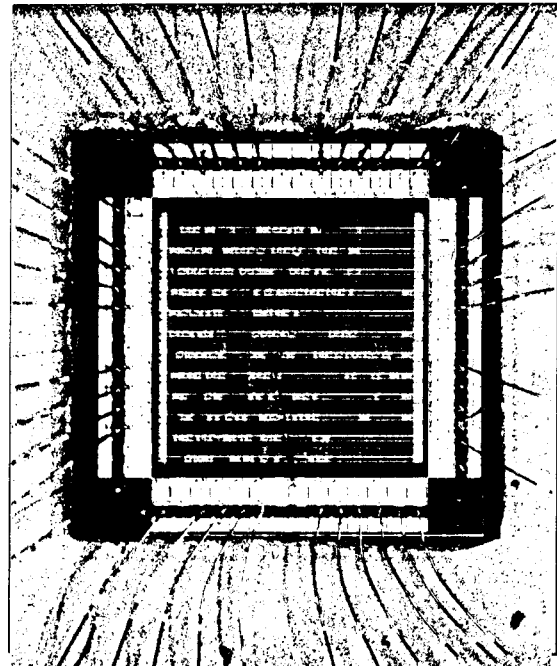


Bezeichnung	
<h1>Elektronischer Würfel 2</h1>	
Funktion: Elektronischer Würfel mit Anzeige der Augenzahl über 7 LEDs, die direkt angesteuert werden. Der Chip wird durch die Start - Taste erst aktiviert und schaltet sich automatisch nach etwa 10 Sekunden wieder ab. Ein Uhrenquarz dient als Zeitbasis. Der Würfel "rollt" langsam aus, d.h. die Augenzahl wird nicht sofort erreicht, sondern erst nach einer Ausrollzeit von etwa 2 Sekunden. Mit dem Rollen wird ein "Klick" generiert, der durch einen Piezo- Summer hörbar gemacht wird. Das Ergebnis wird mit einem aus 2 Klängen bestehenden Ton signalisiert. Bei den Ziffern "6" und "1" wird jeweils eine kurze Melodie gespielt. Der Chip arbeitet mit 3 V aus einer Lithiumbatterie und nimmt extrem wenig Strom auf. Es werden nur eine Taste, der Uhrenquarz, der Piezo -Speaker und die LEDs als externe Bauteile benötigt.	
Technologie: ES 2 1.5 um CMOS Standardzellendesign auf Mentor 7.05 CAE - Tools mit EUROCHIP - ES2 Design Kit	
Entwicklungsbeginn: Juli 1993	Lieferung: EUROCHIP Run #66 Sept. 93
Anzahl: 50	gut /schlecht: 46/4
	geprüft: Schweiker, Dez. 93
Entwurfshistorie: 1. Basiert auf ursprünglichem diskreten Aufbau und IMS - Version (Bernd Reinke, 1991) 2. Neuentwurf im Rahmen einer Studienarbeit Juni 1993 (Schweiker) 3. Designabgabe Juli 1993 (Feißt) 4. Lieferung Muster durch EUROCHIP September 1993 5. Geprüft und auf SMD - Platinen montiert: Dez 93/Jan 1994	
Bemerkungen: Entwurfszeit betrug nach funktioneller Vorsimulation in Hochsprache nur 2 Wochen! Chip war auf Anhieb voll funktionsfähig.	
Dimensionen: 2.6 x 2.6 mm ²	Komplexität: ca 4000 Gatter
Package: CLCC 64	
Anschlußzahl: 64 Pin	
	Foto des Chips 

Bezeichnung		
<h1>Hardmacro - Chip</h1>		
Funktion: Der Chip enthält eine Sammlung von Hardmacros, die jeweils für sich geroutet und als eigene Macros platziert wurden. Damit wurden die hierarchischen Routemöglichkeiten der Mentor - Software ausprobiert. Als Zellen wurden realisiert: ALU 8 bit, Schieberegister 8 bit, Zählermodule 4bit und 8 bit, Multiplexer und Decoder. Der Chip wurde so konfiguriert, daß alle einzelnen Macros für sich getestet werden können.		
Technologie: ES 2 1.5 um Standardzellen, hierarchisch geroutet, von Hand platziert auf Mentor 7.05 CAE		
Entwicklungsbeginn: Oktober 1992	Lieferung: EUROCHIP RUN #62 Juli 93	
Anzahl: 10 Muster	gut /schlecht: noch nicht geprüft	geprüft: siehe unten
Entwurfshistorie: 1. Entwurf im Rahmen einer Diplomarbeit (Detlef Pouw) 2. Große Probleme mit der Software, deshalb Beschränkung auf einfache Funktionen 3. Teile des Chips müssen Full Custom realisiert werden, DRC erforderlich 4. Design endgültig abgegeben März 1993 (Feißt, Mackensen) 5. Lieferung im Juni 1993		
Bemerkungen: Test konnte noch nicht durchgeführt werden wegen Problemen mit der Pin - Belegung und dem Sockel. Verhalten einzelner Macros noch ungeklärt. Stromaufnahme normal. Bearbeiter nicht mehr verfügbar.		
Dimensionen: 4.9 x 4.7 mm ²	Komplexität: ca 3500 Gatter	
Package: CLCC 68		
Anschlußzahl: 68 Pin		



Bezeichnung		
<h1>Elektronischer Würfel 1</h1>		
Funktion:		
<p>Elektronischer Würfel mit Anzeige der Würfelaugen durch 7 LEDs. Enthält Taktgenerator mit Uhrenquarz, Start/Stop - Logik und eine Ausrolllogik, d.h die Augen werden nicht sofort angezeigt, sondern erst nach einem Ausrollvorgang von etwa 2 sek.</p> <p>So konfiguriert, daß nur minimale externe Bauteile benötigt werden.</p> <p>Versorgung mit 3 V aus Lithiumbatterie, sehr geringer Stromverbrauch.</p>		
Technologie:		
IMS Gatearray , Gate - Forrest 2.5 um		
Entwicklungsbeginn:	Lieferung:	
1990	Sommer 1991	
Anzahl:	gut /schlecht:	geprüft:
20	16/4	Reinke
Entwurfshistorie:		
<ol style="list-style-type: none"> 1. Basiert auf diskretem Aufbau mit SMD - Logik (Reinke) 2. Integriert alle Einzelbausteine und zusätzlich Ausrollverhalten (Reinke) 3. Entwurf als Gatearray mit IMS - Designkit (Reinke) 4. Fertigung beim IMS (Individualisierung), erste Version wies Kurzschluß in der oberen Zellreihe auf 5. Durch Korektur an 2. Metallisierungsmaske konnte Kurzschluß behoben werden (Nacharbeit IMS) 6. Chips schließlich einwandfrei in Funktion und auf SMD - Platinen montiert (Reinke) 		
Bemerkungen:		
<p>Fehler im Designkit führte zu Kurzschluß in der obersten Plazierungsreihe. Konnte durch Nacharbeit im IMS an noch verfügbaren Wafern beseitigt werden. Schaltung arbeitete danach einwandfrei.</p> <p>Erster an der FHO entwickelte CMOS - Chip!</p>		
Dimensionen:	Komplexität:	
4,5 x 4,5 mm ²	ca 2000 Gatter, 50 % voll	
Package:		
CLCC 44		
Anschlußzahl:		
44 Pins		



Fachhochschule Ravensburg Weingarten

PIO-Baustein mit 48 freiprogrammierbaren Pins

Verfasser : Ender Karl-Heinz

Betreuer : Dr.Ing. Ludescher

Inhalt

1. Überblick
2. Verwendungszweck
3. Aufbau
4. Zusammenschaltung
5. Schreib/Lesezyklen

1.Übersicht

- PIO -Baustein

- Jeder PIO-Baustein besitzt 6 * 8 freiprogrammierbare Pins (Eingang oder Ausgang).
- Es können bis zu 8 PIO-Bausteine zusammenschaltet werden.
- Durch den Steuerbefehl "Ausgeben" können alle Pins an allen 8 PIO-Bausteinen gleichzeitig ihre Zustände ändern.
- Durch den Steuerbefehl "Einlesen" können alle Signale an den Pins an allen 8 PIO-Bausteinen gleichzeitig in Einleseregistern zwischengespeichert werden.
- Ein Anschluß an die Centronicsschnittstelle oder an den PC-Bus ist ohne größeren Aufwand möglich.

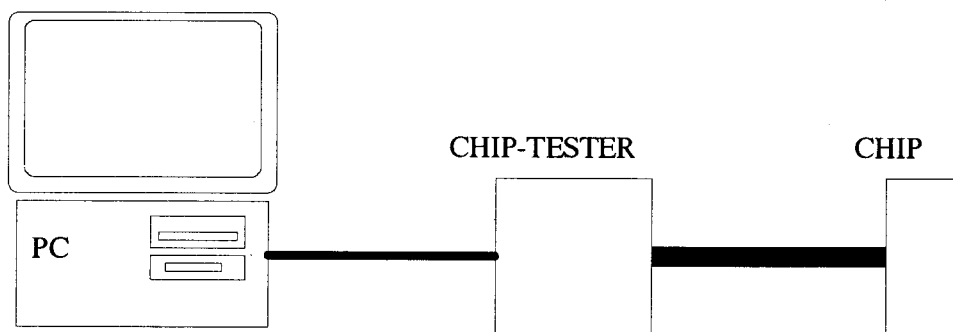
2.Verwendungszweck

Als Verwendungszweck des PIO-Bausteins war bei der Entwicklung der Aufbau eines einfachen Chip-Testers vorgesehen. Durch das Zusammenschalten von bis zu 8 PIO-Bausteinen können dann Bitmuster mit einer Breite von 384 Bit erzeugt und auch eingelesen werden.

Durch den Steuerbefehl "Ausgeben" ist es möglich, das gesamte Bitmuster zu einem bestimmten Zeitpunkt zu ändern, ohne daß unerwünschte Zwischenzustände an den Ports auftreten.

Der Steuerbefehl "Einlesen" erlaubt es, zu einem bestimmten Zeitpunkt alle Signale an den 384 Pins in die Einleseregister zwischenzuspeichern. Damit wird vermieden, daß sich Änderungen während des Einlesens der bis zu 48 Ports auf das Ergebnis auswirken.

Für den Chip-Tester sollte noch eine Software entwickelt werden, die es ermöglicht, direkt aus dem Ergebnis von Simulationsprogrammen (Hilo oder Quicksim) die entsprechenden Bitmuster zu erzeugen. Dadurch wäre ein IC-Test mit den Simulationsdaten möglich.



3.Aufbau des Chips :

Gehäuse

84 Pin PLCC oder CLCC

Master GF4G1

72 Pad-cells

22 950 Site-Cells

Padring

56 Bidirektionale Treiber (12mA)

-8*6 Ports

-8 Bit Datenbus

6 Eingänge

-Steuersignale (WR,RD,A0,EN,RES)

-Nandtree (TREEIN)

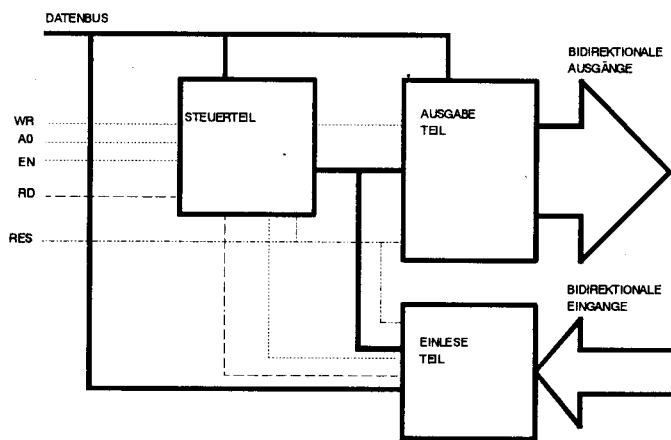
1 Ausgang

-Nandtree (TREEOUT)

3 Eingänge mit Pulldown

-Chipadresse (CS5A,CS6A,CS7A)

Core :



Steuerteil

-Adressspeicherung , Adressdecodierung , Auswahl der Bausteine

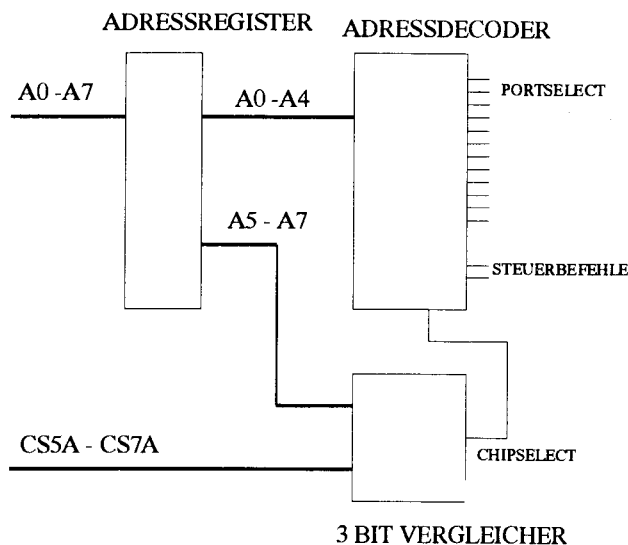
Ausgabeteil

-Zwischenspeichern der Daten , Ansteuerung der bidirektionalen Treiber

Einleseteil

-Speichern der Daten an den Ports , Ansteuern der Datenbustreiber

Übersicht der Gesamtschaltung



Adressregister :

- Das Adressregister speichert die Adresse bei der steigenden Flanke von WR-Signal und dem A0-Signal auf Low.

Adressdecoder :

- Aus den Adressen A0 bis A4 werden die Portselectsignale PS1-PS12 decodiert, welche die Register der Ports im Ausgabeteil und im Einleseteil ansteuern.
- Es werden außerdem noch die Steuerbefehle ,Ausgeben und Einlesen im Adressdecoder erzeugt.

3 Bit Vergleicher :

- Vergleich der Signale CS5A-CS7A mit A5-A7 zum Auswählen eines von 8 möglichen Chips.

Belegung der Adressregister

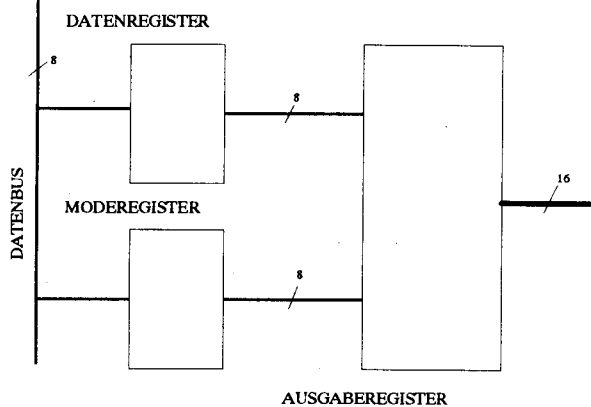
	Writezyklus	Readzyklus
A0	0 = Datenregister 1 = Moderegister	0 = Einleseregister 1 = Ports direkt
A1	Portadresse 0	
A2	Portadresse 1	
A3	Portadresse 2	
A4	Steuerbefehl 0=Adressen	1 = Steuerbefehle
A5	Chipadresse (Vergleich mit CS5A)	
A6	Chipadresse (Vergleich mit CS6A)	
A7	Chipadresse (Vergleich mit CS7A)	

Mit dem Steuersignal A0 = Low wird das Adressregister selektiert . Die Adresse A0 unterscheidet bei einem Writezyklus zwischen Daten- und Moderegister .Bei einem Readzyklus unterscheidet A0 zwischen dem Einleseregister und dem Port direkt.

Mit Adresse A1 - A3 wird einer von 6 möglichen Ports ausgewählt.

Adresse A4 unterscheidet zwischen einem Steuerbefehl und einem Write- oder Readzyklus.

Aus den Adressen A5-A7 wird einer von 8 Chips selektiert.

Ausgabeteil

Jeder Port besitzt ein Daten-, Mode- und Ausgaberegister. Durch den Steuerbefehl "Ausgeben" werden sämtliche Daten der Daten- und Moderegister in das Ausgaberegister übernommen. Die Daten sind dann an den Ports gleichzeitig vorhanden.

Datenregister : (8 Bit)

- Zum Zwischenspeichern der Daten

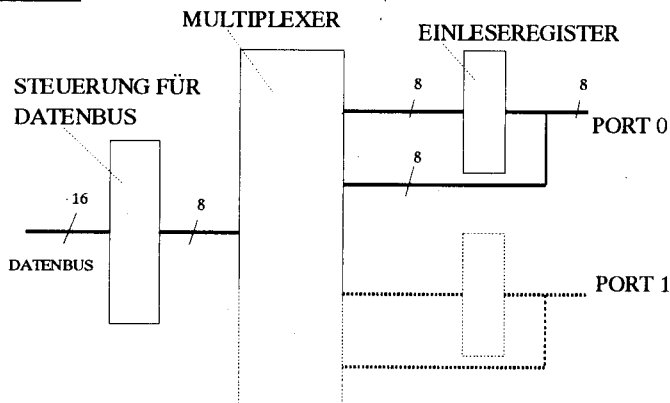
Moderegister : (8 Bit)

- Zum Zwischenspeichern der Modedaten (0 = Eingang , 1 = Ausgänge)

Ausgaberegister : (16 Bit)

- Ausgeben der Daten und Modedaten

- Ansteuern der bidirektionalen Treiber.

Einleseteil

Einleseregister :

- In den 6 Einleseregistern werden die Daten der Ports beim Steuerbefehl "Einlesen" gleichzeitig von allen Ports zwischengespeichert.

Multiplexer :

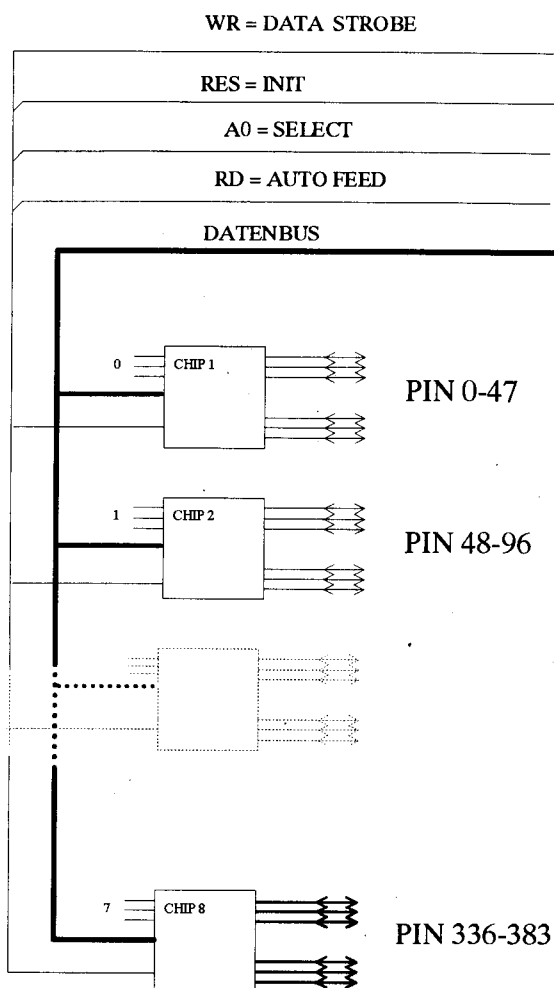
- Mit dem 12 * 1 Multiplexer kann jedes der 6 Einleseregister ausgewählt werden oder jeder der 6 Ports direkt ausgewählt werden.

Steuerung für Datenbus :

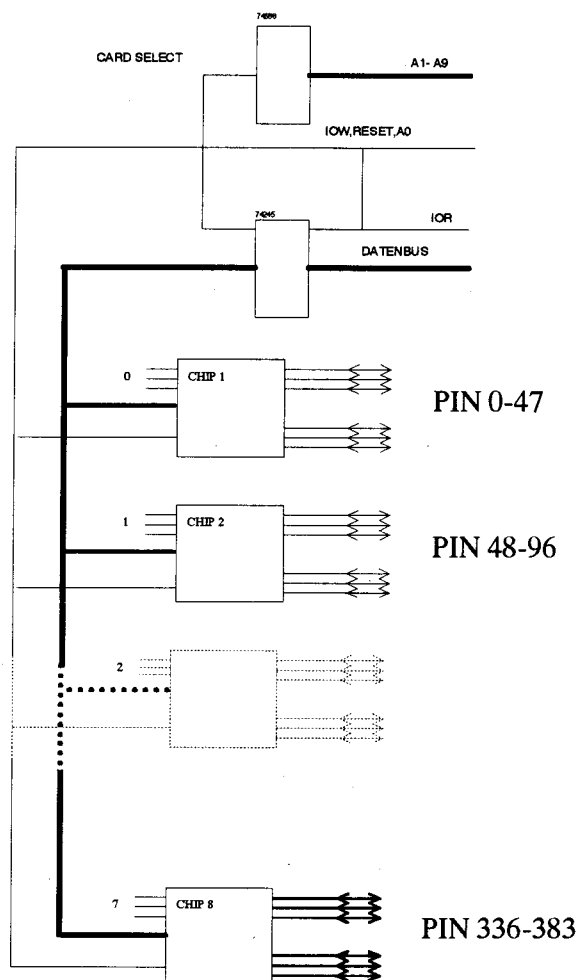
- Bei Low-Signal auf dem RD-Eingang werden die Daten des Multiplexers auf den Datenbus gelegt.

4. Zusammenschaltung

Anschluß an die Centronics



Anschluß an den PC-Bus



Es können bis zu 8 Chips parallel geschaltet werden.

Dadurch sind maximal 384 frei programmierbare Pins möglich.

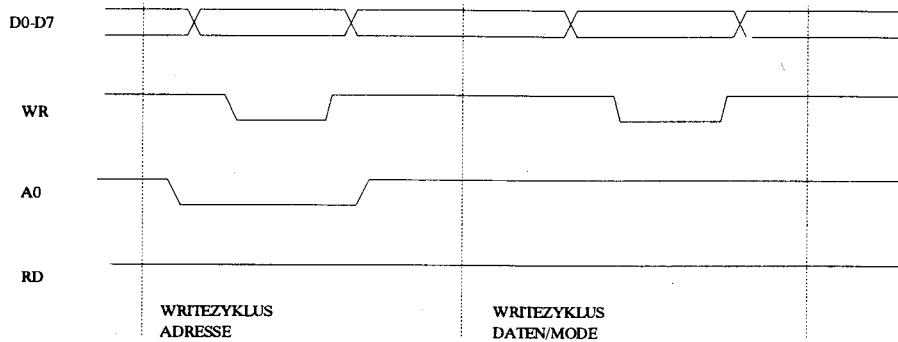
Beim Anschluß an den PC-Bus werden nur 2 Adressen belegt.

Jeder Pin kann mit 12 mA belastet werden.

Die maximale Geschwindigkeit ist abhängig von der Anzahl der Ports, die geändert werden müssen. Wird zum Beispiel ein Port geändert und nur ein Port direkt eingelesen, so sind auf dem PC-Bus oder auf der Parallelschnittstelle 4 Writezyklen und 1 Readzyklus nötig. Dadurch wird die Geschwindigkeit eines Chip-Testers kleiner als 1 MHz pro Erzeugtem Bitmuster. Dadurch ist nur ein sehr langsamer Test mit statischen Signalen möglich.

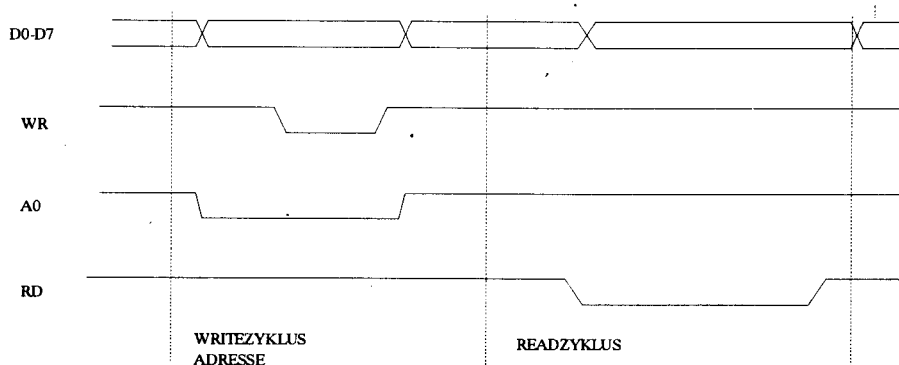
5.Schreib/Lesezyklen

Schreibzyklus in ein Register



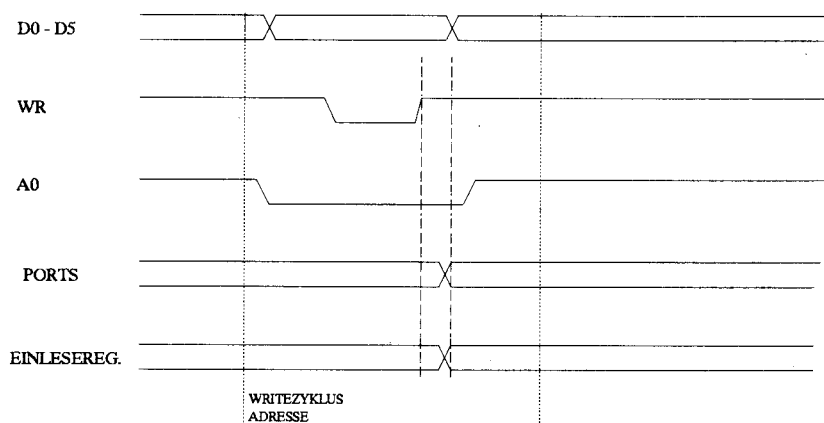
Bei der steigenden Flanke vom WR-Signal werden die Daten in das Adressregister oder Daten/Moderegister übernommen.
Das Signal A0 unterscheidet zwischen Daten und Adressen.

Lesezyklus aus einem Einleseregister oder von einem Port



Ist das RD-Signal auf 0 werden die Daten an den Datenbus gelegt .

Steuerbefehle



Bei den Steuerbefehlen handelt es sich nur um bestimmte Adressen.
Ausgeben "xxx1111" und Einlesen "xxx1000"

Simulation und Design eines Breitbandverstärkers in BiCMOS-Technik

T. Ehrenbeck, M. Schmid, G. Forster
Labor Schaltungsintegration der FH Ulm

Einleitung

Für die Herstellung integrierter Schaltungen haben sich zwei Technologien durchgesetzt, die Silizium-Bipolartechnik und die MOS-Feldeffekttransistor-Technik. Mit diesen beiden Technologien wird der größte Teil aller integrierten Bausteine hergestellt. Naheliegender ist daher die Entwicklung einer kombinierten Bipolar-CMOS-Technologie, die es auch ermöglicht, immer komplexere Schaltungen auf einem Chip zu integrieren. Ein weiterer Vorteil ist der geringere Entwicklungsaufwand bei Verwendung einer BiCMOS-Technologie, der in der industriellen Produktion einen großen Anteil an den Produktionskosten darstellt.

Zusätzliche Aufgaben entstehen bei der Entwicklung völlig neuer Schaltungskonzepte. Bei immer größeren Anforderungen an Geschwindigkeit, Packungsdichte und Schaltungs-Präzision ist es wahrscheinlich, daß die BiCMOS-Technologie in Zukunft an Bedeutung gewinnen wird.

Aufgabe war es einen Oszilloskop-Eingangsverstärker auf einem Chip zu integrieren. Da es bei einem Oszilloskop viel Anwendungen gibt, in denen Signale umgeschaltet werden müssen, und auch Digitalteile vielfach vorhanden sind, wurde für diese Anwendung ein BiCMOS-Prozeß gewählt. Es handelt sich um einen 2 µm BiCMOS-Prozeß von SGS-Thomson namens HF2CMOS.

Der HF2CMOS-Prozeß von SGS-Thomson

Eigenschaften

Der HF2CMOS-Prozeß von SGS Thomson ist ein Full-Custom-BiCMOS-Prozeß, der auch eine Standardzellen-Bibliothek bereitstellt und eine Vielfalt an Möglichkeiten zum Entwurf einer Schaltung bietet.

Es ist ein auf 14 Layern aufgebauter double-metal-Prozeß, der zusätzlich noch fünf weitere Optional-Layer zur Verfügung stellt. Hierunter sind auch ein zweiter Poly-, ein dritter Metall- und ein, an Stelle

des Buried-Layers einsetzbarer, für das Design und die Isolation eines vertikalen PNP-Transistors notwendiger, Niso-Layer. Die Versorgungsspannung kann, je nach Bedarf, zwischen $V_{DD} = 2.7, 5, 10$ und 11.5 V gewählt werden.

Übersicht :

- 14 Layer + 5 Optional-Layer
- 2 Metalle (drittes Metall optional)
- Poly / Poly - Kondensatoren (optional, 500 pF/mm^2)
- Poly / N-Wanne Kondensator (450 pF/mm^2 beim 10 V-Prozeß)
- low distortion poly Widerstand (optional)
- n - Isolation (NMOS-Bulk und Vertikal-PNP-Kollektor)
- Dünnoxid (100 \AA) für EEPROM
- Widerstände mit $R_s = 60 \text{ } \Omega/\text{sq} \dots 7000 \text{ } \Omega/\text{sq}$
- Versorgungsspannungen $2.5 \dots 11.5 \text{ V}$
- umfangreiche Analogbibliothek (Transistoren, D/A-Wandler, Verstärker usw.)

Leider sind noch nicht alle dieser Optional-Layer über EuroChip für Hochschulen verfügbar.

Charakteristiken des CMOS-Teils :

- $2 \text{ } \mu\text{m}$ (gezeichnet) Silizium-Gate N-Wannen-Technologie
- 475 \AA Dünnoxid
- Doppel-Diffusion des NMOS-Transistors
- Bulkanschlußisolation bei PMOS-Transistoren im Standard-Prozeß
- Bulkanschlußisolation bei NMOS-Transistoren mittels extra Maske, ansonsten liegt dieser auf Substratpotential

Charakteristiken des Bipolar-Teils :

- NPN mit $f_{Tmax} = 6 \text{ GHz}$
- Lateraler PNP mit $f_{Tmax} = 50 \text{ MHz}$
- Vertikaler, isolierter PNP mit $f_{Tmax} = 2.5 \text{ GHz}$
- Zener-Diode aus dem Basis-Emitter-Übergang eines NPN-Transistors

Transistoren

NPN-Transistor

Bild 1 stellt den Minimaltyp eines NPN-Transistors dar. Zu sehen ist hier der vertikale Aufbau (Buried-Layer-Kollektor zu Emitter), der in Zusammenhang mit der Oxidisation (schraffierte Fläche) eine hohe Grenzfrequenz ermöglicht. Die Stromverstärkung für $I_c = 100 \mu\text{A}$ liegt bei 90-150, halbe Stromverstärkung für $I_c = 5 \text{ mA}$. Die Transitfrequenz für $I_c = 2 \text{ mA}$ liegt bei 6 GHz. Der Flächenbedarf dieses Transistors beträgt ca. $600 \mu\text{m}^2$.

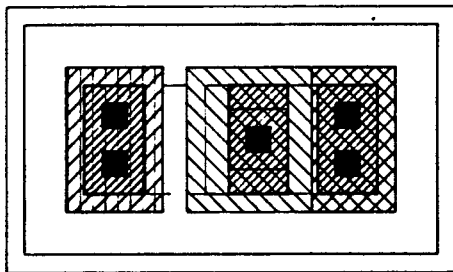
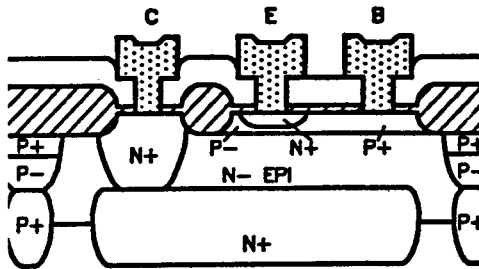


Bild 1 : Querschnitt und Layout eines NPN-Transistors

PNP-Transistor, vertikal

Um einen solchen vertikalen PNP-Transistor zu realisieren, ist ein zusätzlicher Buried-p-Schritt notwendig. Gleichzeitig bedarf es einer geringen Anhebung der Epitaxieschichtdicke. Dieser Transistor benötigt eigene Prozessschritte, denn anstelle des normalen Buried-Layer bedarf es eines Buried-Isolation-Layers, der in diesem Prozeß mit N-iso bezeichnet wird (Bild 2). Die entstehende, vom Substrat isolierte, P-Wanne bildet den Kollektor und die innere Epitaxieinsel die Basis. Die Basisweite kommt hierbei in die Größenordnung der NPN-Transistoren, ist jedoch größeren Streuungen unterworfen, wobei am

stärksten die Epitaxieschichtdicke eingeht. Zu kleineren Werten streuende Epitaxiedicken führen schneller zu U_{CE} -Ausfällen als bei den NPN-Transistoren. Bild 2 zeigt das Layout dieses Transistors mit einer Fläche von ca. $2900 \mu\text{m}^2$.

Die Stromverstärkung liegt bei $I_c = 10 \mu\text{A}$ im Bereich 90-150 und sinkt bei $I_c = 1.8 \text{ mA}$ auf ca. die Hälfte ab. Die Transitfrequenz liegt bei 2,5 GHz. (Zum Vergleich: Grenzfrequenz des lateralen PNP-Transistors : 50 MHz)

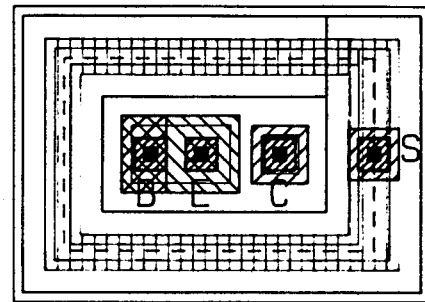
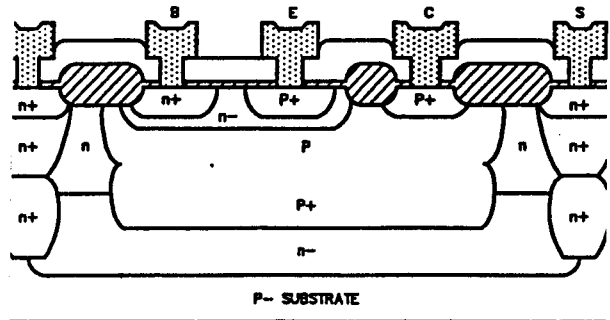


Bild 2 : Querschnitt und Layout eines vertikalen PNP-Transistors

NMOS-Transistor

Die Doppeldiffusion der n-Gebiete (siehe Eigenschaften) bewirkt, daß die Sperrschichtkapazität sinkt. Zudem wird dadurch erreicht, daß die Kanallänge relativ spannungsunabhängig wird, da sich Sperrschichten bevorzugt in schwach dotierte Gebiete ausdehnen. Die kleinen p^+ -Gebiete an den äußeren Rändern von Drain und Source haben die Funktion von 'Channelstoppem', die verhindern sollen, daß sich hier Kanäle bilden. In Bild 3 ist die Kanallänge L zu erkennen, die von der Breite des Poly-Layers beschrieben wird. Die Kanalweite W wird durch die Länge des Active-Layers, der vom Poly überquert wird, beschrieben. Die Kanallänge ist in diesem Fall für eine Versorgungsspannung von 10 V dimensioniert.

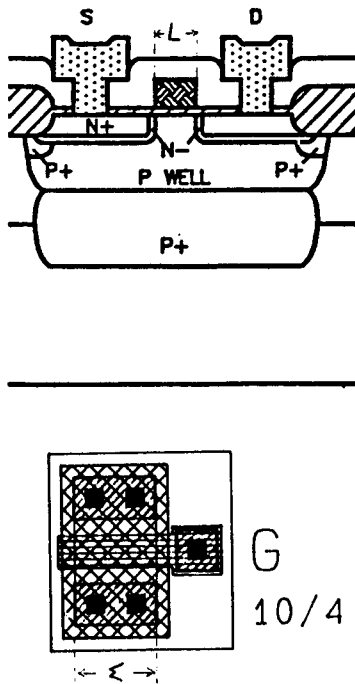


Bild 3 : Querschnitt und Layout eines NMOS-Transistors

PMOS-Transistor

Die beiden Darstellungen (Bild 4) zeigen den in einer N-Wanne liegenden PMOS-Transistor. Aus Isolationsgründen liegt hier der N+Buried-Layer unter dem Transistor. Auch hier beschreibt die Breite des Poly-Layers die Kanallänge, während die Breite des Active-Gebiets die Kanalweite definiert.

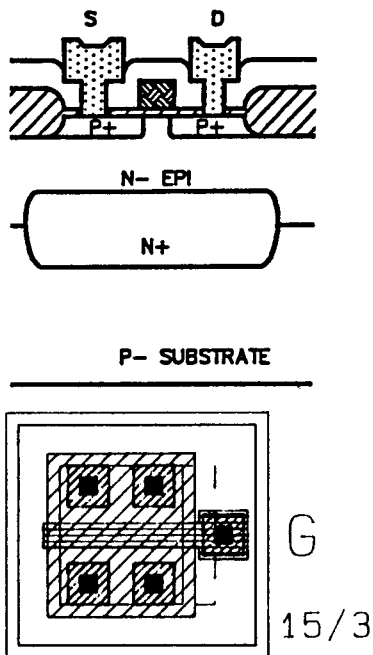


Bild 4 : Querschnitt und Layout eines PMOS-Transistors

Oszilloskop-Eingangsverstärker

Leistungsmerkmale eines Oszilloskop-Eingangsverstärkers:

- große Bandbreite (> 100 MHz) um auch hohe Frequenzen darstellen zu können
- DC-Kopplung vom Eingang auf den Ausgang, damit auch DC-Signale verarbeitet werden können
- umschaltbare Verstärkung, um einen großen Spannungsbereich darstellen zu können
- geringes Rauschen
- unsymmetrisches Ein- und Auskoppeln bei hoher Frequenz

Schaltungsbeschreibung

Der Oszilloskop-Eingangsverstärker ist als zweistufiger Verstärker konzipiert. Die erste Stufe heißt amplifizier 1 und die zweite Stufe amplifizier 2.

amplifizier 1

Ein Oszilloskop-Eingangsverstärker muß eine einstellbare Verstärkung haben. Diese einstellbare Verstärkung soll mit der 1. Verstärkerstufe realisiert werden. Dabei war die Zielsetzung, 8 verschiedene Verstärkungen im Bereich von 0.25...10 zu realisieren. Die Vorteile der BICMOS-Technik können ausgenutzt werden, indem die zeitkritischen Pfade des Breitbandverstärkers in Bipolartechnik, die Umschaltung der einzelnen Verstärkungen jedoch in MOS-Technik realisiert werden.

Differenzverstärker mit Kaskodeschaltung

Bei der Schaltung amplifizier 1, die den ersten Teil des Eingangsverstärkers bildet, handelt es sich im wesentlichen um vier parallelgeschaltete Differenzverstärker, die mit MOSFETs ein- und ausgeschaltet werden können. Differenzverstärker bilden jeweils die Transistorpaare Q3 und Q10, Q4 und Q9, Q5 und Q8, sowie Q6 und Q7. Die Transistoren Q1 und Q2 werden in Basisschaltung betrieben. Es handelt sich also um einen Differenzverstärker mit Kaskodeschaltung. Da in jedem Zweig des Differenzverstärkers vier Transistoren parallelgeschaltet sind, wirken, auch wenn jeweils nur ein Transistor arbeitet, alle Basis-Kollektor-Kapazitäten. Es ergibt sich eine wirksame Basis-Kollektor-Kapazität von $4 \cdot C_{\mu}$. Dies erniedrigt die obere Grenzfrequenz erheblich.

Aus diesem Grund werden Neutralisationskondensatoren eingefügt (C1 und C2), welche zum gegenphasigen Punkt geführt sind. Sind diese gleich groß wie $4 \cdot C_{\mu}$, dann wird diese parasitäre Kapazität kompensiert.

Eingangstreiber

Um den geringen Eingangswiderstand der Differenzverstärkerstufe zu erhöhen, wird eine Kollektorschaltung als Eingangstreiber benutzt. Bei der Kollektorschaltung wird der Lastwiderstand, um den Faktor $\approx \beta_0$ vergrößert, an den Eingang transformiert.

Ausgangstreiber

Da mit dieser Schaltung relativ große kapazitive Lasten getrieben werden müssen, ist eine zweistufige Ausgangstreiberschaltung (Emitterfolger) nötig. Die Bandbreite der Schaltung wird fast ausschließlich durch die zu treibende kapazitive Last begrenzt. Um bei kapazitiver Last die Tiefpaßwirkung möglichst gering zu halten, ist ein kleiner Ausgangswiderstand notwendig. Die Kollektorschaltung transformiert den Quellwiderstand, um den Faktor $\approx \beta_0$ verkleinert, an den Ausgang.

Eingangsteilerschaltung

Ein Eingangsteiler ist notwendig, um auch Signale verarbeiten zu können, die so groß sind, daß der Verstärker sonst in die Begrenzung gehen würde.

Der Eingangsteiler, bestehend aus den Widerständen R28 und R29, bzw. R30 und R31, läßt die Teilverhältnisse 1/1 oder 1/4 zu. Die MOSFETs sind in dieser Anwendung einstellbare Widerstände. Dabei arbeiten die MOSFETs als Schalter. Liegen -5V am Gate, ist der Drain-Source-Widerstand einige hundert Megaohm (ideal ∞), bei +5V dagegen nur einige 100 Ohm (ideal 0). Für das Teilverhältnis 1/1 muß die Eingangsspannung direkt am Eingang abgegriffen werden. Es sind also die MOSFETs Q30 und Q32 durchzuschalten, Q29 und Q31 dagegen zu sperren. Für das Teilverhältnis 1/4 ist umgekehrt zu verfahren.

Die Teilerwiderstände sollten einerseits möglichst groß sein, damit der Eingangswiderstand der Schaltung groß wird. Auf der anderen Seite sollten die Teilerwiderstände aber auch klein sein, damit die Bandbreite nicht durch parasitäre Kapazitäten vermindert wird. Als Kompromiß wählte man einen Gesamtwiderstand von $\approx 2 \text{ k}\Omega$. Gewählt wurden aus Layoutgründen Teilerwiderstände von 600Ω und $1,8 \text{ k}\Omega$.

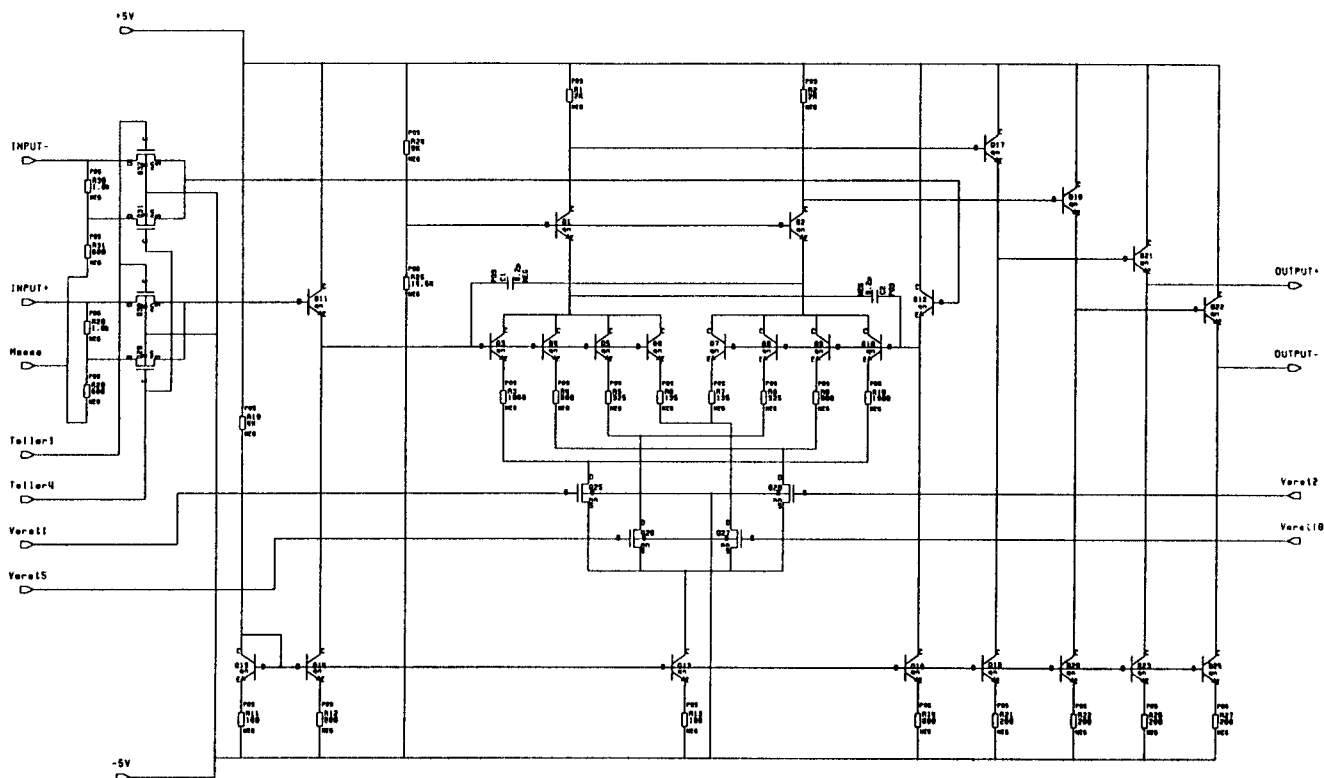


Bild 5: Schaltplan amplifier 1

Gesamtverstärker

Der Verstärker hat 8 einstellbare Verstärkungen. Diese Verstärkungen sind im einzelnen: 1/4, 1/2, 1, 5/4, 2, 5/2, 5 und 10.

amplifier 2

Mit der zweiten Stufe des Eingangsverstärkers soll eine Verstärkung um den Faktor 5, mit gleichzeitiger Wandlung von symmetrischem auf unsymmetrisches Signal, realisiert werden. Weiterhin soll für $U_{id} = 0$ auch $U_o = 0$ sein. Für diesen Verstärker sind schnelle vertikale PNP-Transistoren notwendig, die bei dem verwendeten Prozeß auch vorhanden sind.

Funktion der Schaltung

Bei dieser Schaltung handelt es sich um einen sog. OTA (Operational Transkonduktanz Amplifier). Der Differenzverstärker, bestehend aus den Transistoren Q5 und Q6, arbeitet auf zwei PNP-Stromspiegel, bestehend aus den Transistoren Q1-Q4. Diese spiegeln den jeweiligen Kollektorstrom auf den Stromspiegel, bestehend aus Q7 und Q8. Die Funktion kann man sich also folgendermaßen klarmachen:

Über die Stromspiegel liegt der Kollektorstrom von Q5 auch am Kollektor von Q8, und der Kollektorstrom von Q6 am Kollektor von Q4. Ist die Spannung an input- größer als an input+, ist der Kollektorstrom von Q5 größer als der von Q6. Folglich ist auch der Kollektorstrom von Q8 größer

als der von Q4. Daher muß über R4 ein negativer Strom fließen, d.h. an R4 liegt eine negative Spannung. Ist die Spannung an input+ größer als an input-, verhält es sich umgekehrt. Der Kollektorstrom von Q4 ist größer als der von Q8, über R4 fließt ein positiver Strom, d.h. an R4 liegt eine positive Spannung.

Auch bei dieser Verstärkerstufe sind zwei Emitterfolger als Ausgangstreiber notwendig, damit die Bandbreite auch bei kapazitiver Last ausreichend hoch wird. Um ein unsymmetrisches Ausgangssignal um Masse herum zu erhalten, sind die Dioden, bestehend aus Q14 - Q17 notwendig. Da die beiden Emitterfolger am Ausgang die Spannung an R4/pos um zwei Diodenspannungen niedriger an den Ausgang transformieren, muß die Nullspannung an R4/pos genau zwei Diodenspannungen betragen. Ohne Ansteuerung am Eingang des Verstärkers, fließt über R4 kein nennenswerter Strom, die Spannung über R4 ist null.

Da die Spannung an R4/neg genau zwei Diodenspannungen entspricht, und diese Spannung auch an R4/pos liegt, liegen am Ausgang genau 0V. Da jede beliebige Gleichspannung (statt Masse) an Q16/E angelegt werden kann, kann die Ausgangsnullspannung des Verstärkers eingestellt werden. Dadurch ist die Y-Position des Oszilloskops einstellbar. Durch die Dioden muß ein Gleichstrom fließen, der mindestens so groß wie der max. Kollektorstrom der Transistoren Q5 bzw. Q6 sein muß. Dieser Strom wird durch den Stromspiegel aus Q11, Q12 und Q13 zur Verfügung gestellt.

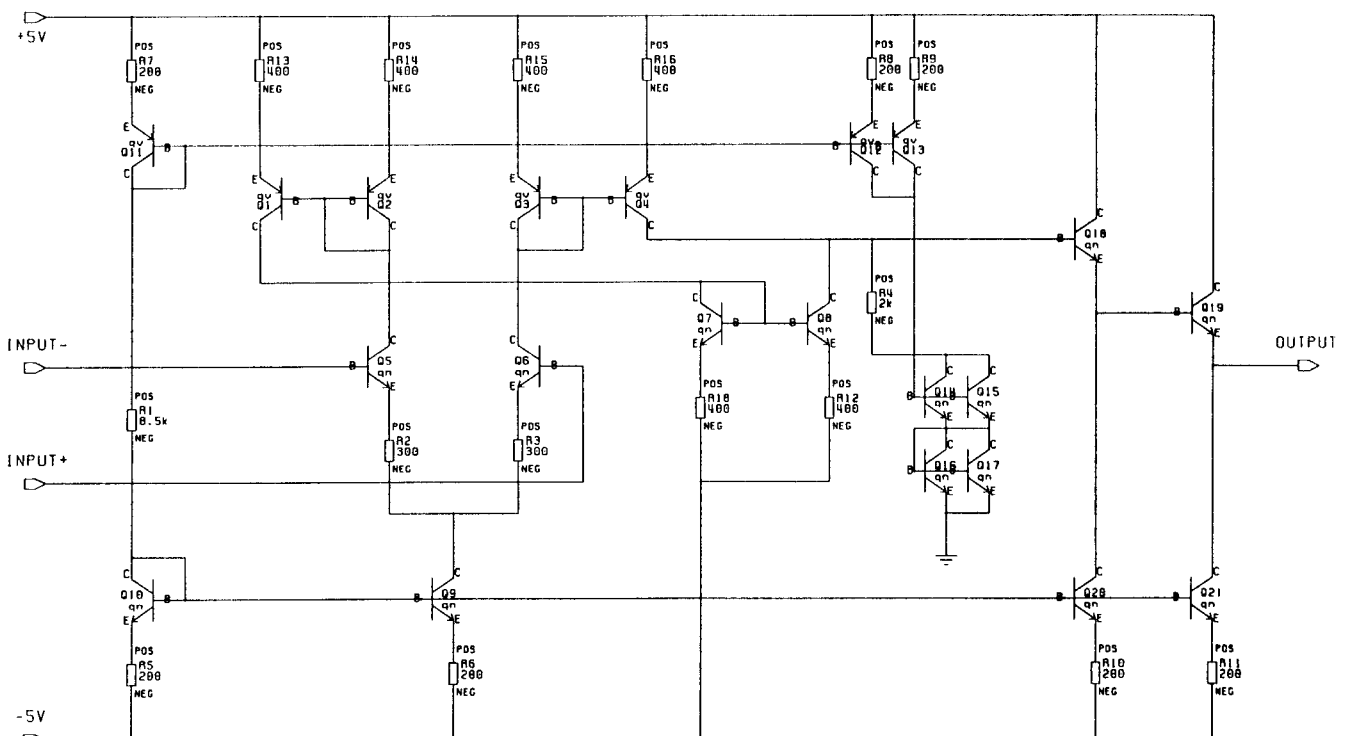


Bild 6: Schaltplan amplifier 2

Simulationsergebnisse

Die SPICE-Parameter für die Simulation der Schaltungen sind von SGS-Thomson. Bereits bei diesen Simulationen wurden die wichtigsten Parasiten eingefügt, um ein realistisches Bild vom tatsächlichen Verhalten der Schaltung auf der späteren Platine zu erhalten. Am Ausgang sind folgende Parasiten anzunehmen:

- Padkapazität (≈ 1 pF)
- Induktivität des Bonddrahtes (≈ 2 nH)
- Kapazität eines Meßstastkopfes bzw. einer Platine (≈ 10 pF)

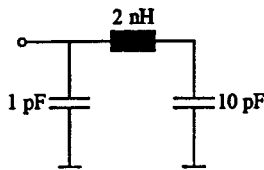


Bild 7: Parasiten am Ausgang

Simulation des amplifiers 1

Großsignalverhalten

Die Großsignalcharakteristiken für $v=2, 5$ und 10 sind in Bild 8 zu sehen. Man erkennt, daß die lineare Aussteuerbarkeit am Ausgang bei ca. $\pm 3V$ liegt, und der Verstärker dann in die Begrenzung geht.

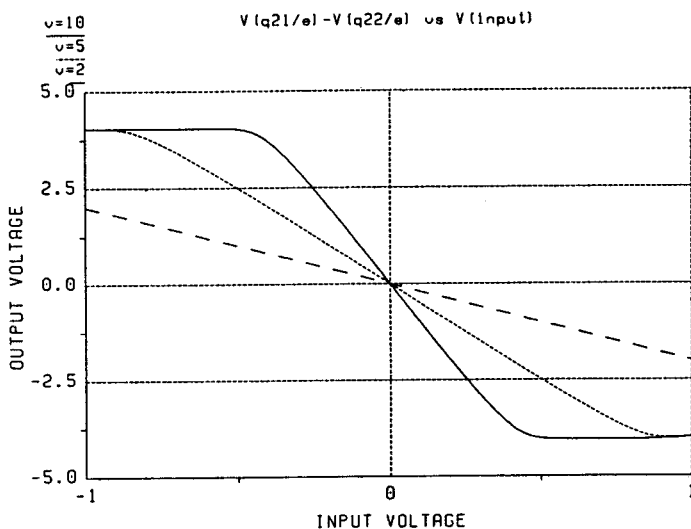


Bild 8: Großsignalverhalten des amplifiers 1

Kleinsignalverhalten

Das Kleinsignalverhalten für die acht verschiedenen Verstärkungen ist in Bild 9 zu sehen.

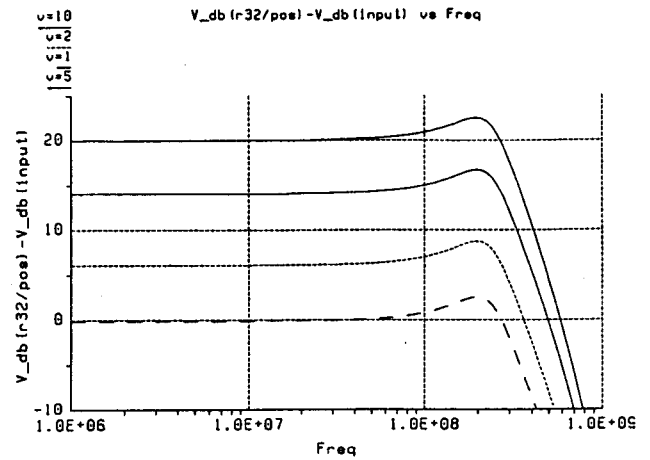


Bild 9a: Kleinsignalanalyse mit Eingangsteiler 1/1

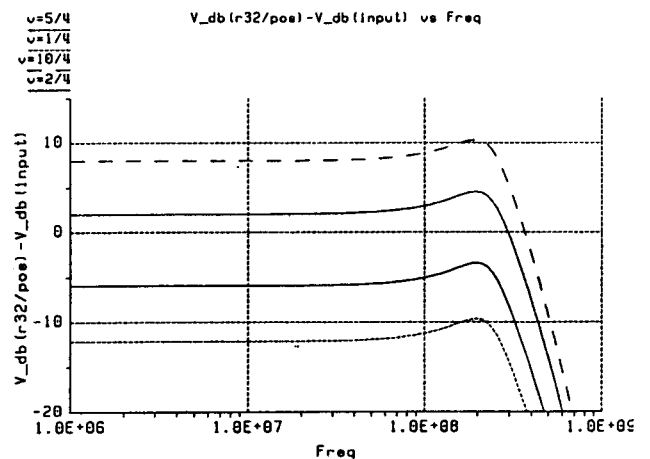


Bild 9b: Kleinsignalanalyse mit Eingangsteiler 1/4

Die -3 dB-Grenzfrequenz liegt bei ≈ 300 MHz. Man erkennt, daß beim Eingangsteilerverhältnis $1/4$ die Bandbreite nur unwesentlich kleiner ist als beim Verhältnis $1/1$.

Die Emitterfolger am Ausgang wirken bei hohen Frequenzen induktiv. Mit der kapazitiven Last am Ausgang bildet sich ein Resonanzkreis, welcher eine Überhöhung verursacht. Diese Überhöhung beträgt bei allen Verstärkungen ca. 2 dB. In der Praxis bildet dieser Verstärker die erste Stufe des Eingangsverstärkers. Am Ausgang des Verstärkers ist lediglich die Eingangskapazität der zweiten Stufe. Diese beträgt ca. 0.55 pF. Eine Simulation für $v=10$ mit dieser Kapazität am Ausgang zeigt Bild 10. Die Bandbreite beträgt hierbei ≈ 900 MHz.

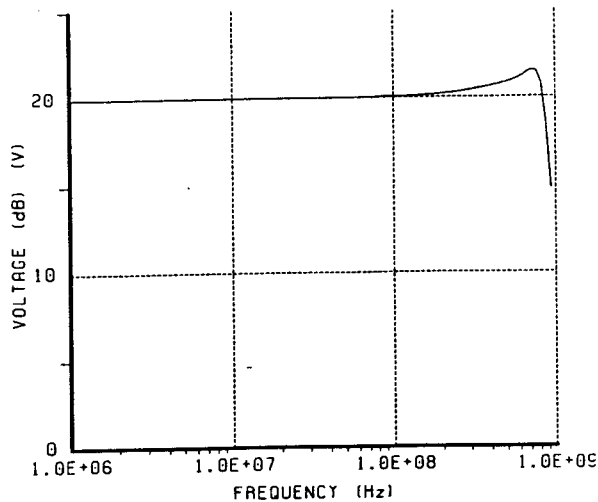


Bild 10: Simulation mit 0.55 pF am Ausgang

Temperaturanalyse

Eine wichtige Eigenschaft einer Schaltung ist ihr Temperaturgang. Für eine Simulation aller Eventualitäten, und unter Berücksichtigung der Erhitzung des Chips, wird eine Temperaturanalyse von -30°C bis $+80^{\circ}\text{C}$ für absolut ausreichend erachtet. Für $v=10$ ergibt sich der in Bild 11 dargestellte Verlauf.

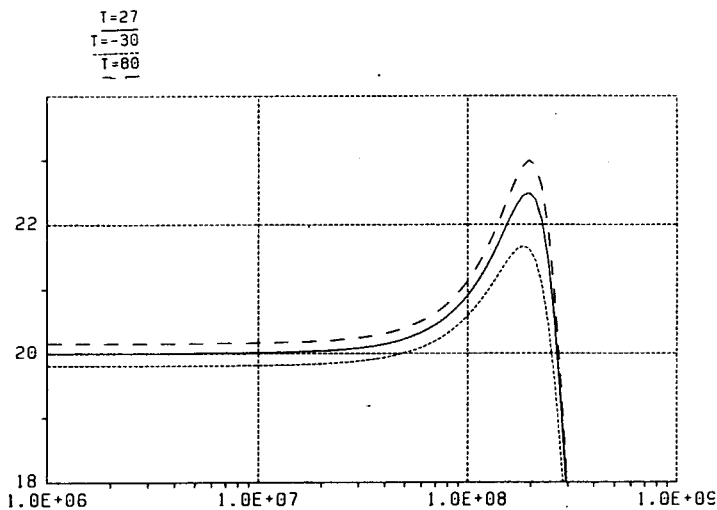


Bild 11: Temperaturanalyse für $v=10$

Man erkennt, daß das Temperaturverhalten der Schaltung sehr gut ist. Die Änderung der Verstärkung mit der Temperatur beträgt ungefähr

$$\frac{dV}{dT} \approx 0.5\%/K.$$

Der Grund hierfür ist einerseits der sehr kleine TK der Widerstände und andererseits die Tatsache, daß die Veränderung der Verstärkung mit der Temperatur teilweise durch eine Änderung des Kollektorstroms kompensiert wird.

Rauschanalyse

Charakteristisch für das Rauschen eines Verstärkers ist das Rauschen am Eingang. Dieses Rauschen muß, um das Rauschen am Ausgang zu erhalten, mit der Spannungsverstärkung multipliziert werden. Für den Signal-Rauschabstand eines mehrstufigen Verstärkers, ist vor allem die erste Stufe entscheidend. Da der Verstärker amplifier 1 die 1. Stufe des Oszilloskop-Eingangsverstärkers ist, ist sein Eingangsrauschen der entscheidende Faktor für den Signal-Rauschabstand des Eingangsverstärkers. Das Eingangsrauschen des Verstärkers beträgt ca. $13 \text{ nV}/\sqrt{\text{Hz}}$.

Simulation des amplifier 2

Großsignalverhalten

In Bild 12 ist die Großsignalkennlinie des Verstärkers zu sehen.

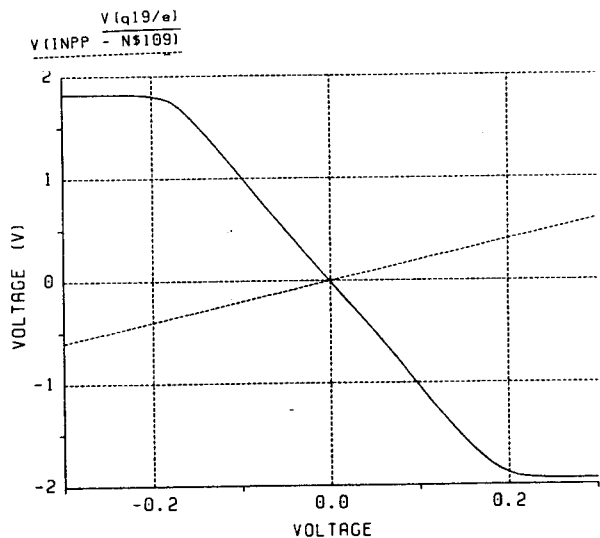


Bild 12: Großsignalkennlinie amplifier2

Es ist erkennbar, daß die lineare Aussteuerbarkeit des Verstärkers bei etwa $\pm 1.7 \text{ V}$ liegt, was einem Eingangssignalhub von c.a. $\pm 0.34 \text{ V}$ entspricht.

Kleinsignalverhalten

Alle Simulationen wurden auch hier mit den berechneten Kapazitäten für Pads und Platine sowie der Induktivität des Bonddrahtes durchgeführt, um die wesentlichen Faktoren für die Bandbreite bereits in der Simulation zu berücksichtigen.

Die Bandbreite des Verstärkers hängt sehr stark vom Strom des Differenzverstärkers ab. Eine Erhöhung des Konstantstroms und damit eine Erhöhung der Kollektorströme von Q1-Q6 erhöht die Bandbreite beträchtlich. Allerdings darf der Strom auch nicht zu groß gemacht werden, da Q2 und Q3 sonst Sättigungseffekte zeigen, und die Funktion der Schaltung negativ beeinflussen. Ein guter Kompromiß wurde mit einem Strom von $I=0.89\text{ mA}$ gefunden. Mit diesem Strom ergibt sich die Kleinsignalkennlinie, die in Bild 13 zu sehen ist.

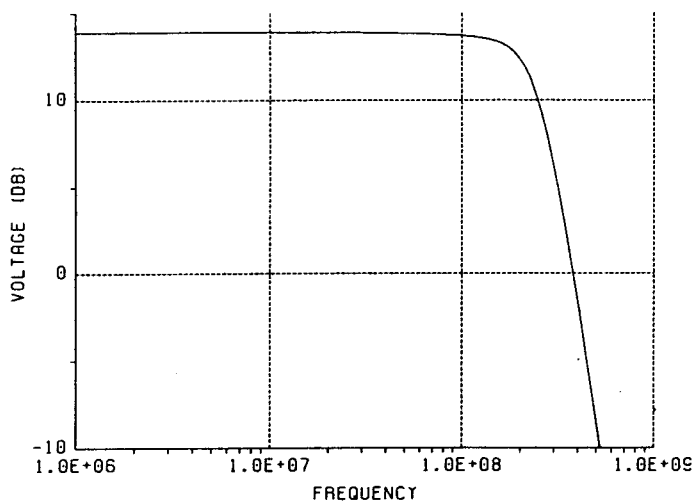


Bild 13: Kleinsignalkennlinie amplifer 2

Layout für amplifer 1

Die Kompensationskondensatoren C1 und C2 werden mit Dummy-Transistoren ohne Emitter realisiert, damit die parasitären Kapazitäten gleich groß sind und sich aufheben. Durch diese Maßnahme wird auch das Temperaturverhalten der Schaltung verbessert, da der Temperaturgang der Dummy-Transistoren gleich dem der Transistoren ist. Durch Löschen des Emitters werden aus den NPN-Transistoren die Dummy-Transistoren hergestellt.

Die Transistorpaare der jeweiligen Differenzverstärker sind beieinander realisiert, um die Toleranzen klein zu halten. Da der kleinste Einheitswiderstand auf $600\ \Omega$ festgelegt wurde, wurde für den Eingangsteiler das Widerstandsverhältnis $600\ \Omega - 1,8\text{ k}\Omega$ gewählt.

Layout für amplifer 2

Da weder die Spannungsquelle, noch die Versorgungsleitungen ideal sind, haben die Ausgangstreiber Q18 und Q19 Spannungsschwankungen am Kollektor. Um Mitkopplungen über die Versorgungsspannungsleitungen zu

vermeiden, sind die Versorgungsleitungen zu den PNP-Transistoren Q1-Q4 extra geführt. Dasselbe gilt sinngemäß für die Versorgungsleitung zu R18 und R12. Um Unsymmetrien zu vermeiden, müssen die Versorgungsleitungen zu R13 und R14, bzw. zu R15 und R16 denselben Widerstand aufweisen. Die Längen der Leitungen wurden entsprechend gewählt. Auch in diesem Layout wurden die, für die Geschwindigkeit relevanten Leitungen, möglichst kurz gehalten, um parasitäre Kapazitäten möglichst zu vermeiden. Die Substratanschlüsse der vertikalen PNP-Transistoren müssen an die positivste Spannung der Schaltung, also an $+5\text{V}$, angeschlossen werden.

Nachsimulation

Für beide Schaltungen wurde eine Nachsimulation mit Parasiten aus dem Layout durchgeführt. Dabei wurde festgestellt, daß diese parasitären Widerstände und Kapazitäten kaum einen Einfluß auf die Funktion der Schaltungen haben. Bei einer Nachsimulation mit max. und min. Transistorparametern ergab sich eine Änderung der Bandbreite von max. 10%.

Chip-Design

Die Schaltungen der Oszilloskop-Eingangsstufe wurden gemeinsam mit weiteren Testschaltungen, u.a. 2 OPV, auf einem Chip integriert. Die Oszilloskop-Eingangsstufe und die OPVs wurden diagonal getrennt voneinander in den Chipecken plziert, um gegenseitige elektrische Wechselwirkungen zu minimieren. Die Signalleitungen können zusätzlich auf diese Weise kurz gehalten werden. Dies vermindert kapazitive Einflüsse und optimiert das Hochfrequenzverhalten. Die einzelnen Schaltungen werden zudem mit separaten, voneinander getrennten Betriebsspannungen versorgt.

Substratanschlüsse mussten so plziert werden, daß der Abstand zu gefährdeten Bauelementen (Latchupgefahr bei MOS-Elementen) nicht mehr als $120\ \mu\text{m}$ betrug.

Der äußere Rahmen beschreibt die Kante des Ritzrahmens, an dem am Ende des Fertigungsprozesses die einzelnen Chips voneinander getrennt werden. Der gesamte Chip hat eine Fläche von $6,25\text{ mm}^2$.

Bei den Testschaltungen, die mit Testpads ausgestattet sind, müssen diese Pads nicht der minimalen Design-Regel-Größe entsprechen, sondern können kleiner entworfen werden. So können sie leichter von Bondpads unterschieden werden und sind zum Ansetzen der Testspitzen noch ausreichend groß.

Chip-Fertigung über EuroChip

Als Gehäuse wurde ein CLCC44 gewählt, da es sich gut für hochfrequente Anwendungen eignet. Die Fertigung solcher Prototypen, die von Hochschulen entwickelt werden, wird über EuroChip abgewickelt. Nach letzten Tests von EuroChip, wurde unser Chip-Entwurf fehlerfrei zur Fertigung nach Grenoble zu SGS Thomson verschickt. Die Dauer eines solchen Fertigungsprozesses liegt bei ca. 8-10 Wochen.

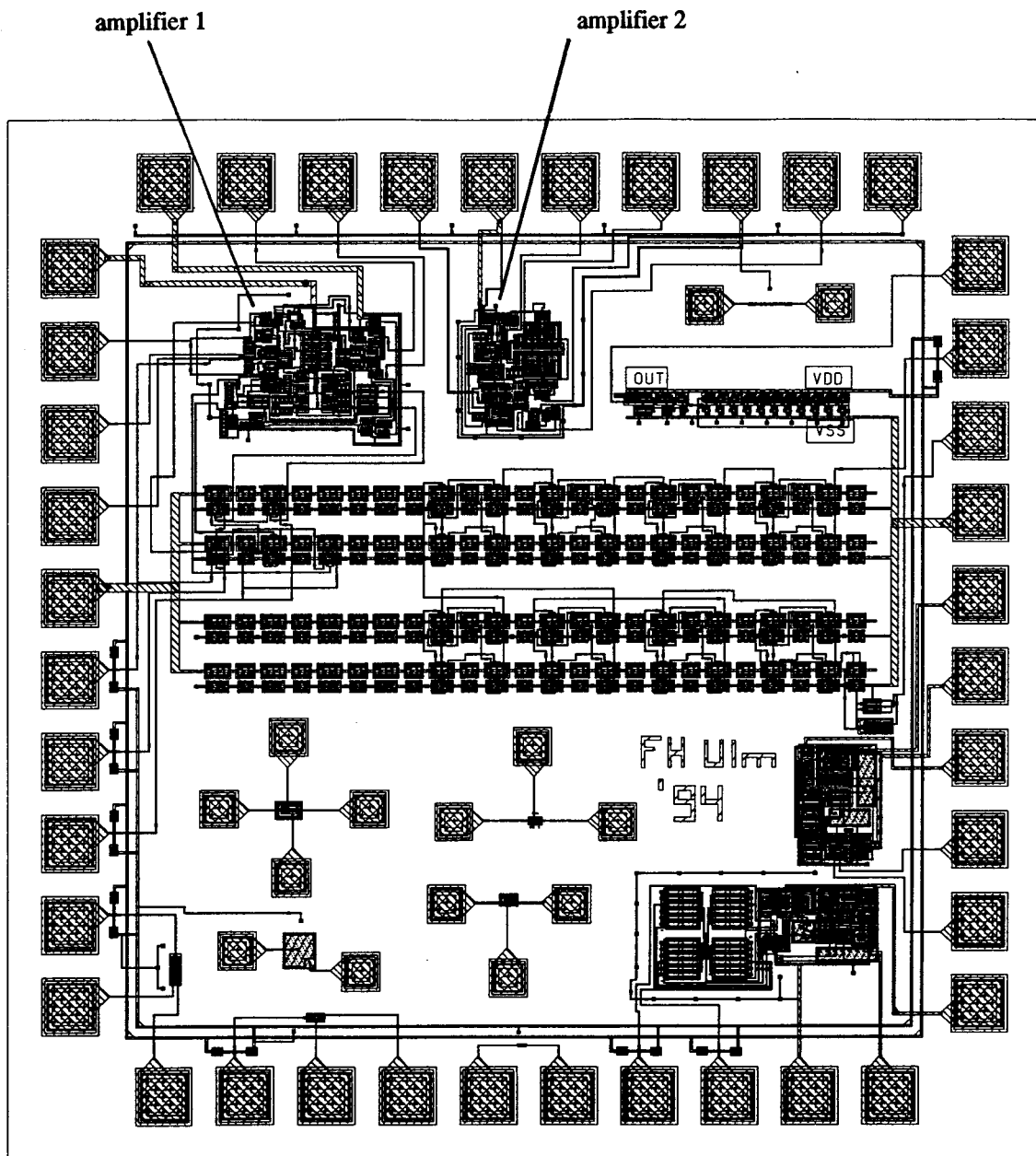
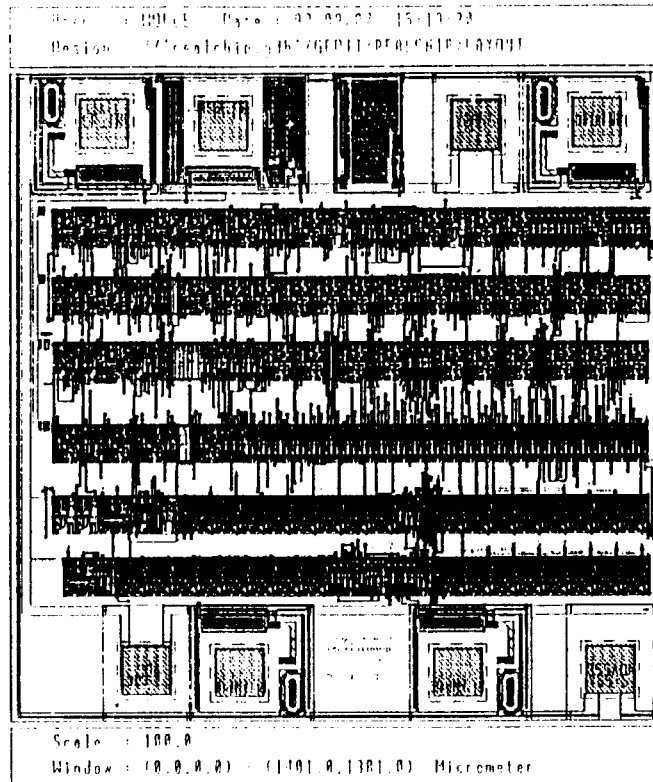


Bild 14: Layout Gesamtchip

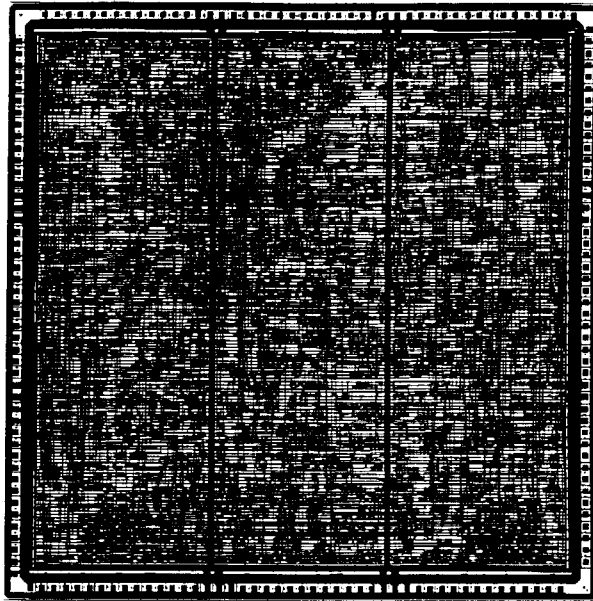
CMOS - ASIC mit niedrigem Leistungsverbrauch unter extremen Einsatzbedingungen

K. Schmidt, B. Röllgen



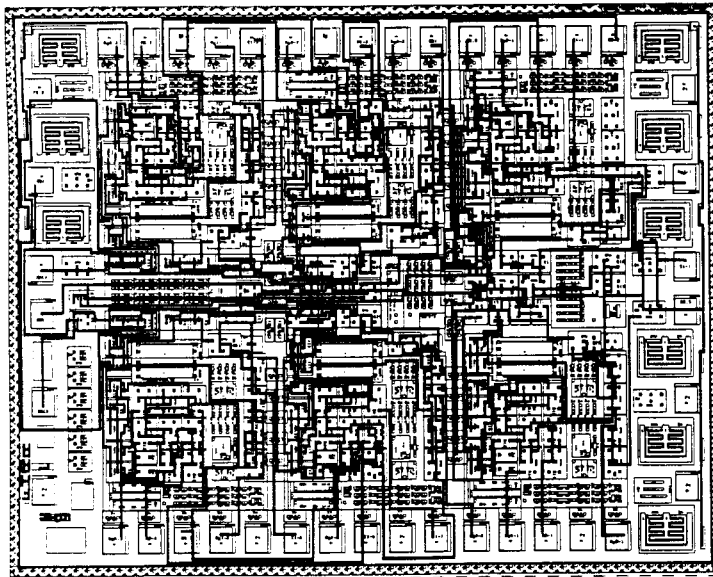
Entwurf:	Fachhochschule Furtwangen, Prof. Dr. Kurt Schmidt und Diplomat Bernd Röllgen,
Layouterstellung:	bei Firma Neutron, Offenbach, durch den Diplomat und Mitarbeiter der Firma,
Chipfertigung:	Firma Elmos,
Herstelldatum:	im Dezember 1993,
Kostenträger:	MPC-Mittel FH-Verbund Baden-Württemberg,
Chipabmessungen:	ca 1,4 x 1,4 mm ² ,
Testergebnisse:	volle Funktion, und bei höchster Betriebstemperatur Leckstrom 100 pA,
Funktion:	Ereigniszähler, Batterie betrieben, mit zuschaltbarer serieller 4-Befehls-Kommunikationsschnittstelle,
Anwendung:	im Überwachungs- und Sicherheitsbereich, eine Erprobung läuft (Stand 2/94),
Datenblatt:	in Vorbereitung.

Parametrisierbarer kaskadierbarer digitaler Filterbaustein



- Entwurf: Fachhochschule Aalen
Bearbeiter: Diplomanden M. Sator und A. Gottscheber
Betreuung: Prof. Dr. B. Kohlhammer, Dipl.-Ing(FH) G. Busch,
Dipl.-Ing(FH) H. Graf
- Layouterstellung
und Chipfertigung: Institut für Mikroelektronik (IMS), Stuttgart
- Herstelldatum: IV. Quartal 1993
- Kostenträger: MPC-Mittel FH Verbund Baden-Württemberg
- Technologie: IMS 1,2 μ m Gate Forest
- Chipdaten: Master: GF α mit 81000 Transistoren, davon 45000 benutzt
Chipgröße: 11,5 x 11,5 mm²
Signalpins : 72
- Funktion: Der Schaltkreis DIGFILT (DIGitaler FILTer) ist ein digitaler Filterbaustein, welcher auch in der digitalen Regelung eingesetzt werden kann.
- Eigenschaften:
- Filtergrad: 2
 - kaskadierbar: bis zu 6 Filterbausteine können in Serie geschaltet werden, d.h. es kann durch Kaskadierung ein digitaler Filter bis zum Grad 12 realisiert werden
 - parametrisierbar: Die Koeffizienten des digitalen Filters werden in Festwert- oder Schreib/Lesespeichern gelesen
 - 16 Bit Datenwortbreite
 - 16 Bit Koeffizientenwortbreite
 - maximaler Rechentakt: 550 kHz (bei 20 MHz Taktversorgung)

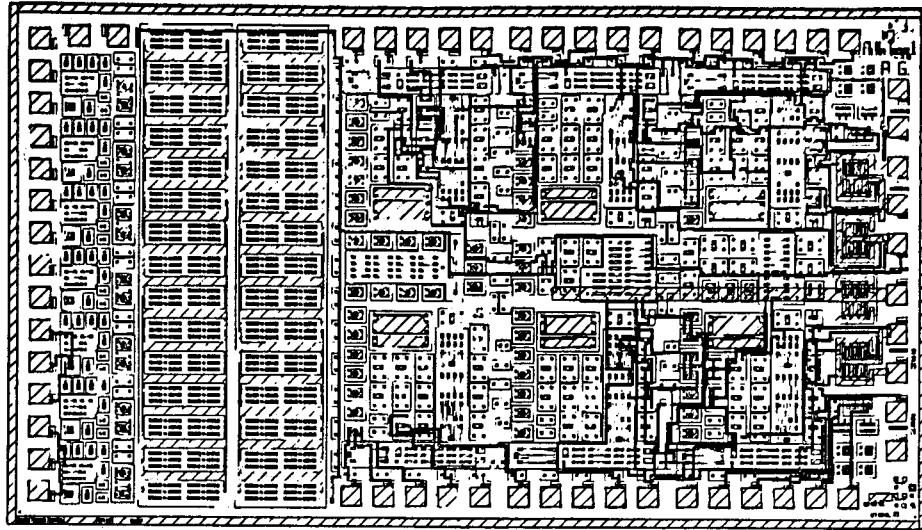
Erdfreier Gyrtator auf B500A Bipolar Array



- Entwurf: Fachhochschule Aalen
Bearbeiter: Diplomanden S. Beck und R. Schrottenholzer
Betreuung: Prof. Dr. B. Kohlhammer, Dipl.-Ing(FH) H. Graf
- Layouerstellung: Fachhochschule Aalen
- Chipfertigung: AEG-Telefunken
- Herstelldatum: I. Quartal 1993
- Kostenträger: MPC-Mittel FH Verbund Baden-Württemberg
- Technologie: Bipolar Array B500A
- Chipdaten: Master: B500A mit 512 Elementen
Chipgröße: 4,1 x 3,2 mm²
Signalpins : 27
- Funktion: 3 Y-Gyratoren mit hoher Polgüte
- Anwendung: Die Gyratoren werden zur Realisierung von hochwertigen analogen Filterschaltungen eingesetzt

Realisierung analoger Grundsaltungen auf einem Bipolaren Transistor-Array

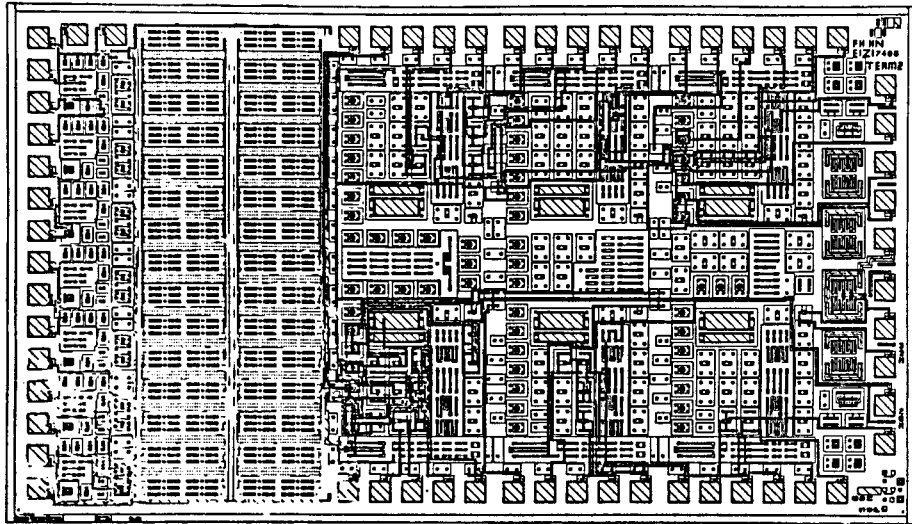
M. Amann, G. Günther



Entwurf:	Fachhochschule Heilbronn, Prof. Dr.-Ing. H. Clauss und Studenten Michael Amann und Götz Günther
Layouerstellung:	Fachhochschule Heilbronn, durch die Studenten
Chipfertigung:	Firma TEMIC
Herstelldatum:	im November 1993
Kostenträger:	MPC-Mittel FH-Verbund Baden-Württemberg,
Chipfläche:	18.4 mm ²
Gehäuse:	keramisch DIL 64 pol.
Testergebnisse:	in Bearbeitung
Funktion:	Realisierung von analogen Grundsaltungen, z. B.: Spannungsregler, Gegentakt-Leistungsendstufe, Rechteckgenerator (1Hz - 1MHz).
Anwendung:	Erstellen einer Bibliothek bereits getesteter analoger Grundsaltungen.

Realisierung analoger Grundsaltungen auf einem Bipolaren Transistor-Array

Stefan Eickert, Michael Zimmermann



Entwurf:	Fachhochschule Heilbronn, Prof. Dr.-Ing. H. Clauss und Studenten Stefan Eickert und Michael Zimmermann
Layouterstellung:	Fachhochschule Heilbronn, durch die Studenten
Chipfertigung:	Firma TEMIC
Herstelldatum:	im November 1993
Kostenträger:	MPC-Mittel FH-Verbund Baden-Württemberg,
Chipfläche:	18.4 mm ²
Gehäuse:	keramisch DIL 64 pol.
Testergebnisse:	volle Funktion
Funktion:	Realisierung von analogen Grundsaltungen, z. B.: verschiedene Stromquellen, Differenzverstärker, Operationsverstärker.
Anwendung:	Erstellen einer Bibliothek bereits getesteter analoger Grundsaltungen.

Baustein zur Datenverschlüsselung

Entwurf:	Fachhochschule Ulm, Diplomarbeit Peter Jonski, Prof. Arnold Führer
Chipfertigung:	ES2 European Silicon Structures
Technologie:	ES2 Cell2 ECPD15 (1,5 um)
Chipabmessungen:	5,182 x 4,758 qmm
Pads:	32, Powerpads enthalten
Gehäuse:	DIL 40
CAD-Software	CADENCE-EDGE mit PLA-Generator auf APOLLO Workstations.
Eigenschaften:	Ein IC für die schnelle Ver- und Entschlüsselung von Daten. Er kann in PCs und Servern verwendet werden. Für das Testen ist ein Scanpath vorgesehen. Eine detaillierte Funktionsbeschreibung ist im MPC-Workshop-Band 'Februar 1993 Reutlingen' erschienen.
Herstelldatum:	April 93
Kostenträger:	EUROCHIP-Mittel

Baustein für eine Fussgängerampel in Gate-Array-Ausführung

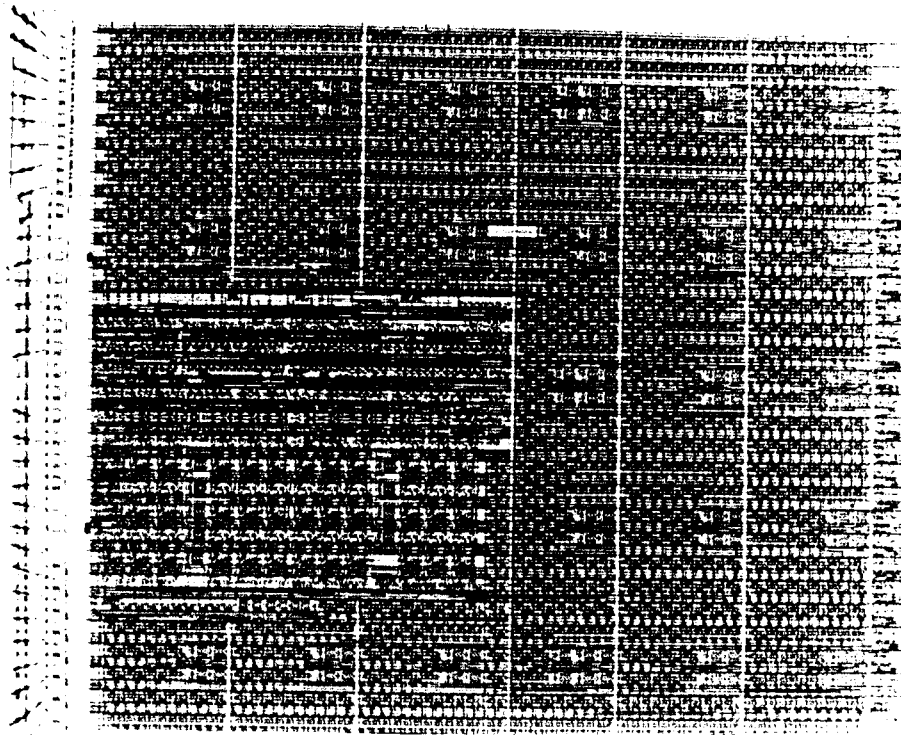
Entwurf:	Fachhochschule Ulm, Praktikumsarbeit Prof. Arnold Führer, Michael Faulhaber
Chipfertigung:	Institut für Mikroelektronik (IMS) Stuttgart
Technologie:	GF12G1 (CMOS 1,2 μm); Gate Array
Bauelemente:	6276 Transistoren
Chipabmessungen:	12,7 qmm
Pads:	44, Powerpads enthalten
Gehäuse:	LCC 44
CAD-Software:	MENTOR GRAPHIC auf APOLLO Workstations.
Eigenschaften:	Ein IC für die die Steuerung einer Fussgängerampel mit digital einstellbaren rot, gelb, grün Ampelphasen. Umschaltbar auf gelben Blinkbetrieb. Die kleinste einstellbare Zeiteinheit beträgt 1 Sekunde. Die Taktfrequenz wird extern erzeugt und beträgt 32kHz. Für das Testen ist ein Scanpath vorgesehen.
Test:	Der Baustein wurde im IMS erfolgreich getestet.
Herstelldatum:	April 93
Kostenträger:	MPC-Mittel

Baustein für eine Fussgängerampel in Standard-Zellen-Ausführung

Entwurf:	Fachhochschule Ulm, Praktikumsarbeit Prof. Arnold Führer, Michael Faulhaber
Chipfertigung:	ES2 European Silicon Structures
Technologie:	ES2 Cell2 ECPD15 (CMOS 1,5 um); Standardzellen
Chipabmessungen:	2,847 x 2.835 qmm
Pads:	48, Powerpads enthalten
Gehäuse:	DIL 48
CAD-Software:	CADENCE-EDGE auf APOLLO Workstations.
Eigenschaften:	Ein IC für die die Steuerung einer Fussgängerampel mit digital einstellbaren rot, gelb, grün Ampelphasen. Umschaltbar auf gelben Blinkbetrieb. Die kleinste einstellbare Zeiteinheit beträgt 1 Sekunde. Die Taktfrequenz wird extern erzeugt und beträgt 32kHz. Für das Testen ist ein Scanpath vorgesehen.
Test:	Der Baustein wurde in eine Demonstrationsplatine eingebaut und arbeitet einwandfrei.
Herstelldatum:	Juni 93
Kostenträger:	EUROCHIP-Mittel

Automatische Meßwertkorrektur für intelligente Sensoren

W. Rülling



Mikroskopische Aufnahme des gefertigten Chips

- Entwurf: Prof. Dr. W. Rülling
IIT-Projekt "Algorithmen für intelligente Sensoren"
- Chipfertigung: IMS, Institut für Mikroelektronik in Stuttgart
- Herstellung: September 93
- Technologie: 2 μ m CMOS-Process, Gate-Forrest GF2- ASIC
Chipdaten: 9.6 x 8.8 mm, 118 000 Transistoren, 72 Pads
- Auslieferung: Mitte Okt.93, 20 ICs
- Kostenträger: MPC FH-Verbund Baden-Württemberg
- Funktion: Prototypchip zur Berechnung von Faltungsintegralen
- Testergebnisse: volle Funktion bei Test mit Logic Analyzer (DAS9200)
volle Funktion als Coprozessor in PC-Umgebung
Testverification mit hp81810S in Arbeit
- Anwendung: Meßwertkorrektur von Sensorsignalen

Kurzbeschreibung Chip "STRA"

Designer: Lorena Castello (Diplomarbeit)

Chipname: STRA

Chiptechnologie: Gate -Array-Master GF9G1 (IMS)

Tag der Datensatzübergabe an das IMS: 3.12.93 Chips erhalten: ca Anfang Januar

Zahl der benutzten Gatter: 426 NAND2-Gatteräquivalente

Funktion: Der Chip ist für den Einsatz in einer Telemetrieschaltung entwickelt worden. Er nimmt die Ausgangssignale von 2 seriellen AD-Wandlern von Typ MAX 190 auf, speichert sie zwischen und gibt sie versehen mit einem Paritybit seriell wieder aus. Außerdem ist die notwendige Ansteuerlogik für die beiden AD_Wandler mit integriert. Es sind zwei Taktgeneratoren auf dem Chip integriert: ein Quarzgenerator und ein RC-Generator.

Chiperprobung: Die Funktion des Chips ist gegeben. Es wurden bisher keine Fehler festgestellt.



Prof. Dr. H. Töpfer

