

# Anhang

## Inhaltsverzeichnis des Anhangs

Inhaltsverzeichnis des Anhangs.....	87
Anhang A Anhang zu Kapitel 1.....	88
A.1 Kanalnachbildung.....	88
A.2 Ersatzschaltbild der "Ti-TP Kette" .....	92
Anhang B Anhang zu Kapitel 2.....	93
B.1 Ersatzschaltbild der Einsteckkarte "HP" .....	93
Anhang C Anhang zu Kapitel 3.....	94
C.1 Messergebnisse der "Spannungsmessung am Ausgang der "Ti-TP Kette" bei OZ1 bis OZ8" .....	94
C.2 Auswirkungen auf das Signalspektrum des periodischen Rechtecks durch die "Ti-TP Kette" bei OZ1 bis OZ8 .....	95
C.3 Spannungs- / Impedanzanalyse für OZ4 und OZ5 .....	96
C.4 Knotenpotentialanalyse.....	97
C.5 Bestimmung des Amplitudenganges der ermittelte Übertragungsfunktionen mit Matlab zum Vergleich mit der Messung am Modell.....	100
C.6 Bestimmung der Grenzfrequenz ab OZ4 .....	101
C.7 Messung des Amplitudenganges mit dem Audioanalysator .....	103
Anhang D Anhang zu Kapitel 4.....	105
D.1 Bestimmung der Übertragungsfunktion mit Hilfe von Matlab.....	105
D.2 Ermittelte Koeffizienten für ein digitales IIR-Filter.....	105
D.3 Testprogramm für ein IIR-Filter mit Umspeichern und Erklärungen .....	106
D.4 Testprogramm für ein IIR-Filter ohne Umspeichern mit dem ATmega32.....	107
Anhang E Anhang zu Kapitel 5 .....	108
E.1 Testprogramm für ein IIR-Filter mit dem "TMS320C6713 DSK-Board" .....	108
E.2 Realisierung der "Ti-TP Kette" mit dem "AIC23" on-board-Codec .....	109
E.3 Realisierung eines IIR-Filters mit dem "ADDA16" Aufsteckmodul .....	113
E.4 Johnson Counter .....	114
E.5 Angebote der Firma GBM.....	115
E.6 Verwendete Wandler-ICs und Bauteile .....	117
Anhang F Anhang zu Kapitel 6 .....	118
F.1 "J3" Connector.....	118
F.2 Register zur Initialisierung der McBSP-Pins als GPIO .....	119
F.3 Register zur Initialisierung der externen Interrupts-Pins als GPIO .....	120
F.4 GPIO-Programm mit Ausgabeoptimierungen .....	122
F.5 "J4" EMIF-Connector.....	126
F.6 Schaltbild des EB-9752 .....	127
F.7 Active Filter Design Tool.....	128
F.8 EMIF-Register .....	129
Anhang H CD mit elektronischer Ausführung der Diplomarbeit .....	in der Rückseite

A.1

Kanalnachbildung bis Seite 91\_1

Kanalnachbildung bis Seite 91\_2

Kanalnachbildung bis Seite 91\_3

Kanalnachbildung bis Seite 91\_4

A2

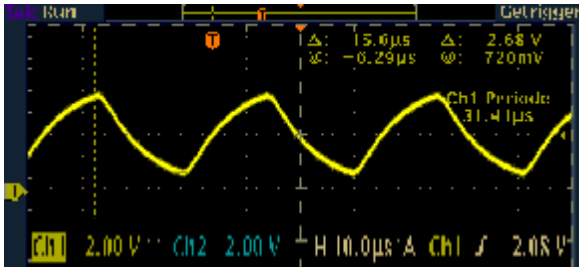
ESB "Ti-TP Kette"

B1

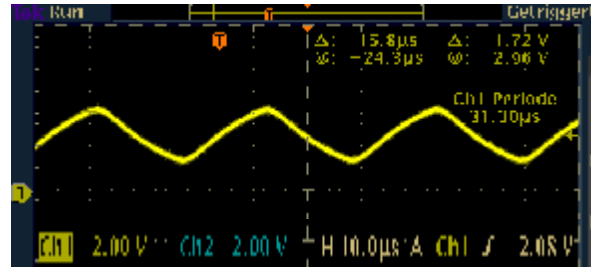
ESB HP

## Anhang C Anhang zu Kapitel 3

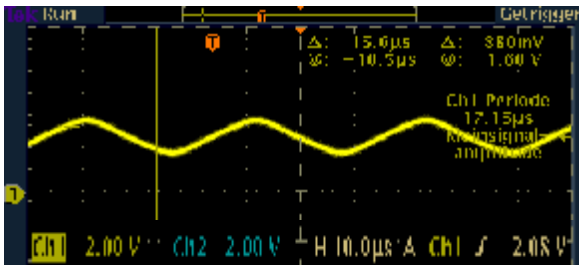
### C.1 Messergebnisse der Spannungsmessung am Ausgang der Ti-TP Kette bei OZ1 bis OZ8:



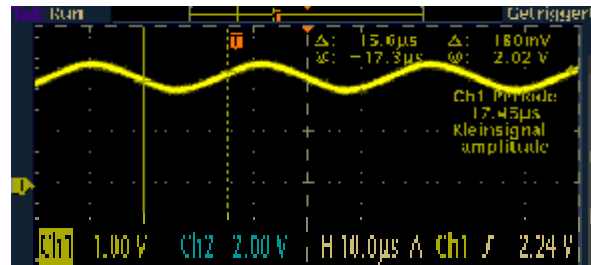
**Abb. C.1:**  
Ausgangssignal der Ti-TP Kette bei OZ3



**Abb. C.2:**  
Ausgangssignal der Ti-TP Kette bei OZ4

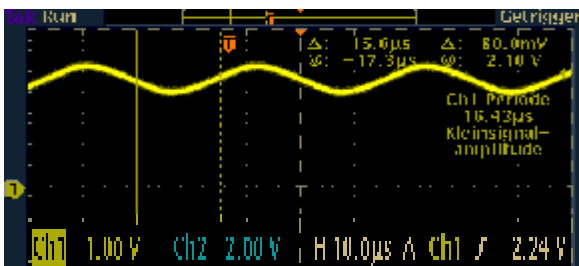


**Abb. C.3:**  
Ausgangssignal der Ti-TP Kette bei OZ5

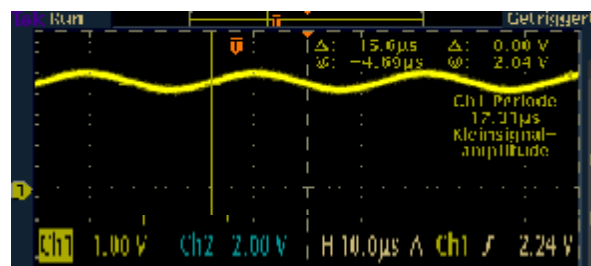


**Abb. C.4:**  
Ausgangssignal der Ti-TP Kette bei OZ6 mit vergrößerter Auflösung

Da die Signaländerung bei der eingestellten Ordnungszahl 6 kaum noch zu erkennen ist, wurde die Auflösung der Ordinate ab der Ordnungszahl 6 auf 1V/div vergrößert.



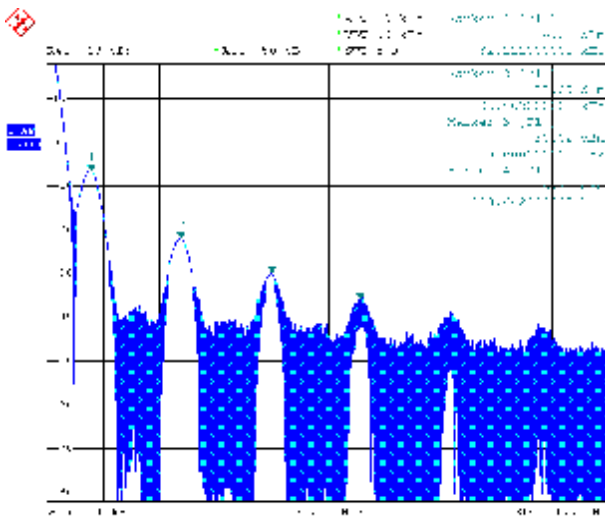
**Abb. C.5:**  
Ausgangssignal der Ti-TP Kette bei OZ7 mit vergrößerter Auflösung



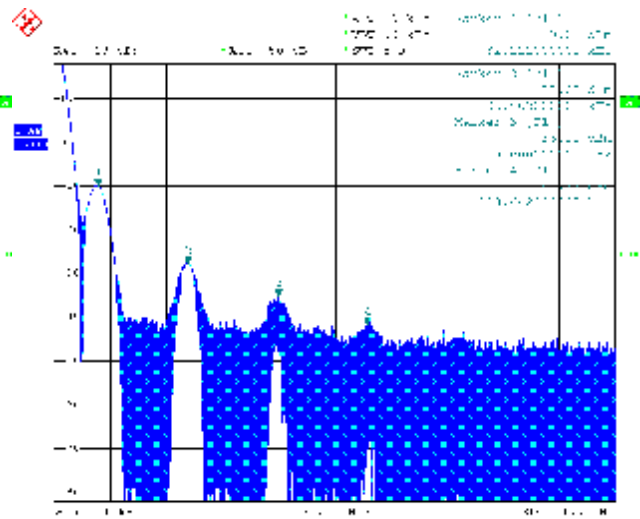
**Abb. C.6:**  
Ausgangssignal der Ti-TP Kette bei OZ7 mit vergrößerter Auflösung



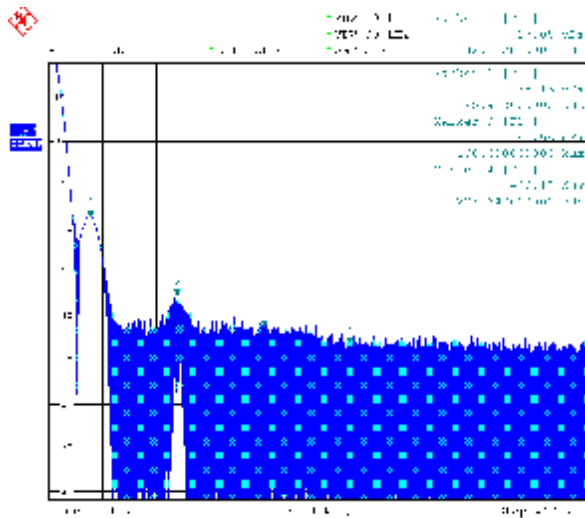
## C.2 Auswirkung auf das Signalspektrum des periodischen Rechtecks durch die Ti-TP Kette bei OZ1 bis OZ8:



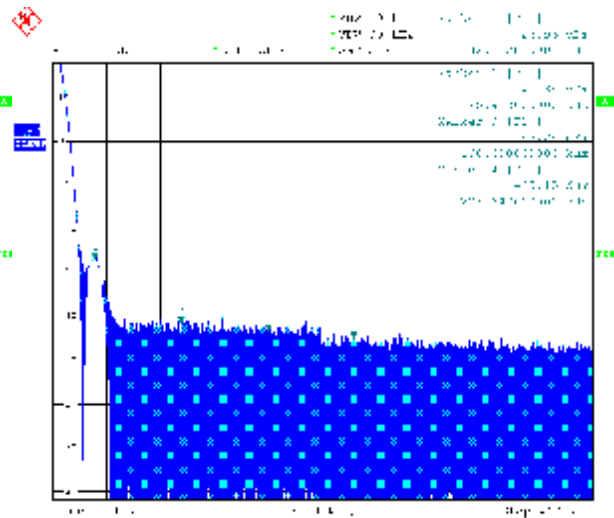
**Abb. C.7:**  
Signalspektrum am Ausgang der "Ti-TP" Kette bei OZ2



**Abb. C.8:**  
Signalspektrum am Ausgang der "Ti-TP" Kette bei OZ3



**Abb. C.9:**  
Signalspektrum am Ausgang der "Ti-TP Kette" bei OZ5



**Abb. C.10:**  
Signalspektrum am Ausgang der "Ti-TP" Kette bei OZ8

### C.3 Spannungs-/ Impedanzanalyse für OZ4 und OZ5

Übertragungsfunktion der nichtentkoppelten RC-Kette 4. Ordnung:

$$U_1 = \frac{U_{IN} \cdot \frac{1}{sC + \frac{1}{R + \frac{1}{sC + \frac{1}{R + \frac{1}{sC + \frac{1}{R + \frac{1}{sC + \frac{1}{R + \frac{1}{sC}}}}}}}}}}{1}$$

Diese Formel lässt sich z.B. mit dem Texas Instruments TI89 bestimmen.

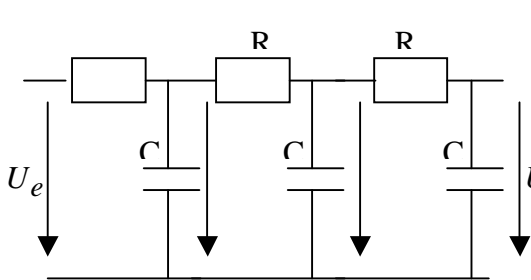
Übertragungsfunktion der nichtentkoppelten RC-Kette 5. Ordnung:

$$U_1 = \frac{U_{IN} \cdot \frac{1}{sC + \frac{1}{R + \frac{1}{sC + \frac{1}{R + \frac{1}{sC + \frac{1}{R + \frac{1}{sC + \frac{1}{R + \frac{1}{sC}}}}}}}}}}{1}$$

Diese Formel lässt sich z.B. mit dem Texas Instruments TI89 bestimmen.

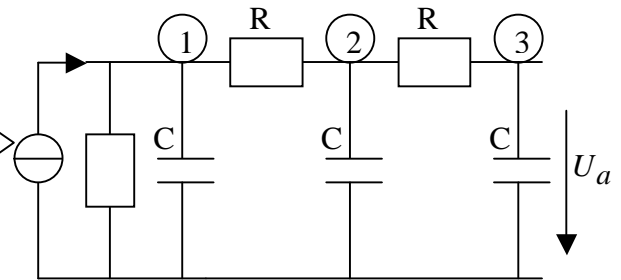
## C.4 Knotenpotentialanalyse

### 3. Ordnung:



**Abb. C.11:**  
RC-Glied 2. Ordnung mit Spannungsquelle

Spannungs-  
stromquellen  
wandeln



**Abb. C.12:**  
RC-Glied 2. Ordnung mit Spannungsquelle

### Aufstellen der Leitwertmatrix:

$$\begin{bmatrix} \frac{2}{R} + s_g C & -\frac{1}{R} & 0 \\ -\frac{1}{R} & \frac{2}{R} + s_g C & -\frac{1}{R} \\ 0 & -\frac{1}{R} & \frac{1}{R} + s_g C \end{bmatrix} = \underline{Y}$$

$$\underline{Y} \underline{g} \underline{U} = \underline{I} = \begin{bmatrix} U_e / R \\ 0 \\ 0 \end{bmatrix}$$

wobei  $\frac{1}{R} + \frac{1}{R} + s_g C$  der Summe der Leitwerte an Knoten 1 und 2,  $\frac{1}{R} + s_g C$  der Summe der Leitwerte an Knoten 2 und  $-\frac{1}{R}$  dem negativen Leitwert zwischen Knoten 1 und 2 bzw. 2 und 3 entspricht.

Mit Hilfe des Software-Tools Mathematica 5.0 wurde die Leitwertmatrix gelöst:

$$\text{Solve}\left[\left\{\left\{\frac{2}{R} + s_g C, -\frac{1}{R}, 0\right\}, \left\{-\frac{1}{R}, \frac{2}{R} + s_g C, -\frac{1}{R}\right\}, \left\{0, -\frac{1}{R}, \frac{1}{R} + s_g C\right\}\right\} \underline{g} \{U_1, U_2, U_a\} = \left\{\frac{U_e}{R}, 0, 0\right\}, \{U_1, U_2, U_a\}\right]$$

Ergebnis:

$$\frac{ua}{ue} = \frac{1}{1 + 6sRC + 5g(sRC)^2 + (sRC)^3}$$

## 5. Ordnung:

Aufstellen der Leitwertmatrix:

$$\begin{bmatrix} \frac{2}{R} + s_g C & -\frac{1}{R} & 0 & 0 & 0 \\ -\frac{1}{R} & \frac{2}{R} + s_g C & -\frac{1}{R} & 0 & 0 \\ 0 & -\frac{1}{R} & \frac{2}{R} + s_g C & -\frac{1}{R} & 0 \\ 0 & 0 & -\frac{1}{R} & \frac{2}{R} + s_g C & -\frac{1}{R} \\ 0 & 0 & 0 & -\frac{1}{R} & \frac{1}{R} + s_g C \end{bmatrix} = \underline{Y} \quad \underline{Y} \underline{g} \underline{U} = \underline{I} = \begin{bmatrix} U_e / R \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

wobei  $\frac{1}{R} + \frac{1}{R} + s_g C$  der Summe der Leitwerte an den Knoten,  
 $\frac{1}{R} + s_g C$  der Summe der Leitwerte am Endknoten und  
 $-\frac{1}{R}$  dem negativen Leitwert zwischen den Knoten entspricht.

Mit Hilfe des Software-Tools Mathematica 5.0 wurde die Leitwertmatrix gelöst:

$$\text{Solve} \left[ \left\{ \left\{ \frac{2}{R} + s_g C, -\frac{1}{R}, 0, 0, 0 \right\}, \left\{ -\frac{1}{R}, \frac{2}{R} + s_g C, -\frac{1}{R}, 0, 0 \right\}, \left\{ 0, -\frac{1}{R}, \frac{2}{R} + s_g C, -\frac{1}{R}, 0 \right\}, \right. \right. \\ \left. \left. \left\{ 0, 0, -\frac{1}{R}, \frac{2}{R} + s_g C, -\frac{1}{R} \right\}, \left\{ 0, 0, 0, -\frac{1}{R}, \frac{1}{R} + s_g C \right\} \right. \right. \\ \left. \left. \underline{g} \{U_1, U_2, U_3, U_4, U_a\} = \left\{ \frac{U_e}{R}, 0, 0, 0, 0 \right\}, \{U_1, U_2, U_3, U_4, U_a\} \right. \right. \left. \left. \right\} \right]$$

Ergebnis:

$$\frac{ua}{ue} = \frac{1}{1 + 15sRC + 35g(sRC)^2 + 28g(sRC)^3 + 9g(sRC)^4 + (sRC)^5}$$

## 8. Ordnung:

Aufstellen der Leitwertmatrix:

$$\begin{bmatrix}
 \frac{2}{R} + s\mathbf{g}C & -\frac{1}{R} & 0 & 0 & 0 & 0 & 0 & 0 \\
 -\frac{1}{R} & \frac{2}{R} + s\mathbf{g}C & -\frac{1}{R} & 0 & 0 & 0 & 0 & 0 \\
 0 & -\frac{1}{R} & \frac{2}{R} + s\mathbf{g}C & -\frac{1}{R} & 0 & 0 & 0 & 0 \\
 0 & 0 & -\frac{1}{R} & \frac{2}{R} + s\mathbf{g}C & -\frac{1}{R} & 0 & 0 & 0 \\
 0 & 0 & 0 & -\frac{1}{R} & \frac{2}{R} + s\mathbf{g}C & -\frac{1}{R} & 0 & 0 \\
 0 & 0 & 0 & 0 & -\frac{1}{R} & \frac{2}{R} + s\mathbf{g}C & -\frac{1}{R} & 0 \\
 0 & 0 & 0 & 0 & 0 & -\frac{1}{R} & \frac{2}{R} + s\mathbf{g}C & -\frac{1}{R} \\
 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{R} & \frac{1}{R} + s\mathbf{g}C
 \end{bmatrix} = \underline{\underline{\mathbf{Y}}}$$

$$\underline{\underline{\mathbf{Y}}} \cdot \underline{\underline{\mathbf{U}}} = \underline{\underline{\mathbf{I}}} = \begin{bmatrix} U_e/R \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

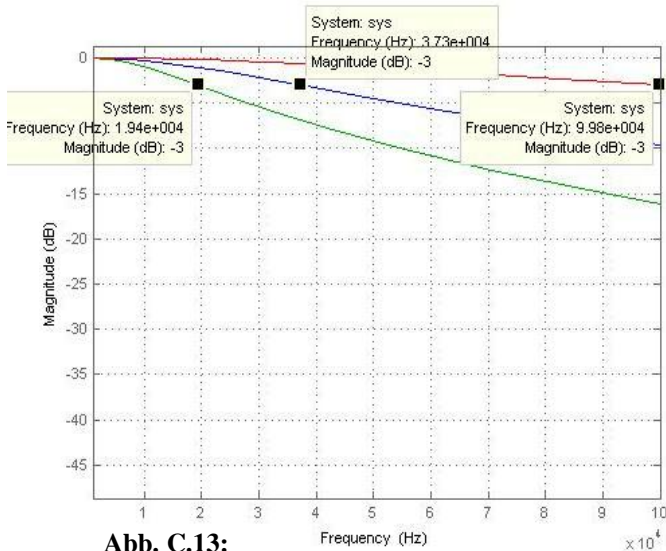
Mit Hilfe des Software-Tools Mathematica 5.0 wurde die Leitwertmatrix gelöst:

$$\text{Solve} \left[ \left\{ \left\{ \frac{2}{R} + s\mathbf{g}C, -\frac{1}{R}, 0, 0, 0, 0, 0, 0 \right\}, \left\{ -\frac{1}{R}, \frac{2}{R} + s\mathbf{g}C, -\frac{1}{R}, 0, 0, 0, 0, 0 \right\}, \left\{ 0, -\frac{1}{R}, \frac{2}{R} + s\mathbf{g}C, -\frac{1}{R}, 0, 0, 0, 0 \right\}, \right. \right. \\
 \left. \left. \left\{ 0, 0, -\frac{1}{R}, \frac{2}{R} + s\mathbf{g}C, -\frac{1}{R}, 0, 0, 0 \right\}, \left\{ 0, 0, 0, -\frac{1}{R}, \frac{2}{R} + s\mathbf{g}C, -\frac{1}{R}, 0, 0 \right\}, \left\{ 0, 0, 0, 0, -\frac{1}{R}, \frac{2}{R} + s\mathbf{g}C, -\frac{1}{R}, 0 \right\}, \right. \right. \\
 \left. \left. \left\{ 0, 0, 0, 0, 0, -\frac{1}{R}, \frac{2}{R} + s\mathbf{g}C, -\frac{1}{R} \right\}, \left\{ 0, 0, 0, 0, 0, 0, -\frac{1}{R}, \frac{1}{R} + s\mathbf{g}C \right\} \right\} \right] \\
 \mathbf{g}\{U_1, U_2, U_3, U_4, U_5, U_6, U_7, U_a\} = \left\{ \frac{U_e}{R}, 0, 0, 0, 0, 0, 0, 0 \right\}, \{U_1, U_2, U_3, U_4, U_5, U_6, U_7, U_a\}$$

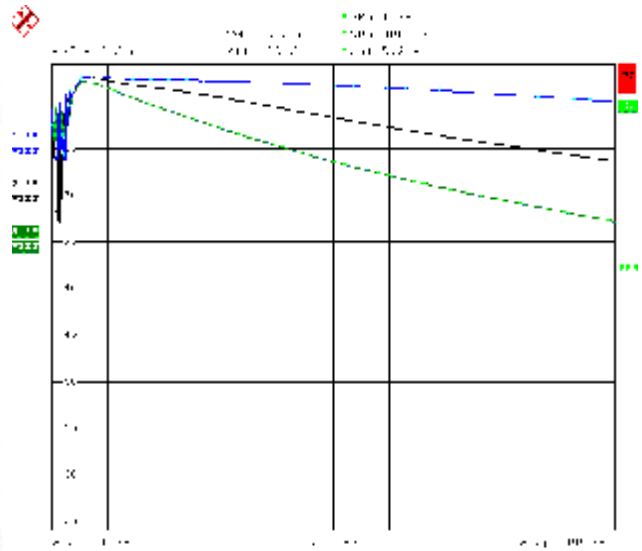
Ergebnis:

$$\frac{ua}{ue} = \frac{1}{1 + 36\mathbf{g}sRC + 210\mathbf{g}(sRC)^2 + 462\mathbf{g}(sRC)^3 + 495\mathbf{g}(sRC)^4 + 286\mathbf{g}(sRC)^5 + 91\mathbf{g}(sRC)^6 + 15\mathbf{g}(sRC)^7 + (sRC)^8}$$

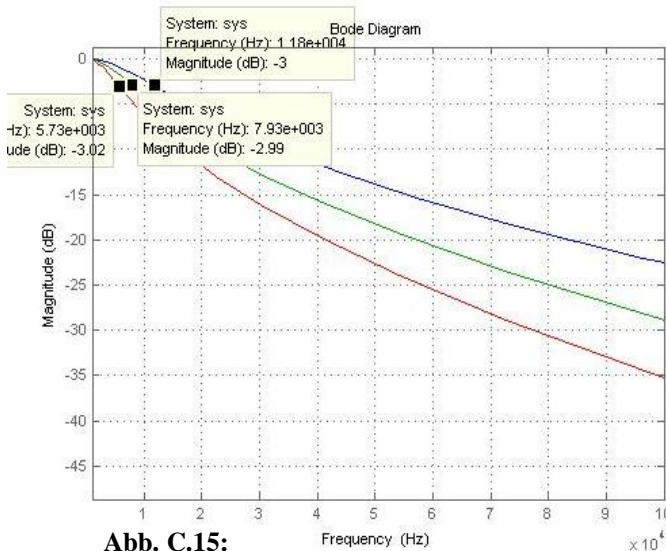
## C.5 Bestimmung des Amplitudenganges der ermittelten Übertragungsfunktionen mit Matlab zum Vergleich mit der Messung am Modell



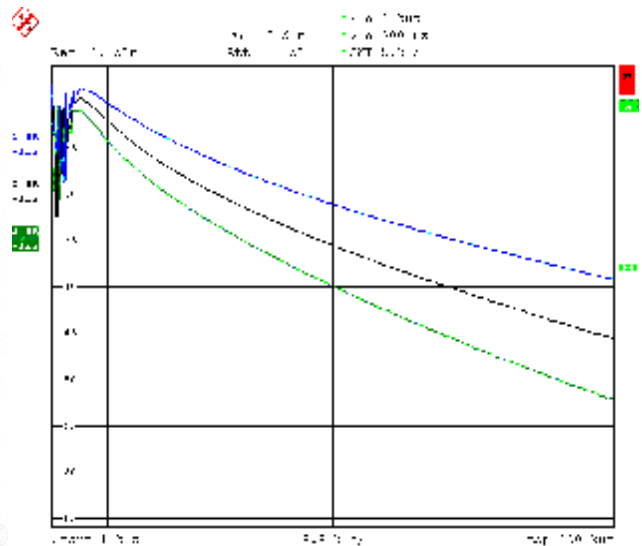
**Abb. C.13:**  
Simulation des Frequenzganges der ermittelten Übertragungsfunktion mit MatLab für OZ1 bis OZ3



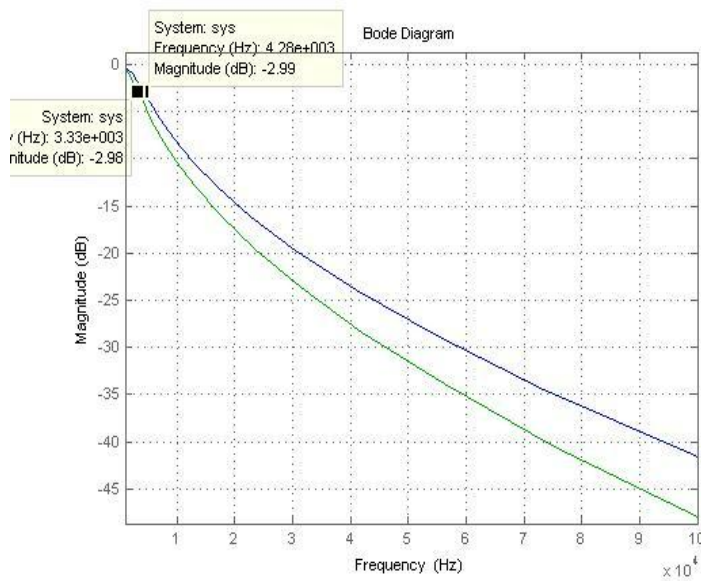
**Abb. C.14:**  
Messung des Frequenzganges der "Ti-TP-Kette" mit dem FSP3 für OZ1 bis OZ3



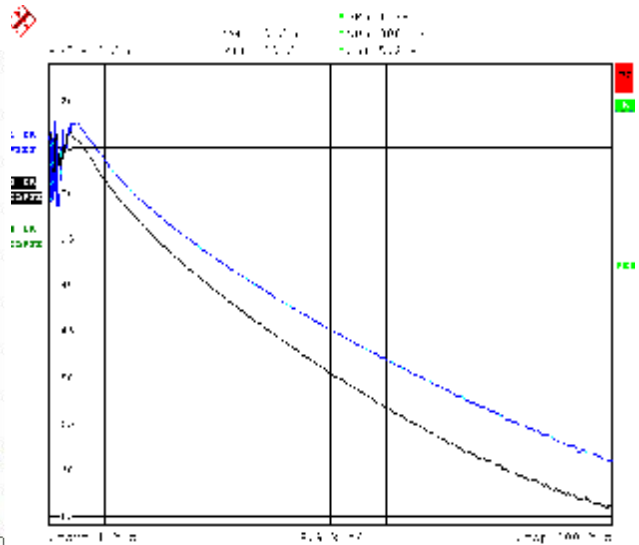
**Abb. C.15:**  
Simulation des Frequenzganges der ermittelten Übertragungsfunktion mit MatLab für OZ4 bis OZ6



**Abb. C.16:**  
Messung des Frequenzganges der Ti-TP-Kette mit dem FSP3 für OZ4 bis OZ6



**Abb. C.17:**  
Simulation des Frequenzganges der ermittelten Übertragungsfunktion mit Matlab für OZ7 und OZ8



**Abb. C.18:**  
Messung des Frequenzganges der Ti-TP-Kette mit dem FSP3 für OZ7 und OZ8

## C.6 Bestimmung der Grenzfrequenz

### 3. Ordnung :

$$G(s) = \frac{1}{(sT)^3 + 5g(sT)^2 + 6gsT + 1} \Rightarrow \frac{1}{\sqrt{2}} = \left| \frac{1}{(j\omega T)^3 + 5g(j\omega T)^2 + 6gj\omega T + 1} \right|$$

$$\frac{1}{\sqrt{2}} = \frac{1}{\sqrt{(1-5g(\omega T)^2)^2 + (6\omega T - (\omega T)^3)^2}} = \frac{1}{\sqrt{(1-10g(\omega T)^2 + 25g(\omega T)^4) + (36g(\omega T)^2 - 12g(\omega T)^4 + (\omega T)^6)}}$$

$$2 = (\omega T)^6 + 13(\omega T)^4 + 26(\omega T)^2 + 1 \quad \left| -2 \right.$$

$$0 = (\omega T)^6 + 13(\omega T)^4 + 26(\omega T)^2 - 1 \quad \left| \text{auflösen nach } f \right.$$

$$f_{1,2} = \pm 19,429 \text{ KHz}$$

### 4. Ordnung :

$$G(s) = \frac{1}{(sT)^4 + 7 \cdot (sT)^3 + 15 \cdot (sT)^2 + 10 \cdot sT + 1} \Rightarrow \frac{1}{\sqrt{2}} = \left| \frac{1}{(j\omega T)^4 + 7 \cdot (j\omega T)^3 + 15 \cdot (j\omega T)^2 + 10 \cdot j\omega T + 1} \right|$$

$$f_{1,2} = \pm 11,849 \text{ KHz}$$

5. Ordnung :

$$G(s) = \frac{1}{(sT)^5 + 9 \cdot (sT)^4 + 28 \cdot (sT)^3 + 35 \cdot (sT)^2 + 15 \cdot sT + 1}$$

$$\frac{1}{\sqrt{2}} = \left| \frac{1}{(j\omega T)^5 + 9 \cdot (j\omega T)^4 + 28 \cdot (j\omega T)^3 + 35 \cdot (j\omega T)^2 + 15 \cdot j\omega T + 1} \right|$$

$$f_{1,2} = \pm 7,967 \text{ KHz}$$

6. Ordnung :

$$G(s) = \frac{1}{(sT)^6 + 11 \cdot (sT)^5 + 45 \cdot (sT)^4 + 84 \cdot (sT)^3 + 70 \cdot (sT)^2 + 21 \cdot sT + 1}$$

$$\frac{1}{\sqrt{2}} = \left| \frac{1}{(j\omega T)^6 + 11 \cdot (j\omega T)^5 + 45 \cdot (j\omega T)^4 + 84 \cdot (j\omega T)^3 + 70 \cdot (j\omega T)^2 + 21 \cdot j\omega T + 1} \right|$$

$$f_{1,2} = \pm 5,719 \text{ KHz}$$

7. Ordnung :

$$G(s) = \frac{1}{(sT)^7 + 13 \cdot (sT)^6 + 66 \cdot (sT)^5 + 165 \cdot (sT)^4 + 210 \cdot (sT)^3 + 126 \cdot (sT)^2 + 28 \cdot sT + 1}$$

$$\frac{1}{\sqrt{2}} = \left| \frac{1}{(j\omega T)^7 + 13 \cdot (j\omega T)^6 + 66 \cdot (j\omega T)^5 + 165 \cdot (j\omega T)^4 + 210 \cdot (j\omega T)^3 + 126 \cdot (j\omega T)^2 + 28 \cdot j\omega T + 1} \right|$$

$$f_{1,2} = \pm 4,302 \text{ KHz}$$

8. Ordnung :

$$G(s) = \frac{1}{(sT)^8 + 15 \cdot (sT)^7 + 91 \cdot (sT)^6 + 286 \cdot (sT)^5 + 495 \cdot (sT)^4 + 462 \cdot (sT)^3 + 210 \cdot (sT)^2 + 36 \cdot sT + 1}$$

$$\frac{1}{\sqrt{2}} = \left| \frac{1}{(j\omega T)^8 + 15 \cdot (j\omega T)^7 + 91 \cdot (j\omega T)^6 + 286 \cdot (j\omega T)^5 + 495 \cdot (j\omega T)^4 + 462 \cdot (j\omega T)^3 + 210 \cdot (j\omega T)^2 + 36 \cdot j\omega T + 1} \right|$$

$$f_{1,2} = \pm 3,353 \text{ KHz}$$



## C.7 Messung des Amplitudenganges mit einem Audioanalysator

1. bis 4. Ordnung :

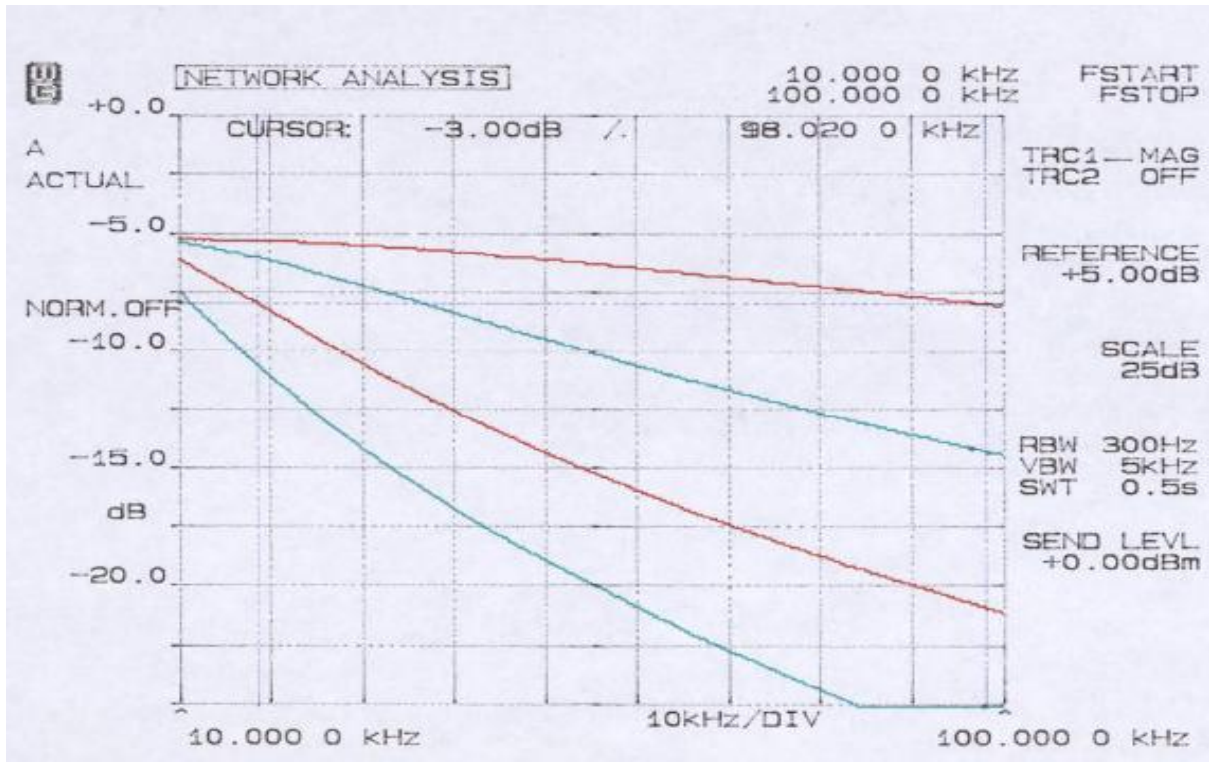


Abb. C.19:

Messergebnis mit dem W&G für OZ1 bis OZ4

5. bis 8. Ordnung :

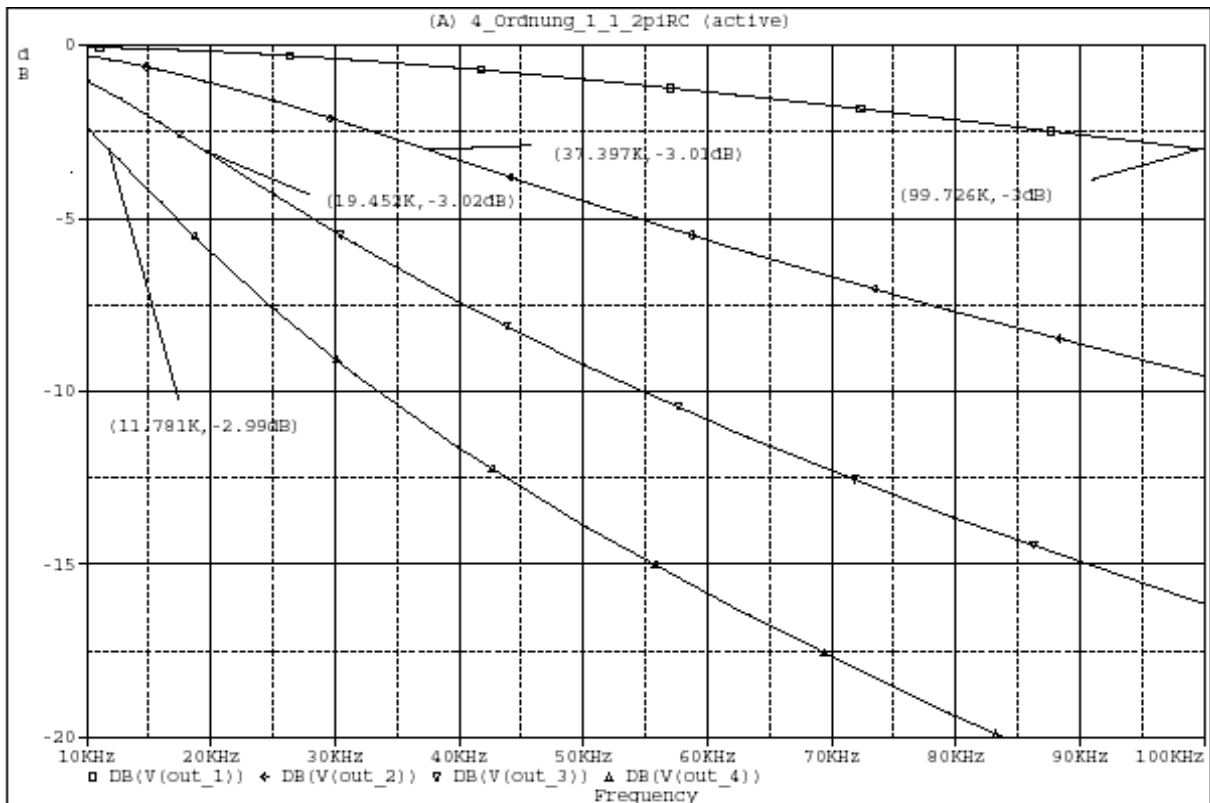


Abb. C.20:

Ergebnis der PSpice Simulation für OZ1 bis OZ4

5. bis 8. Ordnung :

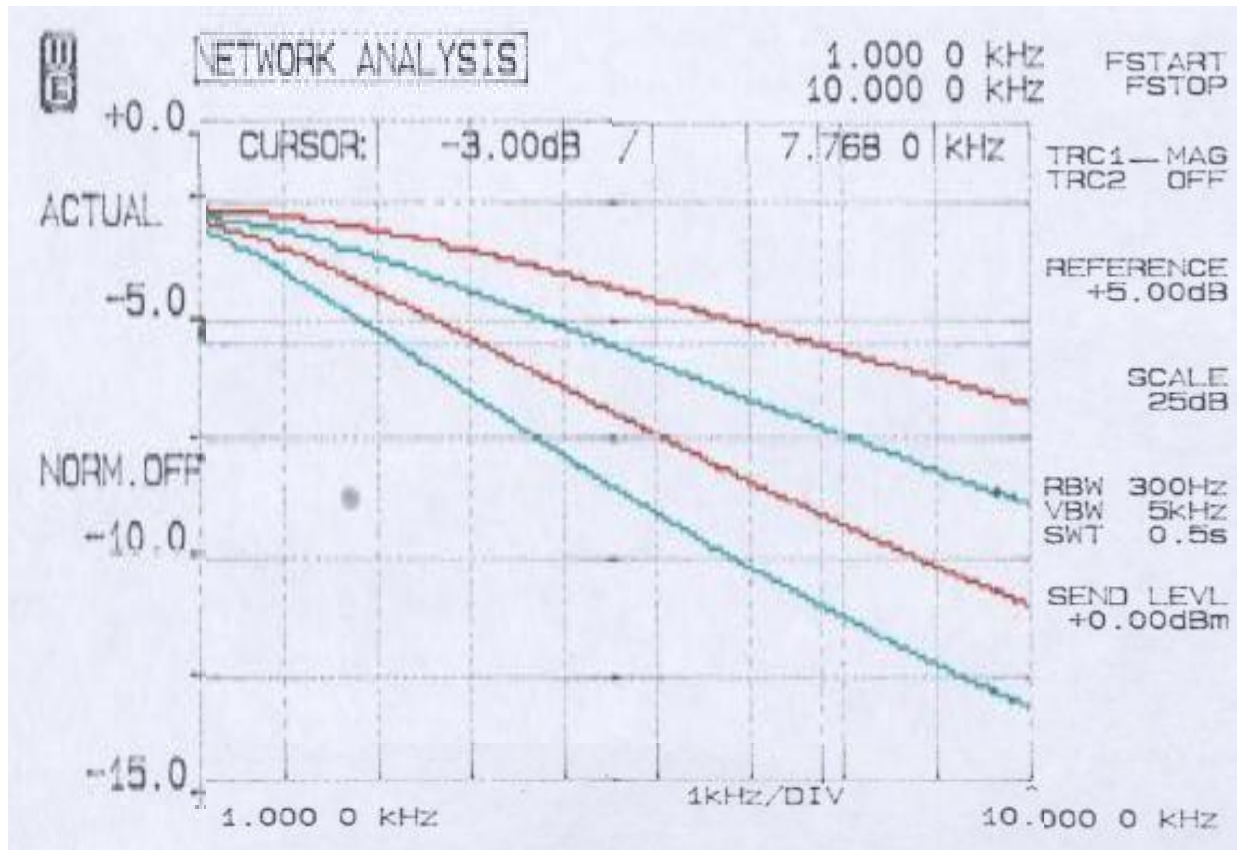


Abb. C.21:  
Messergebnis mit dem W&G für OZ5 bis OZ8

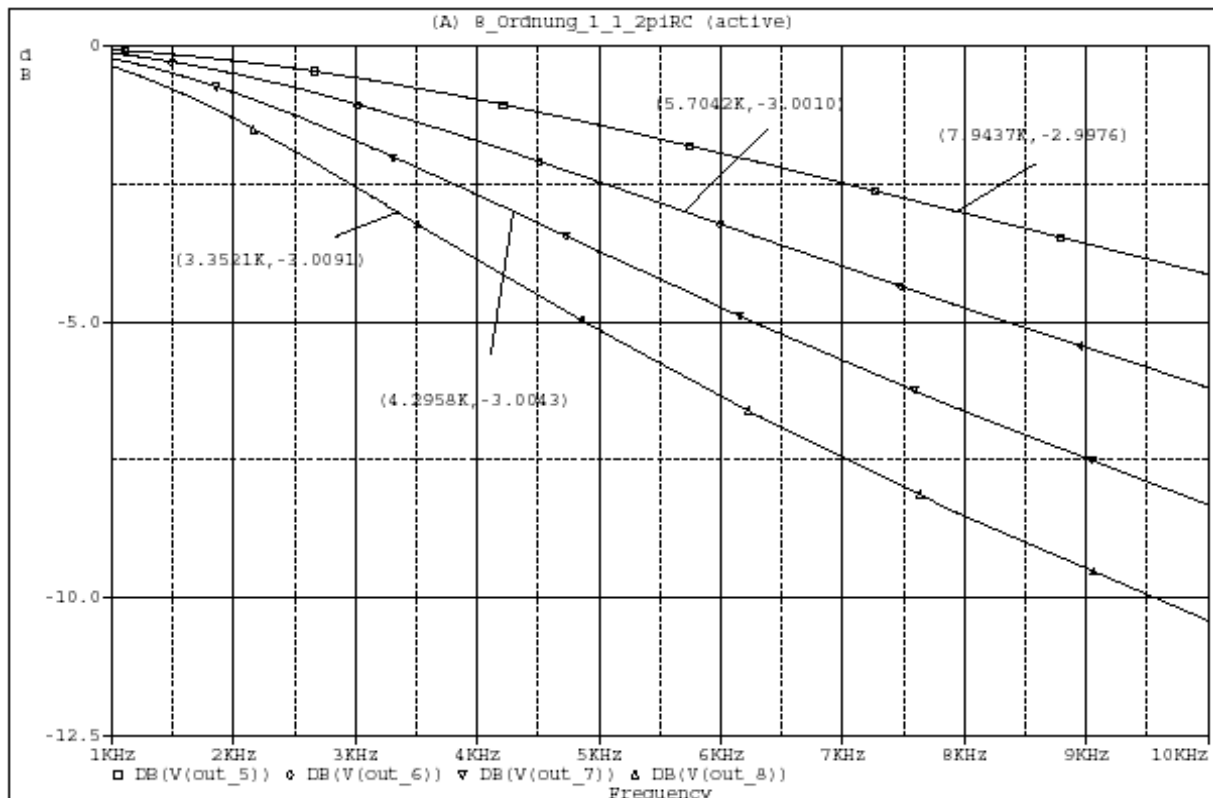


Abb. C.22:  
Ergebnis der PSpice Simulation für OZ5 bis OZ8

## Anhang D Anhang zu Kapitel 4

### D.1 Bestimmung der Übertragungsfunktion mit Hilfe von Matlab

Eingabe In MatLab:

```
[A] = 1
[B] = [(1.59155e-6)^2 3*1.59155e-6 1]
```

Nach der "bilinearen Transformation":

```
[Ab,Bb] = bilinear(A,B,6.4e6)
```

Ergebnis:

```
[Ab] = [0.0021 0.0042 0.0021]
[Bb] = [1 -1.7354 0.7438]
```

Nach der "impulsinvarianten Methode":

```
[Ai,Bi] =impinvar(A,B,6.4e6)
```

Ergebnis:

```
[Ai] = 1e-13* [0 0.2111]
[Bi] = 1e-11* [0.2533 -0.4399 0.1887]
```

jedoch ab der 3. Ordnung folgende Warnung angezeigt wird:

```
>> [Ab,Bb] = bilinear(A,B,6.4e6)
Warning: Matrix is close to singular or badly scaled.
Results may be inaccurate. RCOND = 3.355121e-021.
> In bilinear at 89
Warning: Matrix is close to singular or badly scaled.
Results may be inaccurate. RCOND = 3.355121e-021.
> In bilinear at 90
```

wurde der Befehl "c2dm" (*continious to discrete method*) benutzt, der die gleiche "bilineare Transformation" durchführt:

Eingabe In MatLab:

```
[A] = 1
[B] = [(1.59155e-6)^3 5*(1.59155e-6)^2 6*1.59155e-6 1]
[Ad,Bd] = c2dm(A,B,156.25e-9, 'tustin')
```

### D.2 Ermittelte Koeffizienten für ein digitales IIR-Filter

Die Tabelle zeigt die mit Matlab ermittelten Koeffizienten für das digitale Filter.

OZ	Ü-Funktion 1/X	Bei Samplerate = 6,4MHz Koeffizienten 1.59155µs!!! -> T=2*Pi*R*C=10µs -> R*C =1.59155µs	
		c2dm(A,B,ta,'tustin')	impulsinvariant(A,B,fa)
1	[1 1]	A = [0.0468 0.0468] B = [1 -0.9064]	[1.5625e-7] 1e-5*[0.1592 -0.1443]
2	[1 3 1]	A = [0.0021 0.0042 0.0021] B = [1 -1.7354 0.7438]	1e-13*[0 0.2111] 1e-11*[0.2533 -0.4399 0.1887]

**Tab. D.1a:**

Mit Matlab ermittelte Koeffizienten für das digitale Filter mit  $f_A = 6,4\text{MHz}$

3	[1 5 6 1]	1e-3*[0.0939 0.2816 0.2816 0.0939] B = [1 -2.5640 2.1749 -0.6102]	A = 1e-20*[-0.0 0.1622 0.1378] 1e-16*[0.0403 -0.1035 0.0878 -0.0247]
4	[1 7 15 10 1]	A = 1e-4*[0.0420 0.1682 0.2523 0.1682 0.0420] B = [1 -3.3925 4.2925 -2.4006 0.5006]	A = 1e-27*[-0.0 0.0838 0.2827 0.0594] B = 1e-22*[0.0642 -0.2179 0.2760 -0.1545 0.0323]
5	[1 9 28 35 15 1]	A = 1e-5*[0.0188 0.0942 0.1883 0.1883 0.0942 0.0188] B = [1 -4.2210 7.0966 -5.9399 2.4751 -0.4107]	A = 1e-34*[0 0.0326 0.3009 0.2522 0.0192] B = 1e-28*[0.1021 -0.4315 0.7263 -0.6087 0.2540 -0.0422]
6	[1 11 45 84 70 21 1]	A = 1e-6*[0.0084 0.0506 0.1265 0.1687 0.1265 0.0506 0.0084] B = [1 -5.0495 10.5871 -11.7969 7.3677 -2.4454 0.3370]	A = 1e-41*[0 0.0101 0.2208 0.4687 0.1541 0.0049] B = 1e-33*[0.0163 -0.0822 0.1725 -0.1924 0.1203 -0.0400 0.0055]
7	[1 13 66 165 210 126 28 1]	A = 1e-7*[0.0038 0.0264 0.0793 0.1322 0.1322 0.0793 0.0264 0.0038] B = [1 -5.8781 14.7640 -20.5402 17.0942 -8.5100 2.3465 -0.2764]	A = 1e-48*[-0.0 0.0026 0.1255 0.5549 0.4624 0.0726 0.0011] B = 1e-39*[0.0259 -0.1522 0.3828 -0.5333 0.4444 -0.2215 0.0612 -0.0072]
8	[1 15 91 286 495 462 210 36 1]	A = 1e-8*[0.0017 0.0135 0.0474 0.0947 0.1184 0.0947 0.0474 0.0135 0.0017] B = [1 -6.7066 19.6274 -32.7387 34.0413 -22.5940 9.3480 -2.2043 0.2268]	A = 1e-55*[0 0.0006 0.0587 0.4861 0.8208 0.3364 0.0281 0.0002] B = 1e-44*[0.0041 -0.0276 0.0810 -0.1353 0.1409 -0.0936 0.0388 -0.0092 0.0009]

**Tab. D.1b:**

Mit Matlab ermittelte Koeffizienten für das digitale Filter mit  $f_A = 6,4\text{Mhz}$

### D.3 Testprogramm für ein IIR-Filter mit Umspeichern

Schnell bietet sich an, die mathematische Beschreibung des Filters umzusetzen, indem die Übertragungsfunktion  $H(z)$  mit Hilfe der inversen z-Transformation in die Differenzgleichung  $y(n)$  überführt wird und folgende Zuweisungen vorgenommen werden:

$$DGL: \quad y(n) = A_0 g_x(n) + A_1 g_x(n-1) + A_2 g_x(n-2) - B_1 g_y(n-1) - B_2 g_y(n-2)$$

#### Zuweisungen

```
x0 = x(n)
x1 = x(n-1)
x2 = x(n-2)
y1 = y(n)
y2 = y(n-1)
```

Auf diese Weise lässt sich recht einfach ein Filterprogramm entwickeln.

```
float a0,a1,a2,b1,b2,x0=0,x1=0,x2=0,y0=0,y1=0,y2=0;
//Declare local variables
a0 = 0.2083; //Zählerkoeffizienten
a1 = 0.4165;
a2 = 0.2083;
b1 = 0.0327; //Nennerkoeffizienten
b2 = -0.1996;
```

**Abb. D.1a:**

Testprogramm mit Umspeichern

```

while(1)          //Endlosschleife
{
  x0 = read_adc;  //Eingangswert vom AD

  y0 = a0*x0+ a1*x1+ a2*x2- b1*y1- b2*y2;    //Berechnung
  send_dac = y0;      //Ausgabewert zum DA

  //Umspeichern für die nächste Berechnung
  x2 = x1;
  x1 = x0;
  y2 = y1;
  y1 = y0;
}

```

**Abb. D.1b:**  
Testprogramm mit Umspeichern

Da jedoch bei einem Programm in dieser Form das Umspeichern der Registerinhalte, speziell bei höherer Ordnungszahl, sehr zeitaufwendig ist, sollte dieses vermieden werden, indem im kanonischen Netzwerk für jede Additionsstelle eine Zwischenvariable bzw. Formel aufgestellt wird.

Programm ohne Umspeichern:

Deutlich ist zu erkennen, dass die Anzahl der Multiplikationen und Additionen gleich bleibt, während der Umspeichervorgang komplett entfällt. Wie viel Zeit letztendlich durch diese Vorgehensweise gewonnen wird, ist an der später verwendeten Hardware zu ermitteln.

## D.4 Testprogramm für ein IIR-Filter mit dem Atmega32

```

#include <mega32.h>
// funktionen read_adc und send_dac stehen hier
void main (void)
{
  float a0,a1,a2,b1,b2,u0=0,u1=0,u2=0,u3=0; //Declare local variables
  a0 = 0.2083; //Zählerkoeffizienten
  a1 = 0.4165;
  a2 = 0.2083;
  b1 = 0.0327; //Nennerkoeffizienten
  b2 = -0.1996;

  while(1)
  {
    u0 = read_adc; //Eingangswert vom AD,
                  //Für Testzwecke short Wert direkt zuweisen

    u3 = a0*u0+u2;
    u2 = a1*u0-b1*u3+u1;
    u1 = a2*u0-b2*u3;

    send_dac = u3; //Für Testzwecke löschen
  };
}

```

**Abb. D.2:**  
Testprogramm mit Umspeichern

Der Befehl "read\_adc" gibt einen Short-Wert vom AD-Wandler, entsprechend der Höhe des Eingangspegel an das Programm zurück.

Bei der Multiplikation der Float-Filterkoeffizienten mit dem Short-Rückgabewert der Funktion treten mit dem ATmega keine Probleme auf, allerdings wird bei der Ausgabe (*send\_dac*) der Rest hinter dem Komma einfach vernachlässigt. Zu Testzwecken wurde der Funktionsaufruf *send\_dac* herausgenommen und *u0* direkt ein short-Wert zugewiesen.

## Anhang E Anhang zu Kapitel 5

### E.1 Testprogramm für ein IIR-Filter mit dem "TMS320C6713 DSK-Board"

Die Verwendung der "DSK6713\_AIC23 header Datei" hat hierbei keinen Einfluss auf die Berechnungszeit!

```
#include "DSK6713_AIC23.h" //codec-DSK file support

void main ()
{
    double a0,a1,a2,b1,b2,u0=0,u1=0,u2=0,u3=0; //Declare local variables
    short u=0, sample_data;

//      T=10µs
    a0 = 0.0957; //Zählerkoeffizienten
    a1 = 0.1915;
    a2 = 0.0957;
    b1 = -0.5143; //Nennerkoeffizienten
    b2 = -0.1028;

    comm_poll(); //init DSK, codec, McBSP
    DSK6713_LED_init();

    while(1)
    {
        sample_data = 10000; //input sample
        u0 = (double)sample_data; //short/integer to double conversion

        DSK6713_LED_on(0);
        u3 = a0*u0+u2;
        u2 = a1*u0-b1*u3+u1;
        u1 = a2*u0-b2*u3;

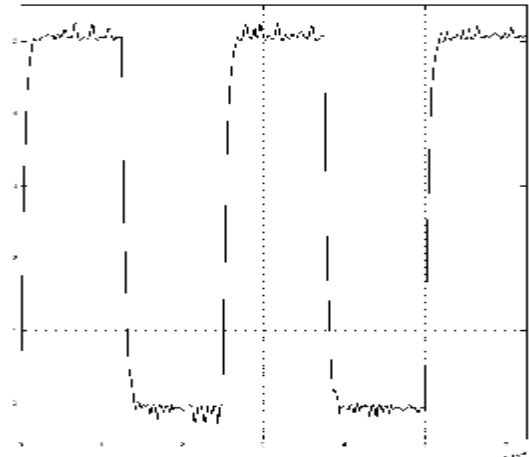
        u = (short)u3; // double to short/integer conversion
        //output_sample(u); //Für Testzwecke löschen
        DSK6713_LED_off(0);
    };
}
```

} ~0,55µs  
=1,818MHz

**Abb. E.1:**  
Testprogramm mit Umspeichern

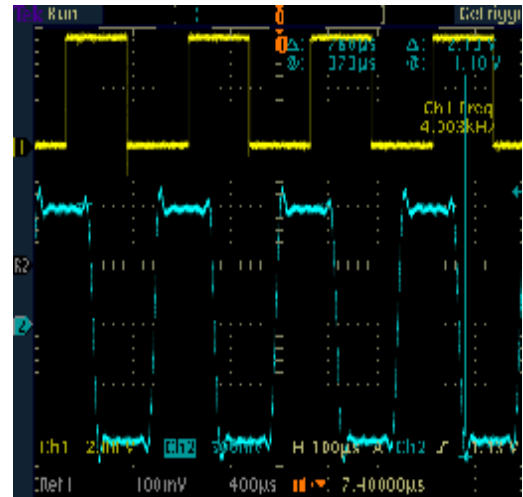
## E2 Realisierung der "Ti-TP Kette" mit dem "AIC23" on-board-Codec

Vor der Optimierung der Berechnungszeit wurde jedoch zu Testzwecken eine Filterversion für ein 8Kbit/s Eingangssignal mit 12-fach-Oversampling ( $\hat{a}$  Ausgangssignal mit 96000Werten/s) programmiert. Da sich der AD- und DA-Wandler beim "AIC23" nicht mit getrennten Abtast- bzw. Update-raten initialisieren lassen, wird einfach jeder 12. Eingangswert ( $96\text{KHz}/8\text{KBit}=12$ ) erfasst. Zuvor wurde diese Version mit Matlab (s.a. Kap. 4.3) simuliert und festgestellt, dass das analoge Filter mit  $T = R \cdot C = 1,59155\mu\text{s}$  so gut wie keinen Einfluss auf das Eingangssignal hat (außer hochfrequente Störungen, (Abb. E. 2 und E. 3). Deshalb wurde die Grenzfrequenz des analogen Filters um den Faktor 8 ( $64\text{KBit}/8\text{KBit}=8$ ) auf 12,5KHz herunterskaliert und der für die Matlab-Rechnung benötigte Faktor  $T=12,7324\mu\text{s}$  ermittelt. Das neu skalierte Filter hat den gleichen Einfluss auf ein 8Kbit/s Eingangssignal, wie das originale Filter auf ein 64Kbit/s Signal. Um dies zu verdeutlichen, wurde das analoge Ausgangssignal aufgenommen



**Abb. E.2:**

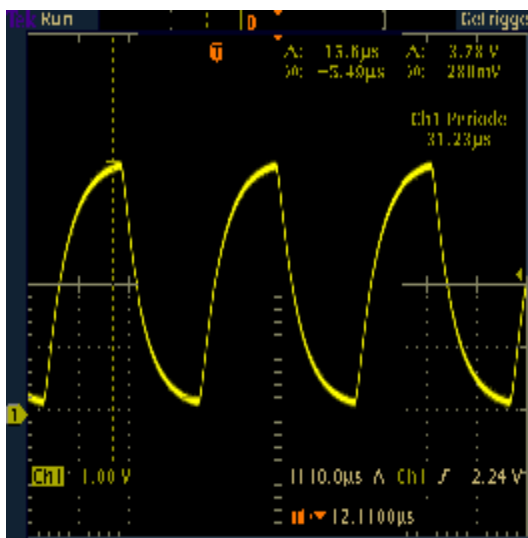
Einfluss des analogen Filters (oz2) mit  $T=1,59155\mu\text{s}$  auf ein 8Kbit/s Eingangssignal ( $100\mu\text{s}/\text{div}$ ) mit MatLab ermittelt



**Abb. E.3:**

Oszilloskopbild des Einflusses vom analogen Filter (oz2) mit  $T=1,59155\mu\text{s}$  auf ein 8Kbit/s Eingangssignal

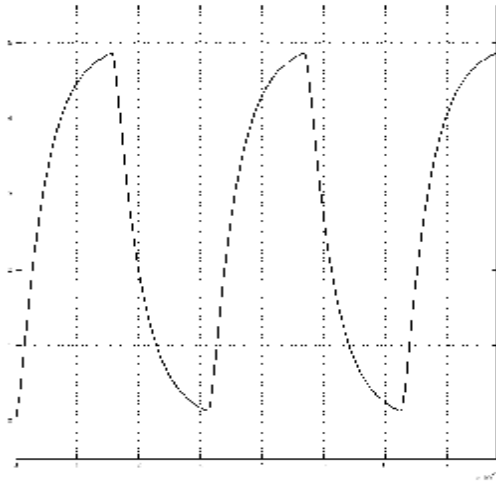
(Abb. E.4), mit dem Ziel exakt das gleiche Verhalten in Matlab zu simulieren.



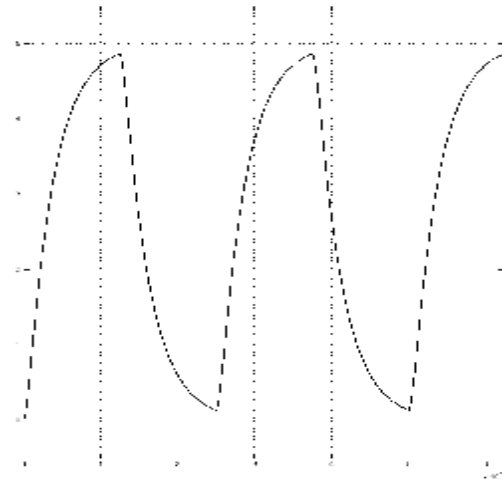
Deutlich erkennt man in der Matlab-Simulation die exakt gleiche Beeinflussung durch den angepassten Filter auf ein 8Kbit/s Eingangssignal (Abb. E.5) wie durch den nachzubauenden Filter auf ein 64Kbit/s Eingangssignal (Abb. E.6). Da zur Zeit noch kein Codec mit ausreichender Ausgabegeschwindigkeit zur Verfügung steht, wurde vorerst dieses neu dimensionierte Filter zu Testzwecken verwendet.

**Abb. E.4:**

Einfluss vom analogen Filter (oz2) mit  $T=1,59155\mu\text{s}$  auf ein 64Kbit/s Eingangssignal



**Abb. E.5:**  
Einfluss des analogen Filters (oz2) mit  $f_g=37,424\text{KHz}$  auf ein  $64\text{Kbit/s}$  Eingangssignal ( $10\mu\text{s/div}$ )



**Abb. E.6:**  
Einfluss des angepassten analogen Filters (oz2) mit  $f_g=4,678\text{KHz}$  auf ein  $8\text{Kbit/s}$  Eingangssignal ( $100\mu\text{s/div}$ )

Es folgen die nach Kap. 4.1 ermittelten Koeffizienten für den neu dimensionierten Filter mit  $T = 12,7324\mu\text{s}$  und  $f_A = 96\text{KHz}$ .

Bei Samplerate = 96KHz			
Koeffizienten $12,7324\mu\text{s}!!! \rightarrow T=2*\pi*R*C=80\mu\text{s} \rightarrow R*C=12,7324\mu\text{s}$			
OZ	Ü-Funktion 1/X	c2dm(A,B,ta,'tustin')	impulsinvariant(A,B,fa)
1	[1 1]	A = [0.2903 0.2903] B = [1 -0.4194]	A = 1.0417e-005 1e-5*[1 -0.3529]
2	[1 3 1]	A = [0.0699 0.1398 0.0699] B = [1 -0.6955 -0.0250]	A = 1e-10*[0 0.3643] B = 1e-9*[0.1621 -0.1376 0.0139]
3	[1 5 6 1]	A = [0.0166 0.0499 0.0499 0.0166] B = [1 -0.9316 0.0379 0.0267]	A = 1e-15*[-0.0 0.1640 0.0421] 1e-14*[0.2064 -0.2479 0.0656 -0.0035]
4	[1 7 15 10 1]	A = [0.0040 0.0158 0.0237 0.0158 0.0040] B = [1 -1.1637 0.1620 0.0663 -0.0014]	A = 1e-21*[-0.0 0.5235 0.5591 0.03] B = 1e-19*[0.2628 -0.4072 0.1788 -0.0241 0.0009]
5	[1 9 28 35 15 1]	A = [0.0009 0.0047 0.0094 0.0094 0.0047 0.0009] B = [1 -1.3955 0.3408 0.0998 -0.0136 -0.0014]	A = 1e-26*[0 0.1290 0.3804 0.0875 0.0016] B = 1e-24*[0.3346 -0.6351 0.3899 -0.0912 0.0079 -0.0002]

**Tab. E.1a:**

Koeffizienten des digitalen Filters mit  $T = 12,7234\mu\text{s}$  bei  $f_A = 96\text{KHz}$



Koeffizienten 1.59155µs!!! --> $T=2\pi R^*C=10\mu s$ --> $R^*C=1.59155\mu s$			
1	[1 1]	A = [0.7659 0.7659] B = [1 0.5319]	A = 1.0417e-5 B = 1e-4*[0.1273 -0.0562]
2	[1 3 1]	A = [0.4975 0.9950 0.4975] B = [1 0.9021 0.0879]	A = 1e-12*[0 0.6086] B = 1e-11*[0.2533 -0.0208 0]

**Tab. E.1b:**

Koeffizienten des digitalen Filters mit  $T = 12,7234\mu s$  bei  $f_A = 96KHz$

Es folgt das Filterprogramm (OZ2) mit dem AIC23-Codec für ein 8KBit/s Eingangssignal mit 12-fach Oversampling, d.h. Ausgangssignal mit 96Kbit/s.

```
#include "DSK6713_AIC23.h" //codec-DSK file support
uint32 fs=DSK6713_AIC23_FREQ_96KHZ; //set sampling rate

void main ()
{
    short sample_data;
    double a0,a1,a2,b1,b2,u0=0,u1=0,u2=0,u3=0; //Declare local variables
    short u=0;
    int j=12,k=0;

//T=12.7324µs bilinear | T=12.7324µs | T=1.59155µs |
| impulsinvariant | bilinear |
a0 = 0.0699; // Zähler- | 0 | 0.4975
a1 = 0.1398; // koeffizienten | 0.3643e-10 | 0.9950
a2 = 0.0699; // | 0.1621e-9 | 0.4975
b1 = -0.6955; // Nenner- | -0.1376e-9 | 0.9021
b2 = -0.0250; // koeffizienten | 0.0139e-9 | 0.0879

    comm_poll(); //init DSK, codec, McBSP

    while(1)
    {
        if (j==12)
        {
            sample_data = input_sample(); //input sample
            u0 = (double)sample_data; //short/integer to double conversion
            //u0 = sample_data; //Eingangswert vom AD,
            j=0;
            k=0;
        };

//IIR 2.Ordnung mit 100-fach Oversampling
        if (k<12)
        {
            //DSK6713_LED_on(0);
            u3 = a0*u0+u2;
            u2 = a1*u0-b1*u3+u1;
            u1 = a2*u0-b2*u3;
            //DSK6713_LED_off(0);

            u = (short)u3; //double to short/integer conversion
            output_sample(u);
            //output_sample(sample_data); //output sample
            k=k+1;
        }
    }
}
```

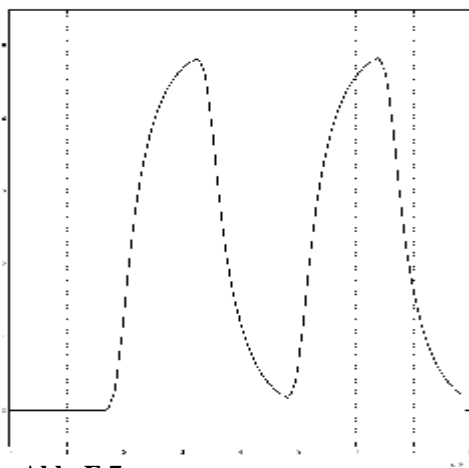
```

};
j=j+1;
};
}

```

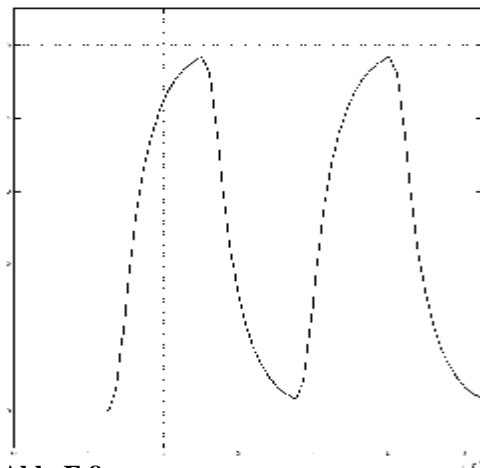
Auch wurde ein Programm ohne Oversamplingfaktor untersucht, da der neu eingelesene Wert  $u_0$  dem Wert, der für  $u_0$  bei der Oversamplingrechnung benutzt wird, entspricht (siehe Abb. 10.9). Dies ist bei Digitalsignalen am Filtereingang immer möglich.

### Funktionsprüfung / Messung (zeitverhalten)



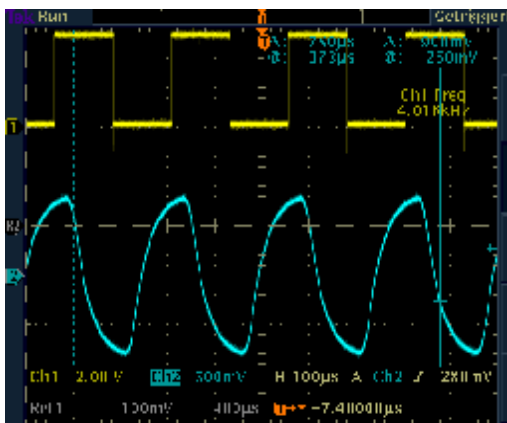
**Abb. E.7:**

Einfluss des digitalen Filters (*oz2, bilinear*,  $f_A = 768\text{KHz}$ ) mit  $f_g=37,424\text{KHz}$  auf ein  $64\text{Kbit/s}$  Eingangssignal ITP:  $f_g=1,5\text{MHz}$ , OZ2, Butterworth ( $10\mu\text{s/div}$ )



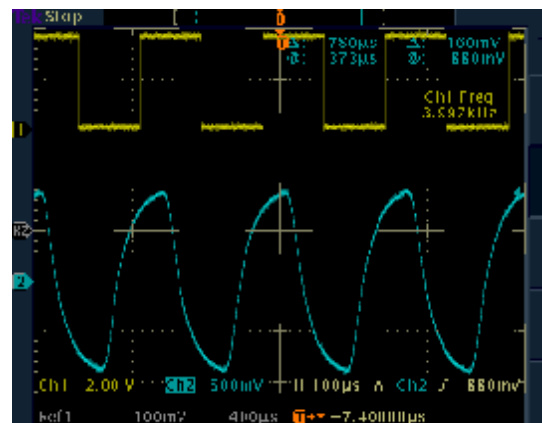
**Abb. E.8:**

Einfluss des digitalen Filters (*oz2, bilinear*,  $f_A = 96\text{KHz}$ ) mit  $f_g=4,678\text{KHz}$  auf ein  $8\text{Kbit/s}$  Eingangssignal ITP:  $f_g=200\text{KHz}$ , OZ2, Butterworth ( $100\mu\text{s/div}$ )



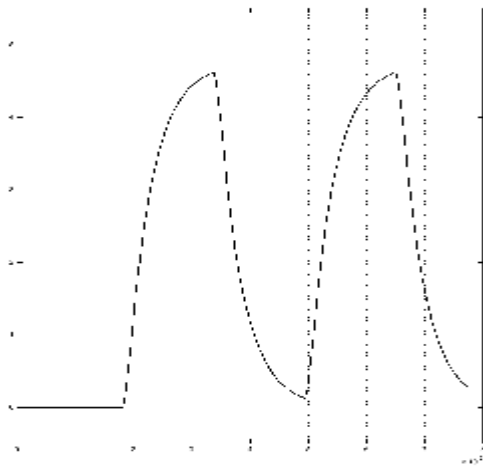
**Abb. E.9:**

Einfluss des digitalen Filters (*oz2, bilinear*) auf ein  $8\text{Kbit/s}$  Eingangssignal ohne Oversampling

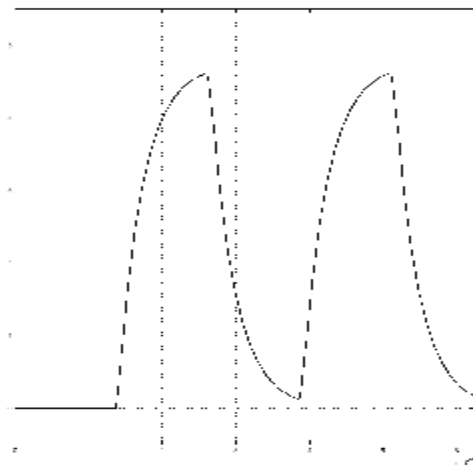


**Abb. E.10:**

Einfluss des digitalen Filters (*oz2, bilinear*) auf ein  $8\text{Kbit/s}$  Eingangssignal mit 12-fach-Oversampling



**Abb. E.11:**  
Einfluss des digitalen Filters (*oz2, impulsinvariant*) mit  $f_g=37,424\text{KHz}$  auf ein  $64\text{Kbit/s}$  Eingangssignal ITP:  $f_g=1,5\text{MHz}$ , OZ2, Butterworth ( $10\mu\text{s/div}$ )



**Abb. E.12**  
Einfluss des digitalen Filters (*oz2, impulsinvariant*) mit  $f_g=4,678\text{KHz}$  auf ein  $8\text{Kbit/s}$  Eingangssignal ITP:  $f_g=200\text{KHz}$ ,

Deutlich erkennt man, wie durch die Simulation erwartet, ein Ausgangssignal ohne das spitze Signalmaxima, was durch den relativ geringen Oversampling-Faktor von 12 entsteht (Abb. 10.10). Auch ein Programm, bei dem mit  $96\text{KHz}$  eingelesen und ausgegeben wird (Abb. 10.9), bringt keine Verbesserung.

### E.3 Realisierung eines IIR-Filters mit dem "ADDA16" Aufsteckmodul

Zur Ermittlung der in der Praxis benötigten Oversamplingrate (*evtl. abweichend von der Matlab-Simulation*) wurde der schnellste vorhandene Codec (*ADDA16*) benutzt und das gleiche Programm wie für den AIC23 mit ein paar Anpassungen verwendet. D.h. ein Eingangssignal von  $8\text{Kbit/s}$ , dann 50-fach-Oversampling  $\hat{a}$  Ausgangssignal mit  $400\text{K}$  Werten/s.

OZ	Ü-Funktion 1/X	Bei Samplerate = 400KHz	
		Koeffizienten $12,7324\mu\text{s}!!! \rightarrow T=2*\pi*R*C=80\mu\text{s} \rightarrow R*C=12,7324\mu\text{s}$	
1	[1 1]	$c2dm(A,B,ta,'tustin')$ A = [0.0894 0.0894] B = [1 -0.8212]	$impulsinvariant(A,B,fa)$ A = $2.5e-6$ B = $1e-4*[0.1273 -0.1046]$
2	[1 3 1]	A = [0.0074 0.0148 0.0074] B = [1 -1.5188 0.5483]	A = $1e-11*[0 0.4693]$ B = $1e-9*[0.1621 -0.2474 0.0899]$
3	[1 5 6 1]	A = [0.0006 0.0018 0.0018 0.0006] B = [1 -2.2135 1.5837 -0.3653]	A = $1e-17*[-0.0 0.5674 0.4091]$ B = $1e-14*[0.2064 -0.4597 0.3316 -0.0773]$
4	[1 7 15 10 1]	A = $1e-3*[0.0505 0.2018 0.3027 0.2018 0.0505]$ B = [1 -2.9083 3.1017 -1.4360 0.2433]	A = $1e-22*[-0.0 0.0465 0.1327 0.0234]$ B = $1e-19*[0.2628 -0.7697 0.8281 -0.3874 0.0665]$
5	[1 9 28 35 15 1]	A = $1e-4*[0.0417 0.2084 0.4168 0.4168 0.2084 0.0417]$ B = [1 -3.6030 5.1025	A = $1e-28*[0 0.0287 0.2241 0.1574 0.01]$ B = $1e-23*[0.0335 -0.1215 0.1736$

**Tab. E.2a:**

Digitale Koeffizienten für eine Abtastrate von  $400\text{KHz}$

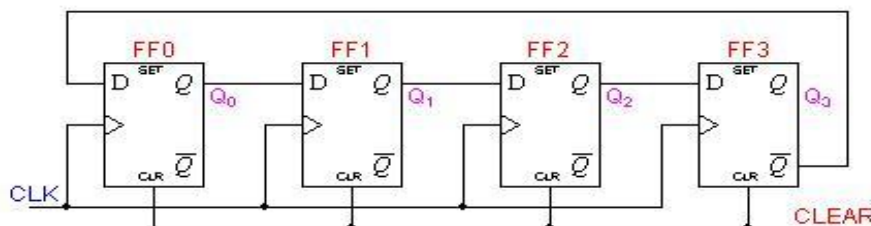
		-3.5476 1.2103 -0.1621]	-0.1219 0.0421 -0.0057]
6	[1 11 45 84 70 21 1]	A = 1e-5*[0.0344 0.2066 0.5166 0.1687 0.1265 0.0506 0.0084]	A = 1e-34*[0 0.0143 0.2614 0.4650 0.1273 0.0034]
		B = [1 -4.2977 7.5859 -7.0355 3.6147 -0.9753 0.1079]	B = 1e-28*[0.0426 -0.1846 0.3287 -0.3080 0.1601 -0.0438 0.0049]
7	[1 13 66 165 210 126 28 1]	A = 1e-6*[0.0285 0.1992 0.5975 0.9959 0.9959 0.5975 0.1992 0.0285]	A = 1e-40*[-0.0 0.0059 0.2365 0.8762 0.6085 0.0792 0.001]
		B = [1 -4.9925 10.5519 -12.2350 8.4031 -3.4179 0.7622 -0.0719]	B = 1e-33*[0.0542 -0.2730 0.5824 -0.6822 0.4738 -0.1951 0.0441 -0.0042]
8	[1 15 91 286 495 462 210 36 1]	A = 1e-6*[0.0024 0.0188 0.0658 0.1317 0.1646 0.1317 0.0658 0.0188 0.0024]	A = 1e-45*[0 0.0002 0.0176 0.1222 0.1721 0.0585 0.0040 0]
		B = [1 -5.6872 14.0006 -19.4814 16.7551 -9.1193 3.0670 -0.5827 0.0479]	B = 1e-37*[1 0.0069 -0.0396 0.0984 -0.1383 0.1203 -0.0663 0.0226 -0.0044 0.0004]

**Tab. E.2b:**

Digitale Koeffizienten für eine Abtastrate von 400KHz

## E.4 Johnson Counter

Eine andere Möglichkeit wäre es, den externen Takteingang am Bitgenerator mit einem zum Abtasttakt synchronen Clock vom C6713/AIC23 einzuspeisen. Dabei könnte ein Takteiler, wie z.B. der Johnson Counter, zur Erzeugung des passenden Taktes zum Einsatz kommen. Der Vorteil wäre ein Generator, dessen Ausgangsgeschwindigkeit durch einen einstellbaren Johnson Counter gesteuert werden könnte.



**Abb. E.13:**  
Johnson Counter

Clock Pulse	Q3	Q2	Q1	Q0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	1	1	1
4	1	1	1	1
5	1	1	1	0
6	1	1	0	0
7	1	0	0	0

CLEAR	FF0	FF1	FF2	FF3
0	0	0	0	0

**Tab. E.3:**  
Wahrheitstabelle zum  
Johnson Counter

## E.5 Angebote der Firma GBM

### Single Board Computer

gbm mbH • Lehmkuhlenweg 16 • 41065 Mönchengladbach  
 Hochschule für Angewandte Wissenschaften Hamburg  
 Labor für digitale Übertragungstechnik  
 Herrn Lang  
 Redderplatz 8 Haus 2

22337 Hamburg



Angebot	
Vorgangsnummer	
Belegnummer	2007-11075
Datum	
Kundennummer	8615
Bearbeiter	H.-D. Breitkopf
Bitte bei allen Rückfragen angeben!	

Versandart	UPS	Bezug	Unsere UStIDNr	DE 811 587 831
Lieferbedingung	Ab Werk	Ihr Zeichen	Unsere SteuerNr	12157200583
		Ihr Beleg	Ihre UStIDNr	

Sehr geehrter Herr Lang,

1	80055-4	SBC6711 DSP Single Board Computer - TMS320C6711 Floating Point DSP (250MHz) - 32 MByte SDRAM - 2 Steckplätze für OMNIBUS-Meßwert erfassungsmodule - 1 RS-232-Port, 1 USB-Interface - 32 digitale I/O	09.08.2007	1 Stk	1.345,00	1.345,00
<b>Alternativ:</b>						
2	80107-1	SBC6713e-I DSP Single Board Computer - TMS320C6713 Floating Point DSP (300MHz) - Spartan IIC 300K Gates FPGA für Meßdaten- vorverarbeitung und Online-Computing - 32 MByte SDRAM - 2 Steckplätze für OMNIBUS-Meßwert erfassungsmodule - FFDIP Rx- und Tx-Port - 10/100-BaseT Ethernet-Port - 32 digitale I/O	09.08.2007	1 Stk	1.550,00	1.550,00
3	80020-24	RF - Analog-I/O-Modul - 2 unabhängige A/D-Wandler (65 MHz, 12 Bit) - 2 unabhängige D/A-Wandler (65 MHz, 12 Bit)	09.08.2007	1 Stk	1.605,00	1.605,00

Pos.	Artikelnr.	Bezeichnung	Termin	Menge ME	Einzelpreis	Gesamtpreis
<b>Alternativ :</b>						
4	80020-12	A/D1 - Analog-I/O-Modul - 4 unabhängige 14 Bit A/D Wandler mit je 10 MHz Abtastrate - 1k FIFO-Speicher für jeden Kanal - 1 D/A-Wandler mit 7,5 MHz Updaterate	09.08.2007	1 Stk	1.965,00	1.965,00
5	80020-23D	Servo16 - Analog-I/O-Modul - 16 unabhängige A/D-Wandler (je 100 kHz Abtastrate, 16 Bit Auflösung) - 4-Pol Anti-Aliasing Filter bei 50 kHz - 16 unabhängige D/A-Wandler (je 100 kHz Abtastrate, 16 Bit Auflösung)	09.08.2007	1 Stk	2.010,00	2.010,00

Auch bei den Angeboten für PCI-Einsteckkarten wird die Preislage nicht freundlicher:

### PCI-Einsteckkarten

Pos.	Artikelnr.	Bezeichnung	Termin	Menge ME	Einzelpreis	Gesamtpreis
1	80098-1	Oruga-32DA DSP-Einsteckkarte - TMS320C6713 Floating Point DSP (225 MHz) - 32 Mbyte SDRAM - 32 Analogeingänge mit 16 Bit Auflösung (2 A/D-Wandler mit je 2,2 MHz Abtastrate und jeweils 16 über Multiplexer vorgeschalteten Eingängen) - 4 unabhängige D/A-Wandler mit 16 Bit Auflösung und 2 MHz Updaterate - 64 digitale I/O-Ports - 64/32 Bit PCI-Interface	26.07.2007	1 Stk	5.260,00	5.260,00

#### **Alternativ :**

2	80169-1	P25M DSP-Einsteckkarte - TMS320C6713 Floating Point DSP (300 MHz) - 64 MByte SDRAM - 4 synchrone Analogeingänge mit je 25 MHz und 16 Bit Auflösung - 4 synchrone Analogausgänge mit je 50 MHz und 16 Bit Auflösung - programmierbares FPGA Xilinx Spartan3 XC3S1000-4FG456C (1.000K Gates) für Meßdatenvorverarbeitung und Online-Computing - 64 digitale I/O-Ports - 64 Bit / 66 MHz PCI-Interface	26.07.2007	1 Stk	2.580,00	2.580,00
---	---------	--	------------	-------	----------	----------

Pos.	Artikelnr.	Bezeichnung	Termin	Menge ME	Einzelpreis	Gesamtpreis
------	------------	-------------	--------	----------	-------------	-------------

#### **Alternativ :**

3	80117-1	M6713 DSP-Einsteckkarte - TMS320C6713 Floating Point DSP (300 MHz) - 128 MByte SDRAM - 2 Steckplätze für OMNIBUS-Meßwerterfassungsmodule - programmierbares FPGA für Meßdatenvorverarbeitung und Online-Computing - 32 digitale I/O-Ports, FPDP-Port - 64/32 Bit PCI-Interface	26.07.2007	1 Stk	2.570,00	2.570,00
---	---------	--	------------	-------	----------	----------

#### **Alternativ :**

4	80070-0	Conejo DSP-Einsteckkarte - TMS320C6711 Floating Point DSP (150 MHz) mit 900 MFLOPS - 32 Mbyte SDRAM - 4 synchrone Analogeingänge mit je 10 MHz und 14 Bit Auflösung - 4 synchrone Analogausgänge mit je 15 MHz und 16 Bit Auflösung (alternativ 2x 30MHz ; 1x 50MHz) - 64 digitale I/O-Ports, FIFO-Port - 64/32 Bit PCI-Interface	26.07.2007	1 Stk	3.850,00	3.850,00
---	---------	---	------------	-------	----------	----------

## E.6 Verwendete Wandler ICs

Wird von GBM verwendet:

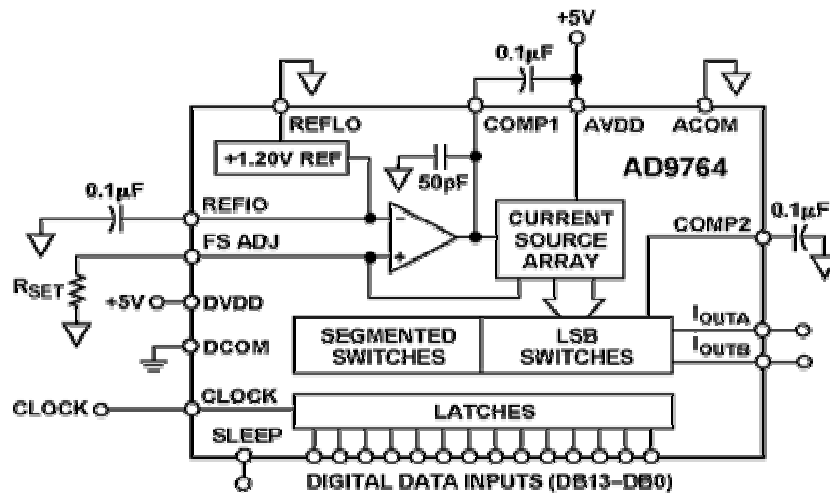


Abb. E.14:  
"AD9764"

Selbst herausgesucht:

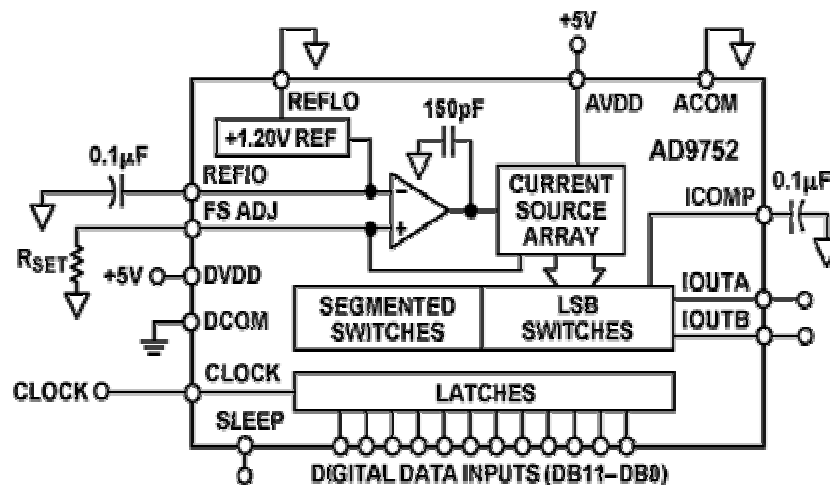


Abb. E.15:  
"AD9752"

Wird von GBM verwendet:

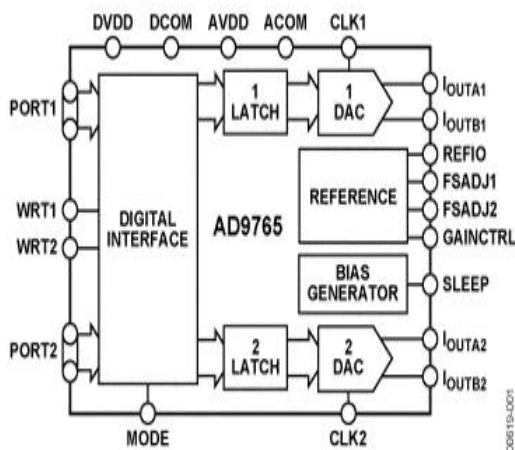


Abb. E.16:  
"AD9765"

Wird von GBM verwendet

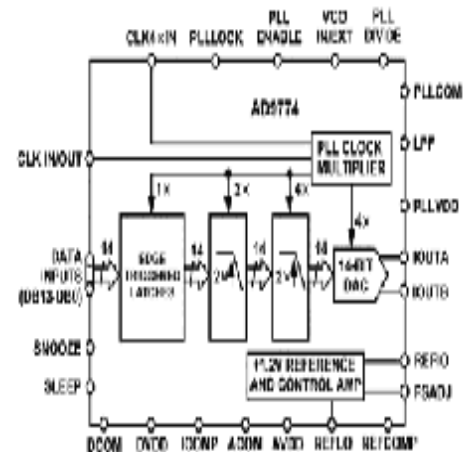


Abb. E.17:  
"AD9774"

## Anhang F Anhang zu Kapitel 6

### F.1 J3 Connector

#### 3.3.2 J3, Peripheral Expansion Connector

Table 3: J3, Peripheral Expansion Connector

Pin	Signal	I/O	Description	Pin	Signal	I/O	Description
1	12V	Vcc	12V voltage supply pin	2	-12V	Vcc	-12V voltage supply pin
3	GND	Vss	System ground	4	GND	Vss	System ground
5	5V	Vcc	5V voltage supply pin	6	5V	Vcc	5V voltage supply pin
7	GND	Vss	System ground	8	GND	Vss	System ground
9	5V	Vcc	5V voltage supply pin	10	5V	Vcc	5V voltage supply pin
11	N/C	-	No connect	12	N/C	-	No connect
13	N/C	-	No connect	14	N/C	-	No connect
15	N/C	-	No connect	16	N/C	-	No connect
17	N/C	-	No connect	18	N/C	-	No connect
19	3.3V	Vcc	3.3V voltage supply pin	20	3.3V	Vcc	3.3V voltage supply pin
21	CLKX0	I/O	McBSP0 transmit clock	22	CLKS0	I	McBSP0 clock source
23	FSX0	I/O	McBSP0 transmit frame sync	24	DX0	O	McBSP0 transmit data
25	GND	Vss	System ground	26	GND	Vss	System ground
27	CLKR0	I/O	McBSP0 receive clock	28	N/C	-	No connect
29	FSR0	I/O	McBSP0 receive frame sync	30	DR0	I	McBSP0 receive data
31	GND	Vss	System ground	32	GND	Vss	System ground
33	CLKX1	I/O	McBSP1 transmit clock	34	CLKS1	I	McBSP1 clock source
35	FSX1	I/O	McBSP1 transmit frame sync	36	DX1	O	McBSP1 transmit data
37	GND	Vss	System ground	38	GND	Vss	System ground
39	CLKR1	I/O	McBSP1 receive clock	40	N/C	-	No connect
41	FSR1	I/O	McBSP1 receive frame sync	42	DR1	I	McBSP1 receive data
43	GND	Vss	System ground	44	GND	Vss	System ground
45	TOUT0	O	Timer 0 output	46	TINP0	I	Timer 0 input
47	N/C	-	No connect	48	EXT_INT5	I	External interrupt 5
49	TOUT1	O	Timer 1 output	50	TINP1	I	Timer 1 input
51	GND	Vss	System ground	52	GND	Vss	System ground
53	EXT_INT4	I	External interrupt 4	54	N/C	-	No connect
55	N/C	-	No connect	56	N/C	-	No connect
57	N/C	-	No connect	58	N/C	-	No connect
59	RESET	O	System reset	60	N/C	-	No connect
61	GND	Vss	System ground	62	GND	Vss	System ground
63	CNTL1	O	Daughtercard control 1	64	CNTL0	O	Daughtercard control
65	STAT1	I	Daughtercard status 1	66	STAT0	I	Daughtercard status
67	EXT_INT6	I	External interrupt 6	68	EXT_INT7	I	External interrupt 7
69	ACE3#	O	Chip enable 3	70	N/C	-	No connect
71	N/C	-	No connect	72	N/C	-	No connect
73	N/C	-	No connect	74	N/C	-	No connect
75	DC_DET#	Vss	System ground	76	GND	Vss	System ground
77	GND	Vss	System ground	78	ECL KOUT	O	EMIF Clock
79	GND	Vss	System ground	80	GND	Vss	System ground

Tab. F.1:  
"J3" Connector



## F.2 Register zur Initialisierung der McBSP-Pins als GPIO

Diese beiden Register (*SPCR*, *PCR*) sind in Abb. 6.3 und Abb. 6.4 dargestellt.

(Auszug *spru190d.pdf* Seite 592+667)

Figure 12–2. Serial Port Control Register (*SPCR*)

31	26	25	24	23	22	21	20	19	18	17	16			
reserved		FREE <sup>‡</sup>	SOFT <sup>‡</sup>	$\overline{\text{FRST}}$	$\overline{\text{GRST}}$	XINTM	XSYNCERR <sup>†</sup>	$\overline{\text{XEMPTY}}$	XRDY	$\overline{\text{XRST}}$				
R, +0		RW, +0	RW, +0	RW, +0	RW, +0	RW, +0	RW, +0	R, +0	R, +0	RW, +0				
15	14	13	12	11	10	8	7	6	5	4	3	2	1	0
DLB	RJUST	CLKSTP	Rsvd	DXENA <sup>‡</sup>	Rsvd	RINTM	RSYNCERR <sup>†</sup>	RFULL	RRDY	$\overline{\text{RRST}}$				
RW,+0	RW, +0	RW,+0	R, +0	RW, +0	R, +0	RW, +0	RW, +0	R, +0	R, +0	RW, +0				

<sup>†</sup> Writing a 1 to XSYNCERR or RSYNCERR will set the error condition when the transmitter or receiver ( $\overline{\text{XRST}}=1$  or  $\overline{\text{RRST}}=1$ ), respectively, are enabled. Thus, it is used mainly for testing purposes or if this operation is desired.

<sup>‡</sup> Available in the C621x/C671x/C64x only.

**Abb. F.1:**

Serial Port Control Register (*SPCR*)

Erklärung zu den benötigten Bits im SPCR:

0       $\overline{\text{RRST}}$       Receiver reset. This resets or enables receiver.  
 $\overline{\text{RRST}} = 0$ : Serial port receiver is disabled and is in reset state.  
 $\overline{\text{RRST}} = 1$ : Serial port receiver is enabled.

16       $\overline{\text{XRST}}$       Transmitter reset. This resets or enables transmitter.  
 $\overline{\text{XRST}} = 0$ : Serial port transmitter is disabled and is in reset state.  
 $\overline{\text{XRST}} = 1$ : Serial port transmitter is enabled.

31	reserved														16
R, +0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsvd	XIOEN	RIOEN	FSXM	FSRM	CLKXM	CLKRM	Rsvd	CLKS_STAT	DX_STAT	DR_STAT	FSXP	FSRP	CLKXP	CLKRP	
R,+0	RW,+0	RW,+0	RW,+0	RW,+0	RW,+0	RW,+0	R,+0	RW,+0	RW,+0	R,+0	RW,+0	RW,+0	RW,+0	RW,+0	

**Abb. F.2:**

Pin Control Register (*PCR*)

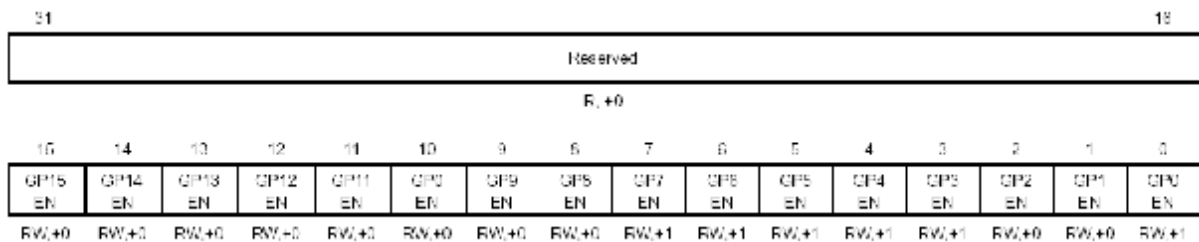
Erklärung zu den benötigten Bits im PCR:

13	XIOEN	Transmitter in general-purpose I/O mode <i>only</i> when $\overline{XRST} = 0$ in SPCR XIOEN = 1: CLKX pin is a general-purpose input. DX pin is a general-purpose output. FSX and CLKX are general-purpose I/Os. These serial port pins do not perform serial port operation.
12	RIOEN	Receiver in general-purpose I/O mode <i>only</i> when $\overline{RRST} = 0$ in SPCR RIOEN = 1: DR and CLKX pins are general-purpose inputs. FSR and CLKR are general-purpose I/Os. These serial port pins do not perform serial port operation.

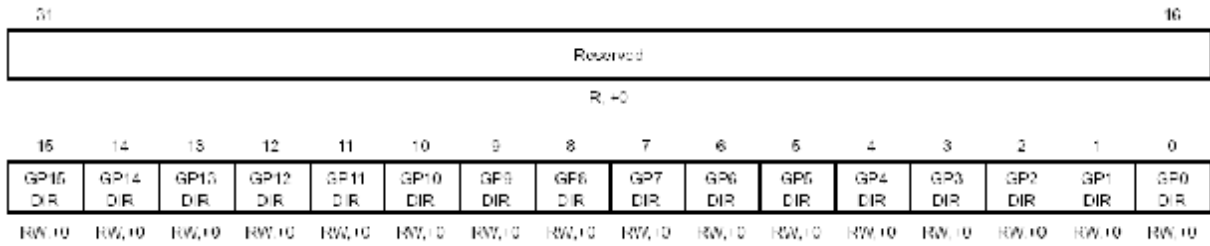
Pin	General-Purpose I/O Enabled When...	Selected as Output When...	Output Value Driven From	Selected as Input When ...	Input Value Readable on
CLKX	$\overline{XRST} = 0$ XIOEN = 1	CLKXM = 1	CLKXP	CLKXM = 0	CLKXP
FSX	$\overline{XRST} = 0$ XIOEN = 1	FSXM = 1	FSXP	FSXM = 0	FSXP
DX	$\overline{XRST} = 0$ XIOEN = 1	Always	DX STAT	Never	N/A
CLKR	$\overline{RRST} = 0$ RIOEN = 1	CLKRM = 1	CLKRP	CLKRM = 0	CLKRP
FSR	$\overline{RRST} = 0$ RIOEN = 1	FSRM = 1	FSRP	FSRM = 0	FSRP
DR	$\overline{RRST} = 0$ RIOEN = 1	Never	N/A	Always	DR STAT
CLKS	$\overline{RRST} = \overline{XRST} = 0$ RIOEN = XIOEN = 1	Never	N/A	Always	CLKS STAT

**Abb. F.3:**  
Configuration of Pins as General Purpose I/O

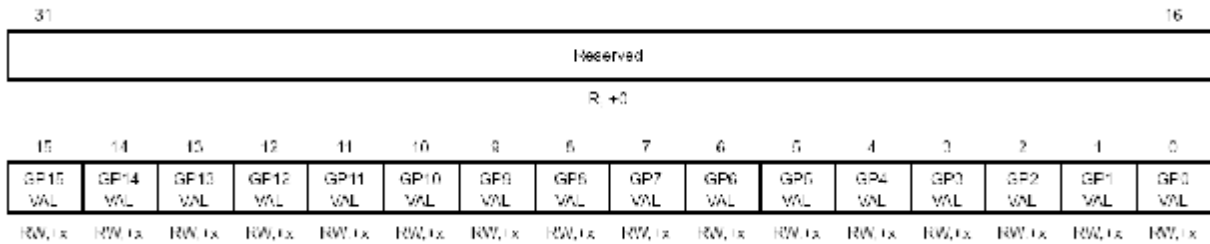
### F.3 Register zur Initialisierung der externen Interrupt-Pins als GPIO



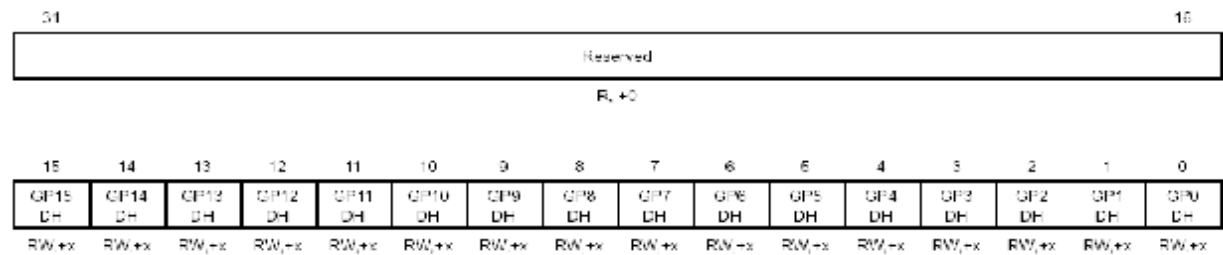
**Abb. F.4:**  
GPIO Enable Register (GPEN)



**Abb. F.5:**  
GPIO Direction Register (*GPDIR*)



**Abb. F.6:**  
GPIO Value Register (*GPVAL*)



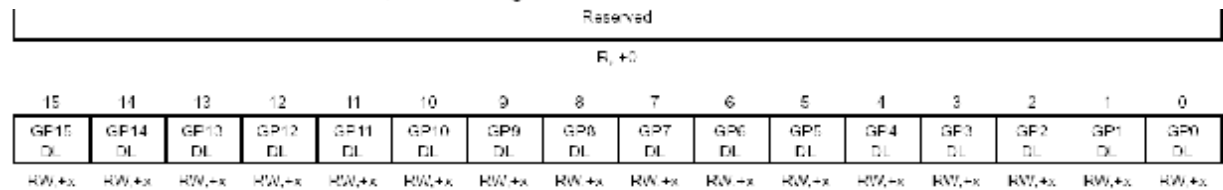
**Abb. F.7:**  
GPIO Delta High Register (*GPDH*)

**Erklärung zum GPDH:**

15:0 GPxDH GPx Delta High. A low-to-high transition is detected on the GPx input. Applies when the corresponding GPx pin is enabled as an input (GPxEN = 1, GPxDIR = 0)

GPxDH = 0; a low-to-high transition is not detected on GPx

GPxDH = 1; a low-to-high transition is detected on GPx



**Abb. F.8:**  
GPIO Delta Low Register (*GDDL*)

15:0 GPxDL GPx Delta Low. A high-to-low transition is detected on the GPx input. Applies when the corresponding GPx pin is enabled as an input (GPxEN = 1, GPxDIR = 0).

GPxDL = 0; a high-to-low transition is not detected on GPx.

GPxDL = 1; a high-to-low transition is detected on GPx.

## F.4 Programm mit Interrupt und Ausgabe über GPIO\_Pins an "J3"

```
//IO_Ausgabe_2_2_12Bit_test
//loop_poll.c Loop program using polling. Output=delayed input

#include "DSK6713_AIC23.h" //codec-DSK file support
#include <dsk6713.h>
#include <stdio.h> // only required for puts
#include <csl.h>
#include <csl_irq.h>
#include <stdlib.h> // only required for puts
#include <csl_mcbasp.h>
#include <math.h>
#include <csl_gpio.h>
#define WRITE_GPVAL(val) (*(volatile short*)(0x01B00008) = val)
//#include <delay.h>
Uint32 fs=DSK6713_AIC23_FREQ_96KHZ; //set sampling rate

// Global variable
int i=0 ,daten_vorhanden=0,io_daten=0;
unsigned short u=0;
MCBSP_Handle hMcbasp0;
MCBSP_Handle hMcbasp1;
GPIO_Handle hGpio0;

MCBSP_Config My_MCBSP_Config = {
    0x02000000, /* Serial Port Control Reg. (SPCR) 0x02000000 */
    0x00000000, /* Receiver Control Reg. (RCR) 0x00000000*/
    0x00000000, /* Transmitter Control Reg. (XCR) 0x00000000 */
    0x20000000, /* Sample-Rate Generator Reg. (SRGR) 0x20000000*/
    0x00000000, /* Multichannel Control Reg. (MCR) 0x00000000 */
    0x00000000, /* Receiver Channel Enable(RCER) 0x00000000 */
    0x00000000, /* Transmitter Channel Enable(XCER) 0x00000000 */
    0x00003F00 /* Pin Control Reg. (PCR) 0x00003F00*/
};

GPIO_Config My_GPIO_Config = {
    0x00000000, // Uint32 gpgc;
    0x000000F0, // Uint32 gpen;
    0x000000E0, // Uint32 gpdir;
    0x00000000, // Uint32 gpval;
    0x00000010, // Uint32 gphm;
    0x00000000, // Uint32 gplm;
    0x00000010 // Uint32 gppol;
};

// Interrupt Service-Routine
interrupt void intser(void)
{
    io_daten=MCBSP_FGET(PCR1,DRSTAT); //INPUT: DR0 & DR1 (pin 30 & 42)
    daten_vorhanden=1;
}

void main ()
{
    short sample_data;

    double a0,a1,a2,b1,b2,u0=0,u1=0,u2=0,u3=0; //Declare local variables
    int j=12,k=0,tmp=0,g,h,i=0,u_par=0;
    long ux,CLKXP0 ;
```

```

int gpint_pol,bbla=0;
char buffer[20];

in Vectors_poll.asm Int4_V mit int5_V tauschen
    CSL_init(); // Init CSL
    IRQ_map(IRQ_EVT_EXTINT4, 5); // Map external INT4 to Interrupt 5
    IRQ_reset(IRQ_EVT_EXTINT4); // Disable and Clear old Requests
    IRQ_nmiEnable(); // Required for interrupt system
    IRQ_enable(IRQ_EVT_EXTINT4); // Enable external INT4
    IRQ_globalEnable(); // Set GIE-Flag

/*
Aus ADDA16 Höfs
    IRQ_EXTPOL = 0x00000008u; // interrupt 7 active on falling edge
    IRQ_map(IRQ_EVT_EXTINT7, 7); // map EXTINT7 to CPU interrupt 7
    IRQ_reset(IRQ_EVT_EXTINT7); // disable and clear old interrupts
    IRQ_enable(IRQ_EVT_EXTINT7); // enable EXTINT7 interrupt
    IRQ_nmiEnable(); // required for interrupt sys.
    IRQ_globalEnable(); // set GIE-Flag
*/

// gpint_pol=lesen(0x01b00024); //lese GPPOL register
// gpint_pol=gpint_pol|0x10;
// schreiben(0x01b00024,gpint_pol); //setze EXT_INT4 auf triggern
//bei aufsteigender Flanke

//T=12.7324µs bilinear
//impulsinvariant
//T=1.59155µs bilinear
a0 = 0.0699; // Zähler-
a1 = 0.1398; // koeffizienten
a2 = 0.0699; //
b1 = -0.6955; // Nenner-
b2 = -0.0250; // koeffizienten

```

	T=12.7324µs impulsinvariant	T=1.59155µs bilinear
a0 = 0.0699;	0	0.4975
a1 = 0.1398;	0.3643e-10	0.9950
a2 = 0.0699;	0.1621e-9	0.4975
b1 = -0.6955;	-0.1376e-9	0.9021
b2 = -0.0250;	0.0139e-9	0.0879

```

DSK6713_rset(DSK6713_MISC,0x03); // sets MCBSP0 and MCBSP1 in off board mode
MCBSP_resetAll();
hMcbSP0 = MCBSP_open(MCBSP_DEV0, MCBSP_OPEN_RESET);
hMcbSP1 = MCBSP_open(MCBSP_DEV1, MCBSP_OPEN_RESET);
MCBSP_config(hMcbSP0, &My_MCBSP_Config);
MCBSP_config(hMcbSP1, &My_MCBSP_Config);

hGpio0 = GPIO_open(GPIO_DEV0, GPIO_OPEN_RESET);
GPIO_config(hGpio0, &My_GPIO_Config);

while(1) //infinite loop
{
    if (daten_vorhanden==1)
    {
        u0 = (double)io_daten;
        daten_vorhanden=0;
    };
    k=0;

//IIR 2.Ordnung mit 100fach Oversampling
while (k<100)
{
    GPIO_pinWrite(hGpio0,GPIO_PIN5,0);

    u3 = a0*u0+u2;
    u2 = a1*u0-b1*u3+u1;
    u1 = a2*u0-b2*u3;
}
}

```

```

    u = (unsigned short)u3; //double to short/integer conversion
    u = (u>>4);           //funktioniert, aber noch zu langsam
                           //(z.B. 3062,4 wird dann 191)

    par_out(u);

    k=k+1;
  };
  k=0;
}

int par_out()
{
    MCBSP_FSET(PCR0,CLKXP,(u & 0x01));
    MCBSP_FSET(PCR0,FSXP,((u & 0x02)>>1));
    MCBSP_FSET(PCR0,DXSTAT,((u & 0x04)>>2));
    MCBSP_FSET(PCR0,CLKRP,((u & 0x08)>>3));
    MCBSP_FSET(PCR0,FSRP,((u & 0x010)>>4));

    MCBSP_FSET(PCR1,CLKXP,((u & 0x020)>>5));
    MCBSP_FSET(PCR1,FSXP,((u & 0x040)>>6));
    MCBSP_FSET(PCR1,DXSTAT,((u & 0x080)>>7));
    MCBSP_FSET(PCR1,CLKRP,((u & 0x0100)>>8));
    MCBSP_FSET(PCR1,FSRP,((u & 0x0200)>>9));
    GPIO_pinWrite(hGpio0,GPIO_PIN6,((u & 0x0400)>>10)); //Ext_Int 6 Pin 67 1
    GPIO_pinWrite(hGpio0,GPIO_PIN7,((u & 0x0800)>>11)); //Ext_Int 7 Pin 68 0
    GPIO_pinWrite(hGpio0,GPIO_PIN5,1); //Takt setzen evtl, kurz warten
    return 0;
}
//Ausgabe ändern!!!!!!-- sonst nur 330KHz möglich

```

### Optimierung 1:

```

int par_out()
{
    u= (u>>4); //funktioniert, aber zu langsam (3062,4 wird short, dann 191!!

/*
    MCBSP_RSET(PCR0,((u & 0x2F)+0x3F00));
    GPIO_pinWrite(hGpio0,GPIO_PIN6,((u & 0x10)>>4));
    u= (u>>6);
    MCBSP_RSET(PCR1,((u & 0x2F)+0x3F00));
    GPIO_pinWrite(hGpio0,GPIO_PIN7,((u & 0x10)>>4));
*/
//Nun schon 666Khz möglich. Mit -o3 sind es 870Khz(830) -> (mit ISR)

```

### Optimierung 2:

```

    MCBSP_RSET(PCR0,((u & 0x2F)+0x3F00));
    GPIO_RSET(GPVAL,((u & 0x10)<<2) +((u & 0x400)>>3)+0xE10));
    u= (u>>6);
    MCBSP_RSET(PCR1,((u & 0x2F)+0x3F00));

    GPIO_pinWrite(hGpio0,GPIO_PIN5,1);
    oder WRITE_GPVAL(0xE30); und WRITE_GPVAL(0xE10); //als Takt

//Nun 830KHz ;Mit -o3 und -o2 läuft das Programm nicht mehr mit ISR; aber ohne ISR
mit 1,2MHz
//Mit -o1 Optimierung ca.1MHz

```

Es folgt die Pinbelegung des EMIF zum Anschluss an den DAC:

```
/*
CLKX0          Pin 21  0 bit1
FSX0          Pin 23  0 bit3
DX0           Pin 24  0 bit5
CLKR0         Pin 27  1 bit0
FSR0          Pin 29  0 bit2
CLKX1         Pin 33  1 bit7
FSX1          Pin 35  1 bit9
DX1           Pin 36  1 bit11 beachte DA-Wandler positiv und
                    negativ? nur 1 wenn u0=65535 also unsigned !?!
CLKR1         Pin 39  1 bit6
FSR1          Pin 41  1 bit8
Ext_Int 6     Pin 67  1 bit4
Ext_Int 7     Pin 68  0 bit10
*/

return 0;
}
```

**Tab. F.2:**  
"J3" Connector

## F.5 "J4" EMIF Connector

Spectrum Digital, Inc

### 3.3.1 J4, Memory Expansion Connector

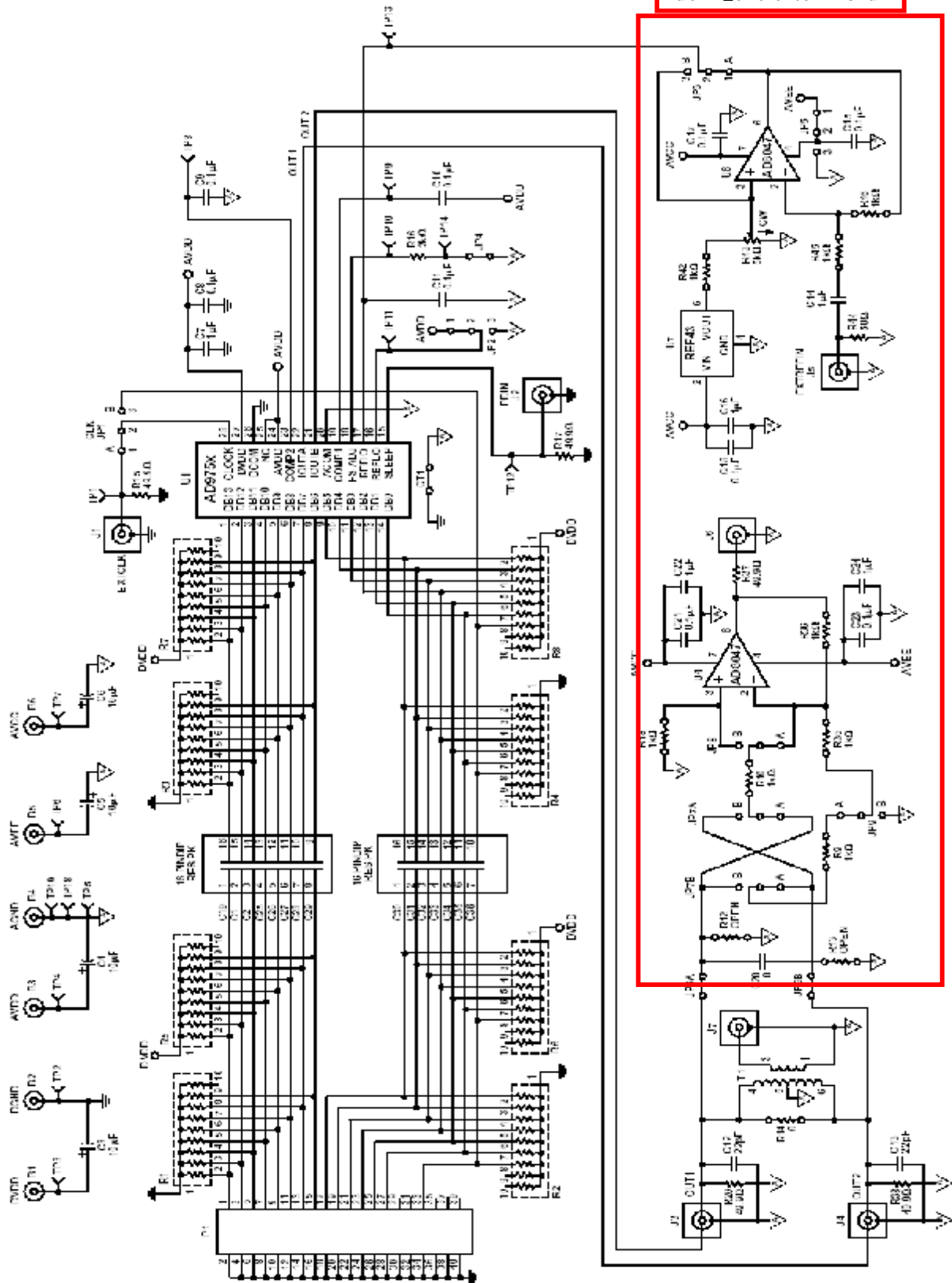
Table 2: J4, Memory Expansion Connector

Pin	Signal	I/O	Description	Pin	Signal	I/O	Description
1	5V	Vcc	5V voltage supply pin	2	5V	Vcc	5V voltage supply pin
3	AEA21	O	EMIF address pin 21	4	AEA20	O	EMIF address pin 20
5	AEA10	O	EMIF address pin 10	6	AEA18	O	EMIF address pin 18
7	AEA17	O	EMIF address pin 17	8	AEA16	O	EMIF address pin 16
9	AEA15	O	EMIF address pin 15	10	AEA14	O	EMIF address pin 14
11	GND	Vss	System ground	12	GND	Vss	System ground
13	AEA13	O	EMIF address pin 13	14	AEA12	O	EMIF address pin 12
15	AEA11	O	EMIF address pin 11	16	AEA10	O	EMIF address pin 10
17	AEA9	O	EMIF address pin 9	18	AEA8	O	EMIF address pin 8
19	AEA7	O	EMIF address pin 7	20	AEA6	O	EMIF address pin 6
21	5V	Vcc	5V voltage supply pin	22	5V	Vcc	5V voltage supply pin
23	AEA5	O	EMIF address pin 5	24	AEA4	O	EMIF address pin 4
25	AEA3	O	EMIF address pin 3	26	AEA2	O	EMIF address pin 2
27	ABE3#	O	EMIF byte enable 3	28	ABE2#	O	EMIF byte enable 2
29	ABE1#	O	EMIF byte enable 1	30	ABE0#	O	EMIF byte enable 0
31	GND	Vss	System ground	32	GND	Vss	System ground
33	AED31	I/O	EMIF data pin 31	34	AED30	I/O	EMIF data pin 30
35	AED29	I/O	EMIF data pin 29	36	AED28	I/O	EMIF data pin 28
37	AED27	I/O	EMIF data pin 27	38	AED26	I/O	EMIF data pin 26
39	AED25	I/O	EMIF data pin 25	40	AED24	I/O	EMIF data pin 24
41	3.3V	Vcc	3.3V voltage supply pin	42	3.3V	Vcc	3.3V voltage supply pin
43	AED23	I/O	EMIF data pin 23	44	AED22	I/O	EMIF data pin 22
45	AED21	I/O	EMIF data pin 21	46	AED20	I/O	EMIF data pin 20
47	AED19	I/O	EMIF data pin 19	48	AED18	I/O	EMIF data pin 18
49	AED17	I/O	EMIF data pin 17	50	AED16	I/O	EMIF data pin 16
51	GND	Vss	System ground	52	GND	Vss	System ground
53	AED15	I/O	EMIF data pin 15	54	AED14	I/O	EMIF data pin 14
55	AED13	I/O	EMIF data pin 13	56	AED12	I/O	EMIF data pin 12
57	AED11	I/O	EMIF data pin 11	58	AED10	I/O	EMIF data pin 10
59	AED9	I/O	EMIF data pin 9	60	AED8	I/O	EMIF data pin 8
61	GND	Vss	System ground	62	GND	Vss	System ground
63	AED7	I/O	EMIF data pin 7	64	AED6	I/O	EMIF data pin 6
65	AED5	I/O	EMIF data pin 5	66	AED4	I/O	EMIF data pin 4
67	AED3	I/O	EMIF data pin 3	68	AED2	I/O	EMIF data pin 2
69	AED1	I/O	EMIF data pin 1	70	AED0	I/O	EMIF data pin 0
71	GND	Vss	System ground	72	GND	Vss	System ground
73	AARE#	O	EMIF async read enable	74	AAWE#	O	EMIF async write enable
75	AAOE#	O	EMIF async output enable	76	AARDY	I	EMIF asynchronous ready
77	ACE3#	O	Chip enable 3	78	ACE2#	O	Chip enable 2
79	GND	Vss	System ground	80	GND	Vss	System ground

Tab. F.3:  
"J4" Connector



## F.6 Schaltplan zum EB-9752



Dieser Teil fehlt auf dem gelieferten Bord

Abb. F.9:  
EB-9752

## F.7 Active Filter Design Tool

### Interactive Design Tools: OpAmps : Active Filter Synthesis

A simple tool for designing active filters using voltage-feedback opamps.

[Instructions](#) | [Troubleshooting](#) | [Related Information](#) | [Send this Link to a Colleague](#)

The screenshot shows the Analog Devices Active Filter Tool interface. At the top, the filter is configured as a Lowpass Butterworth filter of order 2 with a cutoff frequency  $f_c = 2 \times 10^6$  Hz. A highlighted 'Stage 1' section shows a Sallen-Key LP configuration with a corner frequency  $F_0 = 2.000 \times 10^6$  Hz and a quality factor  $Q = 0.7071$ . The circuit diagram shows an op-amp based Sallen-Key LP filter with components R1, R2, R3, R4, C1, and C2. The component values are listed as: Gain = 1.0, C1 = 0.010 nF, R3 = 22.0 K Ohms. Below the diagram, a table provides the calculated component values: R1 = 11.3 K Ohms, R2 = 11.3 K Ohms, C1 = 0.01000 nF, C2 = 0.005100 nF, R3 = 22.1 K Ohms, and R4 = Infinity Ohms. The tolerance is set to 1.0% for resistors and 5.0% for capacitors. At the bottom, two plots are shown: the System Magnitude Response (green line) and the System Phase Response (red line), both plotted on a log-log scale from 1 Hz to 100e6 Hz.

**Bauteile für den ITP (Sallen-Key, LP,  $f_c = 2 \times 10^6$  Hz, ButterLP = OZ2,  $Q = 0,7071$ )**

R1=11,25K, R2=11,25K, C1=10pF, C2=5pF,

R3=22K, R4=unendlich

**Abb. F.10:**

Filter Tool von "Analog Devices" mit Bestimmungsalgorithmus nach Sallen-Key

## F.8 EMIF-Register

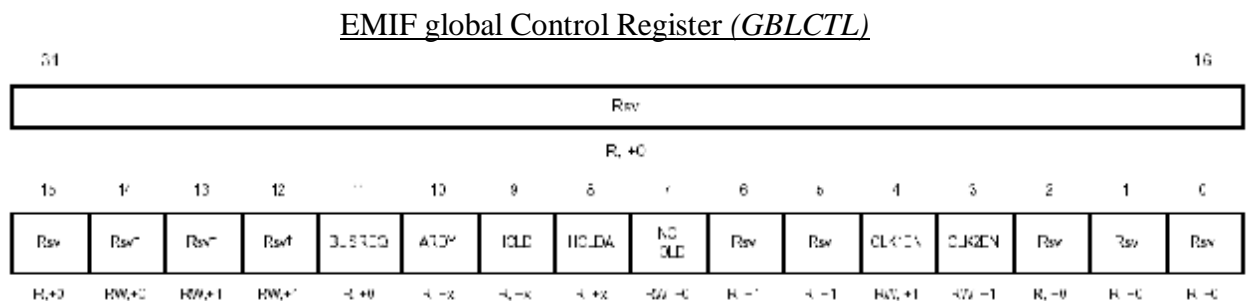
### EMIF-Register

Die Steuerung des EMIF wird von Speicheradressen gebundenen Registern übernommen. Ein Zugriff auf diese Register macht einen EMIF-clock erforderlich.

Acronym	Register Name	Section
GBLCTL	EMIF global control register	<a href="#">Section 3.7.1</a>
CECTL0-3	EMIF CE space control registers	<a href="#">Section 3.7.2</a>
SDCTL	EMIF SDRAM control register	<a href="#">Section 3.7.3</a>
SDTIM	EMIF SDRAM refresh control register	<a href="#">Section 3.7.4</a>
SDEXT	EMIF SDRAM extension register	<a href="#">Section 3.7.5</a>

### Felderbeschreibungen für die entsprechenden Register:

Das "EMIF global Control Register" (*GBLCTL*) konfiguriert Parameter für alle "Commande enable" (*CE*) Bereiche.



† Die reservierten Bitfelder sollten immer mit den entsprechenden "Standardwerten" beschrieben werden, wenn das GBLCTL verändert wird.

**Abb. F.11:**  
GBLCTL-Register

## Feldbeschreibung für das GBLCTL-Register:

Bit	Feld	symval	Wert	Beschreibung
31 - 12	Reserviert	-	0	Die reservierten Bits werden immer als 0 gelesen. Ein Wert, in diesen Bereich geschrieben, hat keine Wirkung.
11	BUSREQ			Das Bus-Anfrage (BUSREQ) Ausgangs Bit zeigt an, ob der EMIF einen Zugriff oder "refresh" zulässt.
		LOW	0	Kein Zugang/"refresh" möglich
		HIGH	1	Access / refresh anhängig ( <i>pending</i> ) oder in progress. ????????
10	ARDY			ARDY Eingangsbit
		LOW	0	Externes Gerät ist nicht bereit
		HIGH	1	Externes Gerät ist bereit
9	HOLD			HOLD Eingangsbit
		LOW	0	Externes Gerät stellt Anfrage über EMIF ( <i>Ext. device requesting EMIF???</i> )
		HIGH	1	No external request pending ????
8	HOLDA			HOLDA Ausgangsbit
		LOW	0	Externes Gerät verwendet EMIF
		HIGH	1	Kein externes Gerät verwendet das EMIF
7	NOHOLD			externes NOHOLD Freigabebit
		DISABLE	0	"No hold" ist deaktiviert. Halteanfragen über den HOLD Eingang werden über den HOLDA Ausgang zum frühest, möglichen Zeitpunkt erkannt.
		ENABLE	1	"No hold" ist aktiviert. Halteanfragen über den HOLD Eingang werden ignoriert.
6	Reserviert	-	1	Die reservierten Bits werden immer als 1 gelesen. Ein Wert, in diesen Bereich geschrieben, hat keine Wirkung.
5	EKEN <i>ECLKOUT does not turn off/on glitch free via EKEN</i>			ECLKOUT Aktivierungsbit
			0	ECLKOUT ist "low"
			1	ECLKOUT ist aktiviert um extern getaktet werden zu können ( <i>standard</i> )
4	CLK1EN			CLKOUT1 Freigabebit. Beim C6713 muss diese Position mit 0 beschrieben, um einen ordnungsgemäßen Betrieb zu gewährleisten.
		DISABLE	0	CLKOUT1 ist "high"
		ENABLE	1	CLKOUT1 ist aktiviert um Takt zu erzeugen
3	CLK2EN			CLKOUT2 ist aktiviert/deaktiviert ( <i>Benutzung des SSCEN/SDCEN Bits erf.</i> ).
		DISABLE	0	CLKOUT2 ist "high"
		ENABLE	1	CLKOUT2 ist aktiviert um Takt zu erzeugen
2-0	Reserviert	-	0	Die reservierten Bits werden immer als 0 gelesen. Ein Wert, in diesen Bereich geschrieben, hat keine Wirkung.

**Tab. F.4:**  
GBLCTL-Register Bitbeschreibung

Die "EMIF CE Space Control Control Register" (*CECTL0-3*) entsprechen den CE Speicherbereichen "supported by the EMIF".

Es gibt vier dieser Register, die zu den 4 externen CE Signalen korrespondieren. Das MTYPE Feld identifiziert den Speichertyp für den entsprechenden CE Bereich. Wird ein synchroner Speicher erkannt, haben die anderen Bits keinen Effekt. Da für die geforderte Anwendung der asynchrone Modus erkannt wird, spezifizieren die restlichen Registerfelder die Gestaltung der Adresse und Steuersignale für den Zugriff auf diesen Bereich. Das MTYPE Feld sollte nur einmal bei der Initialisierung gesetzt werden.

### EMIF CE Space Control Register (CECTL)

31	28	27	22	21	20	19	18	
WRSETUP		WRSTRB			WRHLD		RDSETUP	
R/W-1111		R/W-11 1111			R/W-11		R/W-1111	
15	14	13	8	7	4	3	2	0
IA		RDSTRB			MTYPE		Reserved	RDHLD
R/W-11		R/W-11 1111			R/W-0010		R-0	R/W-011

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Abb. F.12:**  
CECTL-Register

## Feldbeschreibung für das CECTL-Register:

Bit	Feld	symval	Wert	Beschreibung
31 - 28	WRSETUP	OF (value)	hex 0-F	"Write Setup width". Anzahl der Taktzyklen für die Zeit bis zur Inbetriebnahme der Adresse (EA), Chip enable (CE) und Byte enable (BE), bevor "write strobe" ( <i>schreiben Stroboskop oder Blitz ???</i> ) fällt. Für asynchrone Lesezugriffe ist dies auch die Zeit, bis zur Inbetriebnahme des "AOE" bevor "ARE" fällt.
27 - 22	WRSTRB	OF (value)	hex 0-3F	"Write Strobe width". Die Länge des "write strobe" (AWE) in Taktzyklen.
21 - 20	WRHLD	OF (value)	hex 0-3	"Write Hold width". Hier werden die Anzahl der Taktzyklen eingetragen, je nachdem wie lange die Werte der Adresse (EA) und des "byte strobes" (BE) gehalten werden sollen, nachdem "write strobes" angestiegen ist. Für asynchronen Lesezugriff ist dies auch die Zeit, bis zur Inbetriebnahme des "AOE" nachdem "ARE" steigt.
19 - 16	RDSETUP	OF (value)	hex 0-F	"Read Setup width". Anzahl der Taktzyklen für die Zeit bis zur Inbetriebnahme der Adresse (EA), Chip enable (CE) und Byte enable (BE), bevor "read strobe" ( <i>lesen Stroboskop oder Blitz ???</i> ) fällt. Für asynchrone Lesezugriffe ist dies auch die Zeit, bis zur Inbetriebnahme des "AOE" bevor "ARE" fällt.
15 - 14	TA	OF (value)	hex 0-3	"Minimum Turn-Around time". TA steuert die minimale Anzahl von ECLKOUT Zyklen zwischen einem Lesen, gefolgt von einem Schreiben (gleiche oder verschiedene CE-Bereiche), oder zwischen zwei Lesevorgängen aus verschiedenen CE-Bereichen. Gilt nur für asynchrone Speicher-Typen.
13 - 8	RDSTRB	OF (value)	hex 0-3F	"Read Strobe width". Die Länge des "read strobe" (ARE) in Taktzyklen.
7 - 4	MYTYPE		hex 0-F	Speichertyp für den entsprechenden CE Bereich
		ASYNC8	hex 0	8-bit-wide asynchronous interface
		ASYNC16	hex 1	16-bit-wide asynchronous interface
		ASYNC32	hex 2	32-bit-wide asynchronous interface
		SDRAM32	hex 3	32-bit-wide SDRAM
		SBSRAM32	hex 4	32-bit-wide SBSRAM
		-	hex 5-7	Reserviert
		SDRAM8	hex 8	8-bit-wide SDRAM
		SDRAM16	hex 9	16-bit-wide SDRAM
		SBSRAM8	hex A	8-bit-wide SBSRAM
		SBSRAM16	hex B	16-bit-wide SBSRAM
		-	hex C-F	Reserviert
3	Reserviert	-	0	Die reservierten Bits werden immer als 0 gelesen. Ein Wert, in diesen Bereich geschrieben, hat keine Wirkung.
2-0	RDHLD	OF (value)	hex 0-7	"Read Hold width". Hier werden die Anzahl der Taktzyklen eingetragen, je nachdem wie lange die Werte der Adresse (EA) und des "byte strobes" (BE) gehalten werden sollen, nachdem "read strobes" angestiegen ist. Für asynchronen Lesezugriff ist dies auch die Zeit, bis zur Inbetriebnahme des "AOE" nachdem "ARE" steigt.

**Tab. F.5:**  
CECTL-Register Bitbeschreibung

Die Kalkulation der CECTL-Register kann auch von einem automatischen "EMIF Calculator" übernommen werden, dessen Nutzung z.B. von der Firma "dsignT" [14] angeboten wird.

**Device**  
C621X, C671X

**Memory Type**  
asynchronous, 16-bit  
64-bit wide memory is only supported by C64XX and DM64X devices,  
C621X, C671X: synchronous memory = SBSRAM  
C620X, C670X: synchronous memory (SBSRAM) and SDRAM: 32-bit only

**Asynchronous Timing**  
Specify setup, strobe, and hold as number of EMIF clock cycles. Turn Around Time is the minimum number of clock cycles between a read followed by a write or between reads from different CE spaces.

Write Setup	<input type="text" value="1"/>	0..15	Read Setup	<input type="text" value="1"/>	0..15
Write Strobe	<input type="text" value="2"/>	0..63	Read Strobe	<input type="text" value="2"/>	0..63
Write Hold	<input type="text" value="1"/>	0..3 C64XX,DM642: 0..7	Read Hold	<input type="text" value="1"/>	C620X, C670X: 0..3 others: 0..7
			Turn Around Time	<input type="text" value="1"/>	C6X0X:0, others:0..3

**Synchronous Timing (C64XX, DM64X only)**

Clock Source	<input type="text" value="ECLKOUT1"/>	
SADS/SRE Function	<input type="text" value="ADS"/>	ADS mode is used for SBSRAM and ZBT SRAM Read Enable Mode is used for Fifos and generic synchronous interfaces
CE Extension	<input type="text" value="inactive"/>	inactive: CE goes inactive after final command active: CE active while SOE active (determined by read latency)
Write Latency	<input type="text" value="0"/>	0..3 clock cycles from write command to data valid
Read Latency	<input type="text" value="0"/>	0..3 clock cycles from read command to data read

**Result**

Clear      Calculate

CECTLx Register:

CESECx Register:  (C64XX, DM64X only)

**Abb. F.13:**  
EMIF CECTL Calculator

Das "EMIF SDRAM Control Register" (*SDCTL*) steuert SDRAM Parameter für alle CE Bereiche, in denen ein SDRAM-Speicher-typ im MTYPE Feld für das gewählte CE-Bereichs Kontrollregister (CECTL) definiert ist.

Da das SDCTL alle SDRAM bereiche steuert, muss jeder Bereich SDRAM mit den gleichen Erneuerungs-, Zeit- und Seiteneigenschaften enthalten. Das SDCTL sollte während eines SDRAM-Zugriffs nicht verändert werden

### EMIF SDRAM Control Register (*SDCTL*)

31	30	29	28	27	26	25	24	23	20	19	16
Reserved	SDBSZ	SDRSZ	SDCSZ	RFEN	INIT	TRCD			TRP		
R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0100			R/W-1000		
15	12	11									0
TRC				Reserved							
R/W-1111				R/W-0							

LEGEND: R/W = Read/Write, R = Read only; -n = value after reset

**Abb. F.14:**  
SDCTL-Register

Feldbeschreibung für das SDCTL-Register:

Bit	Feld	symval	Wert	Beschreibung
31	Reserviert	-	0	Die reservierten Bits werden immer als 0 gelesen. Ein Wert, in diesen Bereich geschrieben, hat keine Wirkung.
30	SDBSZ			SDRAM "bank size bit"
		2 BANKS	0	"One bank select" Pin (2 BANKS)
		4 BANKS	1	"Two bank select" Pin (4 BANKS)
29 - 28	SDRSZ		hex 0-3	SDRAM "row size bit"
		11 ROW	hex 0	11 Zeilen Adresspins (2048 Zeilen per <i>bank</i> )
		12 ROW	hex 1	12 Zeilen Adresspins (4096 Zeilen per <i>bank</i> )
		13 ROW	hex 2	13 Zeilen Adresspins (8192 Zeilen per <i>bank</i> )
		-	hex 3	Reserviert
27 - 26	SDCSZ		hex 0-3	SDRAM "column size bit"
		9 COL	hex 0	9 Spalten Adresspins (512 Elemente pro Zeile)
		8 COL	hex 1	8 Spalten Adresspins (256 Elemente pro Zeile)
		10 COL	hex 2	10 Spalten Adresspins (1024 Elemente pro Zeile)
		-	hex 3	Reserviert
25	RFEN			"Refresh enable bit". Wenn kein SDRAM verwendet wird muss RFEN = 0 sein; Andernfalls kann sich der BUSREQ durchsetzen, wenn ein SDRAM timer bis 0 herunter gezählt hat.
		disable	0	SDRAM refresh is disabled
		enable	1	SDRAM refresh is enabled

**Tab. F.6a:**  
SDCTL-Register Bitbeschreibung



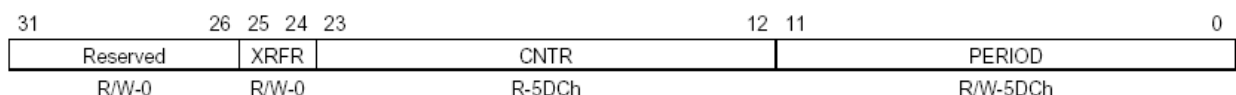
24	INIT			"Initialization bit". Dieses Bit sichert die Initialisierung aller vorhandener SDRAM
		NO	0	Keine Effekt
		YES	1	Initialisiert SDRAM für jeden dafür eingerichteten CE Speicherbereich. Die CPU (DSP) sollte alle CE Bereichs Kontroll Register und SDRAM "extension" Register initialisieren, bevor INIT auf 1 gesetzt wird.
23 - 20	TRCD	OF (value)	hex 0-F	Legt den Wert für die Zeit trcd des SDRAM in EMIF-Takt-Zyklen fest. TRCD = trcd / tcyc - 1
19 - 16	TRP	OF (value)	hex 0-F	Legt den Wert für die Zeit trp des SDRAM in EMIF-Takt-Zyklen fest. TRP = trp / tcyc - 1
15 - 12	TRC	OF (value)	hex 0-F	Legt den Wert für die Zeit trc des SDRAM in EMIF-Takt-Zyklen fest. TRCD = trc / tcyc - 1
11 - 0	Reserviert	-	0	Die reservierten Bits werden immer als 0 gelesen. Ein Wert, in diesen Bereich geschrieben, hat keine Wirkung.

**Tab. F.6b:**  
SDCTL-Register Bitbeschreibung

Das "EMIF SDRAM Timing Register" (*SDTIM*) legt die "refresh"-Periodendauer in EMIF Taktzyklen fest. Optional kann das "PERIOD"-Feld einen Interrupt an die CPU senden. So kann dieser Zähler als Takt für allgemeine Zwecke (*general purpose*) genutzt werden, falls vom System kein SDRAM benutzt wird. Die CPU kann den Zählerstand im "CNTR"-Feld auslesen. Wenn der Zähler 0 erreicht, wird er automatisch, mit dem ursprünglich für die Periode angegebenen Wert, neu geladen und "SDINT" wird (*wieder*) mit höherer Priorität behandelt.

Das "XRFR"-Feld steuert die Anzahl der "Refresh"-Wiederholungen wenn der entsprechende Zähler 0 erreicht hat. Da diese Anwendung für das zu erstellende System nicht verwendet wird, wurde nicht näher darauf eingegangen.

### EMIF SDRAM Timing Register (*SDTIM*)



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Abb. F.15:**  
SDTIM-Register

### Feldbeschreibung für das SDTIM-Register:

Bit	Feld	symval	Wert	Beschreibung
31 - 24	Reserviert	-	0	Die reservierten Bits werden immer als 0 gelesen. Ein Wert, in diesen Bereich geschrieben, hat keine Wirkung.
25 - 24	XRFR	OF (value)	hex 0-3 hex 0 hex 1 hex 2 hex 3	"Extra refreshes" steuert die Anzahl der "refresh"-Wiederholungen des SDRAM wenn der entsprechende counter 0 reicht. 1 refresh 2 refresh 3 refresh 4 refresh (unter bestimmten Voraussetzungen auch mehr als 4 möglich)
23 - 12	CNTR	OF (value)	hex 0-FFF	Aktueller Wert des "refresh"-Zählers
11 - 0	PERIOD	OF (value)	hex 0-FFF	"refresh"-Periodendauer in EMIF Taktzyklen

**Tab. F.7:**  
SDTIM-Register Bitbeschreibung

Das "EMIF SDRAM Extension Register" (*SDEXT*) ermöglicht die Programmierung von vielen SDRAM Parametern, was wiederum zwei entscheidende Vorteile mit sich bringt:

- Ermöglicht ein Interface zu einer Vielzahl von SDRAMs und beschränkt sich nicht nur auf ein paar Konfigurationen oder Geschwindigkeitsmerkmale.
- Ermöglicht dem EMIF einen ununterbrochenen Datentransfer von externem SDRAM durch "Features", wie z.B. versteckte "precharge" und mehrere "Bänke" öffnen.

Man bedenke, dass das "SDCTL"-Register nach dem "SDEXT"-Register zu konfigurieren ist.

### EMIF SDRAM Extension Register (*SDEXT*)

31	21	20	19	18	17	16	15	14	12	
Reserved			WR2RD	WR2DEAC	WR2WR	R2WDQM	RD2WR			
R/W-0			R/W-1	R/W-10	R/W-1	R/W-11		R/W-101		
11	10	9	8	7	6	5	4	3	1	0
RD2DEAC		RD2RD	THZP	TWR	TRRD	TRAS		TCL		
R/W-11		R/W-1	R/W-10	R/W-01	R/W-1	R/W-111		R/W-1		

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Abb. F.16:**  
SDEXT-Register

## Feldbeschreibung für das SDEXT-Register:

Bit	Feld	symval	Wert	Beschreibung
31 - 21	Reserviert	-	0	Die reservierten Bits werden immer als 0 gelesen. Ein Wert, in diesen Bereich geschrieben, hat keine Wirkung.
20	WR2RD	OF (value)	0 - 1	Legt die minimale Anzahl der Zyklen zwischen einem Schreib- und Lesekommando des SDRAM in EMIF Taktzyklen fest. $WR2RD = (\# \text{ of cycles WRITE to READ}) - 1$
19 - 18	WR2DEAC	OF (value)	hex 0-3	Legt die minimale Anzahl der Zyklen zwischen einem Schreib- und "DEAC/DCAB"-kommando des SDRAM in EMIF Taktzyklen fest. $WR2DEAC = (\# \text{ of cycles WRITE to "DEAC/DCAB"}) - 1$
17	WR2WR	OF (value)	0 - 1	Legt die minimale Anzahl der Zyklen zwischen einem Schreib- und Schreibkommando des SDRAM in EMIF Taktzyklen fest. $WR2WR = (\# \text{ of cycles WRITE to WRITE}) - 1$
16 - 15	R2WDQM	OF (value)	hex 0-3	Legt die Anzahl der Zyklen fest, die ein "BEx"-Signal auf "high" sein muss, bevor WRITE einen "READ" unterbricht. $R2WDQM = (\# \text{ of cycles Bex high}) - 1$
14 - 12	RD2WR	OF (value)	hex 0-7	Legt die Anzahl der Zyklen zwischen einem Lese- und Schreibkommando des SDRAM in EMIF Taktzyklen fest. $RD2WR = (\# \text{ of cycles READ to WRITE}) - 1$
11 - 10	RD2DEAC	OF (value)	hex 0-3	Legt die Anzahl der Zyklen zwischen einem Lese- und "DEAC/DCAB"-kommando des SDRAM in EMIF Taktzyklen fest. $RD2DEAC = (\# \text{ of cycles READ to "DEAC/DCAB"}) - 1$
9	RD2RD	OF (value)		Legt die Anzahl der Zyklen zwischen einem Lese- und Lesekommando ( <i>gleicher CE Bereich</i> ) des SDRAM in EMIF Taktzyklen fest. 0 READ to READ = 1 EMIF Taktzyklus 1 READ to READ = 2 EMIF Taktzyklen
8 - 7	THZP	OF (value)	hex 0-3	Legt den Wert für die Zeit thzP ( <i>auch tROH</i> ) des SDRAM in EMIF-Taktzyklen fest. $THZP = thzP / tcyc - 1$
6 - 5	TWR	OF (value)	hex 0-3	Legt den Wert für die Zeit tWR des SDRAM in EMIF-Taktzyklen fest. $TWR = tWR / tcyc - 1$
4	TRRD	OF (value)		Legt den Wert für die Zeit trRD des SDRAM in EMIF-Taktzyklen fest. 0 TRRD = 2 EMIF-Taktzyklen 1 TRRD = 3 EMIF-Taktzyklen
3 - 1	TRAS	OF (value)	hex 0-7	Legt den Wert für die Zeit tRAS des SDRAM in EMIF-Taktzyklen fest. $TRAS = tRAS / tcyc - 1$
0	TCL	OF (value)		Legt die CAS-Latenzzeit des SDRAM in EMIF-Taktzyklen fest. 0 CAS Latenz = 2 EMIF-Taktzyklen 1 CAS Latenz = 3 EMIF-Taktzyklen

**Tab. F.8:**  
SDEXT-Register Bitbeschreibung

## Quellenverzeichnis

- [1] Prof. Dr.-Ing. Hans Jürgen Micheel: Unterlagen zum Praktikumsversuch “Entzerrung von Digitalsignalen“ (*Versuch 56*), 22.04.2002
- [2] Werner, Martin: Nachrichtentechnik, Vieweg Verlag, 4. Auflage 2003, ISBN: 3-528-37433-0
- [3] Müller-Wichards, Dieter: Transformationen und Signale, B.G. Teubner Verlag, 1999, ISBN: 3-519-02742-9
- [4] Kröger, Peter; Gerdsen, Peter: Digitale Signalverarbeitung in der Nachrichtenübertragung, Springer Verlag, 2. Auflage 1997, ISBN: 3-540-61194-0
- [5] Görth, Joachim: Bauelemente und Grundsaltungen, B.G. Teubner Verlag, 1999, ISBN: 3-519-06258-5
- [6] Chassaing, Rulph: Digital Signal Processing and Applications with the C6713 and C6416 DSK, Wiley-Interscience Verlag, 2005, ISBN: 0-471-69007-4
- [7] Tietze / Schenk: Halbleiterschaltungstechnik, Springer Verlag, 2002, ISBN: 978-3-540-42849-7
- [8] Lancaster, Don: Das Aktiv-Filter-Kochbuch, IWT Verlag, 1982, ISBN: 3-88322-007-8
- [9] Kammermeyer, Karl-Dirk; Kroschel, Kristian: Digitale Signalverarbeitung, Teubner Verlag, 2006, ISBN: 3-8351-0072-6
- [10] Morgan-Voyce Algorithmus  
<http://mathworld.wolfram.com/Morgan-VoycePolynomials.html> (16.12.2007)  
<http://www.research.att.com/%7Enjas/sequences/table?a=85478&fmt=312>
- [11] spru190d.pdf, sprs266e.pdf, spra186l.pdf, TMS320C6713techref .pdf, Programming reference  
§install dir (Hilfeordner der installierten CCS Version)
- [12] AD9764, AD5444, AD9752, EB-9752, AD8047, AD9765, AD9774  
<http://www.analog.com/en/subCat/0,2879,761%255F795%255F0%255F%255F0%255F,00.html> (11.10.07)
- [13] Active Filter Design Tool  
[http://www.analog.com/Analog\\_Root/static/techSupport/designTools/interactiveTools/filter/filter.html](http://www.analog.com/Analog_Root/static/techSupport/designTools/interactiveTools/filter/filter.html) (10.11.2007)
- [14] TMS320C6000 EMIF calculator  
<http://www.dsigt.de/support/tools/c6000emif.html> (23.10.07)

## **Versicherung über die Selbstständigkeit**

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §25(4) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, den

---

(Lang, Christoph)