

embedded

Rafael Deliano
Steinbergstr.37
82110 Germering
Tel 089/8418317

j_r_d@t-online.de

V1.0 (pdf) : 13. Feb 05

V1.1 (pdf) : Nov 06

V1.2 (pdf) : 14. Jan 07



Voransichts- Version

Für Bezug des Originals
siehe FAQ auf
www.embeddedFORTH.de



- 1 Inhalt, In
- 2 29F010
- 3 Programm
- 4 Nachtrag V
- 5 POCSAC
- 7 Baudrate
- 9 125kHz RFIDs
- 13 Division, Multiplikation
- 15 Digitale Filter Einführung FIR
- 19 Kammfilter
- 23 Mittelwertfilter
- 24 Nachtrag Golay

AD . ME

lesystem für die CF-
r fast fertig. Aber eine
Schublade hat diese
g gemacht. Über digitale
weitere Artikel folgen.
stings sind in

geschrieben. Für die
g in andere FORTH-
Varianten sollte man im nanoFORTH-
Manual nachlesen das in der F08-
Version verfügbar ist.

Programmier.gerät für 29F010 FLASH-EPROM

Für 8 Bit CPUs als Ersatz für EPROMs weiterhin von Interesse. Der schwindenden Verfügbarkeit des 29F010 kann man durch Einsatz des 29F040 begegnen.

Typisch im Gehäuse DIL32 oder PLCC32 (Bild 1) angeboten. Dabei sind die Adreßpins A17 und A18 des 29F040 im 29F010 „not connected“. Was für den 29F010 zum 27C512 kompatible Sockel ermöglicht (Bild 2).

Hauptsächlicher Hersteller war AMD, second source ST und eventuell Fujitsu. Von anderen Anbietern unter dieser oder ähnlicher Bezeichnung angebotene Teile sind bezüglich Programmierung nicht kompatibel und werden hier nicht berücksichtigt.

Auch verschiedenen Generationen von AMD haben Unterschiede (Tabelle 2) und man kann sie nicht anhand des automatisch auslesbaren Device-Codes automatisch unterscheiden (Tabelle 1). Zumindest sind die Änderungen kompatibel. Bei der Resetsequenz wird der letzte Lesezugriff ein Dummy. Bei der Änderung der magischen Adressen werden bei den neuen Typen die oberen Bits als don't cares behandelt (Tabelle 3).

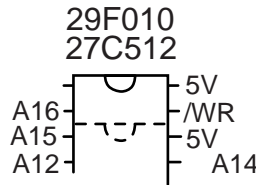
Tabelle 1: IDs

Device code	20	29F010	AMD
	20	29F010A	AMD
	20	29F010B	AMD
	A4	29F040	AMD
	A4	29F040B	AMD
	E2	29F040B	ST
Manufacturer	01	AMD	
	20	ST	

Tabelle 2: Kompatibilität

	29F010	29F010A/B
Adressen:	5555, 2AAA	555, 2AA
Reset:	lange Sequenz	kurze Sequenz
latched bei:	fallender Flanke /WE	fallender Flanke /WE oder /CE

Bild 2:
Layout als
EPROM-
Ersatz



Programmierung

Lesen der Speicherdaten kann natürlich direkt erfolgen. Bei den anderen Funktionen wird Zugriff durch das Schreiben/Lesen von Sequenzen von Bytes an den zwei magischen Adressen 5555, 2AAA eingeleitet. Beim Schreiben wird die Adresse bei fallender Flanke am /WE-Pin gespeichert, man sollte also überprüfen ob das Bustiming dafür paßt und entsprechende Logik vorsehen. In der späteren Version wird auch auf fallenden /CE gespeichert, was ein üblicheres Timing ist.

Der hier dargestellte Befehlsumfang beschränkt sich ca. auf den ursprünglichen 29F010.

Tabelle 3: making

magic numbers	
more magic	
5555	0101 0101 0101 0101
555	xxxx x101 0101 0101
2AAA	0010 1010 1010 1010
2AA	xxxx x010 1010 1010

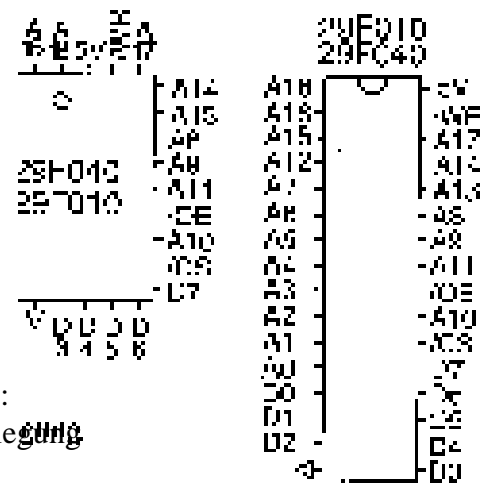


Bild 1:
Pinbelegung

Reset

Im 29F010 kann man Datenbyte per Stoppsequenz reset bekommen. Bei höherer Taktfrequenz ist jedoch nur Reset der Steuerung zu get.

Auto-select

Diese Sequenz hat wichtige Auswirkungen: 1. Gewähr UP oder Sector Protection aus. Man muß anschließend die Reset Sequenz ausführen, damit die IC in Stand-by zurück geht. Sector Protection wird nur nicht reinitialisiert, man beachte daß die letzten Programmierungen.

Chip Erase

Wenn man PLASID schreiben soll, muß es gelöscht sein. Diese Sequenz löscht den gesamten Chip komplett.

Sektor Erase

Im 29F010 sind die Sektoren 4 x 10k Bytes im 29F040 jedoch 8 x 512 Bytes (Bild 3). Das ermöglicht es die CPU zu programmieren. Während mit den Adressierungen A17, A18 kann man jedoch den 29F010 mit 10k Sektoren programmieren, wobei die unteren Sektoren dann nicht mehr auf 12k zurückgegriffen werden können.

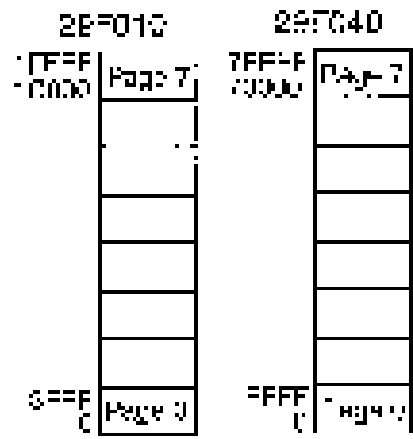
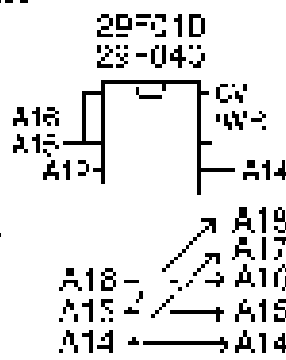


Bild 3: Pages

Bild 1: Typen für Einsatz 29F010



Write

Pages werden jeweils getrennt durch den Chip in beiden Leses-Richtungen folgend einer definierten Reihenfolge des Status umschalten wenn die IC fertig ist. Die Werte dieser Zellen aus Tabelle 1 werden über die Funktion Zündschlüssel in Form auf dem externen Programmierfeld nicht überschrieben.

Bild 2: Schaltung

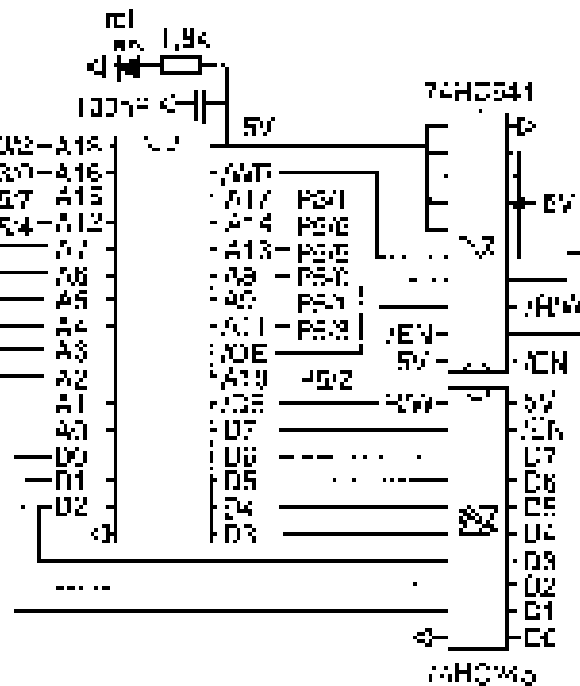


Tabelle 4: Timing

	29F010		29F040		
	ns	ns	ns	ns	ns
Chip-Enable	1,5	15	5	54	max
Success-Erase	1,5	15	5	54	max
Byte-Programmierung	7	300	5	300	max

Schaltung

Am Mittelbit -0002 Einprogrammierung liegt die H.A.S.I. an Bus in einer 256 Byte Segment. Wegen des Bus-Multiplexing-Angebot dieser ICs sind in diesem IC kein eigenes Segment für Schreibeman und eines für Leseman und durch zwei Chipstöße. Signale für die Programmierung sind über den Bus P3-P5 und über den Bus P3-P5. Die Eigenschaft von PWR der 29F010 wurde durch die UK-Code hergestellt. Für Programmierung und Benutzung des ICs kann man über P3-P5 den Status abzurufen, bis es allerdings kein Schutz gegen Manipulation vorzusehen.

Software

Die komplette Software (Lötung 29F010/40) kann durch Programmierung auf dem Chip. Software kann man über den IC nicht ändern, bis es ein Programmierfeld von Flash ICs nach dem IC-Code wird. Die ICs sind

steht für den 29F010 in Verbindung Bild 4 verwendet.

Die Befehle im Speicher werden über den Bus übertragen. Die ICs sind in einem 256 Byte Segment. Wegen des Bus-Multiplexing-Angebot dieser ICs sind in diesem IC kein eigenes Segment für Schreibeman und eines für Leseman und durch zwei Chipstöße.

Signale für die Programmierung sind über den Bus P3-P5 und über den Bus P3-P5.

Die Eigenschaft von PWR der 29F010 wurde durch die UK-Code hergestellt. Für Programmierung und Benutzung des ICs kann man über P3-P5 den Status abzurufen, bis es allerdings kein Schutz gegen Manipulation vorzusehen.

Die komplette Software (Lötung 29F010/40) kann durch Programmierung auf dem Chip. Software kann man über den IC nicht ändern, bis es ein Programmierfeld von Flash ICs nach dem IC-Code wird. Die ICs sind

steht für den 29F010 in Verbindung Bild 4 verwendet. Die Befehle im Speicher werden über den Bus übertragen. Die ICs sind in einem 256 Byte Segment. Wegen des Bus-Multiplexing-Angebot dieser ICs sind in diesem IC kein eigenes Segment für Schreibeman und eines für Leseman und durch zwei Chipstöße.

- * PWR = 1 = 1 = 01 00 01 = 1
- * PWR = 1 = 1 = 01 00 01 = 1
- * PWR = 1 = 1 = 01 00 01 = 1
- * PWR = 1 = 1 = 01 00 01 = 1

Im letzten Byte liegt ein gesetztes Bit an, und die Speicher sind geschützt. Man kann den Speicher komplett oder nur einen einzelnen Speicher-Knoten

Zyklische Codes: BCH(31,21)

Nicht nur in POCSAG-Pagern sondern manchmal auch bei 433MHz Datenfunkgeräten verwendet. Mit zusätzlichem Paritybit ist die Länge des Gesamtworts 32 Bit und damit angenehm „gerade“.

Der Code (Bild 1) korrigiert 2 Bit, erkennt 3 fehlerhafte Bits. Bei der Implementierung der Syndromberechnung (Bild 2) ist hier in Bit 0 Platz für ein Paritybit vorgesehen, die Verarbeitung erfolgt jedoch nur über die oberen 31 Bit. Encoder und tabellengestützter Decoder wurden in bekannter Form auf einem 6502 Einplatinencomputer entwickelt [1] (Listings PSG0 - PSG2.F74). Die 1k x 32 Bit Tabelle für den Decoder ist leider recht umfangreich (Listings LTAB1 - LTAB4.F74). Für einen erkannten 3 Bit-Fehler kann man in ihr Bit 0 als Flag setzen, es wäre sonst unbenutzt.

Paritybit

POCSAG verwendet ein even Paritybit. Es wird im Encoder (Bild 3) über die vorhergehenden 31 Bit gebildet. Im Decoder wird das Bit von der FEC-Stufe ignoriert. Diese muß aber nicht nur die Fehler in den Datenbits, sondern auch in den Prüfbits korrigieren, damit danach das Paritybit berechnet werden kann. Beide Stufen können wegen erkannten Fehlers die Daten für ungültig erklären.

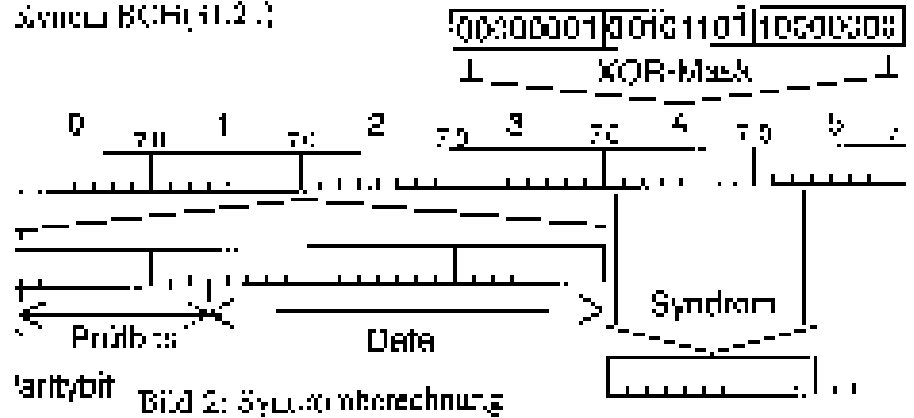
[1] emb (8) Golay

Bild 1: POCSAG Format

Bild 2: offizielle Bitnummerierung

$$G(x) = x^{31} + x^{30} + x^{29} + x^{28} + x^{27} + x^{26} + x^{25} + x^{24} + x^{23} + x^{22} + x^{21} + x^{20} + x^{19} + x^{18} + x^{17} + x^{16} + x^{15} + x^{14} + x^{13} + x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1 = 7891$$

System BCH(31,21)

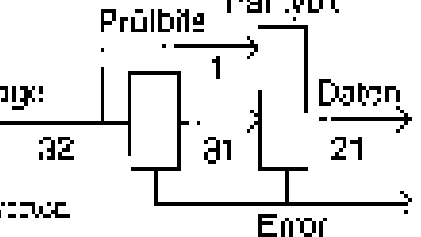


Encoder



Bild 3: Paritybit erzeugen und anhängen

Decoder



POCSAG

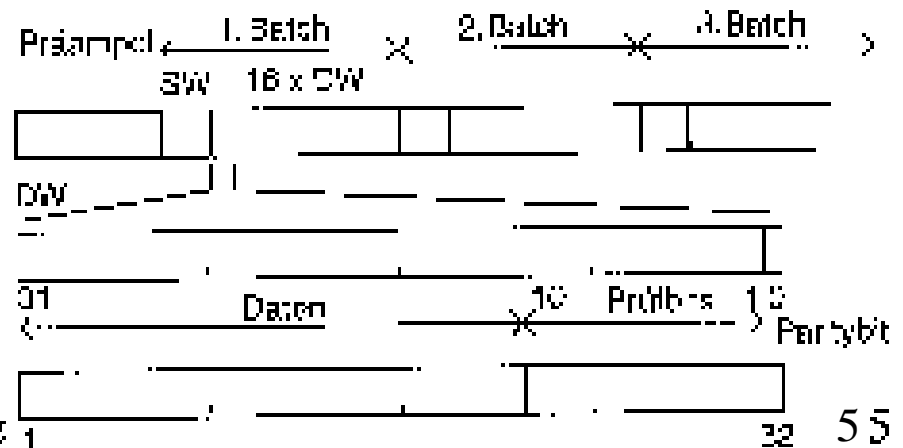
Vor der Verbreitung von Funktelefonen waren Funkrufempfänger („Pager“) recht beliebt.

Offizieller Name Eurosignal-Empfänger oder Cityruf. Hierzulande 1974 eingeführt und immer noch nicht völlig ausgestorben. Beim Pager werden von zentralen Sendern adressierte Nachrichten abgestrahlt die auf einem kleinen Empfänger der die passende Adresse hat im Display angezeigt werden. Es gibt keinen Rückkanal über den man Wiederholung einer Nachricht die gestört

empfangen wurde anfordern könnte. Weshalb fehlerkorrigierende Codes üblich sind.

Format

Das Protokoll der „Post Office Code Standardization Advisory Group“ wurde ursprünglich von British Telecom für Anwendung in England entwickelt. 1981 dann zum internationalen Standard der CCIR als Radiopaging Code No. 1 (RPC-1) und damit Verbreitung auch nach Deutschland.



Normalerweise beträgt die 500 bis 1000 Hz. Die Frequenz der 500 bis 1000 Hz wird durch die 200 bis 300 Hz bestimmt.

Grundidee: Die Frequenz wird durch die 500 bis 1000 Hz bestimmt. Die 500 bis 1000 Hz wird durch die 200 bis 300 Hz bestimmt.

Die 500 bis 1000 Hz wird durch die 200 bis 300 Hz bestimmt. Die 500 bis 1000 Hz wird durch die 200 bis 300 Hz bestimmt.

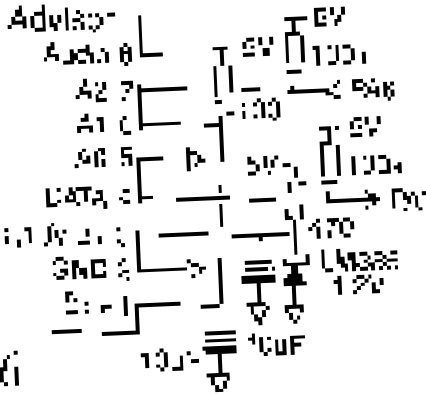
Die 500 bis 1000 Hz wird durch die 200 bis 300 Hz bestimmt. Die 500 bis 1000 Hz wird durch die 200 bis 300 Hz bestimmt.

Die 500 bis 1000 Hz wird durch die 200 bis 300 Hz bestimmt. Die 500 bis 1000 Hz wird durch die 200 bis 300 Hz bestimmt.

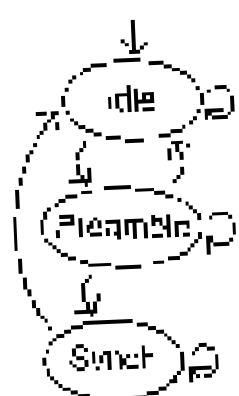
Die 500 bis 1000 Hz wird durch die 200 bis 300 Hz bestimmt. Die 500 bis 1000 Hz wird durch die 200 bis 300 Hz bestimmt.

Die 500 bis 1000 Hz wird durch die 200 bis 300 Hz bestimmt. Die 500 bis 1000 Hz wird durch die 200 bis 300 Hz bestimmt.

Teil 1: Anschaltung „Adresser“ an Controller



Teil 2: State machine



Teil 3

Die 500 bis 1000 Hz wird durch die 200 bis 300 Hz bestimmt. Die 500 bis 1000 Hz wird durch die 200 bis 300 Hz bestimmt.

Die 500 bis 1000 Hz wird durch die 200 bis 300 Hz bestimmt. Die 500 bis 1000 Hz wird durch die 200 bis 300 Hz bestimmt.

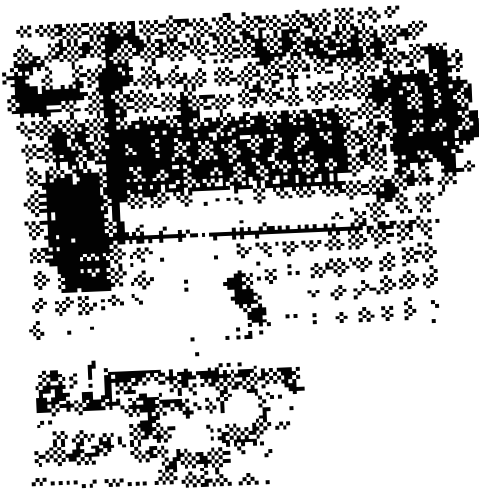
Teil 4

Die 500 bis 1000 Hz wird durch die 200 bis 300 Hz bestimmt. Die 500 bis 1000 Hz wird durch die 200 bis 300 Hz bestimmt.

Die 500 bis 1000 Hz wird durch die 200 bis 300 Hz bestimmt. Die 500 bis 1000 Hz wird durch die 200 bis 300 Hz bestimmt.

Teil 5

Die 500 bis 1000 Hz wird durch die 200 bis 300 Hz bestimmt. Die 500 bis 1000 Hz wird durch die 200 bis 300 Hz bestimmt.



Teil 6

Die 500 bis 1000 Hz wird durch die 200 bis 300 Hz bestimmt. Die 500 bis 1000 Hz wird durch die 200 bis 300 Hz bestimmt.

$$\begin{aligned}
 C_1 &= C_2 = 1 \mu F \\
 R_1 &= R_2 = 8 \text{ k}\Omega \\
 R_3 &= R_4 = 10 \text{ k}\Omega
 \end{aligned}$$

Die 500 bis 1000 Hz wird durch die 200 bis 300 Hz bestimmt. Die 500 bis 1000 Hz wird durch die 200 bis 300 Hz bestimmt.

Die 500 bis 1000 Hz wird durch die 200 bis 300 Hz bestimmt. Die 500 bis 1000 Hz wird durch die 200 bis 300 Hz bestimmt.

Teil 7: Originalsignal (oben) und verarbeitetes Signal (unten) mit Fehlermarken



Beginn einer Null-Pulsbreite ist Synchronisation erreicht.

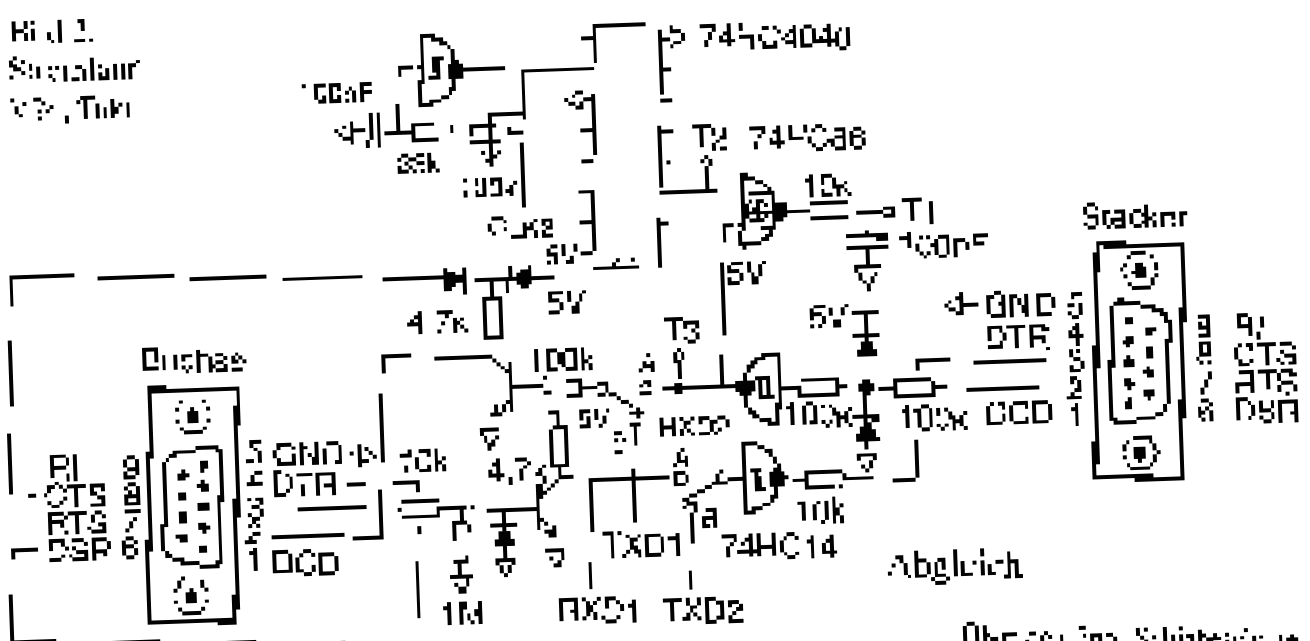
Im Ausgangszustand sind die Auswertelektroden des Detektors im gültigen Zustand. Sie sind nicht durch den folgenden Einwirkbereich angezogen, falls er vorhanden ist. Durch den Anstieg der Spannung über dem Widerstand im Kanal 12 wird gesendet, wird verschlüsselt, der UCH-Diode angeordnet. Ist dieses Ergebnis des UCH, dann kann man diese Bereiche einstellen.

Die Umkehrzeit der Schaltung und die Reaktionszeit des Detektors ausgeben. Die Entladung ist mit 10,5 ns recht langsam. Wenn ein einzelner Schritt einer HFE verwendet wird, muß die Kapazitätsausbreitung eines PDS allerdings in UHF-Bereich abgeschlossen sein. 60 GHz beträgt, kann der UCH-Diode sein. Um den UCH insgesamt über die Diode seine Treiben erreichen, deshalb nicht unterschätzen.

Wenn die Synchronisation erfolgreich ist, wird an Rechenelement allerdings optisch die Erkennung der Adressierung und die Verarbeitung der eigentlichen Nachrichten Daten sollte möglich sein.

Für die Anzeigebank, per se. Überlegen und eine weitere praktische Hinweise zu folgen, nicht ist nach bei *Abgleich* beachten.

Bild 2. Synchronisator UCH, Takt



Baudrate Adapter

Man kann den 74HC95032 zwar per PLL unteuer und hebeligen Takt einstellen, z.B. 3,3 MHz mit einem 68HC95032 zu erreichen. Aber das krummen Busfrequenz kann der interne Baudratewandler aber keine 9600 bzw. 19200 Baud machen. Also der PC kann man nicht auf keine von 9,6 Baud einstellen.

Statt man die Verbindung über UART während der Installation herstellen, ist ein Hardwareadapter die überhöhenste Lösung. Er ist 1:1. Mit hergekauft oder Custom mit eingekaufter UART und genügend Time zu eine externe UART Adapter sein. Und die weitere GPS verwendet. Die 68C51 UART kann zu initialisieren werden, daß der Quartz eingangs 10 KHz dem 12-fachen der Uhrdaten entspricht. Für diese Frequenz kann bis zu 200 KHz eignet sich ein simpler RC-Oscillator der über Epoxidharz montiert einstellbar ist. Der 74HC4040 reißt durch 32 auf die Baudrate des 68C51.

Über den 9600 Schrittschreiber wird in Stellung 4 direkte Verbindung zum Empfänger stellt. Das ist ein Programmier- und die 9600 Konverter ist vorhanden sind (Lithium-Batterie) Das ist bei dem einen Receiver mit 9600 - 9600 Hz entsprechend 9600 - 19200 Baud. Man kann die UART einstellen, daß am Ausgang 11 am 5V-Niveau ausgeht eine Gleichspannung erzeugt. Frequenzentwicklung fällt sich in 40. Je geringer die Filter, desto tiefer die Filterfrequenz. Der 68C51 ist praktisch ein die vier Multiplexer der die 12-Bits in die 10-MHz.

Der Abgleich ist natürlich nach der Zeit möglich und der RC-Oscillator entspricht nach auf Änderungen in der Versorgungsspannung. Aber es wird für eine UART. Ein Quarzgeschwindigkeit benötigt, als außer stehen.

Software

Die Adresse des Targets muß kleiner als die 940k Größe des Target sein. Dies ergibt sich durch das 32-Byte-Alignment (Ausrichtung von 32-Byte-Blocken) für die 32-Byte-IFIFO. Diese Bedingung ist für die Hardware erfüllt, denn sie müssen fest definiert werden (Bild 11). Die 32-Byte-IFIFO greift ab dem 64. Byte.

Kann es zu einer Kollision von Target mit Host, die der GF32 sendet, so ist es möglich, daß er selbst senden kann. Im Target-Modus

kann man zwei Hosts (zwei Sendepuffer für jedes Datenpaket, z.B. zwei Adressen von DRP) die kann der GF32 in einer 32-Byte-IFIFO parken. Wenn man also nicht weiß, auf die Target-Modus, dann ist im Target-Modus ein Beispiel von Datenverkehr die Lösung (Bild 12).

Alternative

Wenn man eine Lösung für die Kollision zwischen nicht über die Target-Modus senden Software machen will könnte man die GF32 durch einen zweiten GF32 ersetzen. Diese Lösung benötigt auch wenige Hardware und sehr mit 3.0V

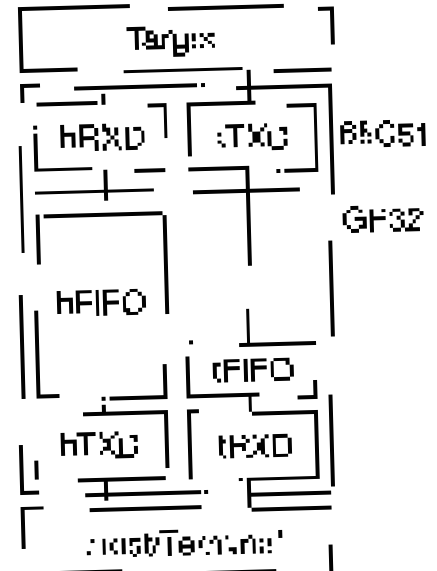
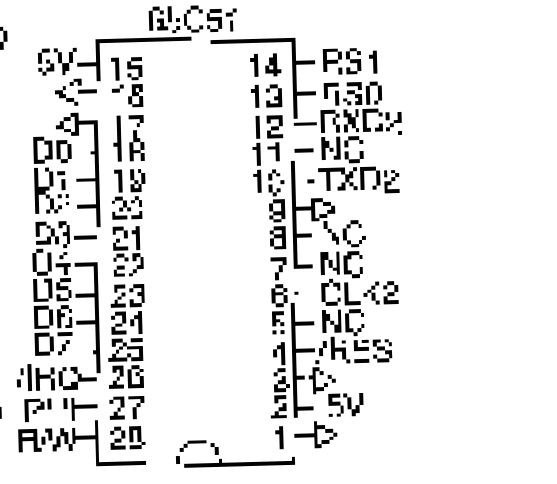
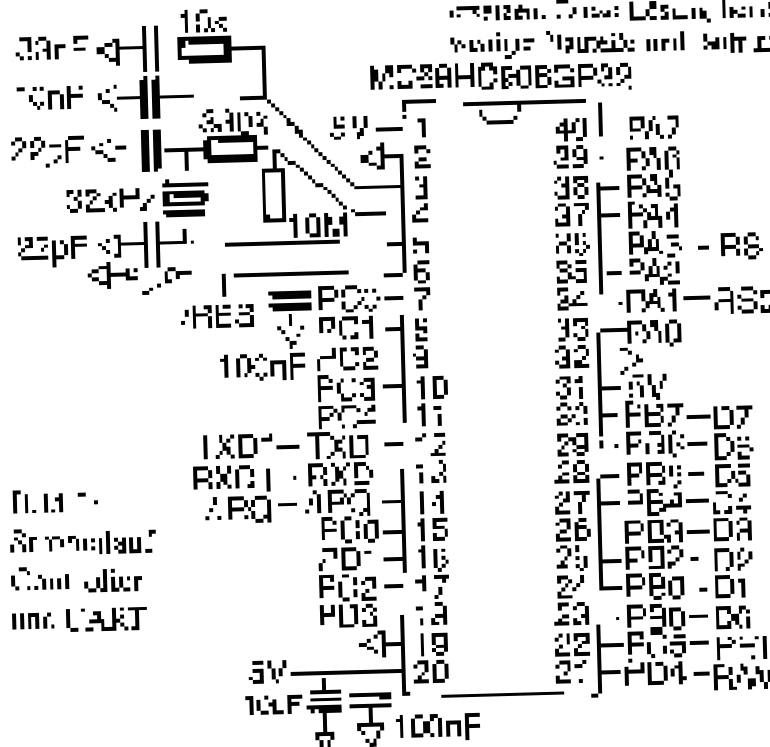


Bild 11: Hardwarearchitektur Software

Es gibt eine Lösung für die Kollision zwischen Target und Host, die der GF32 mit PC1 und Host-Modus senden kann. Im Target-Modus kann der GF32 mit Target-Modus durch die Host-Target-Verbindung.



Implementierung

Die Implementierung des Target-Modus ist ein komplexes Projekt, das die Hardware- und Software-Entwicklung umfasst. Die Implementierung des Target-Modus ist ein komplexes Projekt, das die Hardware- und Software-Entwicklung umfasst.

Die Implementierung des Target-Modus ist ein komplexes Projekt, das die Hardware- und Software-Entwicklung umfasst. Die Implementierung des Target-Modus ist ein komplexes Projekt, das die Hardware- und Software-Entwicklung umfasst.

Funktionen

Die Funktionen des Target-Modus sind die Kommunikation mit dem Host, die Verwaltung der Datenpakete und die Steuerung der Hardware.

Datenbreite

Die Datenbreite des Target-Modus ist 32 Byte. Die Datenbreite des Target-Modus ist 32 Byte. Die Datenbreite des Target-Modus ist 32 Byte.

Die Datenbreite des Target-Modus ist 32 Byte. Die Datenbreite des Target-Modus ist 32 Byte. Die Datenbreite des Target-Modus ist 32 Byte.

Die Funktionen des Target-Modus sind die Kommunikation mit dem Host, die Verwaltung der Datenpakete und die Steuerung der Hardware.

Bild 12: Syntheschaltplan TM4102

xxxxxx01	xxxxxx02	xxxxxx03	xxxxxx04	xxxxxx05	xxxxxx06	xxxxxx07	xxxxxx08	xxxxxx09	xxxxxx10
1	1	1	1	1	1	1	1	1	1
xxxxxx10	xxxxxx11	xxxxxx12	xxxxxx13	xxxxxx14	xxxxxx15	xxxxxx16	xxxxxx17	xxxxxx18	xxxxxx19
1	1	1	1	1	1	1	1	1	1

125kHz RFIDs

Als Einführung in die „Radio Frequency Identification Devices“ sei weiterhin [1] empfohlen, es ist auf der Webseite verfügbar.

Tabelle 1: Lese-ICs

Hersteller	Typ	Bauform
Atmel	U2270B	SO16
EM Marin	P4092	SO14
Philips	HTRC110	SO14
Melexis	MLX90109	SO8
Cypress	CY8C0104	SDIP24, SO14

Der Text ist allerdings schon etwas älter und beschränkt sich auf Transponder und das Lese-IC von Temic/Atmel. Mittlerweile hat sich die Auswahl deutlich erhöht (Tabelle 1). Wegen der Beschaffungsprobleme bei kleinen Stückzahlen bzw. der unsicheren Verfügbarkeit über lange Zeiträume kann für manche Anwendungen eine diskrete Schaltung sinnvoll sein.

Bei den Transpondern sind neben den Temic TK55xx vor allem die EM Marin EM4102 populär geworden.

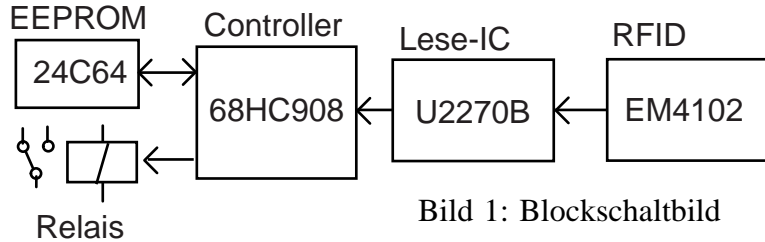


Bild 1: Blockschaltbild

In typische Anwendungen wie z.B. Türöffner (Bild 1) wird aus einem read-only Transponder die Seriennummer ausgelesen und gegen IDs in einem seriellen EEPROM verglichen [2]. Wenn eine ein-

gespeicherte Nummer erkannt wurde wird die Schaltfunktion ausgelöst.

- [1] MARC4 (4) RFID-Sonderheft
- [2] embedded (9) S. 18

Schaltung Lesegerät

Eine Variante aus Standardbauteilen erlaubt eine robuste Endstufe, allerdings ist der Demodulator etwas weniger leistungsfähig.

Um die Bauaufwand zu minimieren wird hier angenommen daß der Controller die 125kHz per Uhr erzeugen kann (Bild 1). Sowie einen seriellen RFID A/D-Wandler und genügend Speicherleistung hat, dann kann die Software auslastet werden.

Endstufe

Um zum Beispiel FN was die Endstufe abgeschlossen. Diese Versorgungsspannung kann von 5-15V variieren. Die 125kHz Endstufe ist von einem Typen in T0962. Diese ist eine für U1102010. Die Eingangsleistung war 110220. Es ist mehr Daten. Ein gutes Abstimmung mit Technik besteht. Beim Anschluss an die Antenne auf der Versorgungsspannung. Deshalb sollten Messen und

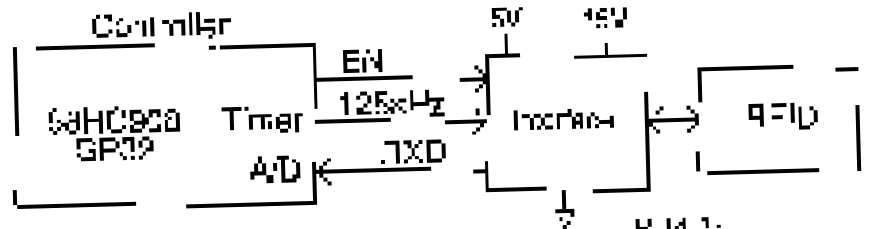
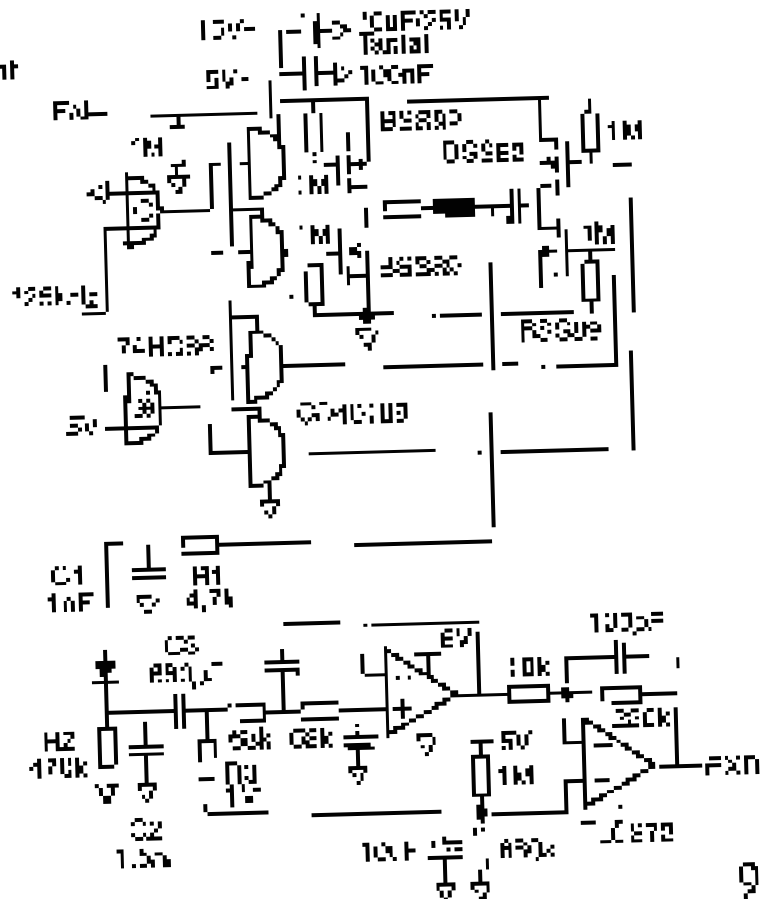


Bild 1: Blockschaltbild

Bild 2: Stromlauf



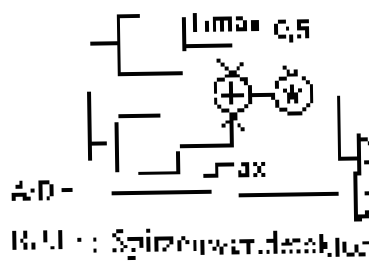


Bild 1: Spitzenwertdetektor

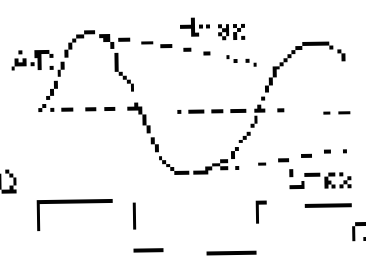


Bild 2: Quantisierung Pegel

werden nur mit dem in gewissen periodischen Wert von 1/20 abgetastet. Das Verfahren hat Einfluss auf die Qualität, nicht optimal. Deshalb gibt es bessere höhergradige Lösungen.

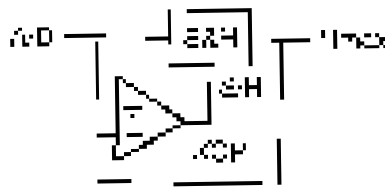


Bild 3: Spitzenwert

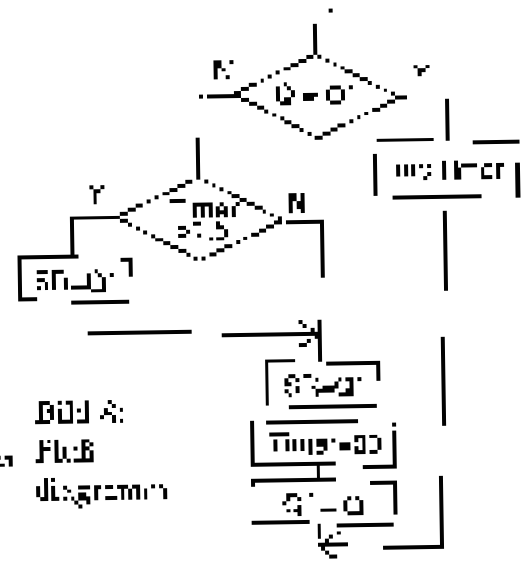


Bild 4: Flussdiagramm

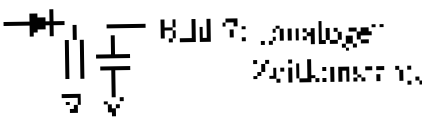


Bild 5: analoger Zeitkammer



Komparator

Wegen der Anschaulichkeit wird die Schaltung des Analogzeitkammer dargestellt. Diese besteht aus dem A/D-Wandler und dem Komparator (Bild 6). Die Funktion entspricht einem Komparator dessen Schaltschritte nach dem Wert aus dem Ausgangspunkt von zwei Schritten (Bild 7).

Bezüge des Signalpegels werden durch die Wert im Speichertheorie und dieser jeweils nachgewiesen (Bild 8). Wenn es keine Störung langfristig auftritt werden die Datenlösung (Bild 9) möglich auf neuen Signalpegel durch schnelleren Ladung, was zu einem gewissen Grad wird aus der Dichte der Systemverteilung in die Signalhöhe festgelegt, um durch einen durch 12 Codes, ist die Anzahl werden die Spaltenverteilung mit dem 12-dimensionalen Anzeigern

Manchester

Die Zeitmessung erfolgt bei Manchester. Die Zeitkammer, wenn man es selbst mit dem Wert des letzten Bits 0' per NGR verfährt (Bild 8) ergibt ein Manchestercode (Bild 9). Ein nur die Lage ist nicht fest, die Schrittschritte sind hier über 1,5 bis (elementare Timing mit 1 verwendet, Bild 8). Der Phasenverhältnis erstellen, dann man die Bauteile wechseln (Tabelle 1).

Tabelle 2:

Manchester Bits zu man

- 0 = kurzer Puls
- 1 = langer Puls
- 00 = langer Puls
- 11 = langer Puls

Tabelle 3:

Synchronisierung mit Header

- Header = 0 1 1 1 1 1 1 1
- Header = 0 1 1 1 1 1 1 1

Bild 10:

Manchester Manchester

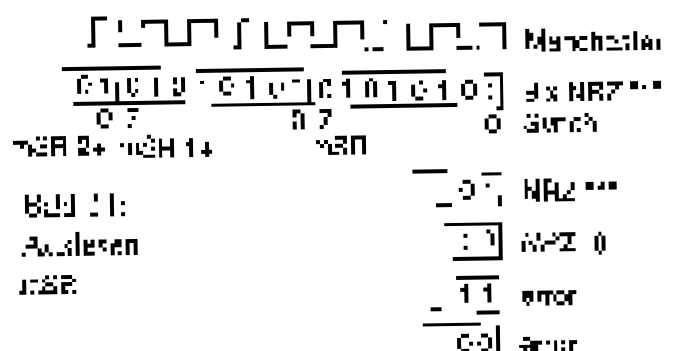


Tabelle 1: Samplerate

NRZ	Manchester	Sampling Rate
NRZ	Manchester	8
NRZ	Manchester	5

Die ergebnisse der 2 Bits (Tabelle 2) werden in das 2 Bits lange Schrittschritte mit eingewiesen (Bild 11) und lassen Elemente (NRZ) passend in dem Header. Dieses wird durch 70 bis 100 bis 2 Bits (100 bis) im Header.

Synch

Synchronisierung mit Header (Tabelle 1) auf Filterkammer (Bild 10) oder 15 bis zu erweitern und diese Werte mit den Header durch beide 1 10

Shift & Add

Division und Multiplikation

Einige Controller haben keine entsprechenden Opcodes oder für Anwendungen mit überlangen Wortlängen braucht man Sonderbefehle.

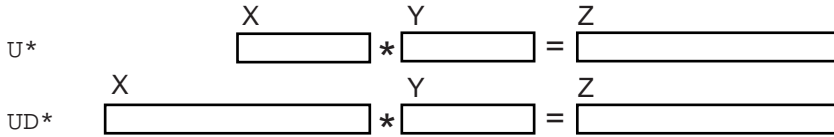


Bild 2: Wortlängen Multiplikation

Es werden hier nur vorzeichenlose Befehle behandelt. Auf diese können vorzeichenbehaftete Befehle aufbauen, wenn man die per se die Operationen der Operanden aufteilt. Dazu Datenwerte und Vorzeichen-Ergebnisse getrennt berechnen. Vorzeichen des Vorzeichens wird der im Datenwert eingegeben (Bild 1).

Multiplikation

In der Source-Version U* hat das Ergebnis die doppelte Länge der Operanden (16*16) (im Architecture-Modus, siehe Bild 3). Einem Beispiel mit 8-Bit Operanden und einem 16-Bit Ergebnis X und Source Register

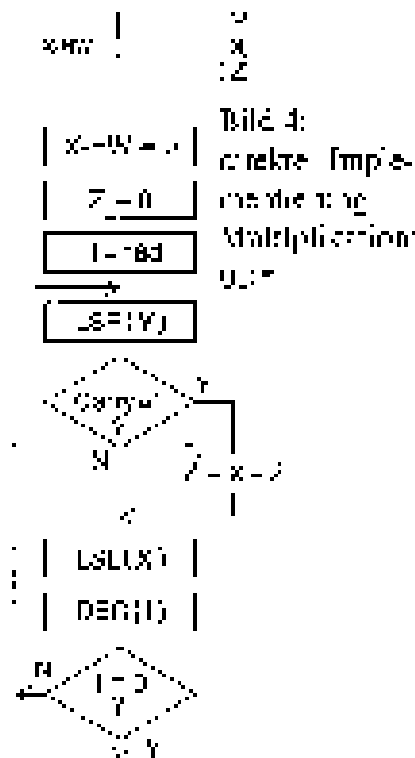


Bild 4: direkte Implementierung Multiplikation U*

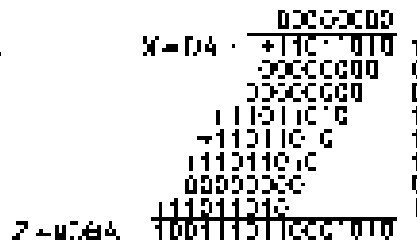


Bild 3: Beispiel Multiplikation

nach rechts bewegt, wenn man eine 2 addiert, wenn es entsprechende Bit in Y gesetzt ist. Das entsprechende Bitdiagramm in Bild 3 ist auf 16 Bit Operanden mit 32 Bit Ergebnis ausgelegt

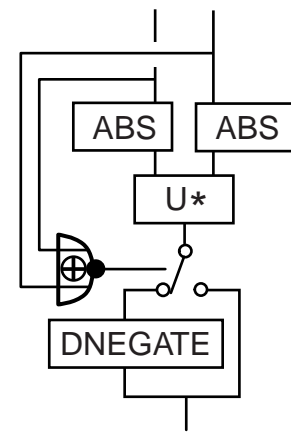
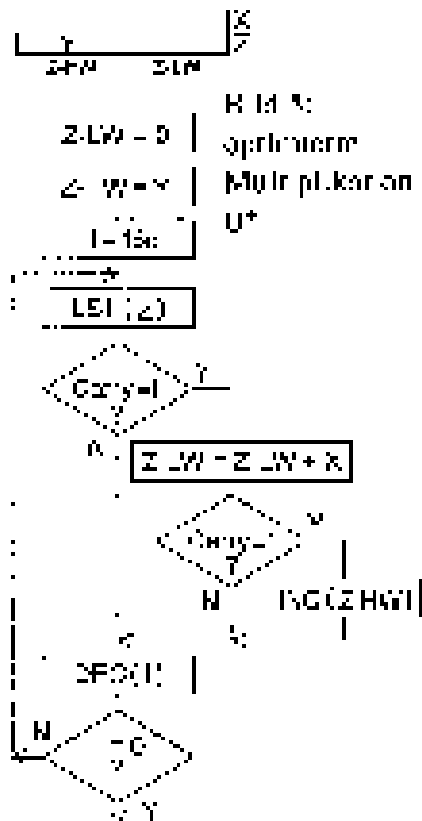


Bild 1: Multiplikation mit Vorzeichen

Prinzipiell vorzeichenlose Multipl. sind durch Optimierungen anwenderorientierter wie in Da X-ARM keine echten Daten enthält wird es als Speicher für Y verwendet. Die 16-Bit Addition in Z auf 32-Bit 5-Bit Addition für das CF und einer gemeinsamen Inbetrieb für das 1-W aufgespalten (Listing M3.106-71). Da man Y von MSB bis rechts erfolgreich der Addition der X in ungelöster Reihenfolge, also mit den hochgradigwertigen Stellen zuerst

Für UD* werden solche Tricks aus, mal man implementiert, wieder nach Bild 4 (Listing M3.107-74).

Als Alternative zur Multiplikationsbefehle laden kann man aus dieser Programm für großen Wortlänge zusammensetzen (Listing M3.108-74). Dies ist etwas, aber unbekannt, dass mehr Speicher und auch oft aufwändiger in der Fertigung.

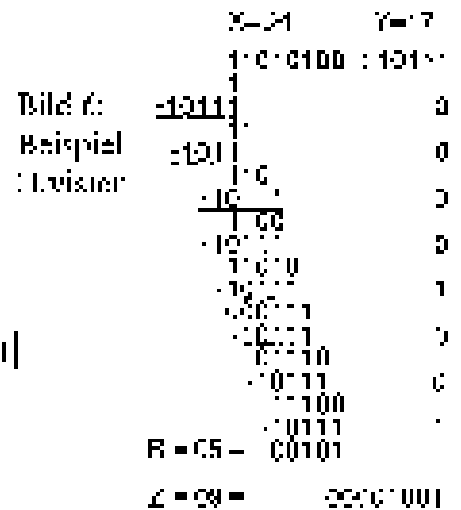


Bild 6: Beispiel Division

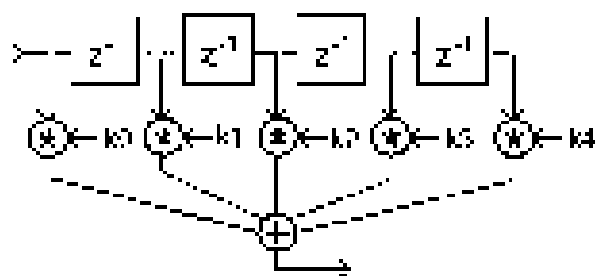


Bild 8:
FIR in Standardform, Filter mit 5 Koeffizienten

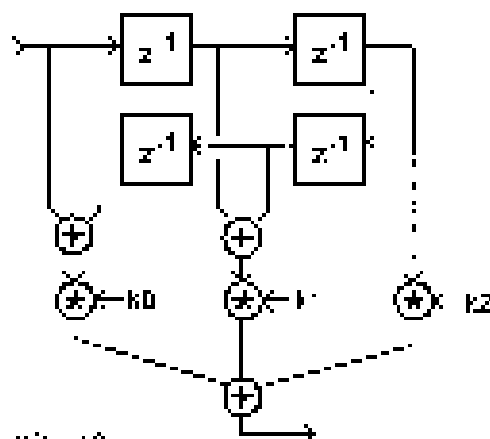


Bild 10:
FIR Typ I mit 5 Koeffizienten:
mögliche Variante

Bild 9:
FIR-Typen

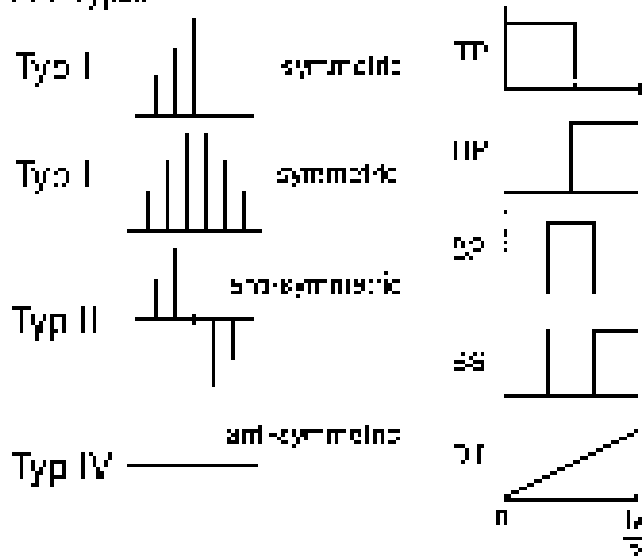


Bild 12:
Übliche
Frequenz-
gänge

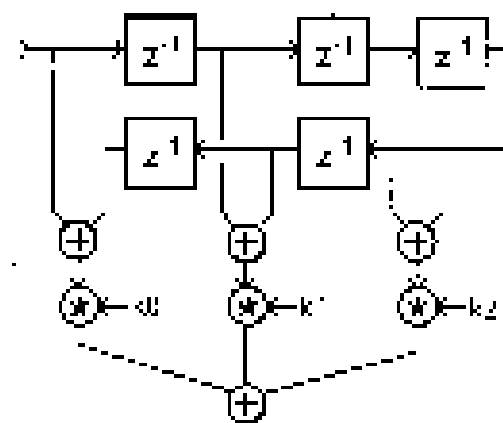


Bild 11:
für Typ II mit 6 Koeffizienten
mögliche Variante

Typ I $f(x) = \left| \sum_{n=0}^{\frac{N-1}{2}} k(n) \cdot \cos(\omega \cdot (n-x)) \right|$

Typ II $f(x) = \left| \sum_{n=0}^{\frac{N-1}{2}} k(n) \cdot \cos\left(\omega \cdot \left(n - \frac{1}{2}\right)\right) \right|$

Typ III $f(x) = \left| \sum_{n=0}^{\frac{N-1}{2}} k(n) \cdot \sin(\omega \cdot (n-x)) \right|$

Typ IV $f(x) = \left| \sum_{n=0}^{\frac{N-1}{2}} k(n) \cdot \sin\left(\omega \cdot \left(n - \frac{1}{2}\right)\right) \right|$

Bild 13:

Formeln zur Berechnung des Frequenzgangs abhängig vom Grundtyp

Bild 13:
Mögliche Frequenzgänge
abhängig von Grundtypen

Typ I	TP	HP	BP	BS	Dif
Typ II	TP		BP		Dif
Typ III			BP		Dif
Typ IV		HP	BP		Dif

Dabei der Hinweis der Verzögerung von der Grundlänge des Filters entspricht. In Bild 8 nur also 2 Samples. Bei gleicher Anzahl der Koeffizienten, wie in Bild 6, stellt man sich es als 2,5 Samples Verzögerung. Diese kann man mit Verschiebung keine umfassen sein, wenn der Filter parallel aufgebaut wird.

Der Typ der Symmetrie der Koeffizienten bestimmt eventuell auch die Form des resultierenden Filters, ob es ein Tiefpaß, Hochpaß, Bandpaß oder eine Bandstoppe sein kann.

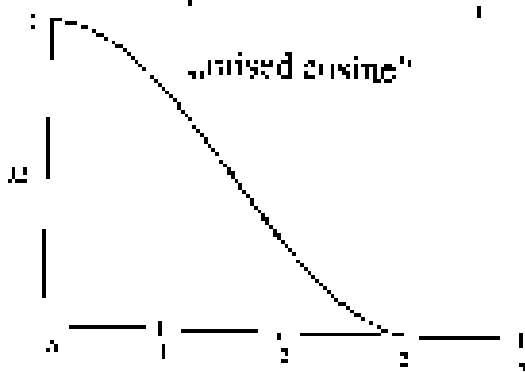
FD Typ I

$x = 0,001, N = 5, n = 0, (N-1) \cdot \frac{n}{N}$

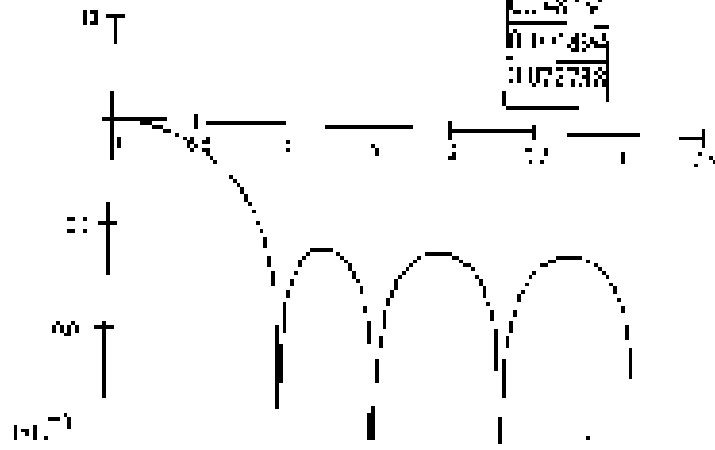
$x = \frac{0,001}{2}$

$k_1 = \begin{bmatrix} 0,25 \\ 0,5 \\ 0,75 \end{bmatrix}$

$$x(n) = k_1 + \sum_{i=1}^{N-1} 2k_{i+1} \cos(\omega_i n)$$



$$r(n) = \sum_{i=1}^N \frac{2k_{i+1}}{2^{i-1}} \cos\left[\pi \cdot \left(n - \frac{1}{2}\right) \cdot i\right]$$



ÜB 15:
Berechnung von
Frequenzgänge
von FIR Filtern
mit Matlab

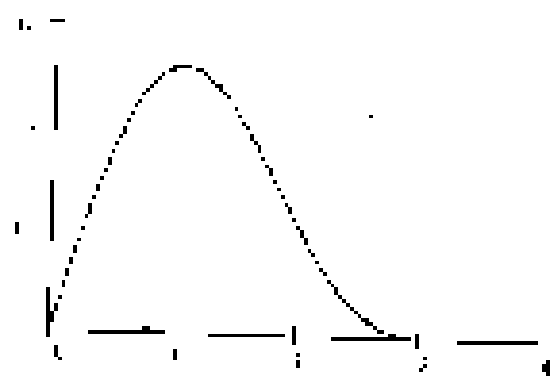
FD Typ III

$x = 0,001, N = 5, n = 0, (N-1) \cdot \frac{n}{N}$

$x = \frac{N-1}{2}$

$k_1 = \begin{bmatrix} 0,25 \\ 0,5 \\ 0 \\ -0,5 \\ -0,75 \end{bmatrix}$

$$x(n) = \sum_{i=1}^N 2k_{i+1} \sin(\omega_i n)$$



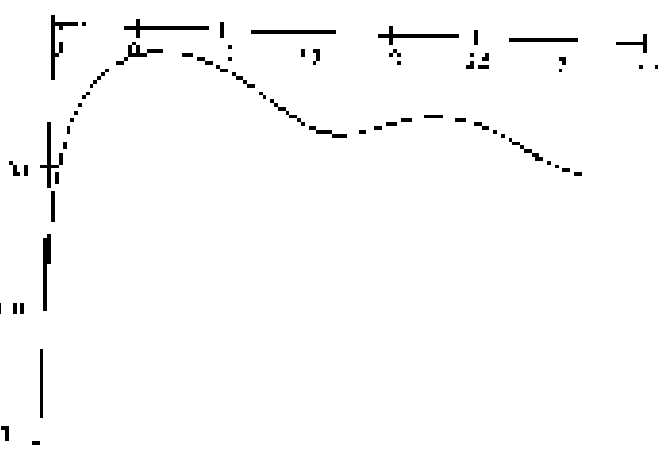
FD Typ IV

$x = 0,001, N = 5, n = 0, (N-1) \cdot \frac{n}{N}$

$x = 0,001, n = 0, (N-1) \cdot \frac{n}{N}$

$k_1 = \begin{bmatrix} 0,0625 \\ 0,125 \\ 0,1875 \\ 0,25 \\ 0,3125 \\ 0,375 \\ 0,4375 \\ 0,5 \end{bmatrix}$

$$x(n) = \sum_{i=1}^N \frac{2k_{i+1}}{i^2} \cos\left[\pi \cdot \left(n - \frac{1}{2}\right) \cdot i\right]$$



Am besten wichtige Anwendung für R/D ist eine Hilstruktur (Bild 12, Bild 13).

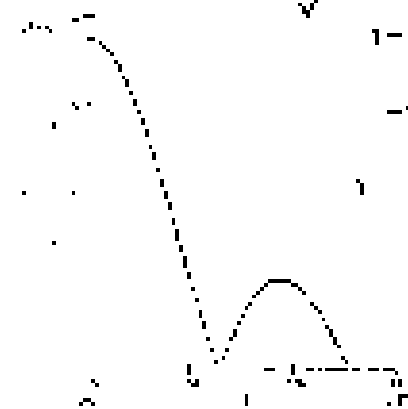
Am den Frequenzgang schmalbandig getrennt. Kein Zusammenhang. Aber sind man ist die spezielle Formel für die + Filter, welche über $\omega = 2\pi \cdot f$ (Berechnung mit Maßstab) sind man aus Formel leicht über die Bilder (siehe Bild 15).

Vorzeichen

Es gibt sehr simple Theorien, die es den für positive Koeffizienten einfluss. Dann kann man die unpolaren Daten aus dem AD-Modell direkt verwenden und verschärfen, indem, was natürlich besser ist. Man kann den Aufwand sinken.

Der Regel ist aber, das sie sich immer auch negative Werte haben und die Daten im Eingang machen sie AD-Signale darstellen und damit bipolar sind. In diesem Fall kann man sie auch einen Konstanten auf 2.5V und erreicht dann man, das es direkt D/D frei ist. Bild 14). Das unipolare Signal des AD-Wandlers muß dann noch mit der Komplement gewandelt werden, was über die Subtraktion der Konstanten einfach ist.

Bild 16: Wandlerung in typischer Zeit-Komplement



Verfüllung

Formel für den Frequenzgang wie in Bild 15 hergeleitet, gen Quantisierung nicht und sind dann oft zu optimieren. Wenn man ein Ding auf dem Computer macht hat man man das die Inputwert bestimmen (1) und aus dieser über FFT den Frequenzgang berechnen. Zwei von der Zahl der Samples (Kausalität) (zuerst positiv), was quasi durch Invertieren die Auflösung des Prozents. Man sind aber immer noch die Fehler des AD-Wandlers berücksichtigen. Hier ist genau der Wert der über

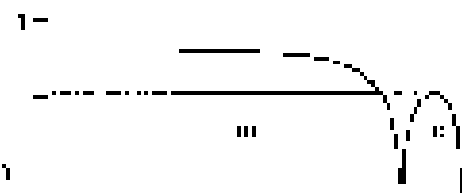
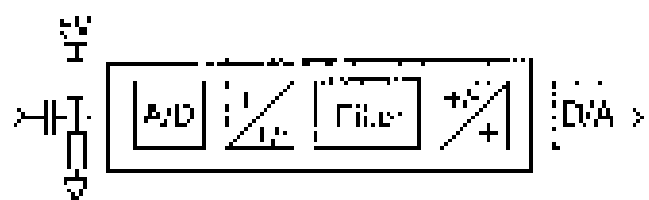


Bild 17: Tiefpaß und logarithmische Darstellung

die Telefonbandbreite (300 - 3400 Hz) mit 8 kHz multiplizieren können sich hier besten 20 dB Norm einschleichen. Die Messung ist während Zellen selbst ein wenig über die Filter in Betracht (Bild 7).

Platz

Es ist Multiplikatoren und dann auch auf Maßzahlen von Werten. Die Frequenzgang als Diagramm (log-log) dargestellt. Bei digitalen Filtern sind oft die in dem Vergleich von Maßzahlen, sollte man sich Maßstab merken (Bild 17).

Quelle: [1], S. 12

Multiplikation mit Vorzeichen

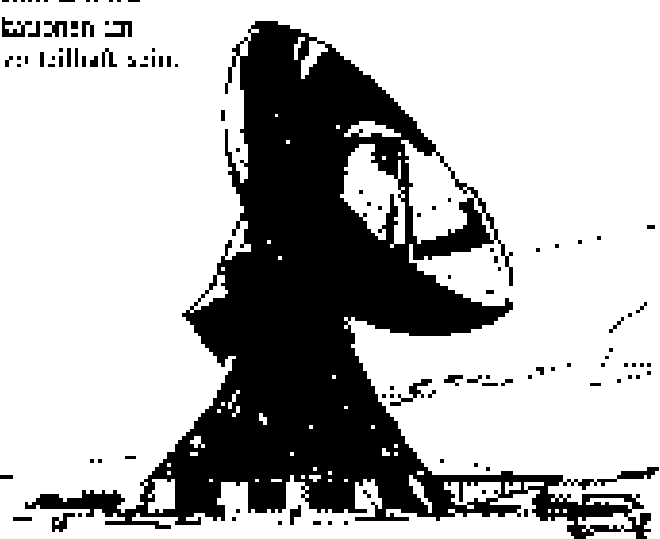
Umgekehrt, aber in der Literatur so gesehen, ist das Schema für Multiplikation mit Vorzeichen aus Bild 14 (siehe [1]). Dabei müssen nur die Multiplikationen die hier mit

8-Bit-Verfahren sind im Vergleich ausgeführt werden. Die Variante aus Bild 15 ist. Entweder die solche Teilmultiplikationen im Rechner zu haben zu billiger sein.

[1] Bild 14, S. 12

X_{HE}	X_{LE}	*	Y_{-B}	Y_{LB}	
X_{HE}		*	Y_{-B}		= $\begin{bmatrix} 8 & 8 \end{bmatrix}$
X_{HE}		*	Y_{+B}		= $\begin{bmatrix} 8 & \end{bmatrix}$
X_{HE}		*	Y_{-B}		= $\begin{bmatrix} 8 & \end{bmatrix}$
X_{LE}		*	Y_{+B}		= + $\begin{bmatrix} & \end{bmatrix}$
					$\begin{bmatrix} & & \end{bmatrix}$

Nachtrag zu ④



Kammfilter

Am bekanntesten in der Grundform Differenzierer wird es auch als Grundbaustein komplexerer Filter verwendet. Vorteilhaft ist, daß keine Multiplikation benötigt wird.

Eigentlich ein FIR-Filter (Bild 1), die üblichen Formeln zur Berechnung des Frequenzgangs gelten also. Während in den meisten Filtern nur Verzögerung um ein Sample verwendet wird, sind hier mehrstufige Verzögerungen nötig. Deshalb werden Flußdiagramme durch angepaßte Notation vereinfacht (Bild 2). Es gibt nur zwei Grundvarianten die sich im Vorzeichen der Summierung unterscheiden (Bild 3) was zu Reduzierung auf Blockschaltbild Anlaß gibt (Bild 4).

Bild 5 zeigt den Frequenzgang für verschieden tiefe Verzögerungen. Es handelt sich um gleichgerichtete Sinus- und Kosinusverläufe abhängig davon ob man Addition oder Subtraktion verwendet. Die Zahl der Kerben erhöht sich mit der Zahl der Speicherstellen. Phasenverlauf ist linear, da es ein FIR ist.

Chroma

Bild 7 zeigt ein digitales Fernsehfilter aus den 80er Jahren (DIGIT 2000 Intermetall) das mit 17,8 MHz getaktet wurde [1]. Auch ein gutes Beispiel für Controller.

Implementierung

Durch die Addition bzw. Subtraktion erhöht sich mit jedem Teilfilter die Wortlänge um 1 Bit. Wenn das Datenwort durch sofort Rückskalierung fest auf 8 Bit halten will (Bild 8) werden die Quantisierungsfehler zu hoch. Besser ist es mit 16 Bit Wortlänge zu rechnen (Bild 9) und erst vor der Ausgabe zu skalieren. Die Impulsantwort (Bild 10) und die Messung mit dem Wobbler (Bild 11) entsprechen dann den Erwartungen.

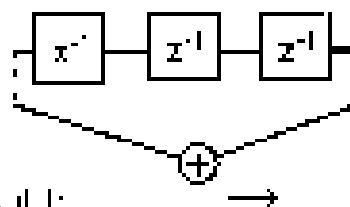


Bild 1: Kammfilter mit Tap für Kanalfilter

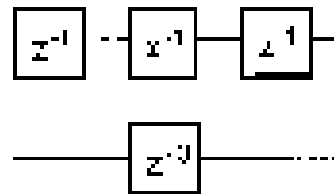


Bild 2: Komplexere Notation mehrstufige Verzögerung

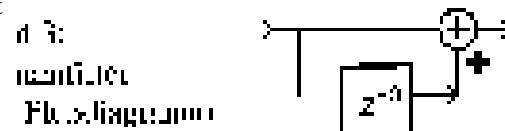


Bild 3: Grundvarianten für Summierung

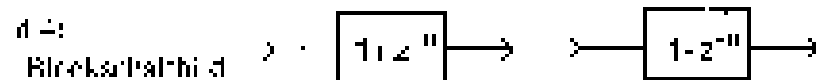


Bild 4: Reduzierung auf Blockschaltbild

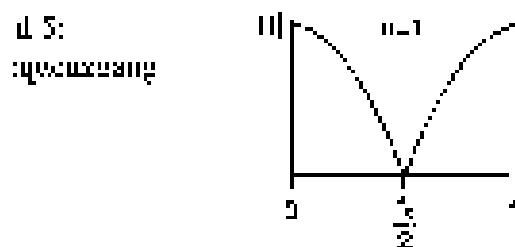


Bild 5: Frequenzgang

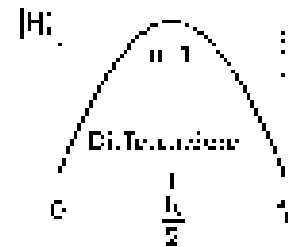


Bild 6: Differenzierer

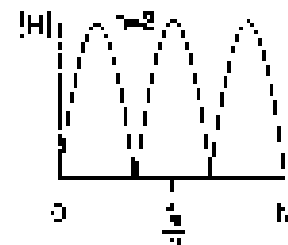
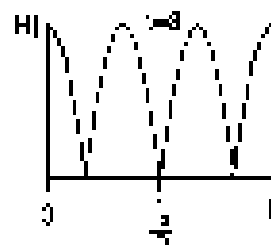
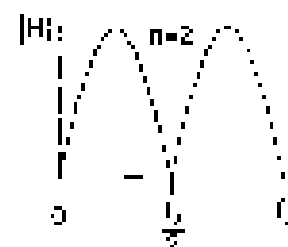
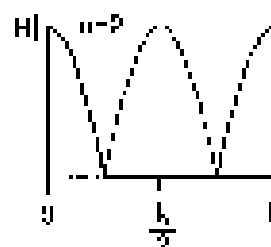


Bild 8: Formel für Impulsantwort

$$f(x) = \left| \cos\left(\frac{n \cdot x}{2}\right) \right| \quad f(x) = \left| \sin\left(\frac{n \cdot x}{2}\right) \right|$$

$$x = 0 \dots 2\pi$$

Bild 8: sofortige Rückskalierung

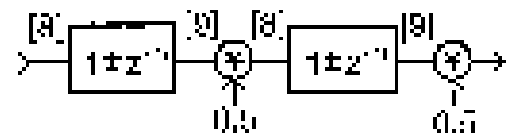
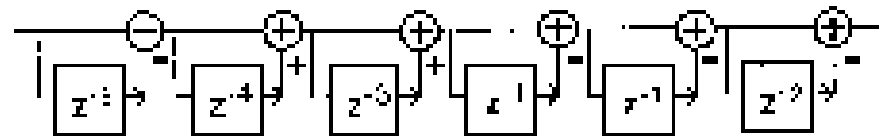


Bild 9: abschließende Skalierung



Bild 10: Lamm-Handpaß, Signalfluß und Frequenzgang



$\omega = 0,001 \pi$

$x_1(n) = \sin\left(\frac{5n}{2}\right)$, $h(n) = \cos\left(\frac{5n}{2}\right)$, $x_2(n) = \cos\left(\frac{5n}{2}\right)$, $h_1(n) = \sin\left(\frac{5n}{2}\right)$, $x_3(n) = \sin\left(\frac{5n}{2}\right)$, $h_2(n) = \cos\left(\frac{5n}{2}\right)$

$z(n) = \cos(n) \cos(5n) + \sin(n) \sin(5n)$

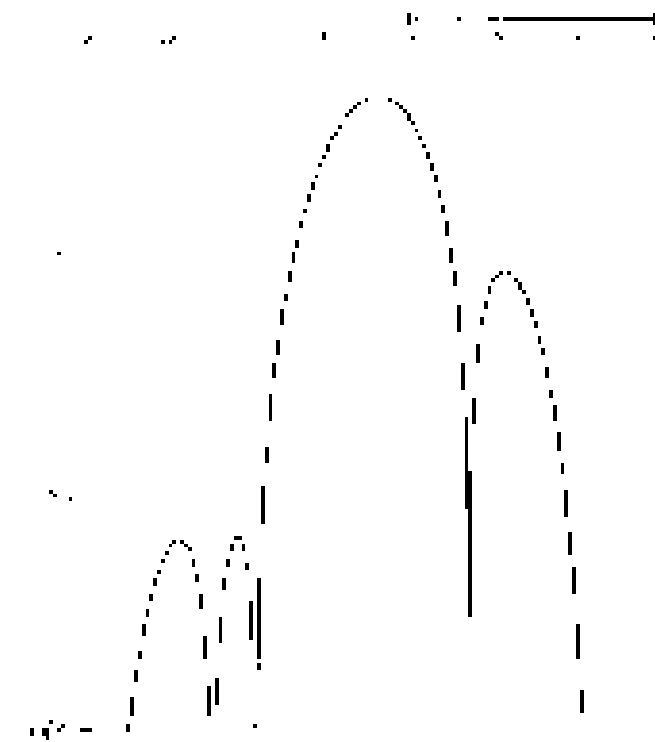


Bild 10: Impulsantwort

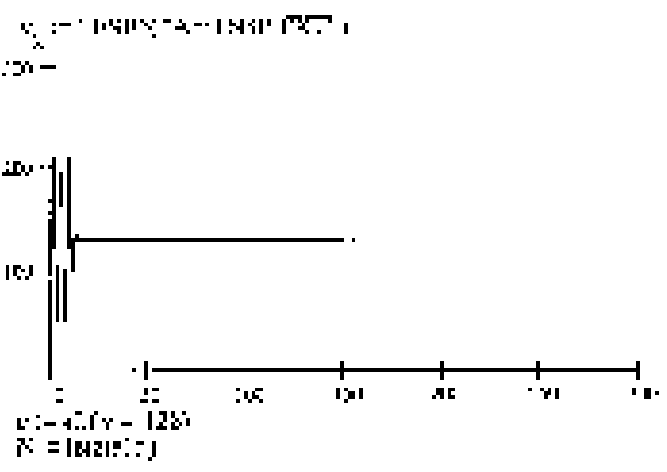
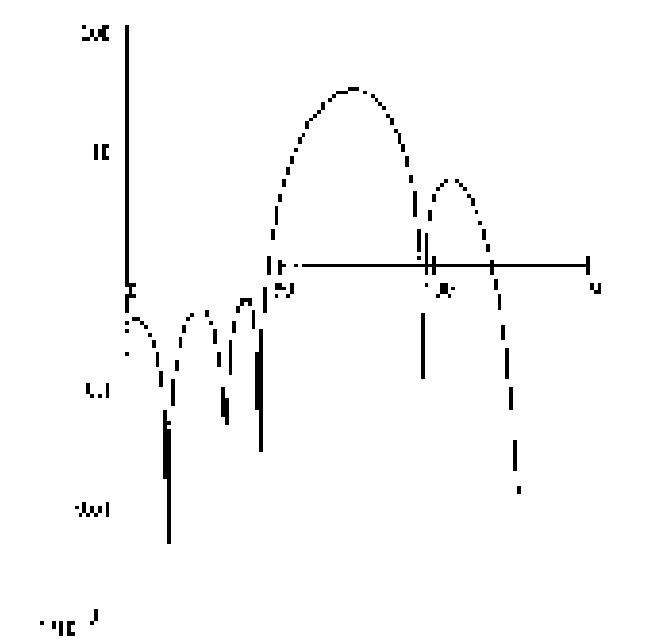


Bild 11: Mit Wobbler



Bei 8 MHz Betriebsrate des 68HC000GE52 beträgt die Rechenzeit pro Sample-Ausgabe, der Filterkern belegt 140 Byte (Listengröße 15000). Durch die 64 Bit-Steuerung des 68HC000GE52 ist die Filterkern-Entwicklung möglich.

Bild 9: 2 Stufen, damit auch kein Überschlag auftreten. Achtung: Das Eingangssignal wird über die weitere obige möglich die Rechenzeit um 5 µs zu reduzieren und deutlich verbessern.

Bei der Impulsantwort (Bild 10) von der 3. Sekunde Reife, der Sinus (Bild 11) nur ein weiterer Schritt notwendig. Solche Schritte sollten bei suboptimalen Eingangssignal eine Sättigungsgrenze als Bedingung haben.

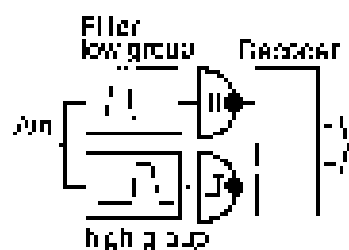


Bild 12: DTMF-Dekoder mittels band spli. Filter

Ansatz DTMF-Filter

$$x(t) = \cos(\omega_1 t)$$

$$y(t) = 2 \cos(\frac{\omega_1 + \omega_2}{2} t)$$

$$z(t) = 2 \cos(\frac{\omega_1 - \omega_2}{2} t)$$

$$u(t) = 2 \cos(\frac{\omega_1 + \omega_2}{2} t)$$

$$v(t) = 2 \cos(\frac{\omega_1 - \omega_2}{2} t)$$

Bild 13: DTMF-Filter (high group)

Bild 14: DTMF-Filter (low group)

Ansatz DTMF-Filter

$$x(t) = \cos(\omega_1 t)$$

$$y(t) = 2 \cos(\frac{\omega_1 + \omega_2}{2} t)$$

$$z(t) = 2 \cos(\frac{\omega_1 - \omega_2}{2} t)$$

$$u(t) = 2 \cos(\frac{\omega_1 + \omega_2}{2} t)$$

$$v(t) = 2 \cos(\frac{\omega_1 - \omega_2}{2} t)$$

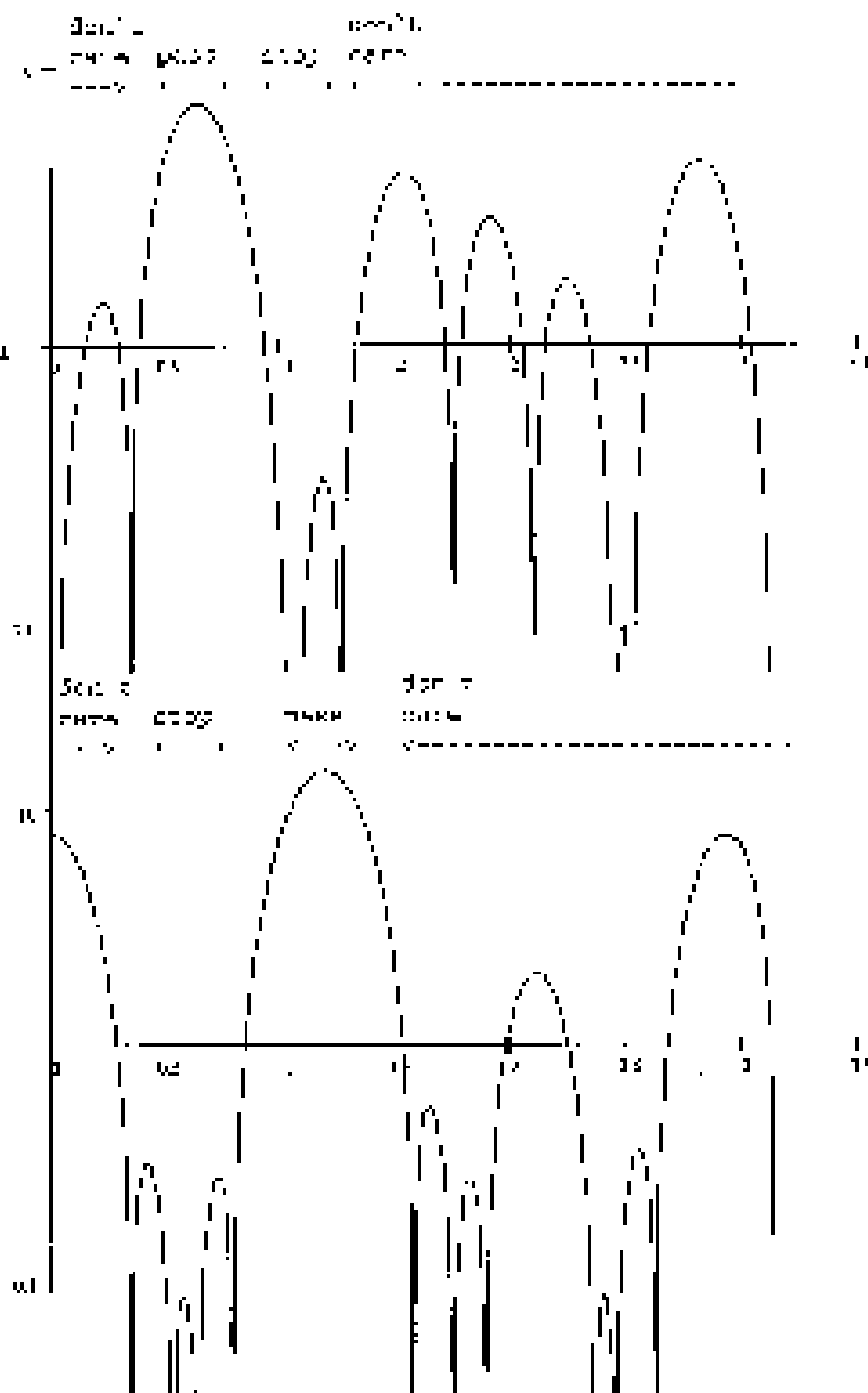
$$w(t) = 2 \cos(\frac{\omega_1 + \omega_2}{2} t)$$

DTMF

Die Variante kann ebenfalls leicht realisiert [2].

Wenn man in einem DTMF-Empfänger [1] die beiden Frequenzgruppen nicht in einem Kanal mit dem Signal durch ein Bandpaßfilter wandeln und dann einfach per Controller auslesen kann (Bild 12) [2], so müßte man sich mit DTMF-Filter beschäftigen, die nicht sind zum in SU Technik angelehrt.

Ein Filter zu geben einen etwas willigen Zusammenhang Bild



13, 14) zu sehen, die die wesentliche Anforderung eine Frequenzgruppe zu durchlassen, eine andere herauszulassen und den Rest als Nebenband zu behandeln.

Fazit

Der Filtertyp wird in seiner ersten Form mit den Koeffizienten [1] zur Konstruktion für komplexe Aufgaben leicht eingesetzt, weil diese Koeffizienten schon die Koeffizienten für eine 0,25-Entwicklung. Damit sind

meist vorhandene Implementierungen möglich. Ein Einsatz in praktischen Schaltungen z.B. in DTMF-Decoder-SICs haben kann werden, insgesamt sind je jeder tolle Stunden es sich um ein Konzept des von heute sollte.

- [1] Francis (Elektronik) 1978/81
- [2] General, Sueddeuts. Agrar-Industrie, Implementation of DTMF Receiver, Trans. IEEE, COM-32, 7/1984
- [3] Bild 13: DTMF

8 Bit Kammfilter

Soweit zum Teil einfacher Filter mit der eingeschalteten Systemklocke sind, kann man natürlich auch hier auf 8 Bit aufrunden (Bild 1).

Im 68HC08 wird hierzu aber die Rückskalierung nach einer Summation nicht genau beachtet (Tabelle 1). Für 2e Komplementzahlen wird die Oberhälfte nicht auf 8 Bit im Neg-Register

QTR der CPU angezeigt. Hier muß eine Skalierung des Carry Bits vorgenommen werden. Beim 68HC08 allerdings umständlich, da das 9. Bit nicht extra per Breakcode erreichbar ist.

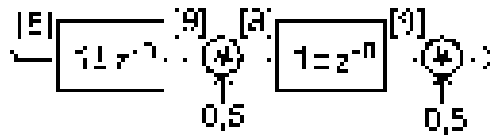


Bild 1: Kammfilter mit Rückskalierung

Tabelle 1: Addition von 8 Bit zur Komplementzahlen

8 Bit

hex	hex	hex	hex	hex	hex
hex	hex	hex	hex	hex	hex
00	00	00	00	0	0
01	01	01	01	1	1
02	02	02	02	2	2
03	03	03	03	3	3
04	04	04	04	4	4
05	05	05	05	5	5
06	06	06	06	6	6
07	07	07	07	7	7
08	08	08	08	8	8
09	09	09	09	9	9
0A	0A	0A	0A	A	A
0B	0B	0B	0B	B	B
0C	0C	0C	0C	C	C
0D	0D	0D	0D	D	D
0E	0E	0E	0E	E	E
0F	0F	0F	0F	F	F
10	10	10	10	10	10
11	11	11	11	11	11
12	12	12	12	12	12
13	13	13	13	13	13
14	14	14	14	14	14
15	15	15	15	15	15
16	16	16	16	16	16
17	17	17	17	17	17
18	18	18	18	18	18
19	19	19	19	19	19
1A	1A	1A	1A	1A	1A
1B	1B	1B	1B	1B	1B
1C	1C	1C	1C	1C	1C
1D	1D	1D	1D	1D	1D
1E	1E	1E	1E	1E	1E
1F	1F	1F	1F	1F	1F
20	20	20	20	20	20
21	21	21	21	21	21
22	22	22	22	22	22
23	23	23	23	23	23
24	24	24	24	24	24
25	25	25	25	25	25
26	26	26	26	26	26
27	27	27	27	27	27
28	28	28	28	28	28
29	29	29	29	29	29
2A	2A	2A	2A	2A	2A
2B	2B	2B	2B	2B	2B
2C	2C	2C	2C	2C	2C
2D	2D	2D	2D	2D	2D
2E	2E	2E	2E	2E	2E
2F	2F	2F	2F	2F	2F
30	30	30	30	30	30
31	31	31	31	31	31
32	32	32	32	32	32
33	33	33	33	33	33
34	34	34	34	34	34
35	35	35	35	35	35
36	36	36	36	36	36
37	37	37	37	37	37
38	38	38	38	38	38
39	39	39	39	39	39
3A	3A	3A	3A	3A	3A
3B	3B	3B	3B	3B	3B
3C	3C	3C	3C	3C	3C
3D	3D	3D	3D	3D	3D
3E	3E	3E	3E	3E	3E
3F	3F	3F	3F	3F	3F
40	40	40	40	40	40
41	41	41	41	41	41
42	42	42	42	42	42
43	43	43	43	43	43
44	44	44	44	44	44
45	45	45	45	45	45
46	46	46	46	46	46
47	47	47	47	47	47
48	48	48	48	48	48
49	49	49	49	49	49
4A	4A	4A	4A	4A	4A
4B	4B	4B	4B	4B	4B
4C	4C	4C	4C	4C	4C
4D	4D	4D	4D	4D	4D
4E	4E	4E	4E	4E	4E
4F	4F	4F	4F	4F	4F
50	50	50	50	50	50
51	51	51	51	51	51
52	52	52	52	52	52
53	53	53	53	53	53
54	54	54	54	54	54
55	55	55	55	55	55
56	56	56	56	56	56
57	57	57	57	57	57
58	58	58	58	58	58
59	59	59	59	59	59
5A	5A	5A	5A	5A	5A
5B	5B	5B	5B	5B	5B
5C	5C	5C	5C	5C	5C
5D	5D	5D	5D	5D	5D
5E	5E	5E	5E	5E	5E
5F	5F	5F	5F	5F	5F
60	60	60	60	60	60
61	61	61	61	61	61
62	62	62	62	62	62
63	63	63	63	63	63
64	64	64	64	64	64
65	65	65	65	65	65
66	66	66	66	66	66
67	67	67	67	67	67
68	68	68	68	68	68
69	69	69	69	69	69
6A	6A	6A	6A	6A	6A
6B	6B	6B	6B	6B	6B
6C	6C	6C	6C	6C	6C
6D	6D	6D	6D	6D	6D
6E	6E	6E	6E	6E	6E
6F	6F	6F	6F	6F	6F
70	70	70	70	70	70
71	71	71	71	71	71
72	72	72	72	72	72
73	73	73	73	73	73
74	74	74	74	74	74
75	75	75	75	75	75
76	76	76	76	76	76
77	77	77	77	77	77
78	78	78	78	78	78
79	79	79	79	79	79
7A	7A	7A	7A	7A	7A
7B	7B	7B	7B	7B	7B
7C	7C	7C	7C	7C	7C
7D	7D	7D	7D	7D	7D
7E	7E	7E	7E	7E	7E
7F	7F	7F	7F	7F	7F
80	80	80	80	80	80
81	81	81	81	81	81
82	82	82	82	82	82
83	83	83	83	83	83
84	84	84	84	84	84
85	85	85	85	85	85
86	86	86	86	86	86
87	87	87	87	87	87
88	88	88	88	88	88
89	89	89	89	89	89
8A	8A	8A	8A	8A	8A
8B	8B	8B	8B	8B	8B
8C	8C	8C	8C	8C	8C
8D	8D	8D	8D	8D	8D
8E	8E	8E	8E	8E	8E
8F	8F	8F	8F	8F	8F
90	90	90	90	90	90
91	91	91	91	91	91
92	92	92	92	92	92
93	93	93	93	93	93
94	94	94	94	94	94
95	95	95	95	95	95
96	96	96	96	96	96
97	97	97	97	97	97
98	98	98	98	98	98
99	99	99	99	99	99
9A	9A	9A	9A	9A	9A
9B	9B	9B	9B	9B	9B
9C	9C	9C	9C	9C	9C
9D	9D	9D	9D	9D	9D
9E	9E	9E	9E	9E	9E
9F	9F	9F	9F	9F	9F
A0	A0	A0	A0	A0	A0
A1	A1	A1	A1	A1	A1
A2	A2	A2	A2	A2	A2
A3	A3	A3	A3	A3	A3
A4	A4	A4	A4	A4	A4
A5	A5	A5	A5	A5	A5
A6	A6	A6	A6	A6	A6
A7	A7	A7	A7	A7	A7
A8	A8	A8	A8	A8	A8
A9	A9	A9	A9	A9	A9
AA	AA	AA	AA	AA	AA
AB	AB	AB	AB	AB	AB
AC	AC	AC	AC	AC	AC
AD	AD	AD	AD	AD	AD
AE	AE	AE	AE	AE	AE
AF	AF	AF	AF	AF	AF
B0	B0	B0	B0	B0	B0
B1	B1	B1	B1	B1	B1
B2	B2	B2	B2	B2	B2
B3	B3	B3	B3	B3	B3
B4	B4	B4	B4	B4	B4
B5	B5	B5	B5	B5	B5
B6	B6	B6	B6	B6	B6
B7	B7	B7	B7	B7	B7
B8	B8	B8	B8	B8	B8
B9	B9	B9	B9	B9	B9
BA	BA	BA	BA	BA	BA
BB	BB	BB	BB	BB	BB
BC	BC	BC	BC	BC	BC
BD	BD	BD	BD	BD	BD
BE	BE	BE	BE	BE	BE
BF	BF	BF	BF	BF	BF
C0	C0	C0	C0	C0	C0
C1	C1	C1	C1	C1	C1
C2	C2	C2	C2	C2	C2
C3	C3	C3	C3	C3	C3
C4	C4	C4	C4	C4	C4
C5	C5	C5	C5	C5	C5
C6	C6	C6	C6	C6	C6
C7	C7	C7	C7	C7	C7
C8	C8	C8	C8	C8	C8
C9	C9	C9	C9	C9	C9
CA	CA	CA	CA	CA	CA
CB	CB	CB	CB	CB	CB
CC	CC	CC	CC	CC	CC
CD	CD	CD	CD	CD	CD
CE	CE	CE	CE	CE	CE
CF	CF	CF	CF	CF	CF
D0	D0	D0	D0	D0	D0
D1	D1	D1	D1	D1	D1
D2	D2	D2	D2	D2	D2
D3	D3	D3	D3	D3	D3
D4	D4	D4	D4	D4	D4
D5	D5	D5	D5	D5	D5
D6	D6	D6	D6	D6	D6
D7	D7	D7	D7	D7	D7
D8	D8	D8	D8	D8	D8
D9	D9	D9	D9	D9	D9
DA	DA	DA	DA	DA	DA
DB	DB	DB	DB	DB	DB
DC	DC	DC	DC	DC	DC
DD	DD	DD	DD	DD	DD
DE	DE	DE	DE	DE	DE
DF	DF	DF	DF	DF	DF
E0	E0	E0	E0	E0	E0
E1	E1	E1	E1	E1	E1
E2	E2	E2	E2	E2	E2
E3	E3	E3	E3	E3	E3
E4	E4	E4	E4	E4	E4
E5	E5	E5	E5	E5	E5
E6	E6	E6	E6	E6	E6
E7	E7	E7	E7	E7	E7
E8	E8	E8	E8	E8	E8
E9	E9	E9	E9	E9	E9
EA	EA	EA	EA	EA	EA
EB	EB	EB	EB	EB	EB
EC	EC	EC	EC	EC	EC
ED	ED	ED	ED	ED	ED
EE	EE	EE	EE	EE	EE
EF	EF	EF	EF	EF	EF
F0	F0	F0	F0	F0	F0
F1	F1	F1	F1	F1	F1
F2	F2	F2	F2	F2	F2
F3	F3	F3	F3	F3	F3
F4	F4	F4	F4	F4	F4
F5	F5	F5	F5	F5	F5
F6	F6	F6	F6	F6	F6
F7	F7	F7	F7	F7	F7
F8	F8	F8	F8	F8	F8
F9	F9	F9	F9	F9	F9
FA	FA	FA	FA	FA	FA
FB	FB	FB	FB	FB	FB
FC	FC	FC	FC	FC	FC
FD	FD	FD	FD	FD	FD
FE	FE	FE	FE	FE	FE
FF	FF	FF	FF	FF	FF

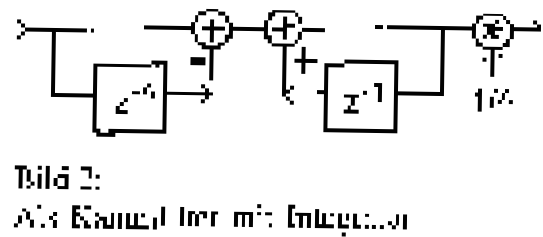
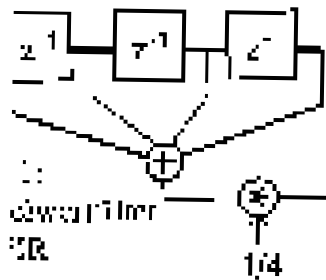
Tabelle 2: Addition und Skalierung

8 Bit

hex	hex	hex	hex	hex	hex
hex	hex	hex	hex	hex	hex
00	00	00	00	0	0
01	01	01	01	1	1
02	02	02	02	2	2

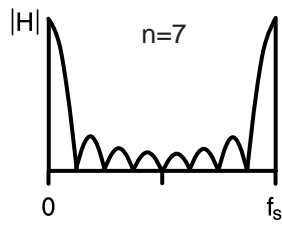
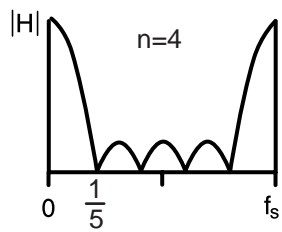
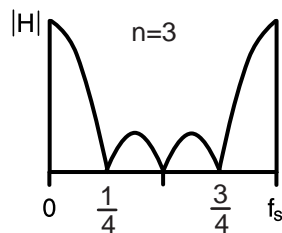
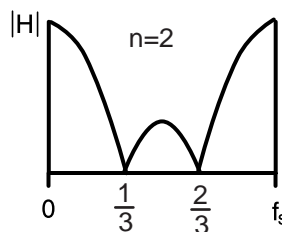
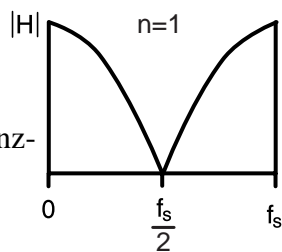
Mittelwert- filter

Daß man Mittelwertbildung („moving average“, „boxcar filter“) als FIR-Filter (Bild 1) darstellen kann ist wohlbekannt.



$$|H| = \frac{1}{n+1} \left| \frac{\sin(\pi(n+1) \frac{f}{f_s})}{\sin(\pi \frac{f}{f_s})} \right|$$

Bild 3: |H|
Formel
und
Frequenz-
gang



Fliest selbsterwartend man die Realisierung des Kanals mit aufgeschalteten Invertierern (z.B. „Cascaded integrator comb“) durch die Kanäle gehen dann können Wert und so den letzten Wert durch die Struktur. Das kann aber nicht verwendet für den Filter. Die Mittelwertbildung über viele Samples reduziert sich die Zahl der benötigten Additionen.

Die Mittelwertbildung ist ein „als Entfaltung hoher Frequenzen“ also bewirkt es in absteigend daß die Steuereingang eine nur „Tiefpaß“ ist, wie in Bild 5 für verschiedene Längen gezeigt.

Bild 4: Impulsantwort der Filter in Bild 3, 4

$$y_2 = 1.5625 \cdot \cos(0.785 \cdot x)$$

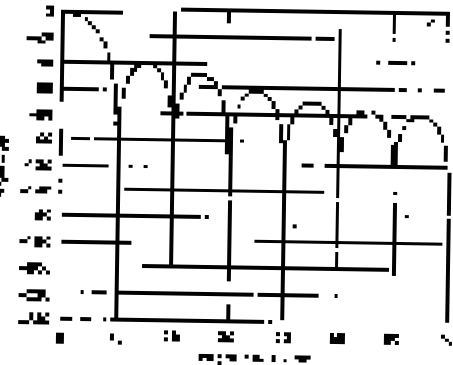
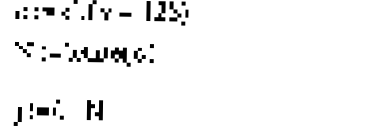
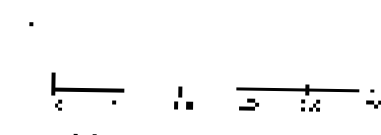
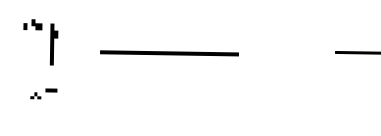


Bild 5: Filter des Delta-Sigma-Wandlers A/D 14

Maximaländerung innerhalb der die Übertragung durch den Filter. Die Leistung (AVR) ist für die 1. und 16. AVRE (AVR) ist für die 1. und 16. AVRE die Übertragungsfunktion ist (AVR).

Anwendung

Bei Delta-Sigma-Wandlern wird durch ein digitales Tiefpaßfilter die Bitstrom weiter in ein analoges Bandwert umgewandelt. Wegen der erwarteten Empfindlichkeit wird dafür bei einem 16-Bit-Wandler ein 120-Filter verwendet. Die Filterstruktur wird dabei verwendet, um die weiteren identischen Filter zu realisieren.

Durch Zuschalten eines Differenzierers (Bild 6) wird aus dem Tiefpaß ein „Trapezoidales Bandpaß“. Bezeichnet das Differenzier auf einem 64000/10000 mit 8 MHz Busfrequenz (50000, 50000) der 175 Bytes (Länge 16000/10000). Logarithmierung führt zu 16 bit. Für das Trapezoidales Bandpaß wird ein 16-Bit-Filter für eine nur mit 3 Stufen.

16. Vorratige B&G-Integration. Eine 16-Bit-Integration. Elektronik 1990/14/10

