

Eingegangen
★ 20 09 84 ★
Erledigt: 2467 St.



elektronik · bauelemente

Nur für den Dienstgebrauch

APPLIKATIVE INFORMATION

Dipl.-Ing. Angela Melde

VEB Robotron-Meßelektronik "Otto Schön" Dresden

Einsatz des U 8032 in einem Rechenwerk zur "Schnellen Fouriertransformation"

1. "Schnelle Fouriertransformation" und ihre Bedeutung in der Meßtechnik

Die Fouriertransformation ist ein wichtiges mathematisches Hilfsmittel in der Meßtechnik. Sie findet auf dem Gebiet der digitalen Signalverarbeitung Anwendung, wo sie zum Beispiel zur Analyse zeitlich veränderlicher Vorgänge genutzt wird, indem sie die Abbildung einer Zeitfunktion auf den Frequenzbereich realisiert.

Bei der "Schnellen Fouriertransformation" (Fast Fourier Transformation) FFT handelt es sich um spezielle Algorithmen zur Realisierung der diskreten Fouriertransformation. Der wesentliche Effekt dieser Algorithmen entsteht durch die geschickte Zerlegung der diskreten Fourierreihe unter Ausnutzung der Periodizitäten der Sinus- und Kosinusfunktion. Mit der FFT kann der mathematische Aufwand für in der Meßtechnik auftretende große Wertemengen beträchtlich reduziert werden. Zum Beispiel ergibt die FFT für einen in der Meßtechnik üblichen Wertebereich von $N = 1024$ Werten eine Reduktion auszuführender Multiplikationen gegenüber der herkömmlichen Fouriertransformation auf ca. 1%! Auch Operationen wie z. B. Faltung und Korrelation können mit der FFT zeitsparend als Multiplikationen im Frequenzbereich ausgeführt werden.

Bisher werden zur FFT vorrangig softwaremäßige Lösungen angeboten, die als Nachteil eine geringe Berechnungsgeschwindigkeit aufweisen. Grundlage für kurze Berechnungszeiten für die FFT ist eine Hardware-Realisierung mit Möglichkeit der schnellen 16-Bit-Multiplikation. Der U 8032 ist die Basis der im weiteren vorgestellten Anordnung zur FFT. Mit seiner Ausführungszeit der 16-Bit-Multiplikation beeinflusst er wesentlich die Geschwindigkeit der Anordnung.

2. Theoretische Grundlagen zur Schaltungsanordnung

2.1. Die diskrete Fouriertransformation

Ausgangspunkt für die FFT ist die diskrete Fouriertransformation nach Gleichung (1).

$$A_{\mu} = \frac{1}{N} \sum_{\nu=0}^{N-1} a_{\nu} e^{-j2\pi \frac{\nu\mu}{N}} \quad (1)$$

Gleichung (1) stellt die Beziehung zwischen den Abtastwerten der Zeitfunktion a_{ν} und den komplexen Fouriertransformierten A_{μ} dar. Zum besseren Verständnis für die weiteren Betrachtungen wird ein w -Faktor nach Gleichung (2) vereinbart, dessen Periodizität durch Gleichung (3) verdeutlicht werden soll.

$$w = e^{-j \frac{2\pi}{N}} \quad (2)$$

$$e^{\pm j\varphi} = \cos \varphi \pm j \sin \varphi \quad (3)$$

Aus den Gleichungen (1) und (2) ergibt sich für die diskrete Fouriertransformation eine Reihenentwicklung nach Gleichung (4), bei der durch Ausklammern von w -Faktoren Summen geschaffen werden, deren Summanden gleiche Potenzen von w enthalten.

$$A_{\mu} = \frac{1}{N} \sum_{\nu=0}^{N-1} a_{\nu} w^{\nu\mu} \quad (4)$$

Aus der beschriebenen Zerlegung der diskreten Fourierreihe entsteht unter Ausnutzung der Periodizität der w -Funktion der Cooley-Tukey-Algorithmus.

2.2. Der Cooley-Tukey-Algorithmus

In der Literatur wird eine Vielfalt von Algorithmen zur FFT vorgestellt, von denen für die gewählte Hardware-Realisierung der Cooley-Tukey-Algorithmus für Zweierpotenzen Verwendung findet. Im Bild 1 ist die Umsetzung der Gleichung (4) für $N = 8$ Werte unter Benutzung des Cooley-Tukey-Algorithmus dargestellt. Dieser Algorithmus wurde ausgewählt, weil er einen geringen Speicherplatzbedarf erfordert und aufgrund auftretender Periodizitäten organisatorisch leicht erfaßbar ist. Die N zu berechnenden Werte werden beim Cooley-Tukey-Algorithmus spaltenweise abgearbeitet, wobei die Anzahl der Spalten $S = \lg N$ beträgt, und die Ergebnisse jeder neu berechneten Spalte auf den Speicherplatz der vorhergehenden Spalte geschrieben werden ("in place"-Berechnung). Das ist von Bedeutung für den Speicherplatzbedarf bei dem in der Meßtechnik auftretenden großen Wertebereich. Die Eingangswerte müssen in bestimmter Reihenfolge für den Algorithmus eingelesen werden. Die N Werte einer Spalte werden über die Grundoperation, auch "Butterfly"-Operation genannt, verarbeitet. Sie tritt in jeder Spalte mit anderer Periodizität auf.

2.3. Die Grundoperation des Cooley-Tukey-Algorithmus

Die Grundoperation ist im Bild 2, aus dem auch die Bezeichnung "Butterfly"-Operation ersichtlich wird, noch einmal explizit dargestellt.

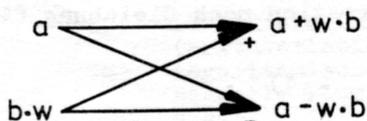


Bild 2: Grundoperation des Cooley-Tukey-Algorithmus

Da es sich bei den w -Faktoren um komplexe Größen handelt, sind a und b im allgemeinen Fall auch komplexe Größen.

Eine Grundfunktion führt demzufolge eine komplexe Multiplikation sowie je eine komplexe Addition und Subtraktion aus. Die Regeln für die komplexe Rechnung ergeben für eine Grundoperation vier reelle Multiplikationen und sechs reelle Additionen bzw. Subtraktionen. Die Ausführung dieser Berechnungen erfolgt im Rechenwerk der Schaltungsanordnung zur FFT, das aus vier U 8032 und einer schnellen ALU (Arithmetik-Logik-Einheit) aufgebaut ist.

2.4. Die Bedingungen für Echtzeitbetrieb

Die Zielstellung für die im weiteren beschriebene Anordnung zur FFT ist minimale Berechnungszeit für den beschriebenen Algorithmus. Voraussetzung dafür ist die schnelle 16-Bit-Multiplikation, die vom U 8032 ausgeführt wird. Die obere Grenzfrequenz für die FFT im Echtzeitbetrieb wird maßgeblich von den Befehlsausführungszeiten des U 8032 bestimmt. Für Echtzeitbetrieb muß folgende Bedingung erfüllt sein:

Während ein Block von N-Werten abgetastet und eingelesen wird, muß für den vorherigen Block von Eingangswerten die FFT berechnet werden. Die für die Berechnung von N-Werten benötigte Zeit muß kleiner sein als die für die Eingabe von N abgetasteten Werten benötigte Zeit. Deshalb bestimmt die Berechnungszeit die maximale Abtastrate und die sich nach dem Abtasttheorem daraus ergebende obere Grenzfrequenz für Echtzeitbetrieb.

3. Schaltungsanordnung zur FFT im Echtzeitbetrieb

3.1. Das Doppelspeicherprinzip

Zwei Speicherkomplexe in der Schaltungsanordnung zur FFT (vergl. Bild 5) dienen der Realisierung des Echtzeitbetriebes. Beide Speicherkomplexe sind über Register sowohl an die Eingangswerte als auch an das Rechenwerk angeschlossen und können abwechselnd als Eingabe- oder Rechenspeicher genutzt werden. Während in einen Speicherkomplex die Eingangswerte eingelesen werden, wird mit dem anderen Speicherkomplex die Berechnung des FFT-Algorithmus ausgeführt. Bild 3 veranschaulicht die Arbeitsweise beider Speicherkomplexe.

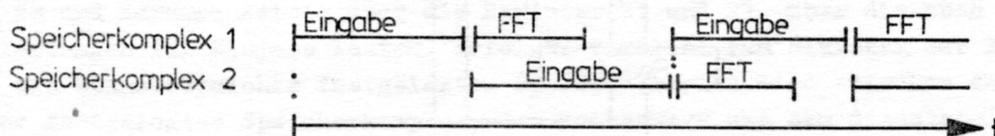


Bild 3: Doppelspeicherprinzip

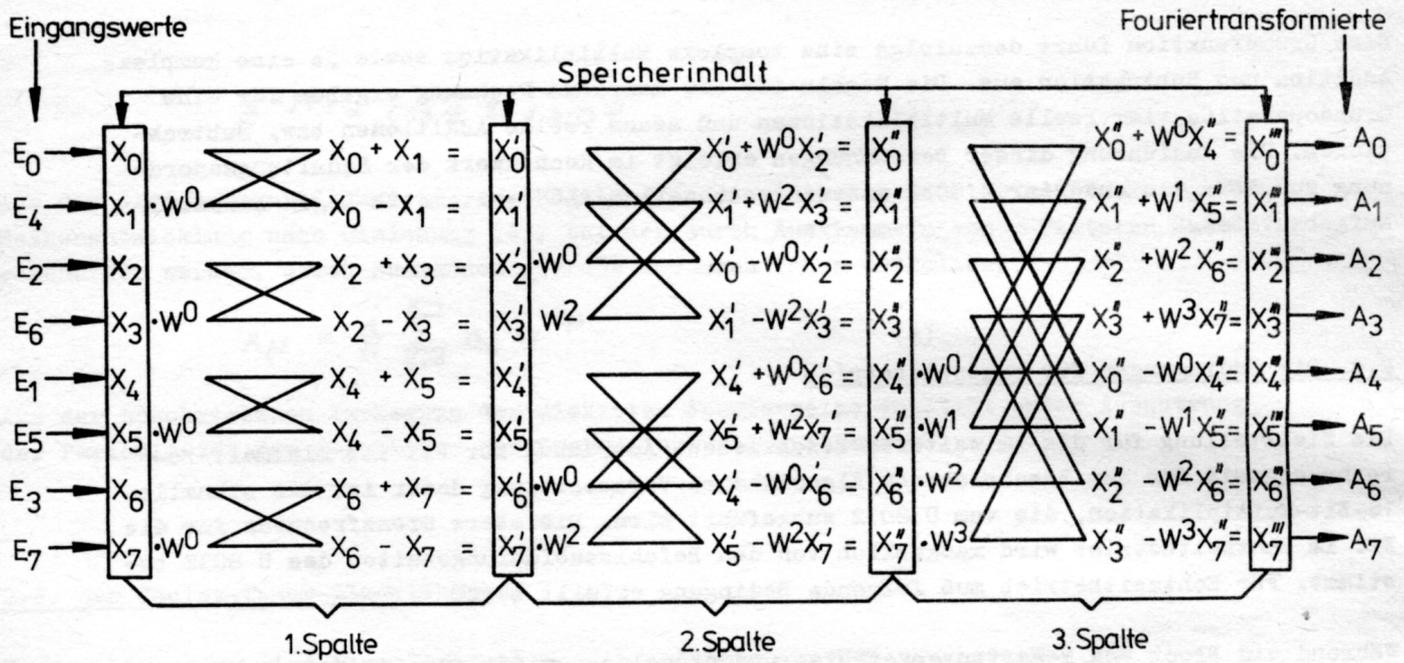


Bild 1: Cooley-Tukey-Algorithmus für N = 8 Werte

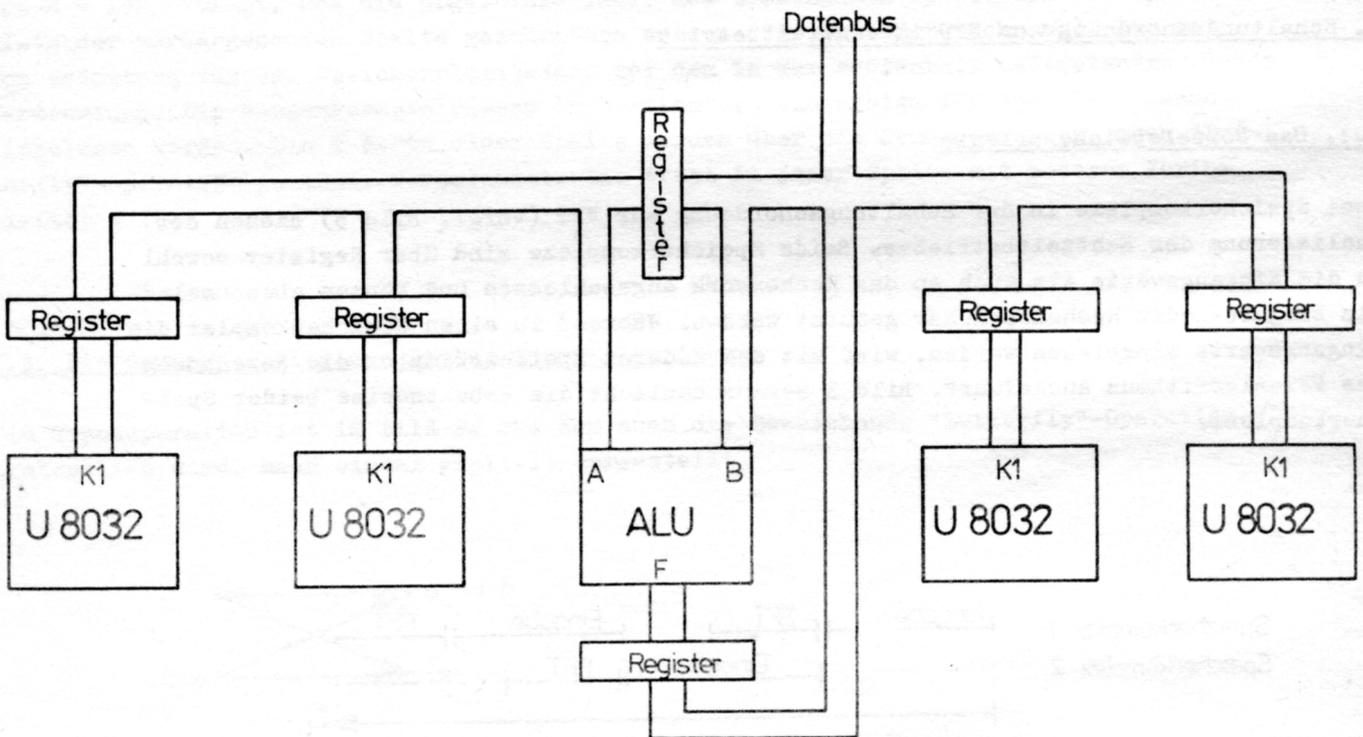


Bild 4: Prinzipieller Aufbau des Rechenwerkes mit U 832

3.2. Aufbau des Rechenwerkes mit U 8032

Den Aufbau des Rechenwerkes zeigt Bild 4. Da die Ausführungsgeschwindigkeit des U 8032 für die 16-Bit-Multiplikation nicht ausreicht, um mit einem Schaltkreis Echtzeitbetrieb zu realisieren, sind für das Rechenwerk vier U 8032 vorgesehen. Die vier U 8032 führen parallel je eine Multiplikation aus. Auf diese Art werden die vier reellen Multiplikationen, die für eine komplexe Multiplikation notwendig sind, in einem Zyklus realisiert. Der U 8032 wird ausschließlich für die komplexe Multiplikation mit dem w-Faktor genutzt, so daß sich ein einfacher Befehlsablauf ergibt. Für alle vier U 8032 kann gleichzeitig derselbe Befehlsaufruf erfolgen. Über die weitere Schaltungsanordnung werden an den Datenkanälen K1 der U 8032 die entsprechenden unterschiedlichen Daten bereitgestellt. Der Befehlsablauf für die in den U 8032 auszuführende Multiplikation hat folgendes Aussehen:

1. Einschreiben des 1. Operanden in RK1 und Transport in internes Register über den Befehl SAC;
2. Einschreiben des 2. Operanden in RK1 und Multiplikation mit dem den 1. Operanden beinhaltenden internen Register über den Befehl MUL;
3. Ausgabe des Ergebnisses über den Befehl LAG.

Der Funktionsablauf wird über die U 8032-Steuersignale /CS, /WE und /RQ gesteuert. Die Arbeitsweise erfolgt im 16-Bit-Festkomma Format. Die Additionen und Subtraktionen werden im Interesse einer minimalen Berechnungszeit in einer schnellen ALU ausgeführt, die eine Berechnung in einer Zeit kleiner 100 ns realisieren kann.

Die Register im Rechenwerk dienen der Zwischenspeicherung von Daten bzw. der Bustrennung. Die Ergebnisse der in den U 8032 ausgeführten Multiplikationen werden in Registern gespeichert und nacheinander der ALU zwecks Addition bzw. Subtraktion zugeführt. Anschließend werden die Ergebnisse in den Rechenspeicher eingelesen.

3.3. Schaltungsanordnung zur FFT

Bild 5 zeigt eine Schaltungskonzeption zur FFT. Ausgangspunkt sind digitale Eingangswerte, die zum Beispiel von einem ADU kommen und digitalisierte Abtastwerte einer Zeitfunktion darstellen. Die Eingangswerte werden in einen der beiden Speicherkomplexe eingeschrieben, wobei die Auswahl der Speicherkomplexe datenseitig über die Register R5 und R8 und adressenseitig über die Register R1 und R3, über die auch die Speichersteuerleitungen für Eingabe laufen, erfolgt. Parallel zum Einlesen der Eingangswerte in den als Eingabespeicher festgelegten Speicherkomplex wird zwischen dem als Rechenspeicher festgelegten Speicherkomplex, dem Rechenwerk und dem Sinustabellenspeicher die Berechnung des Algorithmus durchgeführt. Der Sinustabellenspeicher beinhaltet die zur Berechnung notwendigen w-Faktoren.

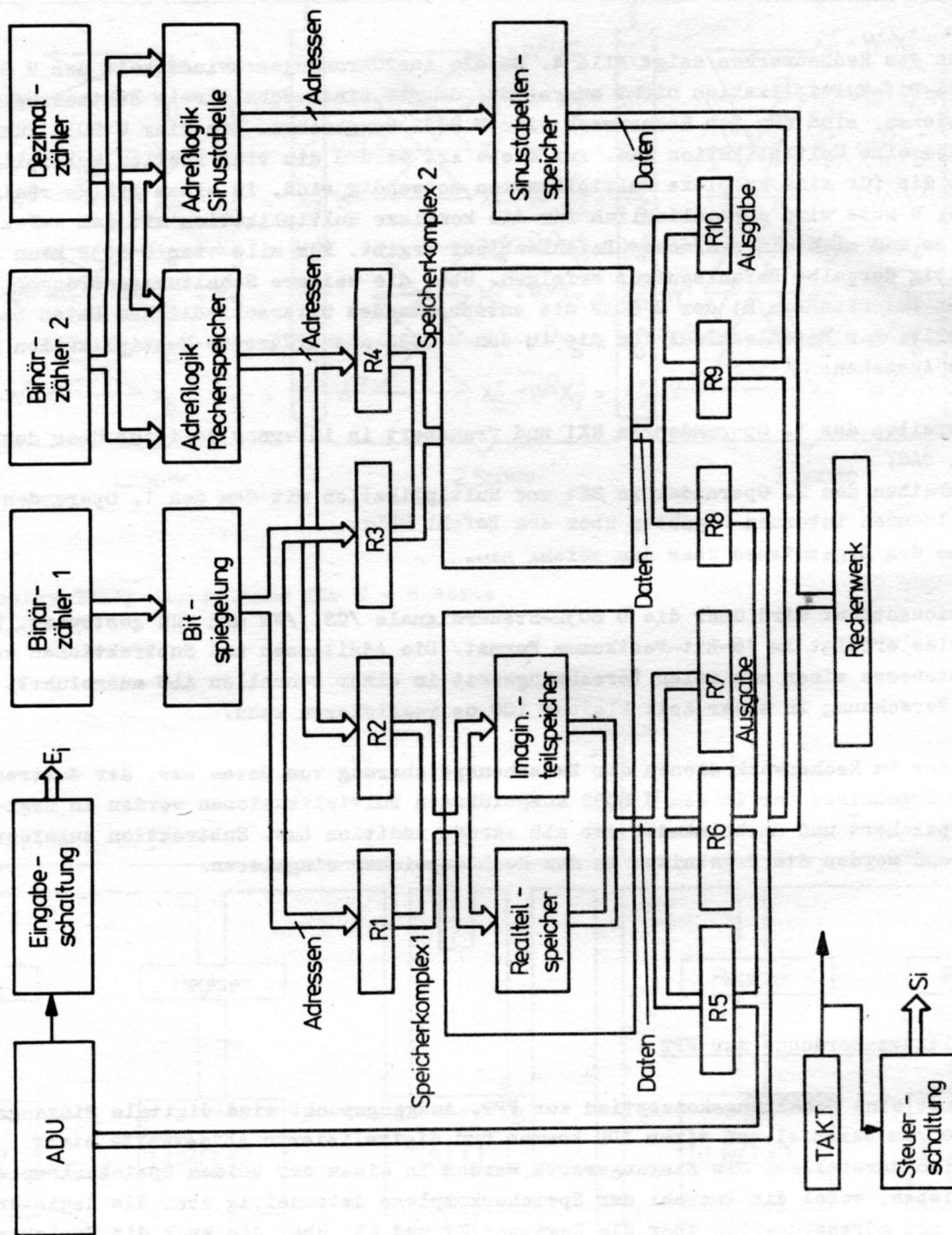


Bild 5: Anordnung zur Realisierung der FFT im Echtzeitbetrieb

Der Algorithmus wird in Spalten abgearbeitet, wobei die Ergebniswerte einer Spalte die Ausgangswerte für die folgende Spalte darstellen. Das heißt, daß die Ergebniswerte im Rechenspeicher auf den Platz der Ausgangswerte geschrieben werden. Da es sich um komplexe Berechnungen handelt, sind die Speicherkomplexe in Realteil- und Imaginärteil-speicher aufgeteilt. Der Adreßaufruf für den Eingabespeicher erfolgt über den Binärzähler 1, der einen Block von N-Eingangswerten zählt und dessen bitgespiegelter Zählerstand die Adresse des Eingangswertes bildet. Die Bitspiegelung ist vom Algorithmus bedingt und stellt hardwaremäßig nur eine Vertauschung der Adreßleitungen dar. Der Adreßaufruf für den Rechenspeicher und den Sinustabellenspeicher erfolgt über Adreßlogik, die ihrerseits von dem Binärzähler 2 und dem Dezimalzähler angesteuert wird. Der Binärzähler 2 zählt die N-Werte einer Spalte, und der Dezimalzähler zählt die Spalten $S = \text{ld}N$. Die Auswertung beider Zählerstände über Logikverknüpfungen ergibt die Adreßansteuerung, wobei für den Rechenspeicher und den Sinustabellenspeicher unterschiedliche Logik realisiert werden muß.

Die Steuersignale für den Eingabevorgang werden in einer Eingabeschaltung erzeugt. Die Steuersignale, die nicht die Eingabe betreffen, werden von einer Steuerschaltung erzeugt. Das umfaßt die Registeransteuerungen für den Berechnungszyklus (R2, R4, R6, R9) und den Ausgabezyklus (R7, R10), die entsprechenden Speichersteuersignale sowie die Steuersignale für das Rechenwerk. Die einfache Befehlsansteuerung für den U 8032 erfordert nur vier Steuersignale. Die Eingabe- und die Steuerschaltung basieren auf PROM-Signalspeichern, die von Zählern angesteuert werden. Der Grundtakt der gesamten Anordnung wird von einem Taktgenerator (10 MHz) geliefert.

4. Zusammenfassende Bemerkungen

Das vorgestellte Schaltungskonzept nach Bild 4 und Bild 5 ist eine vereinfachte Darstellung im Interesse von Übersichtlichkeit und Verständlichkeit. Es realisiert nur den mathematischen Algorithmus der FFT von der Verarbeitung von N-Abtastwerten bis zur Bereitstellung deren Fouriertransformierten.

Literatur:

- /1/ Achilles, D.: "Die Fourier-Transformation in der Signalverarbeitung", Springer-Verlag 1978
- /2/ Graf, G.: "Entwurf und Untersuchungen von Baugruppen zur Schnellen Fouriertransformation", Diplomarbeit HfV Dresden 1983

Dipl.-Ing. Werner Müller

VEB Robotron-Meßelektronik "Otto Schön" Dresden

Digitale Signalverarbeitung mittels U 8032

1. Umfang und Art der Signalverarbeitung

Für die meßtechnische Erfassung und Verarbeitung einmaliger und nichtperiodischer Vorgänge spielt eine leistungsfähige, mikroelektronisch realisierte Rechentechnik eine besondere Rolle. Entsprechend hohe Verarbeitungsgeschwindigkeiten erlauben einen sog. Echtzeitbetrieb, d. h. die rechentechnische Verarbeitung der Meßwerte erfolgt so schnell bzw. ist so organisiert, daß bei Gewährleistung eines kontinuierlichen Meßvorganges kein Meßwert verloren geht. Prinzipiell erfolgt die digitale Signalverarbeitung nach dem Schema gemäß Bild 1.

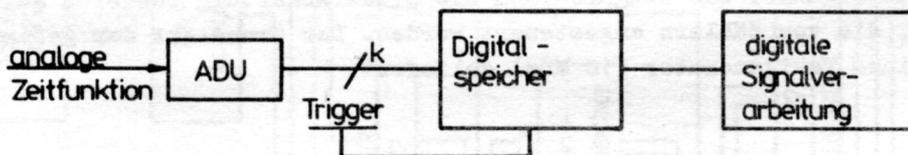


Bild 1: Schema der digitalen Signalverarbeitung

Die analoge Zeitfunktion wird durch den ADU in digitale Abtastwerte zerlegt, die im "Digitalspeicher" für die weitere Verarbeitung aufbewahrt werden. Der "Digitalspeicher" wird umlaufend beschrieben, bis ein Triggerereignis das Einschreiben beendet. Dabei kann der Abstand zwischen Triggerzeitpunkt und "Schreibstop" so gewählt werden, daß entsprechend viel Information vor dem Triggerzeitpunkt oder danach festgehalten wird.

Für die anschließende Signalverarbeitung kommen in Frage:

- a) zyklisches Auslesen des "Digitalspeichers" zur Auswertung einmaliger Vorgänge. Die Anordnung arbeitet als sog. Transientenspeicher.
- b) ständiges Auslesen des "Digitalspeichers" bei gleichzeitigem, zeitgeschichtetem Einschreiben. Dabei liegt die Ausleseadresse jeweils um m Adressen niedriger als die Einschreibadresse. Die Anordnung arbeitet als Verzögerungsleitung, deren Verzögerung durch m einstellbar ist.
- c) ein sich zyklisch wiederholender Vorgang wird zyklisch abgetastet und für jeden Abtastwert der arithmetische Mittelwert über 2^n Zyklen gebildet. Die Anordnung arbe-

tet als sog. "Averager", durch den der Abstand zwischen Nutz- und Störsignalpegel verbessert werden kann.

- d) Berechnung der Autokorrelationsfunktion (AKF)
- e) bei zweikanaliger Auslegung die Berechnung der Kreuzkorrelationsfunktion
- f) die "schnelle Fouriertransformation" (FFT)

Während die Anordnungen nach a) und b) ohne spezielle Verarbeitungseinheiten auskommen (sie benötigen nur eine entsprechende Speichersteuerung), werden für c) ein entsprechend schnelles Addierwerk und die Möglichkeit der Division, für d), e) und f) dazu noch eine entsprechend schnelle Multiplikation benötigt.

Alle diese Operationen sind im Arithmetikschaltkreis U 8032 implementiert, so daß er für die Lösung o. a. Aufgaben einsetzbar ist. Die im folgenden dargestellte Konzeption soll das verdeutlichen, aber auch gleichzeitig die Grenzen seines Einsatzes zeigen. Die FFT wird dabei nicht mitbetrachtet. Ihre Behandlung erfolgt in einem getrennten Beitrag dieser Serie.

2. Mikrobefehlsfolgen zur Realisierung der einzelnen Signalverarbeitungsarten

2.1. Averaging

Der ADU stellt Abtastwerte mit einer Breite von k Bit zur Verfügung. Sie sind mit ihren Werten auf den Bereich $-1 \leq x < 1$ normiert und werden über 2^n Zyklen aufsummiert. Negative Zahlen werden im Zweierkomplement bereitgestellt. Im ungünstigsten Fall kann ein $(k + n)$ Bit breites Ergebnis entstehen. Notwendige und gängige Werte von $k = 12$ und $n = 10$ ergeben eine Ergebnisbreite von 22 Bit.

Der U 8032 besitzt eine Verarbeitungsbreite von 15 Bit (ohne Vorzeichen). Die vorgegebene Aufgabe kann nur gelöst werden durch:

- a) Parallelarbeit von 2 Schaltkreisen U 8032. Damit ergibt sich eine Verarbeitungsbreite von 32 Bit.
- b) Aufteilung jeder Addition auf zwei serielle Teilschritte.

Hier soll nur die Lösung nach b) etwas näher betrachtet werden.

Angenommen der ADU liefert 12 Bit breite Abtastwerte folgender Form (Bild 2):

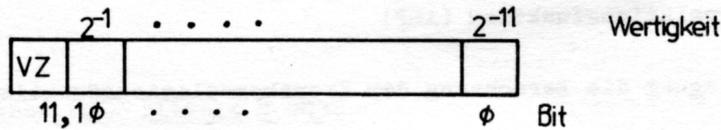


Bild 2: Darstellung der Abtastwerte

dann steht das gedankliche Komma zwischen Bit 11 und 1ϕ . Bei jeder Addition kann der Wert nach links wachsen. Das Komma bleibt dabei an der gleichen Stelle, aber das Vorzeichenbit rückt entsprechend nach links (Bild 3).

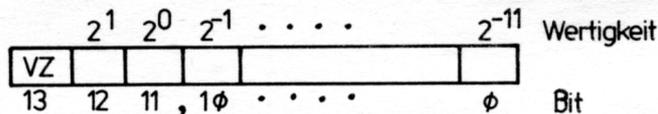


Bild 3: Verschiebung des Vorzeichenbits

Da das Ergebniswort breiter ist als eine Speicherzelle, muß es auf 2 Speicherzellen aufgeteilt werden, die Wertbits 15 ... ϕ z. B. auf Speicherzellen mit ungeraden Adressen und die Wertbits 16 ... 21 und das Vorzeichen auf die darauf folgenden geraden. Für eine vorzeichenrichtige Rechnung ist immer mit dem Wert des Vorzeichenbits auf das volle Wort aufzufüllen. Der jeweils aktuelle Abtastwert wird über den Kanal K1 direkt in den U 8032 übernommen. Die Ergebnisberechnung nutzt den Befehl ADC und läuft gem. Tabelle 1 ab.

Tabelle 1: Mikrobefehlsfolge für Averaging

Befehl	Operation	T/ μ s
SKS 1 1 1 1 SRR $\phi \phi \phi 1$ CLR, A ϕ DEIN R1 SAC, R1, A1	} Anfangsinitialisierung A ϕ mit "alles ϕ " laden ¹⁾ R1 mit "alles 1" laden A1 mit "alles 1" laden ¹⁾	Vorbereitung
SAC, R1, A3 /WE = ϕ	Einstroben des Speicherwertes (W1) aus ungerader Adresse und Schreiben in A3	2,5 (< 2)
SAC, R1, A4 /WE = ϕ	Einstroben des Speicherwertes (W2) aus gerader Adresse und Schreiben in A4	2,5 (< 2)

1) notwendig für das vorzeichenrichtige Auffüllen

(Fortsetzung Tabelle 1)

Befehl	Operation	T/us
$N \rightarrow$ SAC, R1, A5 /WE = \emptyset	Einstroben des aktuellen Abtastwertes (A) und Schreiben in A5. Das hierbei gebildete N-Flag wird abgespeichert zur Steuerung des späteren Auffüllens mit dem Vorzeichenbit	2,5 (<2)
ADD, A5, A3	A + W1, Ergebnis in A3	4 ($<2,5$)
$N = \emptyset$ oder $N \rightarrow$ ADC, A \emptyset , A4	"alles \emptyset " + W2 + Übertrag aus vorangegangener Addition, Ergebnis in A4	4 ($<2,5$)
$N = 1$ ADC, A1, A4	"alles 1" + W2 + Übertrag aus vorangegangener Addition, Ergebnis in A4	
LAG, A3, R1 /RQ = \emptyset	Transport (A3) nach R1 und Ausgabe von W1' zum Digitalspeicher	2,5 (<2)
LAG, A4, R1 /RQ = \emptyset	Transport (A4) nach R1 und Ausgabe von W2' zum Digitalspeicher	2,5 (<2)
	ΣT	20,5 (<15) =====

Die Mikroprogrammschritte der Vorbereitung laufen nur zu Beginn, z. B. beim Einschalten des Gerätes bzw. bei der Wahl der Signalverarbeitungsart, ab, die anschließenden bei jedem Abtastwert. Die dafür benötigte Zeit bestimmt den minimalen Abstand zwischen 2 Abtastwerten. Mit den für den Schaltkreis U 8032 gültigen Maximalwerten (siehe rechte Spalte des Mikroprogramms; Werte in Klammern sind typische Werte) ergibt sich dafür

$$t_{\min} = 20,5 \mu\text{s} \quad (< 15 \mu\text{s})$$

Über das Abtasttheorem

$$2,6 \cdot f_0 = \frac{1}{t_{\min}}$$

ergibt sich damit eine obere Grenzfrequenz $f_0 = 18,7 \text{ kHz}$ ($25,6 \text{ kHz}$) für den Echtzeitbetrieb. Die für die Mittelwertbildung noch notwendige Division durch 2^n wird mittels des Befehles DIV²⁾ ausgeführt.

2.2. Autokorrelation

Bei der Bestimmung von Werten der Autokorrelationsfunktion

$$\varphi(\tau) = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T f(t)f(t-\tau) dt$$

2) Da abweichend von den geforderten Bedingungen für den Befehl DIV die obersten 16 Bit des Zählers nicht Null sind, ist die dadurch gegebene Überlaufproblematik zu beachten.

sind die Produkte $f(t) f(t - \tau)$ zu bilden, zu summieren und über T zu mitteln. $f(t)$ ist dabei immer ein bestimmter Abtastwert, $f(t - \tau)$ der um k Abtastungen frühere. Bei der Summation bestehen die gleichen Probleme wie bei der des Averagings. Sie werden auch hier durch Aufteilung auf 2 serielle Teiladditionen unter Verwendung des Befehles ADC gelöst. Bei der Produktbildung ist zu beachten, daß das Ergebnis stets aus zwei Vorzeichen- und 30 Wertebits besteht. Die niederwertigen Bits (15 ... 0) stehen immer im A7, die höherwertigen (29 ... 16) und die beiden Vorzeichenbits (31, 30) im jeweiligen Zielregister (Bild 4).

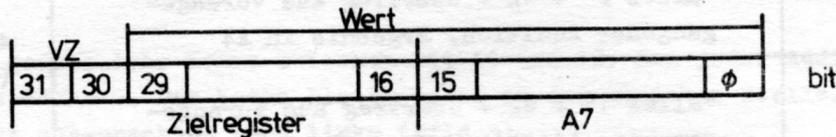


Bild 4: Darstellung des Multiplikationsergebnisses

Benutzt man nur den Wert des Zielregisters für die weitere Verarbeitung, dann tritt ein Fehler von $< 2^{-14}$ auf. Bei einer Breite der Abtastwerte von 12 Bit ist das Ergebnis aber immer noch um 2 Bit genauer und der angegebene Fehler zu vernachlässigen. Durch die 2 Vorzeichenbits steht jetzt das gedankliche Komma zwischen Bit 14 und 13. Das ist zu beachten, wenn mit den Ergebniswerten weiter gearbeitet wird, ohne daß vorher eine 1-Bit-Linksverschiebung (mittels Befehl VLV) durchgeführt wurde. Bei einer folgenden Aufsummierung "läuft das Ergebnis in die Vorzeichenbits hinein". Bit 14 stellt dann den Wert $\pm 2^0$ und Bit 15 den Wert $\pm 2^1$ dar. Über das folgende Mikroprogramm (Tabelle 2) wird jeweils ein Wert $\varphi(\tau)$ der Autokorrelationsfunktion berechnet.

Tabelle 2: Mikroprogramm für Autokorrelationsfunktion

Befehl	Operation	T/ μ s
SKS 1 1 1 1	} Anfangsinitialisierung	Vorbereitung
SRR 0 0 0 1		
CLR, A0		
DEIN R1		
SAC, R1, A1		
CLR, A3		
CLR, A4		
SAC, R1, A2	Einstroben des Abtastwertes $f(t - \tau)$	2,5
/WE = 0	aus Digital Speicher und Einschreiben in A2	(< 2)

1) notwendig für das vorzeichenrichtige Auffüllen

(Fortsetzung Tabelle 2)

Befehl	Operation	T/ μ s
$N \leftarrow$ MUL, R1, A2 /WE = \emptyset	Einstroben des aktuellen Abtastwertes $f(t)$ in R1, Multiplikation dieses Wertes mit (A2), Ergebnis steht im A2 (HW) und A7 (NW). Das hierbei gebildete N-Flag wird abgespeichert zur Steuerung des späteren Auffüllens mit dem Vorzeichenbit.	7 (< 5)
ADD, A2, A3	Addition des Multiplikationsergebnisses (HW) und niederwertigen Teiles der Aufsummation (A3). Ergebnis steht im A3.	4 ($< 2,5$)
$N = \emptyset$ ADC, A \emptyset , A4 oder $N = 1$ ADC, A1, A4	"alles \emptyset " + höherwertiger Teil der Aufsummation (A4) + Übertrag aus der vorangegangenen Addition. "alles 1" + höherwertiger Teil der Aufsummation (A4) + Übertrag aus der vorangegangenen Addition.	4 ($< 2,5$)
$\sum T_1$		17,5 (< 12)

Das Ergebnis der Aufsummation steht nur im U 8032 in den Registern A4 und A3 zur Verfügung (Bild 5) und wird nicht in den Digital Speicher geschrieben.

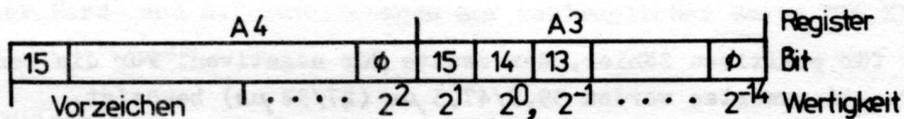


Bild 5: Darstellung der kumulativen Ergebnisse

Eine anschließende Division mittels des Befehles DIV²⁾ liefert den erforderlichen Mittelwert, der dann ausgegeben bzw. in einem Ergebnisspeicher abgelegt werden kann. Bei der Berechnung nur eines Wertes der Korrelationsfunktion ist damit der Vorgang abgeschlossen. Soll die gesamte Funktion berechnet werden, ist die Berechnung mit entsprechend vielen, unterschiedlichen γ -Werten zu wiederholen. Für die Mittelwertbildung und Ausgabe ergibt sich das folgende Mikroprogramm (Tabelle 3).

Tabelle 3: Mikrobefehlsfolge für Mittelwertbildung

Befehl	Operation	T/ μ s
N TST, A4 TST, A3	Vorzeichen- und Zerotest (A4) → Z1	2,5
	Vorzeichen- und Zerotest (A3) → Z2 wenn Z1 = Z2 = \emptyset , dann Abbruch des Programms, da Ergebnis = \emptyset (da Zähler = \emptyset)	(< 2)
N = \emptyset oder N = 1	ZKO, R1 /WE = \emptyset	4 (< 2,5)
	EKO, A4 ZKO, A3 ADC, A \emptyset , A4	12 (< 7,5)
DIV, A4, A3	Division (A4)/(A3) durch (RX), Ergebnis steht in A4	30 (< 20)
LAC, A4, R1	Transport (A4) nach R1	2,5 (< 2)
DAUS (/RQ = \emptyset)	Ausgabe (R1) auf Kanal K1	\approx 0,5
		$\sum T_2$ 39,5/47,5 (27/32) =====

Der erste Wert gilt für positiven Zähler, der zweite für negativen. Für die Berechnung und Ausgabe eines Funktionswertes werden 39,5/47,5 μ s (27/32 μ s) benötigt.