

# RISC+DDS+PLL: Genaue Frequenzen für jedes Budget

Dr.-Ing. MARTIN OSSMANN

Zur Erzeugung genauer Frequenzen kann man sich eines Quarzoszillators bedienen. Durch Frequenzteilung oder mit einem DDS lassen sich dann nahezu beliebige Frequenzen ableiten. Bei geringen Ansprüchen kann ein solches Konzept mit einem schnellen Prozessor realisiert werden. Will man genauer werden (nicht jeder hat das Geld für ein Cäsium- oder Rubidium-Normal), bleibt die phasenstarre Ankopplung an einen (Langwellen-)Normalfrequenzsender.

Die dazu notwendige PLL und der Frequenzteiler können auch per Mikroprozessor realisiert werden. So gelangt man zu einem sehr einfachen, hochgenauen Frequenznormal. Die gleichen Schaltungen kann man auch einsetzen, um die auf Langwelle des öfteren verwendete Phasenmodulation zu demodulieren, wie der Abschluß des Beitrags zeigt.

Die vorgestellten Schaltungen stellen keine perfekte Bauanleitung dar, sie sind eher als Anleitung zum eigenen Experimentieren gedacht. Auch was die spektrale Reinheit und derartige Forderungen angeht, präsentieren wir hier eine Low-End Lösung. Wen dieses Thema interessiert, der sei auf [1] verwiesen.

## ■ AVR-RISC-Prozessor als Schmalspur-DDS-Oszillator

Wie ein Direct Digital frequency Synthesizer (DDS) funktioniert, ist schon mehrfach im FUNKAMATEUR [2] behandelt worden. In Bild 1 ist das Ganze als Blockschaltbild dargestellt. Komplette DDS-Schaltungen gibt es zu kaufen

Gemeinhin verwendet man einen DDS-Baustein, um eine Sinusfunktion zu erzeugen. Will man ein Rechteck-Signal erzeugen, kann man die Sinustabelle und den A/D-Wandler weglassen. Übrig bleibt dann eigentlich nur noch ein N-Bit-Akkumulator und Summierer und der Taktgenerator.

Begnügt man sich damit, relativ niedrige Frequenzen zu erzeugen, kann man die Additionsfunktion einfach mit einem Mikrocontroller (egal welchen Typs) realisieren. So entsteht dann ein „Schmalspur-DDS“ –

Je schneller der Mikrocontroller, um so höhere Frequenzen kann man erzeugen

Das DDS-Synthesepinzip führt dann dazu, daß man Frequenzen mit sehr hoher Auflösung erzeugen kann, auch solche, die sich nicht durch einfache Teilung aus der Taktfrequenz ableiten. Man muß aber einen Jitter in Kauf nehmen, der auch als Phasenrauschen aufgefaßt werden kann.

Was kann man nun praktisch erreichen, wenn man einen preiswerten (ca. 5 bis 10 DM), schnellen Prozessor wie den ATMEL AT90S1200 [3],[4] benutzt? Derzeit ist die 12-MHz-Version erhältlich, eine Zeitlang gab es auch 16-MHz-Versionen, und als Ziel nennt man bei ATMEL 24 MHz.

Bild 2 stellt eine Schaltung für erste Tests dar. Als Taktgenerator findet ein Quarzoszillator Verwendung. Die Pins PB5 bis PB7 sind freigehalten und dienen der (seriellen) (Um)Programmierung des Controllers in der Schaltung, was den Programmtest erleichtert. Listing 1 zeigt ein kurzes Programm, das einen N = 32 Bit DDS realisiert.

Nach der Initialisierung sorgen 4 Befehle für die 32 Bit Summation. Zwei Befehle sorgen dafür, daß das höchstwertigste Bit dieser Summe am Port-Pin PD6 zur Verfügung steht. Ein Rücksprung ist auch noch

nötig (2 Zyklen), so daß ein kompletter DDS-Zyklus 9 Prozessor-Zyklen benötigt. Damit wird dieser DDS durch die folgenden Werte charakterisiert:

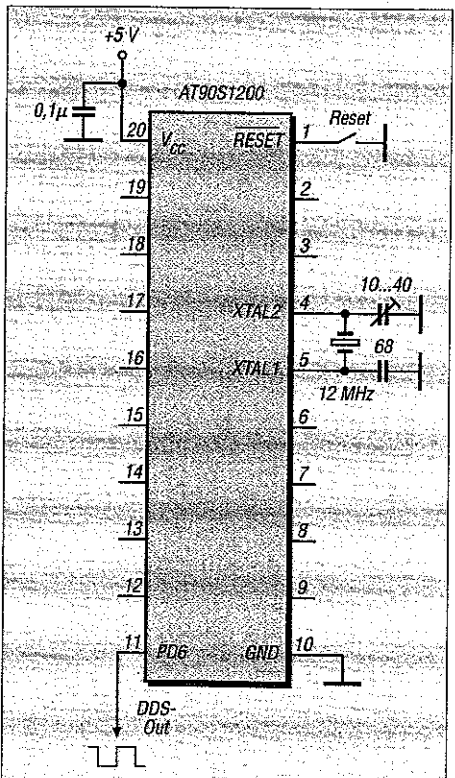
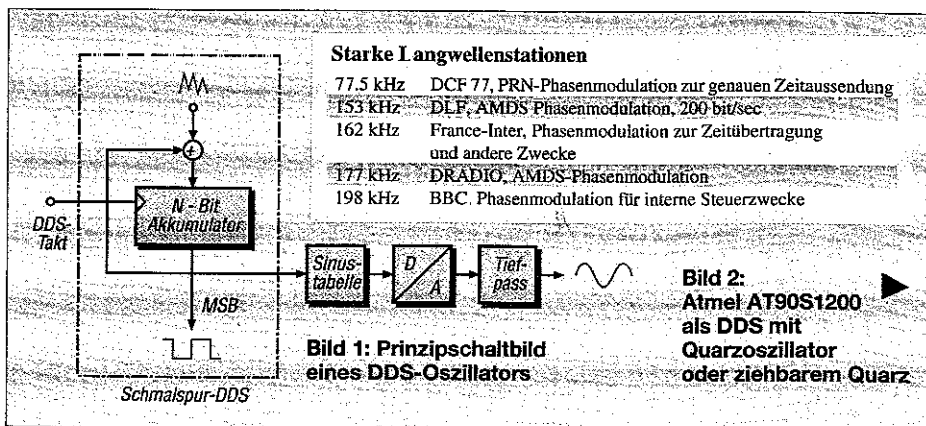
Die Taktfrequenz des Prozessors ist  $F_p = 12$  MHz. Die Taktfrequenz des DDS ist  $f_{DDS} = 12/9$  MHz = 1,333 MHz. Die Bitbreite des Summierers ist  $N = 32$ , und die Auflösung der zu erzeugenden Frequenz ist  $f_{DDS}/2^N = 0,3$  mHz. Der DDS-Oszillator erzeugt die Frequenz  $f_{DDS} \cdot m/2^{32}$ , wobei m der Wert ist, der bei jedem DDS-Zyklus zum Akkumulator addiert wird. Um eine Frequenz von  $f = 77,5$  kHz zu erzeugen, ist  $m = \text{round}(77500 \text{ kHz}/f_{DDS} \cdot 2^{32}) = 249644974 = 0EE147AE_{\text{hex}}$  zu wählen.

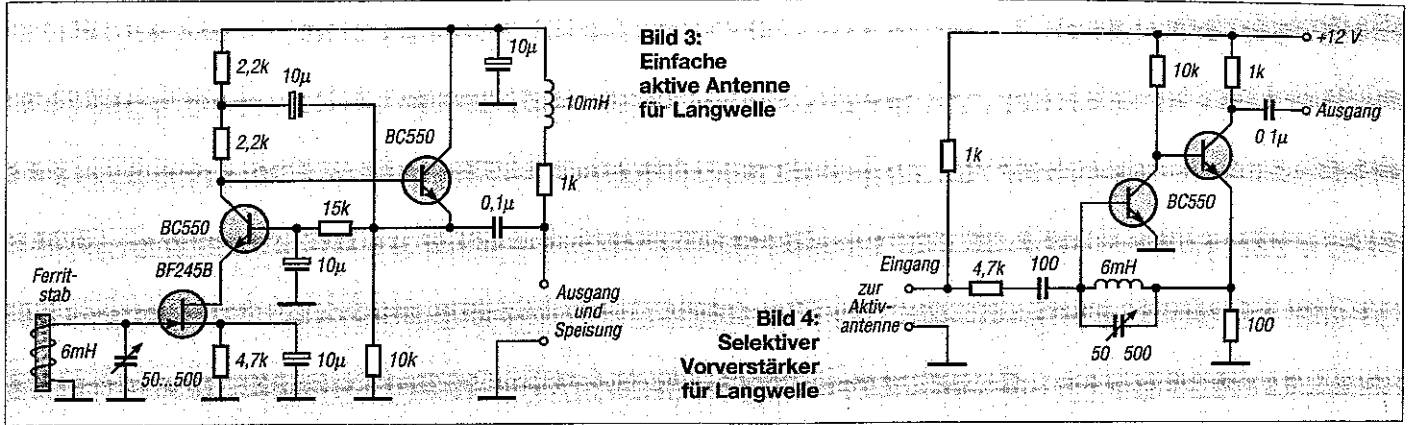
Anstatt zu addieren, wird bei unserem Programm subtrahiert, das fällt dem Prozessor etwas leichter. Der Oszillator läuft dadurch „rückwärts“, was dem ganzen Prinzip aber keinen Abbruch tut.

Wenn wir nun also unseren Schmalspur-DDS aufgebaut haben, messen wir natürlich zuerst seine Frequenz mit einem Zähler nach. Wie kann man nun eine genaue Kontrolle durchführen? Als Einfachstes bietet es sich an, als Referenz das Signal eines starken Langwellensenders zu benutzen, der als Normalfrequenzsender arbeitet, also z.B. den Zeitzeichensender DCF77 auf 77,5 kHz.

## ■ Langwellenempfang

Der Verfasser benutzt zum Langwellenempfang eine aktive Antenne nach Bild 3 mit Ferritstab und Drehkondensator sowie einen selektiven Vorverstärker nach Bild 4. Mit einem Zweikanaloszilloskop kann man leicht kontrollieren, ob der DDS und DCF77





auf der gleichen Frequenz liegen, indem man den Phasenversatz beobachtet. Mit einem trimmbaren Oszillator nach Bild 2 kann man dann ggf die Quarzfrequenz so ziehen, daß die Schwebungsfrequenz möglichst klein wird. Damit hat man dann die Oszillatorfrequenz unseres Prozessors auf 12 MHz eingestellt, und zwar ziemlich genau, und kann nun darangehen, andere Frequenzen mit dem DDS zu erzeugen, deren Genauigkeit dann im wesentlichen nur noch von der Genauigkeit des eben beschriebenen Abgleichs abhängt.

■ Digitaler Mischer mit Tiefpaß

Was sollen nun aber Leute ohne Zweikanal-oszilloskop tun? Nun, auch hier kann mit Software geholfen werden. Jeder kennt noch die Methode, eine Frequenz mit einer bekannten Frequenz zu vergleichen, indem man „auf Schwebungsnul“ abgleicht. Die beiden Frequenzen werden gemischt (siehe Bild 5) und das niederfrequente Mischprodukt mit einem Tiefpaß gefiltert und hörbar gemacht.

Bei unserer Anwendung präziser Frequenzen geht es nun eigentlich um die letzten zehntel Hertz. Das Ohr ist da als Indikator nicht mehr so geeignet. Der Schwebungsindikator nach Bild 5 wird daher nun ähnlich brutal „digitalisiert“ wie der DDS-Oszillator. Das Signal der Referenz (z.B. das empfangene DCF77-Signal) wird mit einem Komparator „digitalisiert“. Unser DDS arbeitet sozusagen als lokaler Oszillator (LO).

Er liefert sowieso nur Nullen und Einsen, die Multiplikation (Mischung) kann also mit einer XOR-Verknüpfung durchgeführt werden. Und zur Tiefpaßfilterung nehmen wir einen Zähler. Dieser wird jeweils 127 DDS-Zyklen lang erhöht, sofern das XOR am Ausgang eine 1 liefert. Danach (also nach 127 Zyklen) wird sein Inhalt an einen D/A-Wandler geliefert (und dort gespeichert). Der Zähler wird dann zurückgesetzt.

Der Inhalt dieses Zählers gibt nach den 127 Zyklen also genau an, wie oft die Referenz und der DDS gleichen Pegel hatten. Wir erhalten so einen digitalen Phasenkomparator,

der mit der Frequenz  $f_{DDS}/127$  Werte zwischen 0 und 127 liefert, die dem Phasenversatz entsprechen. Das Ganze realisieren wir wieder in Software. Das dazugehörige Programm ist ganze 20 Befehle lang. Um das Referenzsignal zu digitalisieren, verwenden wir den im ATMEL-Prozessor enthaltenen Analogkomparator. Als einfachen 7-Bit-D/A-Wandler zur Ausgabe der Phaseninformation verwenden wir einen R-2R-Kettenleiter.

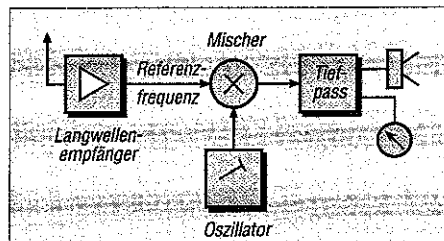


Bild 5: Schwebungsindikator

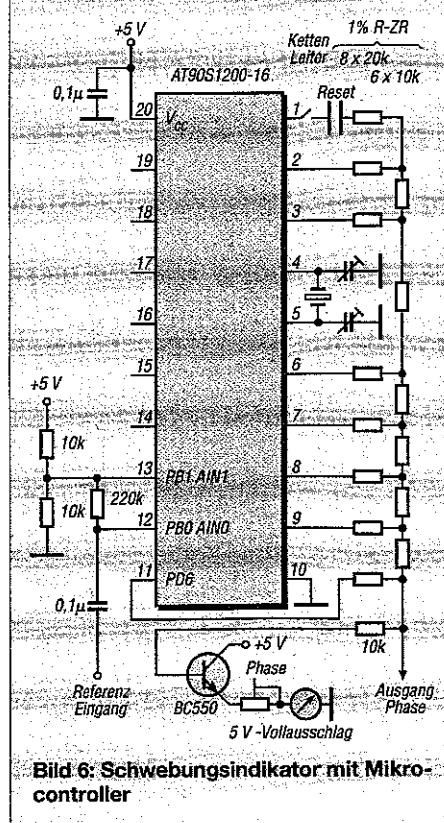


Bild 6: Schwebungsindikator mit Mikrocontroller

Die vollständige Schaltung dieses DDS mit Mischer und Phasenkomparator zeigt Bild 6. Die Phaseninformation wird mit einem (nicht zu trägen) Zeigerinstrument angezeigt oder auf einem Oszilloskop verfolgt.

Nun kann man den Abgleich mit der Referenzfrequenz gut durchführen und auf dem Zeigerinstrument die Schwebung kontrollieren. Der Controller muß mit 12 MHz getaktet werden, als Referenzfrequenz wird das Signal von France-Inter auf 162 kHz verwendet. Bei der Programmierung war die Hauptschwierigkeit, den Ablauf mit möglichst wenig Befehlen zu realisieren, damit der DDS wieder mit einer möglichst hohen Frequenz läuft.

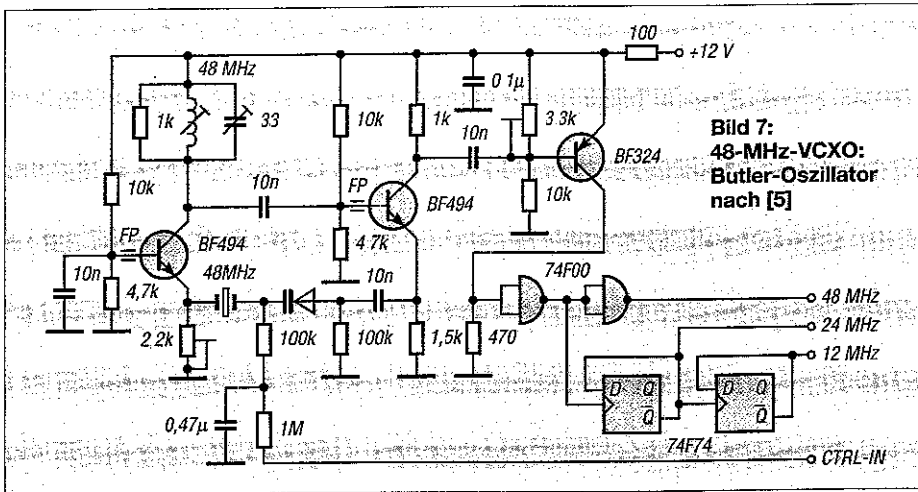
Ein kompletter DDS-Zyklus dauert nun 12 Prozessor-Zyklen. Immer, wenn die Phaseninformation ausgewertet wird (also nach 127 DDS-Takten), wird übrigens keine Multiplikation des Eingangssignals mit dem DDS-Signal vorgenommen, um in diesem Fall Zeit zu sparen. Auch das genaue Auszählen der Taktzyklen ist bei dieser Art von Programmen essentiell.

■ Der Kreis schließt sich: PLL, die erste

Der Handabgleich auf die Referenzfrequenz ist natürlich nicht der Weisheit letzter Schluß. Wie wärs, wenn wir die Schaltung um einen VCO oder VCXO erweitern, der durch die Phaseninformation in seiner Frequenz so nachgeregelt wird, daß automatisch die Referenzfrequenz phasenstarr getroffen wird?

Nun, das geht natürlich auch, und so entsteht dann eine PLL. Der Verfasser hat einen 48-MHz-VCXO nach Bild 7 (aus [5]) aufgebaut (man lese in der entsprechenden Literatur nach, wie man einen Butler-Oszillator in Betrieb nimmt, ein nicht ganz einfaches Verfahren. Wer einen besseren Oszillator für diese Anwendung kennt, teile mir dies bitte mit).

Der VCXO wird so abgeglichen, daß bei 2,5 V Steuerspannung die Ausgangsfrequenz möglichst genau 48 MHz ist. Dazu wird Schalter S in Position I gebracht und auf



**Bild 7:**  
48-MHz-VCXO:  
Butler-Oszillator  
nach [5]

Schwabungsnull mit Hilfe von Instrument II abgeglichen. Dann bringt man Schalter S in Position II, und die PLL regelt automatisch den VCXO so nach, daß die DDS-Ausgangsfrequenz phasenstarr mit dem Langwellenreferenzsignal gekoppelt bleibt

Nun haben wir also ein 48-MHz- und ein 12-MHz-Signal, die schon die Genauigkeit des Langwellensenders haben, das dürfte für viele praktische Zwecke ausreichend sein. Die 48 MHz kann man gut verwenden, um z.B. den DDS-Baustein AD7008 mit einem Takt zu versorgen. So kann man das hochqualitative Sinussignal mit hochgenauer Frequenz erzeugen.

Auch einem 96-MHz-VCXO (oder 100 MHz) steht prinzipiell nichts im Wege. Der Verfasser verwendet diese Kombination nach Bild 8, um die Frequenzen der diversen Langwellensender zu beobachten.

Schon eine leichte Frequenzdifferenz von 0,1 Hz kann man gut beobachten. Auf diese Weise kann man auch das Driftverhalten von selbstgebauten Oszillatoren untersuchen. Schaut man sich übrigens die Regelspannung des VCXO an, wenn er auf France-Inter eingerastet ist, sieht man schön die Phasemodulation, dazu unten mehr

**■ PLL, die zweite**

Nun ist es nicht jedermanns Sache, einen 48-MHz-VCXO zu bauen, und nicht jeder braucht diese High-End Lösung. Wie wär's,

wenn man die PLL-Funktion vollständig mit im Prozessor realisiert, und dazu eventuell auch noch einen zweiten Schmalband-DDS, um neben der Referenzfrequenz noch eine Ausgangsfrequenz zu erzeugen. Das Ganze geht tatsächlich.

Im Prinzip muß man dazu in der Lage sein, die Taktrate unseres DDS-Oszillators zu ändern. Da wir eine quarzgenaue Frequenz an eine andere hochgenaue Referenz ankoppeln wollen, reichen sehr kleine Änderungen schon aus, immerhin stimmt unser Quarz schon um ca. 50 ppm mit der Sollfrequenz überein.

Eine einfache Realisierung zur Nachregelung ist in Listing 3 angewandt. Ein normaler, kompletter DDS-Durchlauf (jetzt zwei DDS-Oszillatoren!) dauert nun 17 Zyklen. Es wird wieder 127 DDS-Zyklen lang die Phase zwischen lokalem DDS und der Referenz (wieder am Analogkomparator) bestimmt.

Danach wird die Phase ausgewertet und ein DDS-Zyklus eingeschoben, der in Abhängigkeit der bestimmten Phase 16 oder 18 Zyklen dauert. Die mittlere Zyklenzahl paßt sich dadurch automatisch so an, daß das Signal des lokalen DDS gleich der Phase der Referenz wird. Während jedes DDS-Zyklus wird der Ausgangs-DDS mit neu berechnet, er liefert die Ausgangsfrequenz. Damit kann man mit der Schaltung nach Bild 9 (Achtung, 16-MHz-Quarz) Frequenzen erzeugen, die von 0 bis ca. 200 kHz so

genau sind, wie die Referenz (plus Ungenauigkeit des DDS, d.h., 32 Bit). Ersetzt man den Ausgangs-DDS durch die Befehle

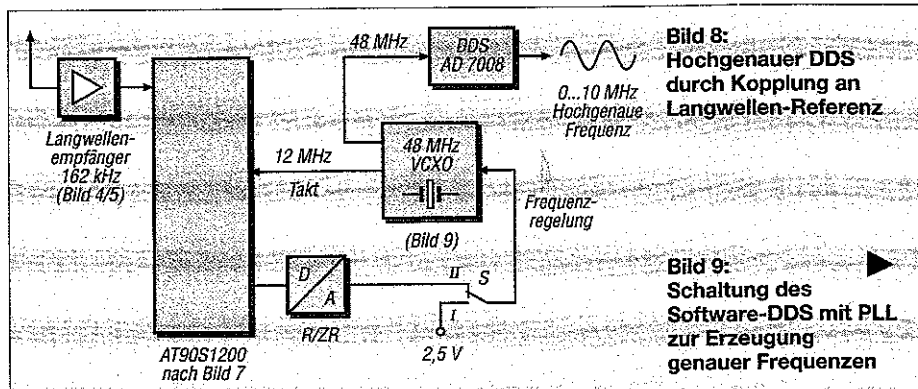
```
inc outreg ;outreg bildet oszillator
out PORTD,outreg ;an port d ausgeben
nop ;auf 4 Zyklen laenge bringen
nop
```

ändert sich die Zykluszahl auf 16, die DDS-Rate gleicht sich also auf genau 1 MHz ab. So hat man auf einfache Weise eine hochstabile 1-MHz-Referenz (an Port Pin PD0). Der Verfasser hat festgestellt, daß auch die 12-MHz-ATMEL-Prozessoren bei 5 V Versorgungsspannung noch gut bei 16 MHz in diesen Anwendungen funktionieren.

**■ Phasendemodulation**

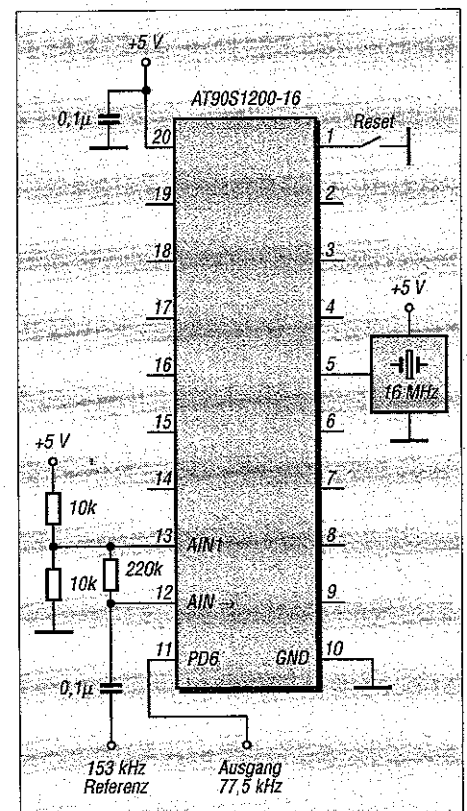
Bisher haben wir gesehen, auf welcher einfachen Weise man eine PLL per Software realisieren kann, um einen Oszillator auf einen anderen zu synchronisieren. Viele Langwellensender senden heutzutage zusätzlich zur Amplitudenmodulation noch phasenmodulierte Signale aus (vgl. Tabelle). Diese Phasemodulation kann man mit den hier vorgestellten Konzepten demodulieren. Entscheidend ist dabei, daß die Regelgeschwindigkeit der PLL so klein ist, daß die Phasemodulation nicht ausgegletzt wird.

Dazu wurde die Software abgewandelt. Die Demodulations-PLL arbeitet wie folgt: Sie schaltet die DDS-Frequenz des Lokalen Oszillators in Abhängigkeit der berechneten Phasendifferenz um, und zwar zwischen zwei Werten, die ein wenig ober- bzw. unterhalb



**Bild 8:**  
Hochgenauer DDS  
durch Kopplung an  
Langwellen-Referenz

**Bild 9:**  
Schaltung des  
Software-DDS mit PLL  
zur Erzeugung  
genauer Frequenzen



**Listing 1: 32-Bit-DDS für 77,5 kHz**  
**XAFU1.ASM Erzeuge 77,5 kHz bei 12 MHz Takt**

```

device at90s1200
include „1200def.inc“
def outreg =r16 ;Hilfs-Ausgaberegister
def tmp =r17 ;Temporaerer Speicherplatz
def dds0 mr18 ;DDS Akkumulator LSB
def dds1 =r19
def dds2 =r20
def dds3 =r21 ;DDS Akkumulator MSB

RESET: ldi tmp,$FF
out DDRD,tmp ;deklariere D als Ausgabeport
DDS: subi dds0,$ae ;Summand OEE147AE ergibt 77 5 kHz
sbc dds1,$47 ;Subtrahieren geht immediate
sbc dds2,$e1 ;32 Bit DDS
sbc dds3,$0e
mov outreg,dds3 ;hole MSB vom DDS
lsr outreg ;schiebe Bit 7 an Position 6
out PORTD,outreg ;DDS MSB an Port ausgeben
rjmp DDS ;Ein DDS Zyklus = 9 Prozessortakte
    
```

**Listing 2: Schwebungsindikator, Referenz: 162 kHz France-Inter**  
**XAFU2.ASM, DDS und Mischer für 162 kHz bei 12 MHz Takt**

```

device at90s1200
include „1200def.inc“
def tmp mr16
def cntr1 =r17 ;Tiefpaß-Zaehler
def phase =r18 ;Phasenwert-Zaehler
def dd80 =r26 ;32 Bit DDS in 4 Bytes
def dds1 =r27
def dds2 =r28
def dds3 =r29 ;hoechstwertiges Byte

RESET: ldi tmp,$FC ;setze Ports auf Ausgabe
out DDRD,tmp
ldi tmp,$7F
out DDRD,tmp ;Mischer Ausgang ist Port D

SAMPLE: mov tmp,dds3 ;Mischer-Bit in tmp Bit 7
sbc ACSR,ACO
com tmp ;wenn Comparator dann invert.Bit
sbrnc tmp,7 ;Wenn Bit dann erhoehe Phasenwert
inc phase

noSAMPLE: subi dds0,$fe ;32 Bit DDS 2978D4FE -- 162 kHz
sbc dds1,$d4 ;1 MHz DDS rate
sbc dds2,$78
sbc dds3,$29
subi cntr1,1 ;Tiefpass-Zaehler
brne SAMPLE ;Teste ob Phasenvergleich zuende
clr cntr1 ;Ruecksetzen Zaehler
isr phase ;Phasenwert nach rechts shiften
out PORTD,phase ;an 7-Bit D/A Wandler ausgeben
clr phase ;Phasenwert loeschen
rjmp noSAMPLE ;ein DDS Zyklus=12 Takte
    
```

**Listing 3: Software-PLL**  
**X@3.ASN; PLL: 16 MHz Takt, DDS-Rate: 16/17 MHz**  
**Referenzfrequenz: 153 kHz, Ausgangsfrequenz: 77,5 kHz an PD6**

```

device at90s1200
include 1200def.inc*

def tmp =r16
def cntr1 =r17
def phase =r18
def outreg =r19

def oDDS0 =r22 ;31 (!) Bit output DDS
def oDDS1 =r23
def oDDS2 =r24
def oDDS3 =r25

def dds0 =r26 ;32 Bit LO-DDS
def dds1 =r27
def dds2 =r28
def dds3 =r29

RESET: ldi tmp,$FC ;Ports als Ausgänge
out DDRB,tmp
ldi tmp,$7F
out DDRD,tmp

SAMPLE: mov tmp,dds3 ;XOR Mischer dds
;mit Analog-Eingang
sbc ACSR,ACO
com tmp
sbrnc tmp,7
inc phase ;Phasenwert berechnen

noSAMPLE: subi dds0,$2D ;LO-DDS: 153kHz 299DB22D
sbc dds1,$B2
sbc dds2,$9D
sbc dds3,$29
subi oDDS0,$09 ;Output DDS: 77 5kHz
;0A8A3D71

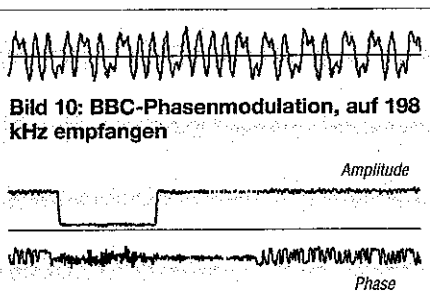
sbc oDDS1,$1B
sbc oDDS2,$1E
sbc oDDS3,$0C ;31 Bit DDS
out PORTD,oDDS3 ;DDS geht an Bit 6
subi cntr1,1 ;Tiefpass zaehlen
brne SAMPLE ;Tiefpaß und
;Phasenvergleich zuende?
;neu starten
;Phasen-Mittelwert
;subtrahieren, C setzen
;Phase loeschen
;16 oder 18 zyklus-
;abhaengig von C

ldi cntr1,126
subi phase,63

clr phase
brcc noSAMPLE

nop
rjmp noSAMPLE
    
```

der zu empfangenen Frequenz liegen. Dadurch kann man extrem schmalbandige PLLs realisieren. Das zugehörige Programm ist ganze 40 Befehle groß. Die DDS-Zykluszeit beträgt dabei 12 Zyklen, d.h., der DDS und Phasenvergleich arbeitet mit 1 MHz, so daß Langwellensignale gut verarbeitet werden können. Damit können alle in der Tabelle erwähnten Phasenmodulationen empfangen werden.



**Bild 10: BBC-Phasenmodulation, auf 198 kHz empfangen**

**Bild 11: DCF-Amplitude (oben) und Phasenmodulation (unten) bei einer 0,1-Sekunden-Absenkung des Trägers**

Bild 10 zeigt die Phasenmodulation der BBC. Bild 11 zeigt die Phasenmodulation des DCF77 zusammen mit der Amplitude. Der Träger wird um 0,1 s abgesenkt, das markiert den Sekundenbeginn. Danach dauert es noch einmal 0,1 s, dann beginnt eine Phasenmodulation mit einer Pseudoräuschsequenz. Mit dieser Phasenmodulation kann man das Zeitsignal wesentlich genauer (besser als 10 Mikrosekunden) auswerten.

Quintessenz: Mit einem Prozessor für 5 DM kann man einen digitalen Langwellen-Datenempfänger realisieren, und aus einem eingefleischten Digital-Fan wird ein Empfangsamateur.

**Ausblick**

Wie gesehen, ist ein einfacher Mikrocontroller von seiner Verarbeitungsleistung in der Lage, in die niedrigsten Gefilde des Rundfunks einzudringen. Nicht immer ist also ein (teurer) Signalprozessor nötig. Die vorgestellten Konzepte lassen sich auch noch

weiter ausbauen. Wie wärs zum Beispiel, wenn man die vorgestellten Programme in programmierbarer Logik mit 100 MHz ablaufen läßt?

Im 4. Teil des Programmierkurses [4] stellte Fred Ziebell übrigens die Frage, wozu wohl all die High-Tech-Prozessoren nütze sind, wenn sie doch meistens nur mit 20 % ausgelastet sind. Nun, die hier vorgestellten Konzepte und Programme schaffen es, so glaube ich, die Prozessoren voll auszunutzen, und jede neue Prozessorgeneration ergibt verbesserte Performance.

**Literatur**

- [1] Rhode, U L., Microwave and Wireless Synthesizers, Wiley and Sons, 1997
- [2] Kernbaum, B., Einfache universelle DDS-Baugruppe, FUNKAMATEUR (47) 1998, H. 1, S. 70
- [3] AVR Enhanced Rise Mikrocontroller Data Book, Atmel Corp, May-1997
- [4] AVR-AT90S1200-Programmierkurs, FUNKAMATEUR (47) 1998, H. 1, S. 40
- [5] The ARRL Handbook for Radio Amateurs, 1995
- [6] Global Positioning, Products Handbook 1996, GEC Plessey 1996