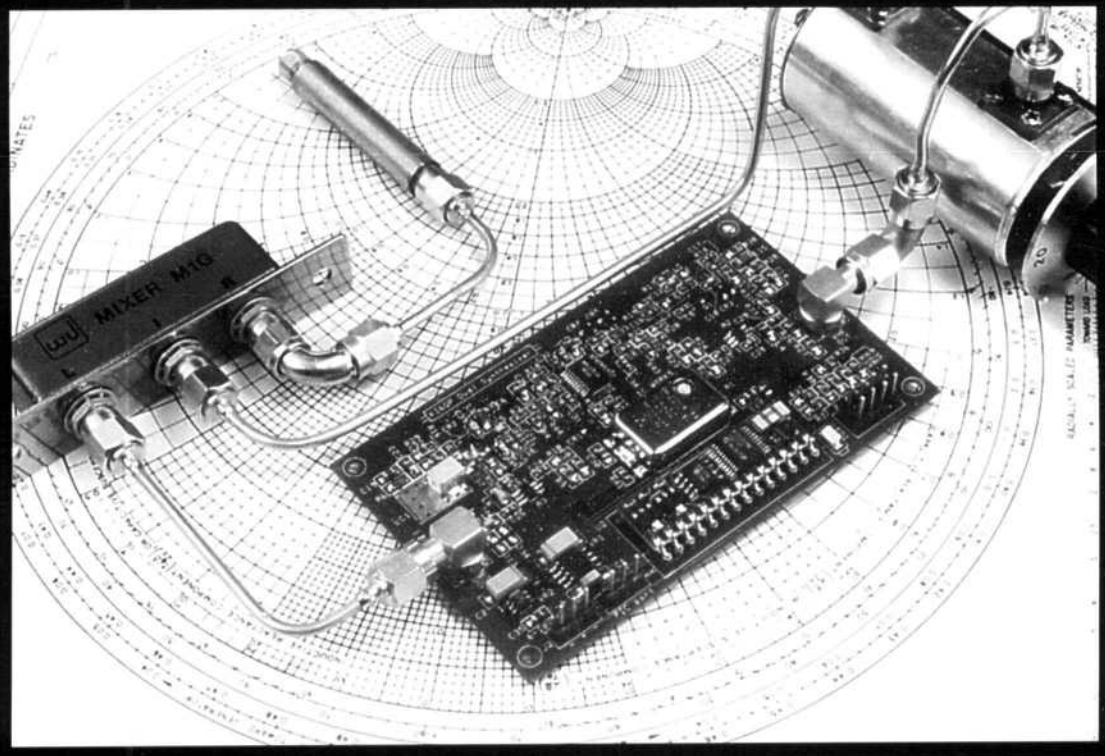
$1.75

QEX

ARRL Experimenter's Exchange

January 1996



A Combined VHF/UHF Synthesizer

QEX: The ARRL
Experimenter's Exchange
American Radio Relay League
225 Main Street
Newington, CT USA 06111

QEX (ISSN: 0886-8093 USPS 011-424) is published monthly by the American Radio Relay League, Newington, CT USA.

Second-class postage paid at Hartford, Connecticut and additional mailing offices.

Offices

225 Main St, Newington, CT 06111-1494 USA
Telephone: 860-594-0200
Telex: 650215-5052 MCI
Fax: 860-594-0259 (24 hour direct line)
Electronic Mail: MCIMAILID: 215-5052
Internet:qex@arrl.org

Subscription rate for 12 issues:

In the US: ARRL Member $15, nonmember $27;

US, Canada and Mexico by First Class Mail: ARRL Member $28, nonmember $40;

Elsewhere by Surface Mail (4-8 week delivery): ARRL Member $20, nonmember $32;

Elsewhere by Airmail: ARRL Member $48, nonmember $60.

QEX subscription orders, changes of address, and reports of missing or damaged copies may be marked: *QEX* Circulation. Postmaster: Form 3579 requested. Send change of address to: American Radio Relay League, 225 Main St, Newington, CT 06111-1494.

Members are asked to include their membership control number or a label from their *QST* wrapper when applying.

About the Cover

Need a VHF/UHF signal source? KI6QP's dual synthesizer operates from 50 to 2000 MHz.

ISSUE NO. 167

Features

Columns

January 1996 QEX Advertising Index

THE AMERICAN RADIO RELAY LEAGUE

The American Radio Relay League, Inc, is a noncommercial association of radio amateurs, organized for the promotion of interests in Amateur Radio communication and experimentation, for the establishment of networks to provide communications in the event of disasters or other emergencies, for the advancement of radio art and of the public welfare, for the representation of the radio amateur in legislative matters, and for the maintenance of fraternalism and a high standard of conduct.

ARRL is an incorporated association without capital stock chartered under the laws of the state of Connecticut, and is an exempt organization under Section 501(c)(3) of the Internal Revenue Code of 1986. Its affairs are governed by a Board of Directors, whose voting members are elected every two years by the general membership. The officers are elected or appointed by the Directors. The League is noncommercial, and no one who could gain financially from the shaping of its affairs is eligible for membership on its Board.

"Of, by, and for the radio amateur, "ARRL numbers within its ranks the vast majority of active amateurs in the nation and has a proud history of achievement as the standard-bearer in amateur affairs.

A bona fide interest in Amateur Radio is the only essential qualification of membership; an Amateur Radio license is not a prerequisite, although full voting membership is granted only to licensed amateurs in the US.

Membership inquiries and general correspondence should be addressed to the administrative headquarters at 225 Main Street, Newington, CT 06111 USA.

Telephone: 860-594-0200
Telex: 650215-5052 MCI
MCIMAIL (electronic mail system) ID: 215-5052
FAX: 860-594-0259 (24-hour direct line)

Officers

President: RODNEY STAFFORD, KB6ZV
5155 Shadow Estates, San Jose, CA 95135

Executive Vice President: DAVID SUMNER, K1ZZ

Purpose of *QEX*:

1) provide a medium for the exchange of ideas and information between Amateur Radio experimenters
2) document advanced technical work in the Amateur Radio field
3) support efforts to advance the state of the Amateur Radio art

All correspondence concerning *QEX* should be addressed to the American Radio Relay League, 225 Main Street, Newington, CT 06111 USA. Envelopes containing manuscripts and correspondence for publication in *QEX* should be marked: Editor, *QEX*.

Both theoretical and practical technical articles are welcomed. Manuscripts should be typed and doubled spaced. Please use the standard ARRL abbreviations found in recent editions of *The ARRL Handbook*. Photos should be glossy, black and white positive prints of good definition and contrast, and should be the same size or larger than the size that is to appear in *QEX*.

Any opinions expressed in *QEX* are those of the authors, not necessarily those of the editor or the League. While we attempt to ensure that all articles are technically valid, authors are expected to defend their own material. Products mentioned in the text are included for your information; no endorsement is implied. The information is believed to be correct, but readers are cautioned to verify availability of the product before sending money to the vendor.

Empirically Speaking

Device Data via the Web

A collection of dog-eared semiconductor device-data books can be found scattered around most experimenters' shacks. And unless you work in a commercial electronics lab, you probably use data books that are at least a couple of years out of date. Even if you *do* have the latest books, what about the reams of update sheets that come out between printings of any data book? There is a solution to this problem, and it will come as no surprise to you that the solution is the World Wide Web.

Most of the companies that produce semiconductor devices make at least some data available for access on the Web. Often, the available data is surprisingly complete. And sometimes there is *more* data available electronically than there is in the data book! Here's a list of sources of device data on the Web. It's necessarily an incomplete list, but it covers many of the major manufacturers. Note that many of these sites provide the data in the form of Adobe PDF files. You'll need the (free) Adobe *Acrobat* reader to view the data; the sites that serve PDF files also provide instructions for obtaining *Acrobat*. These sites are almost all under continual development, so we won't go into specifics as to what's available at each site. We've tried to start you at the page on each site that gets you to the product information fastest.

Analog Devices:
http://www.analog.com/
Cirrus Logic/Crystal Semiconductor:
http://www.cirrus.com/prodtech/
Exar:
http://www.exar.com/products/prodques.htm
Harris:
http://www.semi.harris.com/product_information.html
Hitachi:
http://www.halsp.hitachi.com/tech/tech.html
Maxim:
http://www.mxim.com/
Motorola:
http://Design-net.com/
National Semiconductor:
http://webdirect.national.com/
NEC:
http://www.ic.nec.co.jp/english/products/index.html
Philips:
http://www.semiconductors.philips.com/ps/
Siemens:
http://www.sci.siemens.com/
Texas Instruments:
http://www.ti.com/sc/docs/schome.htm
Xilinx:
http://www.xilinx.com/products.htm
Zilog:
http://www.zilog.com/products.html

Other links to manufacturer product pages are available at **http://www.yahoo.com/Business_and_Economy/Companies/Semiconductors/**.

This Month in *QEX*

Any receiver or transmitter project requires a stable signal source for the receiver LO or the transmitter carrier oscillator. At VHF and UHF, making a stable tunable signal source is most easily done using frequency synthesis. And if you're a VHF/UHF enthusiast, a synthesizer circuit capable of generating a wide range of frequencies would be ideal. Stuart Rumley, KI6QP, presents just that in "The KI6QP Dual Synthesizer Module." His design has both VHF and UHF outputs and operates from 50-2000 MHz or, by mixing the two outputs, up to 2500 MHz.

Among those projects we've published in the past few years, few have sparked more interest than the (almost) all-digital receiver based on a digital down converter. This month, Peter Traneus Anderson, KC1HR, extends the design by adding "A Simple CW Transmit VFO for the DDC-Based Receiver." Using the VFO to drive a high-tech transmitter (a Heath DX-40!), Peter has achieved transceive operation—controlled by a PC.

If you want to design a passive network to match two known impedances, you'll find all of the theory and cookbook solutions easily. But what if you want to know the *range* of impedances a particular network will match? That's quite a bit different! In this month's "RF" column, Zack Lau, KH6CP/1, shows some computer-aided techniques for finding the answers —*KE3Z, email: jbloom@arrl.org.*

The KI6QP Dual Synthesizer Module

RF building blocks for communication systems.

By Stuart Rumley, KI6QP

Monolithic microwave integrated circuits (MMICs) are becoming increasingly available from a number of manufacturers at low cost. MMICs are available as amplifiers, mixers, I-Q modulators/demodulators, LNA/converters, attenuators, buffer/power amplifiers, switches or some combination of all of these. These devices have nominal 50-Ω port impedances, internal coupling and biasing, and use convenient supply voltages. As a result, you can easily and quickly configure a variety of transmitter and receiver system architectures from dc to over 2 GHz with these RF building blocks.

However, in my experience, what is not available is a similarly convenient precision source of RF for the local oscillators and carrier generators that these wonderful new chips require. To experiment with the new devices for 902 MHz, 1296 MHz and 2400 MHz, for example, you will invariably need to either build a complicated crystal oscillator-multiplier or have on hand a couple of expensive laboratory synthesized signal generators—or both. Any attempt to get by with unstable oscillators or generators will be frustrating and will not allow you to fully appreciate the capabilities of these new RF ICs.

The purpose of this project is to provide the missing blocks in the form of a low-cost, synthesized frequency source that can be easily duplicated and is capable of providing programmable outputs in the 50-MHz to 2-GHz range. I also wanted to reduce the cost and complexity of any digital control logic, minimize the required programming and eliminate as much tuning as possible. Furthermore, I needed something that was self-contained and would not always require a personal computer to load the frequency data.

I believe the success of this project is due in large part to the availability of three significant new technologies:

1) Coaxial ceramic dielectric resonators in the form of a λ/4-transmission line. These resonators allow the design of a repeatable VCO in the 800- to 2500-MHz frequency range with a drop-in component. No tuning of multiplier stages is required because the VCO is capable of operation at the output frequency, and a simple broadband MMIC amplifier can be used to buffer the output.

308 Nevada St
Redwood City, CA 94062
e-mail: stuart@itron-ca.com

2) High-integration phase-lock-loop (PLL) integrated circuits. These devices include charge-pump phase detectors, high-frequency prescalers and serially loaded control registers. They use a low-cost BiCMOS process and have input counters capable of clocking at over 2.5 GHz with less than –10 dBm of input.

3) Low-cost BASIC computer modules. Programmable in BASIC, these modules have on board BASIC interpreters with nonvolatile program storage in electronically erasable memory. No compilers, PROM burners

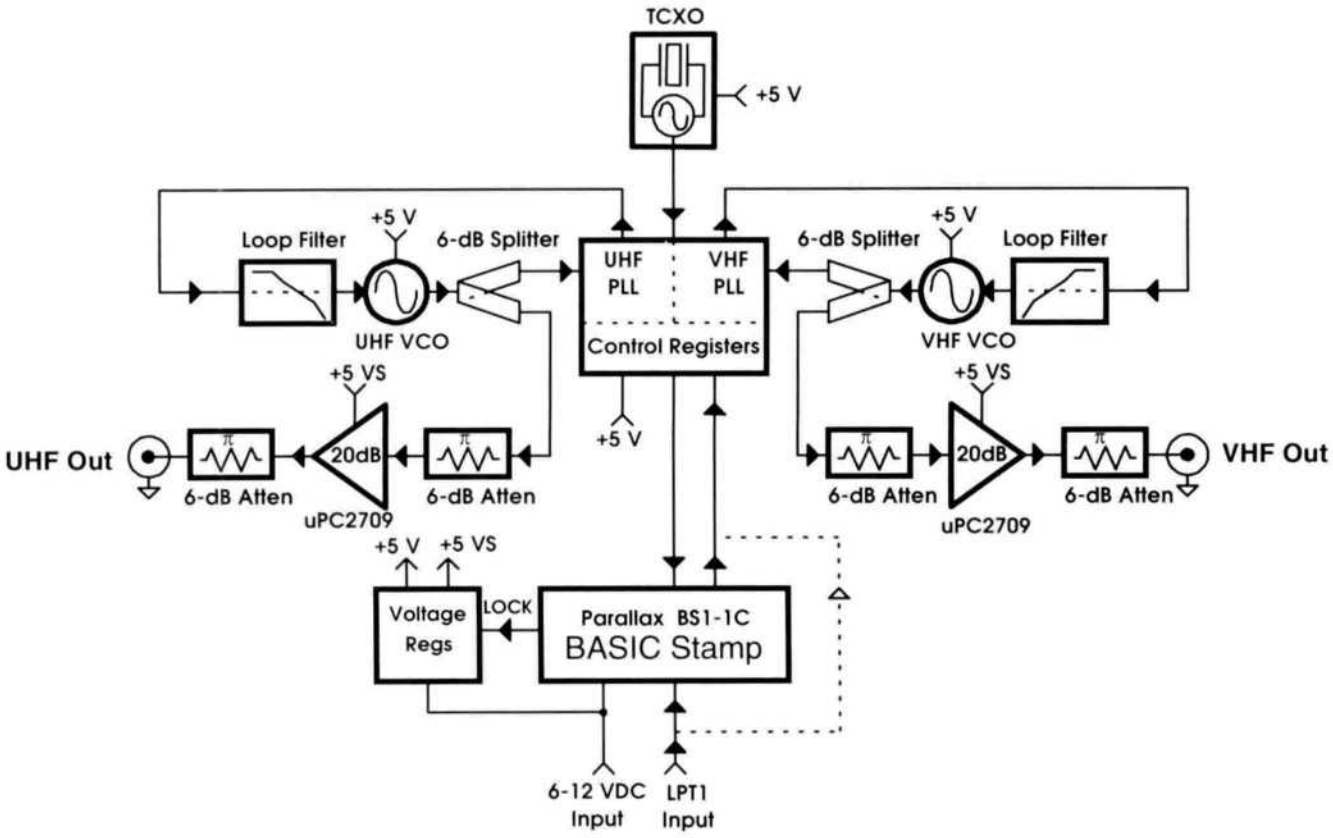


Fig 1—Dual synthesizer module block diagram.

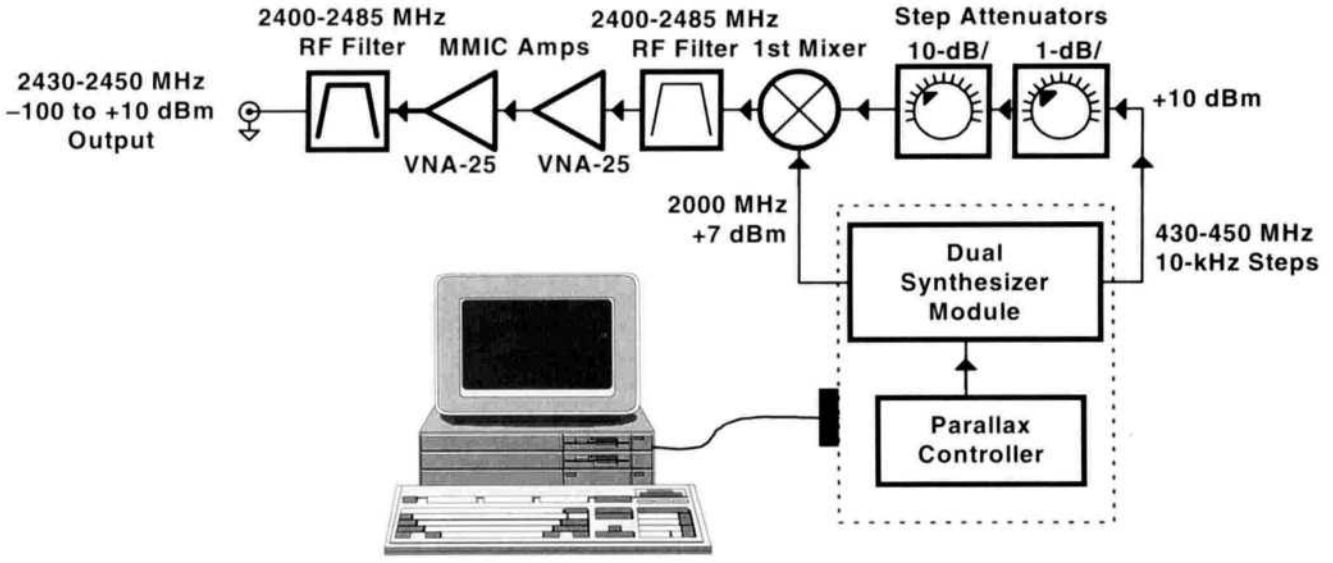


Fig 2—Synthesized signal source.

or assembly language required.

Fig 1 is a block diagram of the dual synthesizer module. There are two VCOs each followed by a passive power splitter and a separate buffer amplifier. The UHF VCO is designed for use with coaxial ceramic resonators in the 800- to 2500-MHz range. The VHF VCO is L-C tuned and designed for operation in the 50- to 500-MHz range.

The high-integration dual-PLL IC is connected to both VCOs and combines two synthesizer functions in one package. A shared reference frequency is provided to individual PLL circuits by a high-stability TCXO module. Passive loop filters are used on both VCOs to stabilize each synthesizer. A microcomputer module with E^2PROM memory, two voltage regulators and some simple interface circuitry provides everything necessary in order to provide a pair of independent, stand-alone, self-contained, synthesized frequency sources.

In many applications only one of the synthesizer outputs may be needed at any one time. Fig 2 demonstrates how both outputs may be used together in order to derive a single output. Here the VHF output is mixed with the UHF output to provide a controlled output at 2.4 GHz. The implications of this configuration are numerous. For example, the step attenuator used here is only accurate to 500 MHz, but this setup gives calibrated output levels at 2.4 GHz. Fig 3 is an example of how I have been using the synthesizer module. Here it provides the first and second LOs in a dual-down-conversion scheme to allow the use of a 2-meter receiver on the 2.4-GHz band. A bilateral scheme is in the works to allow half-duplex operation with a hand-held transceiver for use with repeaters.

There are two programming options available for users of the dual synthesizer module. The first option is to use the evaluation test software that is available from the IC manufacturers (see "Sources" for suppliers). This option requires an IBM personal computer (PC) connection from the PC's parallel port (LPT1) to J4. For convenience, J4 is laid out to accept the same programming cable supplied by the PLL chip manufacturers for use with their evaluation boards. With this configuration, each time the dual synthesizer is powered up you will need to download the PLL register information from the PC. If you are satisfied with this mode then you do not need to install the Stamp module at all.

The second programming option allows for stand alone operation. The on board Parallax BASIC Stamp module reloads the synthesizer registers on each power-up and also tests for lock condition from each PLL. The Stamp has 256 bytes of electronically erasable, programmable, read-only memory (E^2PROM) that can hold the programming information when power is removed. This method frees you from having to have a PC around each time you power up the dual synthesizer module. But you will not need to be a software engineer to program the microprocessor because the Stamp comes with an internal BASIC interpreter.

Circuit Description

The circuit schematic is detailed in Fig 4. U1, the dual frequency synthesizer integrated circuit, commonly referred to as a PLL chip, can be either the National Semiconductor LMX2331A or the IC Works WB1331. These are state-of-the-art, high-performance ICs each capable of operation beyond 2.0 GHz for their UHF prescaler inputs and 500 MHz for their VHF inputs. The dual synthesizer IC has two completely separate sections sharing a common 22-bit serial-to-parallel data register. This data register is loaded with the frequency program data either via a 3-wire interface directly from a personal computer's parallel port connected to J4, or by the on-board BASIC Stamp, U2. Diodes CR4 and CR5 drop the 5-V supply, 5VR, to approximately 3.6 V for the PLL. This lower supply voltage increases the PLL's input prescaler sensitivity and reduces the phase detector's output noise.

The VHF and UHF oscillators use NEC NE68033 silicon bipolar transistors in a form of modified Colpitts configuration. These are low-cost devices with an f_t of 10 GHz and a typical noise figure of 2 dB. Q1 is an L-C tuned VHF oscillator circuit, and Q2 is a coaxial dielectric resonator tuned UHF oscillator circuit. Both of these oscillators rely on a capacitive reactance in their emitter circuits to provide a negative impedance to the resonating elements (see the section on VCO design). The output of the VHF oscillator is fed to a 6-dB resistive power splitter comprising R20, R21 and R22. The output of the UHF oscillator is also fed to a 6-dB splitter, R2, R3 and R4. The output of each splitter is capacitor coupled to the inputs of the dual synthesizer IC. Typically, the UHF VCO has a maximum tuning range of approximately

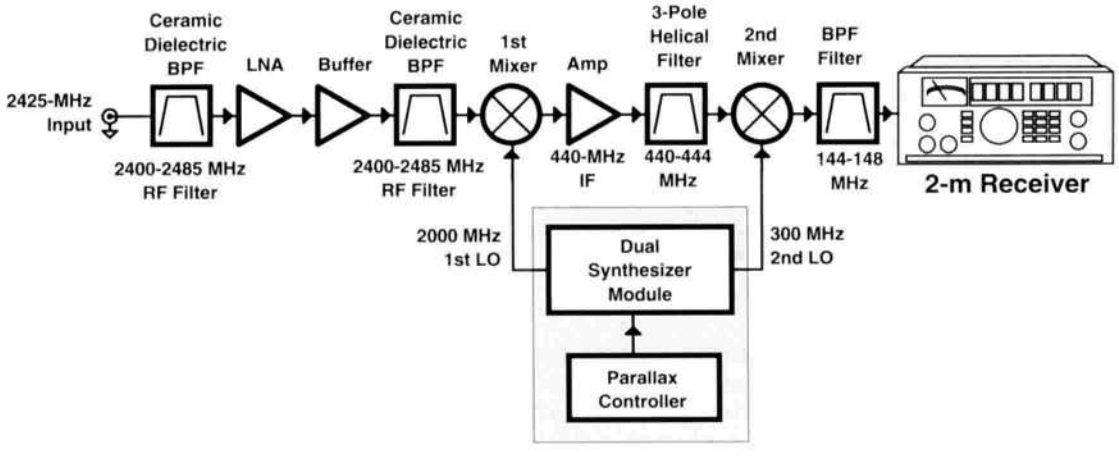


Fig 3—Synthesized signal source.

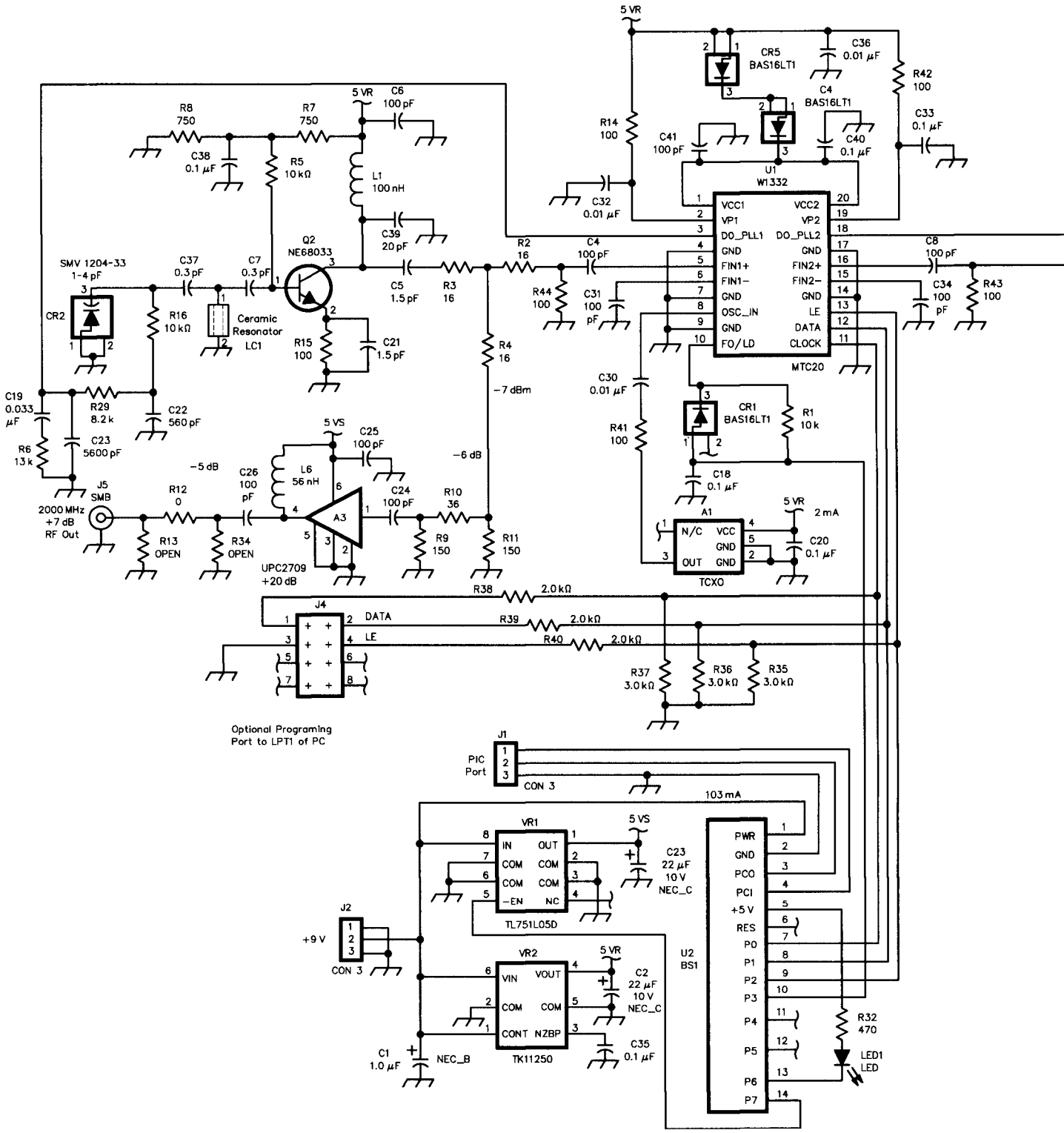
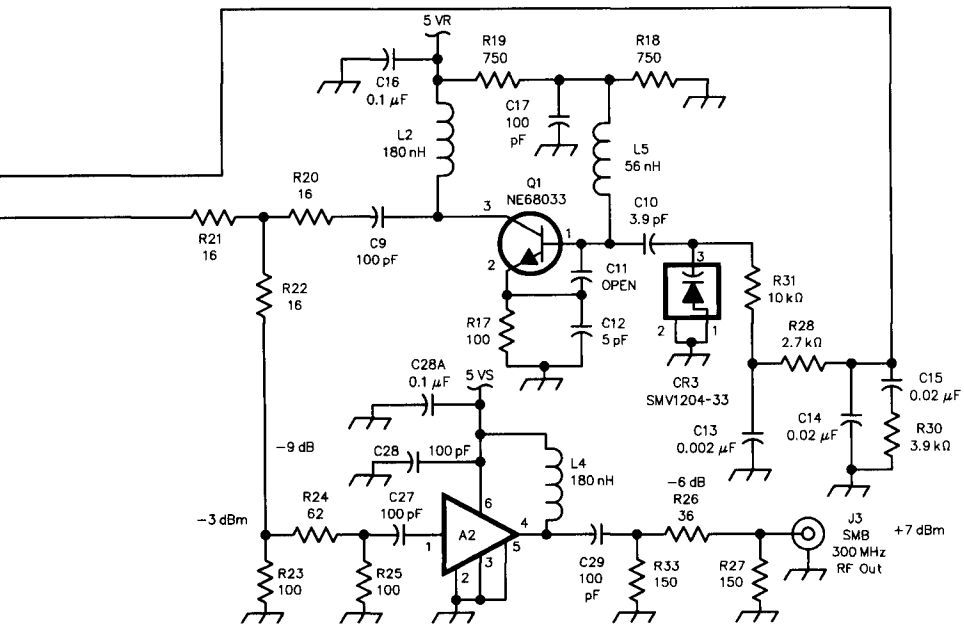


Fig 4—KI6QP dual synthesizer schematic diagram.

±5% as determined by the value of varactor CR2, its coupling capacitor, C37, and the base coupling capacitor, C7. Similarly, the VHF VCO can be designed to have a practical tuning range as large as ±20% as determined by the value of varactor CR3, its coupling capacitor, C10, and the combined effects of C11 and C12.

The other output from each splitter goes to an attenuator-amplifier-attenuator combination. The amplifiers, A2 and A3, are NEC uPC2709s. These MMICs are capable of 20 dB of gain and +10 dBm of output power at 2 GHz. They have a 5-dB noise figure and draw only 25 mA from a 5-V power source. The amplifier input attenua-



tors, R9, R10 and R11 for the UHF side, and R23, R24 and R25 for the VHF side, may be adjusted to provide the desired output power levels. In the case of the UHF side, it may be desirable to remove the output attenuator (R12, R13, R34) in order to get maximum power output. Also, either amplifier output attenuator circuit can easily be modified by replacing the resistors with Ls and Cs to perform a filter function. This may be particularly desirable on the VHF side to minimize harmonics. These attenuators and buffer amplifiers provide a high degree of isolation of the oscillator circuits from the loads. This is desired in order to minimize oscillator phase noise that results from perturbations in the loads.

U2 is the Parallax BASIC Stamp computer module. The unique usefulness of this novel device has been featured in a previous article.[1] This module includes a PIC16C56 microcomputer, a 256-byte E²PROM, a 4-MHz ceramic resonator and a 5-V regulator. The microcomputer is preprogrammed with Parallax's PBASIC interpreter.

The PBASIC language includes familiar BASIC instructions such as FOR...NEXT, IF...THEN and GOTO, as well as SBC instructions, such as SERIN (serial input), PWM (pulse width modulation) and POT (reads a 5- to 50-kΩ variable resistance) to name just a few. This module is used to save the PLL frequency programming register information and reload the bits in the correct sequence on power-up. The Stamp module also checks for a phase-lock condition from the PLL lock detector filter, CR1, R1 and C18. If a lock is detected, VR1 is enabled and the buffer amplifiers are powered on.

A1 is the temperature-compensated crystal oscillator (TCXO) used as the main frequency reference and shared by both internal PLL reference dividers and phase detectors. In the dual PLL, operating from a crystal alone is not possible, but a TCXO of practically any frequency can be used. Select any frequency in the range of 1 MHz to 20 MHz and check that the desired internal reference frequency is obtainable from the reference divider. A small-value resistor and coupling capacitor have been included in series with the reference input to the synthesizer chip to optimize the reference signal level.

The dual PLL IC is connected to each VCO tuning varactor diode by way of a loop filter. The loop filters are primarily responsible for implementing the control law that keeps the synthesizer stable. Each of the loop filters are

[1]Notes appear on page 13.

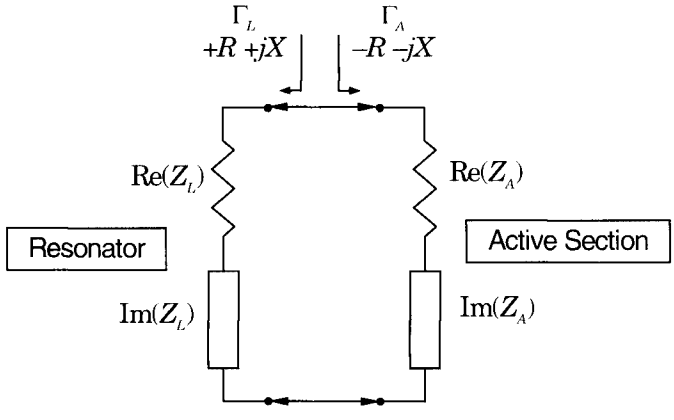


Fig 5—Oscillator block diagram.

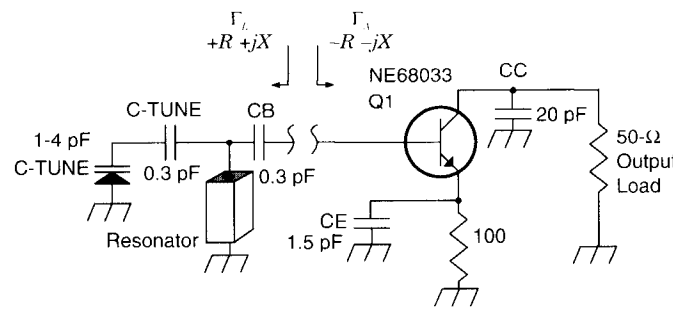


Fig 6—UHF oscillator model.

made up as a second-order integrator-with-zero (IWZ) followed by an additional pole to provide a third-order low-pass function. The second-order IWZ is required for loop stability while the additional low-pass section is needed to remove the spurious reference frequency sideband signals. The IWZ for the UHF side is made up of C19, C23 and R6. The IWZ for the VHF side is C14, C15 and R30. The additional low-pass for the UHF side is set by R29 and C22, while R28 and C13 provide this function for the VHF side.

VR1 and VR2 are low-drop-out voltage regulators. Both of these devices can provide a regulated output with input voltages just a few millivolts higher than their outputs. VR1 is used to provide 5-V regulated current to only the buffer amplifiers. VR1 is enabled by the microcomputer when a "locked" condition is detected. VR2 is a low-noise, 5-V regulator used to power the sensitive VCO and PLL circuits only. VR2 is always enabled. The BASIC Stamp module has its own independent low-drop-out voltage regulator and can operate near 5 V as well.

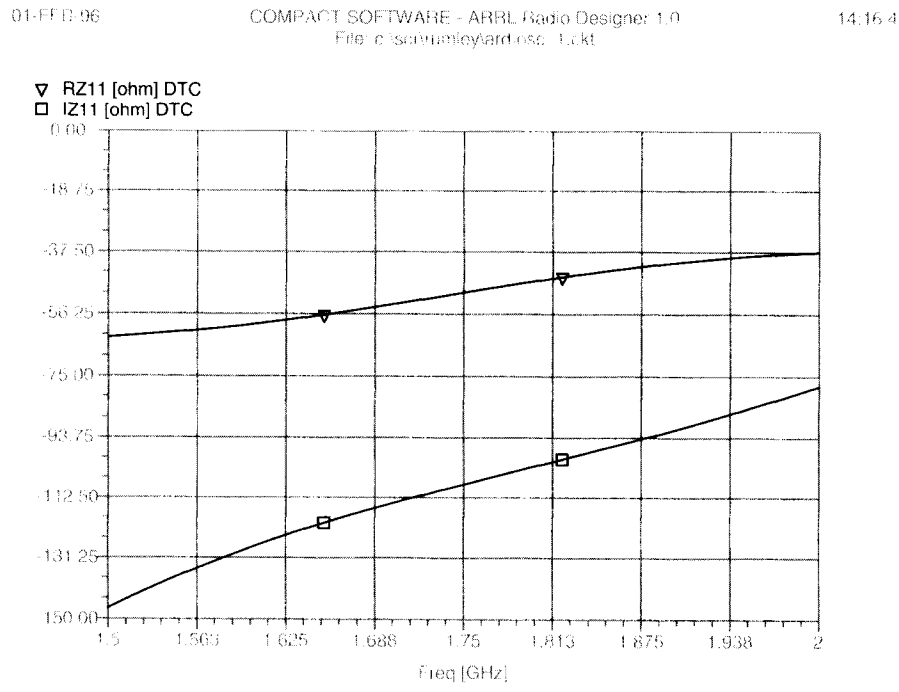


Fig 7—Impedance looking into active section.

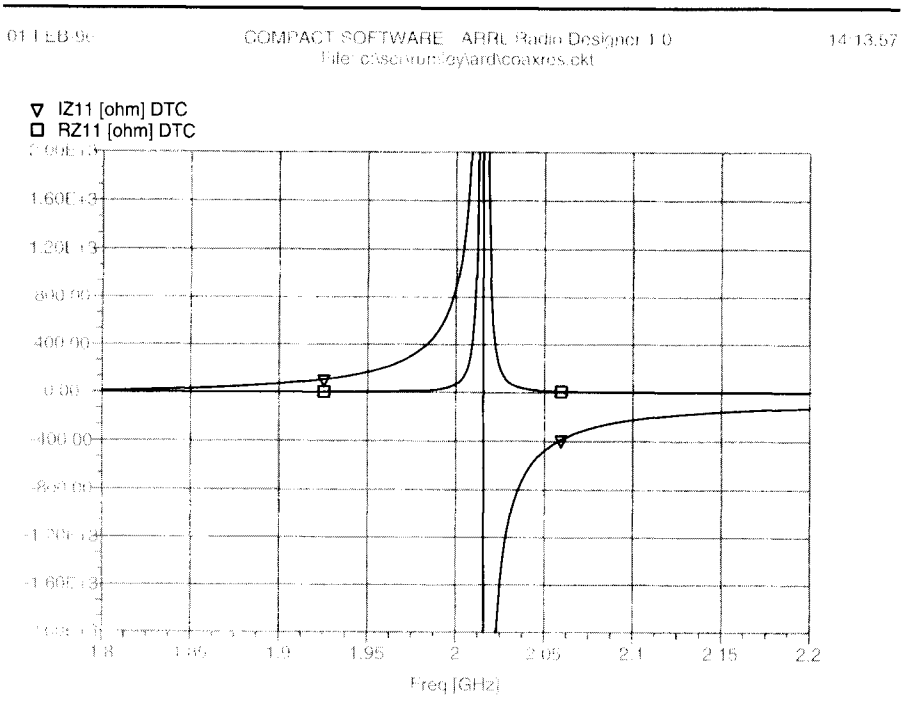


Fig 8— Impedance looking into "load."

VCO Design

Oscillator design techniques, especially those using linear simulator software for analysis, seem to fall into one of two distinct categories. One category uses reflection coefficient techniques and is particularly well suited for microwave oscillators using distributed elements such as microstrip transmission lines. The other category uses the negative resistance technique, and I believe this is more appropriate for oscillators at UHF and below.

Much has been written on oscillator design. The best references I have found for the reflection coefficient technique and the negative resistance technique are listed in the notes.[2,3] I used the negative resistance technique for analysis and simulation of the oscillators used on this project. Both techniques will yield equivalent steady-state results but slightly different start-up conditions.[4] The important thing is to use and perfect whichever method *you* understand best. The dualities that exist between these two methods should be apparent from the summary of oscillator conditions shown below.

Summary of oscillator conditions

Reflection Coefficient Technique

Start-up: $|\Gamma_A| \bullet |\Gamma_L| > 1$, $\angle(\Gamma_A) + \angle(\Gamma_L) = 0$

Steady State: $|\Gamma_A| \bullet |\Gamma_L| = 1$, $\angle(\Gamma_A) + \angle(\Gamma_L) = 0$

Negative Resistance Technique

Start-up: $\mathrm{Re}(Z_A) + \mathrm{Re}(Z_L) < 1$, $\mathrm{Im}(Z_A) + \mathrm{Im}(Z_L) = 0$

Steady State: $\mathrm{Re}(Z_A) + \mathrm{Re}(Z_L) = 1$, $\mathrm{Im}(Z_A) + \mathrm{Im}(Z_L) = 0$

where:

Γ_A is the reflection coefficient of the active section.

Γ_L is the reflection coefficient of the load.

$\mathrm{Re}(Z_A)$ is the real part (although negative) of the active section impedance.

$\mathrm{Re}(Z_L)$ is the real part of the load impedance.

$\mathrm{Im}(Z_A)$ is the imaginary part (these are complex impedances) of the active section.

$\mathrm{Im}(Z_L)$ is the imaginary part of the load impedance.

Keep in mind here that the term *load*, in the above summary, refers to only the resonant elements associated with the oscillator and not necessarily where power is being extracted to serve some useful purpose.

The oscillator designs used in this project were done with the aid of the linear circuit simulator *SuperStar* V4.0 from Eagleware and simulation techniques adopted from Randy Rhea.[5] Other, less expensive linear simulators that give similar results include *Star* 2.0 and *ARRL Radio Designer*, which is a subset of *Super Compact* (see "Sources"). The simulation examples are from *ARRL Radio Designer* and yield similar results to those from *SuperStar*.

Fig 5 shows a block diagram of the negative resistance concept. The active device on the right side provides the negative resistance and in most cases the negative reactance as well. The left side is the tuned element and is generally a net positive reactance (as in an inductance) plus a positive resistance. The left side is what is referred to as the load in the summary above. The positive resistance is the loss associated with the unloaded Q of the resonator. It is this loss that the active section must overcome with its net negative resistance supplied by the active element in the oscillator.

Fig 6 shows how the UHF oscillator is evaluated. First, Z_A is analyzed to ensure that the real part of the active section impedance is negative. Fig 7 shows this to be between –60 and –40 Ω from 1.5 to 2.0 GHz. This is a very desirable value; more would be better, but this is a good starting point. Next, Z_L is analyzed, tuned and then appropriate changes are made to the tuning elements to ensure operation in the desired frequency band. Fig 9 is the complete oscillator circuit analyzed with the source applied to the bottom end of the resonator and the output taken from the collector. The oscillating frequency will be near where the large phase transition occurs and the gain is maximum. Disregard the actual magnitude of the voltage gain (VG) as this is inconsequential because the impedances of the source and load are different. The source impedance in this case is set to 0.5 Ω, see Fig 10.

In the UHF oscillator, Q2 and the feedback capacitor, C21, provide a negative impedance for the 1/4 dielectric resonator, LC1, and its associated tuning elements, C7 and C37, and varactor diode CR2. You can experiment with the value of C21 in your simulation model and in the actual circuit. I found that 1.5 pF was universally the best value for this application. The typical Colpitts oscillator has the collector at RF ground potential and places the load in the emitter circuit. In this modified Colpitts oscillator topology the load is placed in the collector. This isolates the tuning elements from the load by the S_{12} parameter of the transistor, which enhances the oscillator phase noise and stability. However, to ensure sufficient negative resistance, a low value of capacitive reactance must be placed in the collector circuit. I found that optimum output power was achieved with values near 20 pF for C39. Surprisingly, this capacitor has little effect on the oscillator frequency.

The dielectric resonators used in these oscillator circuits actually operate considerably below their 1/4-wave resonant frequency. This is because the active section has a limited amount of negative resistance and a fixed amount of negative reactance. If you analyze Fig 8, you will see that the frequency at which the positive reactance of the "load" is near the negative reactance of the active section is approximately 100 MHz below where the "load" is resonant. More detailed information on these oscillator design techniques can be found in the notes.[6,7]

If you know the dielectric constant of the resonator you have, you can calculate the 1/4-wave resonant frequency from:

$$F_{\lambda/4}GHz = \frac{30\,\mathrm{cm}}{4 \bullet L\sqrt{\varepsilon_r}}$$

Where $F_{\lambda/4}\ GHz$ is the resonant frequency in GHz, L is the resonator physical length in cm (2.54 cm/in) and ε_r is the dielectric constant of the ceramic material used to make the resonator. If you know nothing about the resonators you have you can find the resonant frequency by using a sweep generator and a reflectometer. The test port of the reflectometer is *very* lightly coupled to the resonator and the frequency at which the return loss dips can be taken as

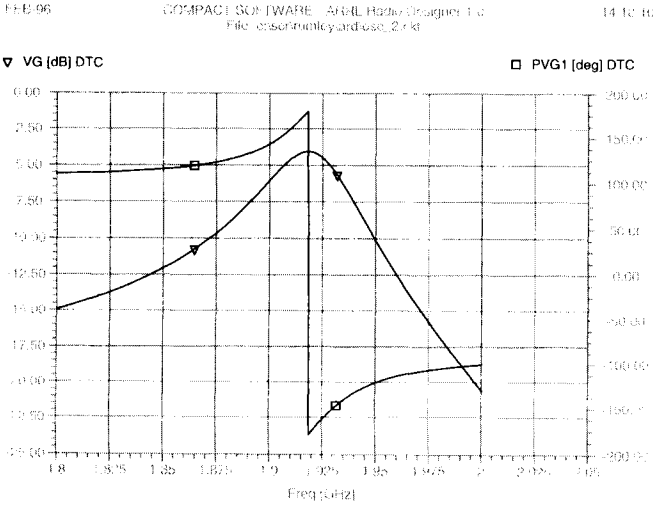


Fig 9—Oscillator simulation.

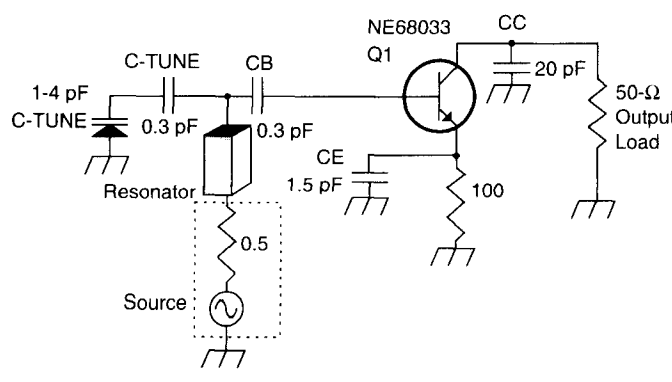


Fig 10—Complete simulation model.

the resonant frequency. From this you should be able to derive the dielectric constant. Typically you will find only three common dielectric materials in use. These are ε_r = 21, 36 and 88.

All of the techniques used in the UHF oscillator design and analysis can be applied to the VHF oscillator. The VHF oscillator consists of Q1 and the feedback capacitors C11 and C12 and provides a negative impedance to tuning elements L5, C10 and varactor diode CR3. In some cases C11 will not need to be installed due to sufficient parasitic capacitance provided by the transistor and circuit board.

The component values shown in the schematic will give a center frequency of approximately 300 MHz. By changing the value of L5, C11, C12, C10 and the value of the varactor CR3, you should be able to modify the operating frequency of the VHF VCO anywhere from 50-500 MHz.

PLL Compensation

The component values in the loop filters shown in the schematic are for PLLs operating with a specific set of parameters. If your loop parameters are not similar to those shown in Table 1 you will need to calculate new loop filter component values. There are already many good references for PLL design and compensation.[7,8,9] I will present a more cookbook approach to determining the values for the resistors and capacitors to use in the loop filters. This approach was developed from the National Semiconductor Corporation application note on passive loop filter design.[10]

In order to proceed with finding the loop filter values, which will establish $Kf(s)$, you will have to either set or determine the parameters for the other functional blocks shown in the mathematical model of Fig 11. The process is a compromise between those parameters you can not change, those parameters you will want to change and those parameters you may need to change. For example, you might decide you need a certain loop bandwidth or a particular channel spacing (reference frequency) only to discover that the capacitors or resistors are excessively large or otherwise impractical. You may then be able to change the phase margin or VCO tuning rate, for example, in order to realize a more practical solution.

To start your own design, begin by editing Table 1 with your initial parameters, some of the parameters may need to be changed as a result of the calculations yielding impractical component values. Follow the example below for the VHF PLL operating with the parameters shown in Table 1. When you have completed the VHF design, repeat the process for the UHF PLL. This example will begin by only calculating the second-order filter components as if the filter were configured as shown in Fig 12.

1) Measure the tuning rate of the VCO; you need two tuning voltage points and two corresponding frequencies.

Then: K_{v_low} = 290 MHz at v_low = 0.5 V; and K_{v_high} = 3.10 MHz at v_high = 4.5 V; the tuning rate is then:

$$\frac{K_{v\_high} - K_{v\_low}}{v\_high - v\_low} = K_{vco} = 5\frac{\text{MHz}}{\text{volt}}$$

angular form:

$$K_v = 2\pi \bullet K_{vco} = 3.142 \bullet 10^7 \frac{\text{rad}}{\text{sec.volt}}$$

2) Find the phase detector gain constant. This can be obtained (usually with some effort) from the manufacturer's data sheet information. This constant is a function of supply voltage and programming mode and may differ

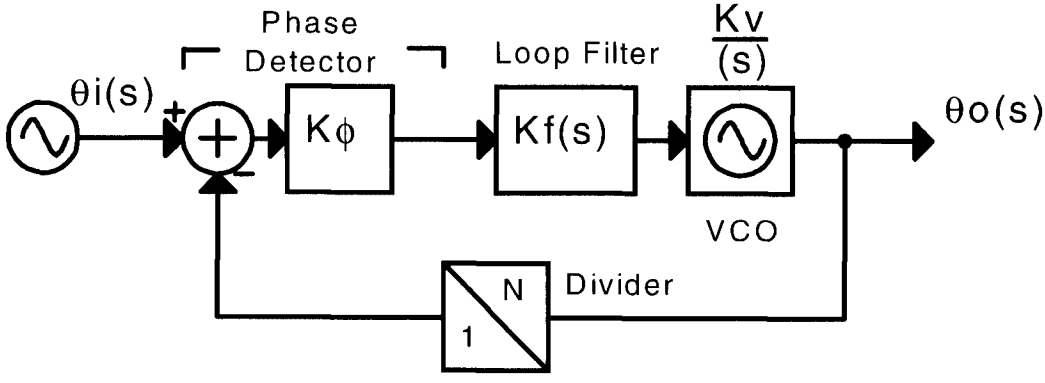


Fig 11—Block diagram of PLL.

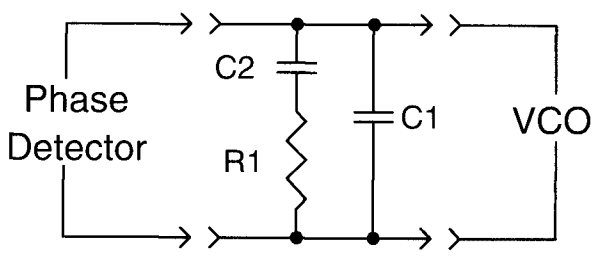


Fig 12—Second-order loop filter.

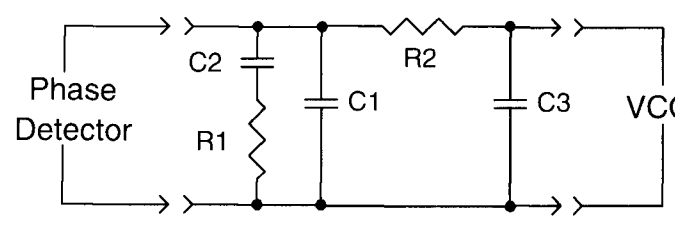


Fig 13—Third-order loop filter.

from one manufacturer to the next. This constant is not generally a tabular parameter, but rather must be determined from performance curves. In most all cases, however, you find these phase detectors will have a range of $\pm 2\pi$ radians.

$$K_{dmin} = -1\,\text{mA}; K_{dmax} = +1.0\,\text{mA} \quad K\phi = \frac{K_{dmax} - K_{dmin}}{2\pi - (-2\pi)} = 0.159\frac{\text{mA}}{\text{rad}}$$

3) Select a loop bandwidth, f_p; this example uses 1000 Hz. A lower loop bandwidth will give better phase noise characteristic but results in longer settling time and larger capacitor values. Now calculate:

$$\omega_p = 2\pi \bullet f_p = 6283\frac{\text{rad}}{\text{sec}}$$

4) Select a reference frequency. Use the largest reference frequency that is consistent with the desired channel spacing and can be accommodated by the PLL's reference divider. The higher the reference frequency, the lower the reference sideband spurs.

5) Calculate the main loop division ratio, N;

$$N = \frac{RF_{opt}}{F_{ref}} = 3000.$$

A typical value for RF_{opt} is the middle of the VCO tuning range, $(K_{v\_high}+K_{v\_low})/2$.

6) Select a phase margin. In this example ϕ_p=45°.

7) Now solve for the two time constants:

$$T1 = \frac{\sec(\phi_p) - \tan(\phi_p)}{\omega_p} = 65.9\,\mu\text{sec}$$

and

$$T2 = \frac{1}{\omega_p^2 \bullet T1} = 384.2\,\mu\text{sec}$$

8) And now we can find all of the component values for the second order loop:

$$C1 = \frac{T1}{T2} \bullet \frac{K_\phi \bullet K_v}{\omega_p^2 \bullet N} \sqrt{\frac{1+(\omega_p \bullet T2)^2}{1+(\omega_p \bullet T2)^2}} = 0.017\,\mu\text{F} \cong 0.018\,\mu\text{F};$$

$$C2 = C1\left(\frac{T2}{T1} - 1\right) = 0.084\,\mu\text{F} \cong 0.1\,\mu\text{F}; \quad R1 = \frac{T2}{C2} = 4.55\,\text{k}\Omega \cong 4.7\,\text{k}\Omega.$$

That should now provide you with component values that will give a stable PLL. If you decide that further reference-spur suppression is needed, add a low-pass section following the second-order filter. Fig 13 shows the relationship of the low-pass section, R2 and C3, to the original loop filter.

9) Amend the second-order design by selecting the amount of additional reference-spurious reduction desired. Let's just say we desire an additional 20 dB of reference-spur reduction. Select a value for C3 that is no larger than 1/10 the value of C1 previously calculated. Then find the new time constant T3:

$$T3 = \sqrt{\frac{10\left(\frac{attenuation}{20}\right)}{2\pi \bullet F_{ref}}}$$

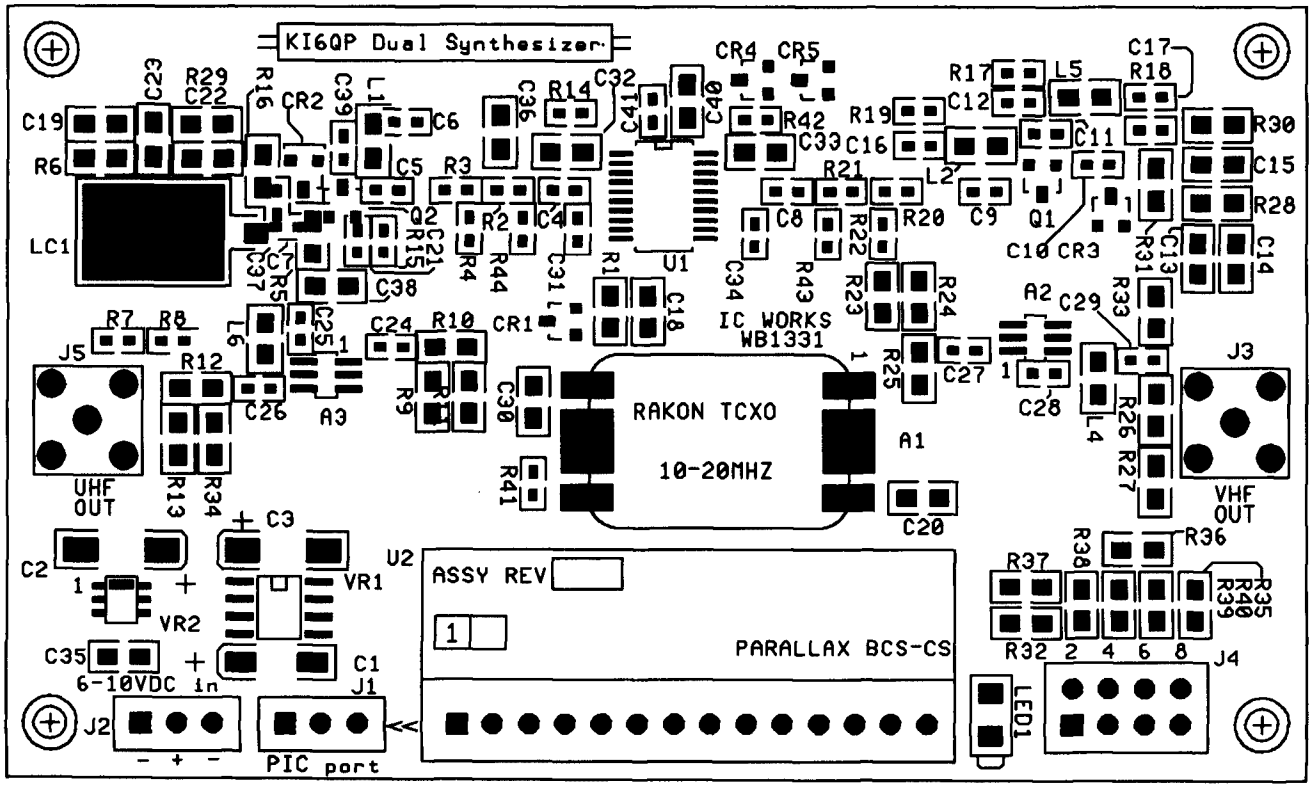


Fig 14—Surface-mount printed-circuit board.

This now gives a slightly lower loop bandwidth:

$$\omega_c = \frac{\tan(\phi_p)\bullet(T1+T3)}{\left[(T1+T3)^2+T1\bullet T3\right]}\bullet\left[\sqrt{1+\frac{(T1+T3)^2+T1\bullet T3}{\left[\tan(\phi_p)\bullet(T1+T3)\right]^2}}-1\right]$$

$$= 5806\frac{\text{rad}}{\text{sec}}$$

The second-order filter time constant, T2, will change slightly to:

$$T2 = \frac{1}{\omega_c{}^2\bullet(T1+T3)} = 419.6\mu\text{sec}$$

which will change the value of C1, C2 and R2 slightly as well.

$$C1 = \frac{T1}{T2}\bullet\frac{K_\phi K_v}{\omega_c{}^2\bullet N}\sqrt{\frac{1+(\omega_c\bullet T2)^2}{\left[1+(\omega_c\bullet T1)^2\right]\bullet\left[1+(\omega_c\bullet T3)^2\right]}}$$

$$= 0.019\mu\text{F} \cong 0.02\mu\text{F};$$

$$C2 = C1\left(\frac{T2}{T1}-1\right) = 0.102\mu\text{F} \cong 0.1\mu\text{F};$$

$$R1 = \frac{T2}{C2} = 4.09\text{k}\Omega \cong 3.9\text{k}\Omega;$$

and finally $R2 = \frac{T3}{C3} = 2.73\text{k}\Omega \cong 2.7\text{k}\Omega$.

This completes the calculations for all the loop filter component values. You will probably discover that great latitude can be taken with the actual values before the loop will be completely unstable and incapable of remaining locked. Now you are ready to proceed with programming the PLL and putting the synthesizer on frequency.

A template of these calculations, using MathSoft's *Mathcad*, is provided in the appendix. I highly recommend using the systematic approach that the template gives you because not only does it eliminate the manual calculations but it also allows you to document your design. If you don't have *Mathcad*, you can also use Microsoft's *Excel* spreadsheet to perform the calculations and documentation.

Programming

Wait!! Don't worry this will be easy...it's not *that* kind of programming. Again, you have two options here: 1) use your PC (IBM or compatible) and program the PLL using a handy screen menu, or 2) write a *simple* BASIC program (using the listing shown as a template) that will reside in the on-board Stamp module non-volatile memory.

The synthesizer IC manufacturers will supply programming software (free) that runs on a PC and programs the PLL directly through the parallel (LPT1) port connection via J4. This software allows you to independently set each bit of the four 20-bit PLL registers. The IC Works software has a very flexible interface scheme that allows you to set the frequency, and it will calculate the register values or you can set the register values and the software will calculate the frequencies. This software also supports frequency jump and sweep routines, that are very useful in testing the dynamic response of the PLL.

If you want your PLL registers to be automatically loaded on power up, you will want to install the BASIC Stamp module and program it following the sample program listed below. The software is available free from Parallax's BBS (see "Sources") but you might want to get the complete BASIC Stamp kit in order to get all of the documentation and applications.

Either programming option will require a separate special cable depending on whether you're connecting directly through J4 or programming the Stamp module through J1. Table 2 lists the pin connections you will need to make for either cable.

Even if you opt for the BASIC Stamp mode of operation you probably will want to use the IC manufacturers' supplied software for your initial debugging phase. Also, you will want to have the synthesizer data sheet on hand so that you will better understand all of the options and how to program.

To program the PLL you must first solve the frequency equation in order to determine the values for the registers. The general form is:

$$RF_{opt} = \left[(P\bullet B)+A\right]\bullet\frac{ExFref}{R}$$

where:

P = Prescaler; 8 or 16 for VHF, 64 or 128 for UHF.
B = Program counter; $3 \le B \le 2047$ also $B \ge A$
A = Swallow counter; 0..15 for VHF, 0..127 for UHF.
R = Reference divider; $3 \le R \le 32676$.
RF_{opt} = the output frequency desired.
ExFref = the external reference oscillator or TCXO input, usually between 2 MHz and 20 MHz.

Consult the manufacturers' data sheets on the uses of the *Rprog* and *Nprog* registers. These registers allow a number of optional operating and diagnostic modes to be configured in the PLL. The last few lines of the main program put the Stamp module in an endless loop of checking the PLL lock detector. If either PLL fails to lock, the Stamp will prevent the buffer amplifiers from powering on by disabling the voltage regulator. You may want to modify this part of the program to put the Stamp module in the End mode. This will completely disable the processor and it will not function again until reset by power up. This may prove desirable in those applications where the utmost reduction in residual FM is wanted, as the Stamp module can contribute some FM noise if it remains running in the lock loop.

Assembly and Construction

The dual synthesizer was fabricated on a two-layer, copper-clad, FR-4 glass-epoxy circuit board. With the exception of the Parallax module and the connectors, all of the components are surface-mount devices. The resistors and capacitors are 0603- and 0805-size code parts. Surface-mount technology was required in order to maintain control of the parasitic circuit elements and ensure repeatable construction. As a result, very consistent performance from unit to unit has been observed. High-frequency RF techniques were used to lay out the PC board. All of the components and most of the circuit traces are on the top layer. Areas between traces are filled with copper and connect to the bottom layer with a large number of vias.

Conclusion

I hope you find this article enabling by providing you the tools and courage to develop your own communications projects based on these recent RF device technologies. Assembled and tested boards are available for 1975 MHz and 325 MHz for $185. Other frequencies can be made available for additional cost.

Table 1—Loop Parameters

Loop Parameter		*UHF*	*VHF*
Kvco	The tuning rate of the VCO	10 MHz/V	5 MHz/V
KΦ	The phase detector gain constant Check the MFG data sheet	0.159 mA/rad	0.159 mA/rad
RFopt	The output operating frequency	2000 MHz	300 MHz
Fref	The frequency the phase detector sees	100 kHz	100 kHz
N	The division ratio = RFopt/Fref	20000	3000
fp	The loop bandwidth, wider for faster lock times—lower for lower phase noise	1000 Hz	1000 Hz
Φp	Determines the PLL response to step change. Usually set to 45° but can range from 25°-70° if need be.	45°	45°

Table 2—Parallel Port Connections

LPT1 25-pin D	*J4 direct port*	*J1 PIC port*
2	1 Clock	
3	2 Data	1 PCI
11		2 PCO
18	3 Ground	
4	4 Load Enable	
25		3 Ground

You will need a separate cable for each port.

Sources

National Semiconductor Corporation, 2900 Semiconductor Drive, PO Box 58090, Santa Clara, CA 95052-8090. Tel: 800-272-9959 www.nsc.com. Data sheets, samples and application notes are available for the LMX233X PLLatinum family of dual frequency synthesizer integrated circuits.

IC Works, 70 N First Street, Santa Clara, CA. Tel: 408-922-0202. Data sheets, samples, evaluation boards and application notes are available for the WB133X series of synthesizer integrated circuits.

Rakon Limited, 1 Pacific Rise, Mt Wellington, Private Bag 99943, Newmarket, Aukland, New Zealand. Tel: 09-573-5554. From the US dial 011-649-573-5554. High-quality low-cost TCXO available in a number of standard frequencies will supply TCXOs in single quantities for radio amateurs and experimenters.

NE68033 transistors and UPC2709 MMIC are manufactured by NEC. Their US agent is California Eastern Laboratories, 4590 Patrick Henry Drive, Santa Clara, CA 95054. Tel: 408-988-7846.

Digi-Key Corporation, 701 Brooks Avenue South, PO Box 677, Thief River Falls, MN 56701-0677. Tel: 1-800-344-4539. No minimum purchase required, service charge on orders under $25.

Parallax, Inc, 3805 Atherton Road #102, Rocklin, CA 95765. Tel: 916-624-8333. BASIC Stamp microprocessor modules. At the time of this writing the BS1-1C module was available for $34 from Parallax, Inc. BBS tel: 916-624-7107.

Pico-Farad Corp, 237-D N Euclid Ave, Anaheim, CA 92801. Tel: 714-553 3880. Ceramic dielectric resonators. Samples and small purchase orders can be obtained from Val Jackson & Associates, Inc (KD6KXA), 12 Willis Road, Scotts Valley, CA 95066. Tel: 408-438-5442.

CAPAX Technologies, Inc. Hi-Q porcelain capacitors used for the tuning elements of the VCOs. Tel: 805-257-7666.

SuperStar linear simulator is available from Eagleware Corporation, 1750 Mountain Glen, Stone Mountain, GA 30087. Tel: 404-939-0156. *Oscillator Design & Computer Simulation* (Note 5) text book includes *Star* 2.0.

ARRL Radio Designer 1.0 is available from ARRL 225 Main St, Newington, CT 06111-1994. Tel: 860-594-0200. Order No. 4882 $150. *This is probably the best bargain in CAE ever.*

Notes

[1]Price, Harold E., NK6K, Digital Communications, *QEX*, August 1995.

[2]Vendelin, George D., *Microwave Circuit Design*, John Wiley & Sons, 1990.

[3]Rohde, Ulrich L., *Digital PLL Frequency Synthesizers Theory and Design*, Prentice-Hall Inc, 1983. (Out of print. See also: Rohde, Ulrich L., KA2WEU, "Designing Low-Phase-Noise Oscillators," *QEX*, October 1994, pp 3-12.

[4]Savaria, Sylvain and Champagne, Patrick, "Linear Simulators for use in Oscillator Design," *Microwave Journal*, May 1995.

[5]Rhea, Randall W., *Oscillator Design and Computer Simulation*, (Second Edition), Eagleware Corporation, Noble Publishing, 1995.

[6]Kelly, Brendan, Evans, Dr. Noel and Burns, Brian, "1.8 GHz Direct Frequency VCO With CAD Assessment," *RF Design*, February 1993.

[7]Henicle, Ed, "VCO Design using Coaxial Resonators," *RF Design*, November 1995.

[8]Best, Roland E., *Phase-Locked Loops Theory, Design, and Applications*, Second Edition, McGraw Hill, Inc, 1992. (This is a classical work.)

[9]Crawford, James A.,*Frequency Synthesizer Design Handbook*, Artech House, Inc, 1994. (Modern and practical.)

[10]Nash, Garth, "Phase-Locked Loop Design Fundamentals," AN-535 Application Note, Motorola Semiconductor Products Inc, Motorola Inc, 1970. (Cornerstone document everyone should have.)

[11]"An Analysis and Performance Evaluation of a Passive Filter Design Technique for Charge Pump Phase-Locked Loops," National Semiconductor Corporation, Wireless Communications Group, 2900 Semiconductor Drive, Santa Clara, CA 95052.

Sample Program

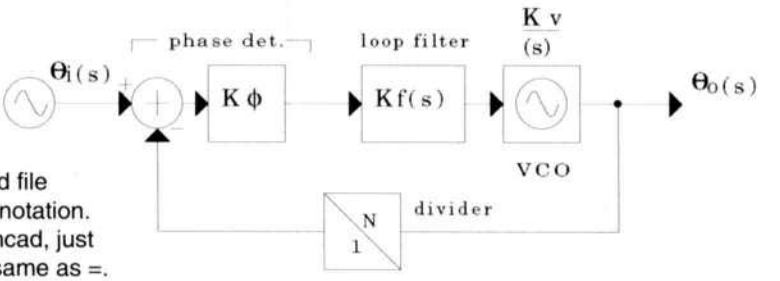
```
'Sample program to be used as a template to program the BASIC Stamp module for the KI6QP dual
'synthesizer module.
'File qp_2g3.bas,  sdr 12/16/95
'This example is for 2000-MHz RF (UHF side) and 300-MHz IF (VHF) side. RF and IF are notation
'conventions of the IC MFGs.

symbol clk          =      0       'The clock pin
symbol data         =      pin1    'The data pin
symbol strb         =      2       'The strobe pin

'You calculate the values for the PLL IF registers and enter them into the template shown below.
'This example is for 300 MHz/100 KHz PD reference. F = [(P*B)+A](Fref/R)

symbol IF_Rprog     =      %01001 'R20..R16; IF program modes, 5-bits.
symbol IF_R         =      100    'R15..R01; IF Reference divider, 15-bits. R = 3 to 32676
symbol IF_Nprog     =      %01    'N20..N19; IF power down(N20) and prescaler (N19)
                                  'N20=0/1 pwrd-up/pwrd-down, N19=0/1 prescaler = 8/16
symbol IF_B         =      187    'N18..N08; IF B counter, 3 to 2047 B>=A.
symbol IF_A         =      8      'N07..N01; IF A counter, 0 to 15.

'You calculate the values for the PLL RF registers and enter them into the template shown below.
'This example is for 2000MHz/100KHz PD reference. F = [(P*B)+A](Fref/R)

symbol RF_Rprog     =      %01001 'R20..R16; RF program modes, 5-bits.
symbol RF_R         =      100    'R15..R01; RF Reference divider, 15-bits. R = 3 to 32676.
symbol RF_Nprog     =      %00    'N20..N19; RF power down(N20) and prescaler (N19)
                                  'N20=0/1 pwrd-up/pwrd-down,  N19=0/1 prescale=64/128
symbol RF_B         =      312    'N18..N08; RF B counter. 3 to 2047 B>=A.
symbol RF_A         =      32     'N07..N01; RF swallow A counter, 0 to 127.

Dirs   =           %11110111     'set input/output pins for direction.
pins   =           %11100000     'preset pin conditions.
'-------------------------------------
'send IF reference programmable modes, R register and control bits
w0     =           IF_Rprog
gosub  send_5
w0     =           IF_R
gosub  send_15
w0     =           %00000000     'register control bits.
gosub  send_2
toggle strb                      'latch enable pulse hi.
toggle strb                      'latch enable pulse low.
'-------------------------------------
'send IF N programmable modes, B register, A register and control bits
w0     =           IF_Nprog
gosub  send_2
w0     =           IF_B
gosub  send_11
w0     =           IF_A
gosub  send_7
w0     =           %00000001     'register control bits.
gosub  send_2
toggle strb                      'latch enable pulse hi.
toggle strb                      'latch enable pulse low.
'-------------------------------------

'send RF reference programmable modes, R register and control bits
w0     =           RF_Rprog
gosub  send_5
w0     =           RF_R
gosub  send_15
w0     =           %00000010     'register control bits.
gosub  send_2
toggle strb                      'latch enable pulse hi.
```

```
toggle strb          'latch enable pulse low.
'-------------------------------------
'send RF N programmable modes, B register,
'A register and control bits
w0      =  RF_Nprog
gosub send_2
w0      =  RF_B
gosub send_11
w0      =  RF_A
gosub send_7
w0      =  %00000011 'register control bits.
gosub send_2
toggle strb          'latch enable pulse hi.
toggle strb          'latch enable pulse low.
'-------------------------------------
'Lock detector, enable amplifiers when locked.
lock:
   input 3
   if pin3= 1 then pwr_on
   high 7            'turn of amplifier VCC.
   toggle 6          'flash the lock LED.
   pause 100         'wait 100mS.
   goto lock         'keep checking.

pwr_on:
   low 7             'turn on amplifier VCC.
   low 6             'turn on lock LED.
   goto lock         'keep checking

'-------------------------------------
send_5:              '5 bit data sender.
   forb10     =  0 to 4
   data       =  bit4
   toggle clk
   toggle clk
   w0         =  w0 * 2
   next
   data       =  0
   return

send_15:             '15 bit data sender.
   forb10     =  0 to 14
   data       =  bit14
   toggle clk
   toggle clk
   w0         =  w0 * 2
   next
   data       =  0
   return

send_2:              '2 bit data sender.
   forb10     =  0 to 1
   data       =  bit1
   toggle clk
   toggle clk
   w0         =  w0 * 2
   next
   data       =  0
   return

send_11:             '11 bit data sender.
   forb10     =  0 to 10
   data       =  bit10
   toggle clk
   toggle clk
   w0         =  w0 * 2
   next
   data       =  0
   return

send_7:              '7 bit data sender.
   forb10     =  0 to 6
   data       =  bit6
   toggle clk
   toggle clk
   w0         =  w0 * 2
   next
   data       =  0
   return
```

Appendix

File: pll_cal2.mcd

This is a template for calculating the component values for a passive loop filter like the ones used on the dual PLL. Select a loop bandwidth, desired phase margin, Kpd, Kv, and N as shown below. This applies to passive loop filters driven by a charge pump phase detector and have the general block diagram form shown here.



This document is a Mathcad file which explains the strange notation. If your not familar with Mathcad, just keep in mind that := is the same as =.

1) Measure the tuning characteristics of the VCO and calculate Kv as shown

$K_{v\_low} := 290 \cdot MHz$ $v\_low := 0.5 \cdot volt$ Low frequency and corresponding tuning voltage

$K_{v\_high} := 310 \cdot MHz$ $v\_high := 4.5 \cdot volt$ High frequency and corresponding tuning voltage

$$K_{vco} := \frac{K_{v\_high} - K_{v\_low}}{v\_high - v\_low} \qquad K_{vco} = 5 \cdot \frac{MHz}{volt}$$ This is the VCO tuning rate constant.

$$K_v := K_{vco} \cdot 2 \cdot \pi \qquad K_v = 3.142 \cdot 10^7 \cdot \frac{rad}{sec \cdot volt}$$ Same as above but in angular form.

2) Determine the phase detector gain constant. This is from the manufacurer's data sheet information.

$K_{dmin} := -1.0 \cdot mA$ $K_{dmax} := 1.0 \cdot mA$ Minimum and maximum current from the phase detector.

$$K_\phi := \frac{K_{dmax} - K_{dmin}}{2 \cdot \pi - (-2 \cdot \pi)} \qquad K_\phi = 0.159 \cdot \frac{mA}{rad}$$ Phase detector gain constant in angular form.

3) Select a loop bandwidth. 200Hz to 2000Hz is a good starting point for most fixed frequency applications using pasive loop filters.

$f_p := 1 \cdot KHz$ $\omega_p := f_p \cdot 2 \cdot \pi \cdot rad$ $\omega_p = 6.283 \cdot 10^3 \cdot \frac{rad}{sec}$

4) Select a reference frequency.

$F_{ref} := 100 \cdot 10^3 \cdot Hz$

5) Calculate the division ratio. In this example the middle of the VCO frequency range is used as the ouput frequency. This is divided by the reference frequency to give the division ratio.

$$N := \frac{(K_{v\_high} + K_{v\_low})}{2 \cdot F_{ref}} \qquad N = 3000$$

6) Select an open loop phase margin. 45 degrees is a good value to start with and should work well.

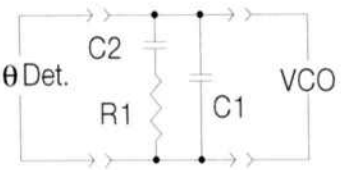
$\phi_p := 45 \cdot deg$

The time constants can now be calculated and the circuit component values derived. $usec := 1 \cdot 10^{-6} \cdot sec$

$$T1 := \frac{\sec(\phi_p) - \tan(\phi_p)}{\omega_p} \qquad T2 := \frac{1}{\omega_p^{\,2} \cdot T1} \qquad T2 = 384.234 \cdot usec$$

$T1 = 65.924 \cdot usec$

This is the relationship of the components in a typical passive loop filter circuit.



$$C1 := \frac{T1}{T2} \cdot \frac{K_\phi \cdot K_v}{\omega_p^{\,2} \cdot N} \cdot \sqrt{\frac{1 + (\omega_p \cdot T2)^2}{1 + (\omega_p \cdot T1)^2}}$$

$$C2 := C1 \cdot \left(\frac{T2}{T1} - 1\right) \qquad R1 := \frac{T2}{C2}$$

Here are the values for this design:

$C1 = 0.017 \cdot \mu F$

$C2 = 0.084 \cdot \mu F$

$R1 = 4.551 \cdot K\Omega$

$i := 20, 21 .. 80 \qquad f_i := 10^{\frac{i}{20}}$

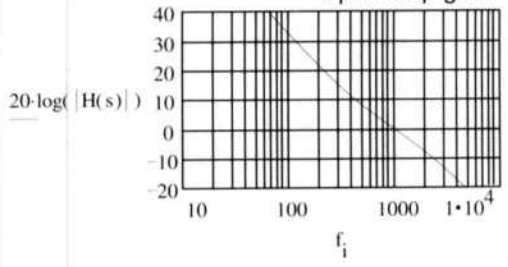
$s_i := j \cdot 2 \cdot \pi \cdot f_i \cdot \frac{rad}{sec}$

$$K_f(s) := \frac{s_i \cdot T2 + 1}{s_i \cdot (C1 + C2) \cdot (s_i \cdot T1 + 1)}$$ Loop filter function

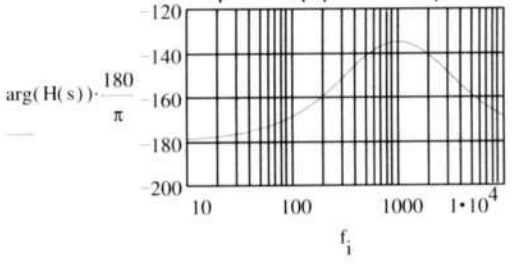
$$H(s) := \frac{K_\phi \cdot K_v \cdot K_f(s)}{N \cdot s_i}$$ Open loop gain

$$\theta(s) := \frac{K_\phi \cdot K_v \cdot K_f(s)}{s_i \cdot (1 + H(s))}$$ Closed loop response

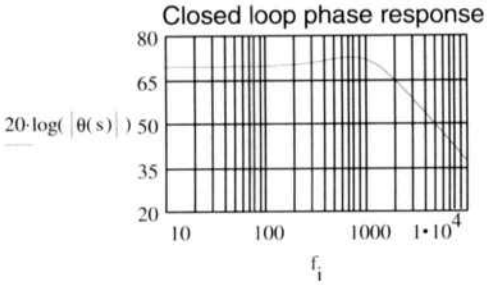
Open loop gain



Open loop phase response



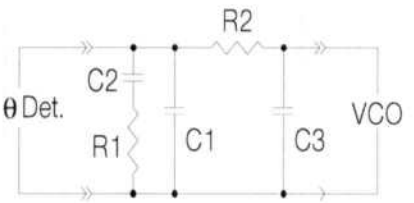
Closed loop phase response



Additional reference spur suppression can be obtained with the third order loop filter. In this case, start by selecting the desired spur attenuation and select a value for C3. C3 should be less than C1/10.

$$\text{Atten} := 20 \qquad C3 := \frac{C1}{10} \qquad C3 = 1.749 \cdot 10^{3} \cdot \text{pF}$$

$$T3 := \sqrt{\frac{10^{\left(\frac{\text{Atten}}{20}\right)} - 1}{\left(2 \cdot \pi \cdot F_{ref}\right)^{2}}}$$



This is the third order filter configuration

The new loop bandwidth

$$\omega_c := \frac{\tan\left(\phi_p\right) \cdot (T1 + T3)}{(T1 + T3)^2 + T1 \cdot T3} \cdot \left[\sqrt{1 + \frac{(T1 + T3)^2 + T1 \cdot T3}{\left[\tan\left(\phi_p\right) \cdot (T1 + T3)\right]^2}} - 1 \right] \qquad \omega_c = 5.806 \cdot 10^{3} \cdot \frac{\text{rad}}{\text{sec}}$$

$$f_c := \frac{\omega_c}{2 \cdot \pi} \qquad f_c = 924 \cdot \text{Hz} \qquad \frac{f_c}{f_p} = 0.924$$

Reducion in loop bandwidth due to additional pole

$$T2 := \frac{1}{\omega_c^{\,2} \cdot (T1 + T3)} \qquad T2 = 419.639 \cdot \text{usec}$$

$$C1 := \frac{T1}{T2} \cdot \frac{K_\phi \cdot K_v}{\omega_c^{\,2} \cdot N} \cdot \sqrt{\frac{1 + \left(\omega_c \cdot T2\right)^2}{\left[1 + \left(\omega_c \cdot T1\right)^2\right] \cdot \left[1 + \left(\omega_c \cdot T3\right)^2\right]}} \qquad C2 := C1 \cdot \left(\frac{T2}{T1} - 1\right) \qquad R1 := \frac{T2}{C2} \qquad R2 := \frac{T3}{C3}$$

Here are the new filter circuit component values

$C1 = 0.019 \cdot \mu\text{F}$ $R1 = 4.095 \cdot \text{K}\Omega$

$C2 = 0.102 \cdot \mu\text{F}$ $R2 = 2.73 \cdot \text{K}\Omega$

$C3 = 0.002 \cdot \mu\text{F}$

In order to get a more precise prediction of the loop response, the phase detector sampling delay must be considered.

$$K_{\phi\Delta}(s) := K_\phi \cdot e^{-\left(\frac{s_i}{2 \cdot F_{ref}}\right)}$$

$$K_f(s) := \frac{s_i \cdot T2 + 1}{s_i \cdot C1 \cdot \left(s_i \cdot T3 + 1\right) \cdot \left(s_i \cdot T2 + 1\right) + s_i \cdot C2 \cdot \left(s_i \cdot T3 + 1\right) + s_i \cdot C3 \cdot \left(s_i \cdot T2 + 1\right)}$$

Loop filter

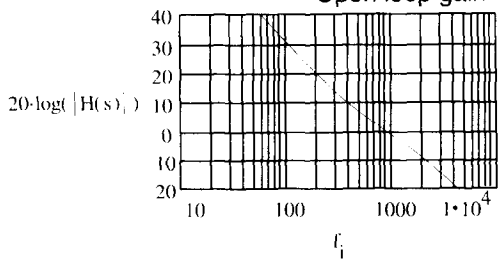
$$H(s) := \frac{K_{\phi\Delta}(s) \cdot K_v \cdot K_f(s)}{s_i \cdot N}$$

Open loop gain

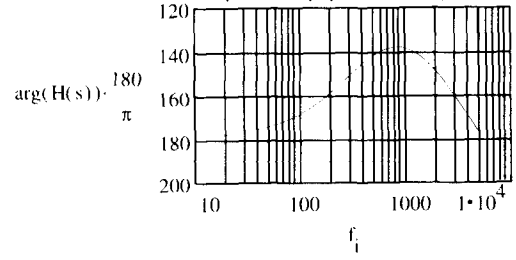
$$\theta(s) := \frac{K_{\phi\Delta}(s) \cdot K_v \cdot K_f(s)}{s_i \cdot (1 + H(s))}$$

Closed loop phase response

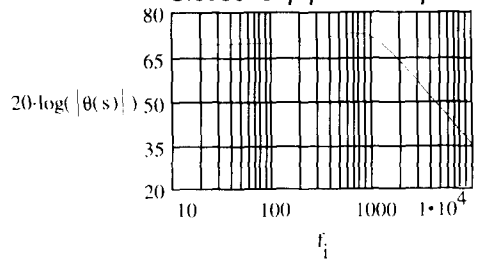
Open loop gain



Open loop phase response



Closed loop phase response



A Simple CW Transmit VFO for the DDC-Based Receiver

Adding a DDS chip to the receiver produces a VFO usable for transceive operation.

By Peter Traneus Anderson, KC1HR

I have been restoring an old Heathkit DX-40 transmitter for CW operation. The DX-40 uses either crystal control or an external VFO, and it occurred to me that an Analog Devices AD7008 single-chip direct digital synthesizer (DDS) would make a dandy VFO for the DX-40. By adding three integrated circuits to my DDC-based receiver, I am able to have a software-controlled VFO running in transceive mode with the receiver.[1,2] Previous articles in *QST* and *QEX* have also described simple DDS circuits for VFO applications.[3,4]

Like many older CW/AM transmitters, the DX-40 uses the VFO fundamental frequency on the 80- and 40-m bands and harmonics of the VFO for higher bands. I am interested in the 80- and 40-m bands only (for now), so the VFO frequency simply equals the transmit frequency.

The DDS is controlled from the printer port of an IBM-compatible PC along with the rest of the receiver. Fig 1 shows the DDS, U31, and its associated circuitry. Fig 2 shows the interface circuitry, which has been slightly modified from that of the original receiver. These figures alone are enough to build a stand-alone VFO. Refer to Notes 1 and 2 for details on the rest of the receiver. Also slightly modified is the control software, shown in Listing 1.

The AD7008 is a complete DDS containing both the necessary digital circuitry and a 10-bit digital-to-analog converter (DAC). As used here, external circuitry provides a 50-MHz clock, the configuration bits in serial form and an output amplifier.

The four NOR gates of U30 in Fig 1, a 74LS02, convert the control signals from the PC printer-port interface (Fig 2) to the form needed by the DDS. NOR gate U30B inverts the $\overline{\text{RESET}}$ signal to give an active-high RESET signal to the DDS. When signal LDDS is low, NOR gate U30A clocks serial data into the DDS serial register. When signal LDDS is high, NOR gate U30D loads the DDS frequency register from the DDS serial register.

The DDS has a parallel input port that is not used here, so all of the parallel input pins are grounded. For an example of the use of the parallel input, see Note 4.

[1]Notes appear on page 00.

990 Pine Street
Burlington, VT 05401
email: traneus@emba.uvm.edu

The DDS provides 1 V p-p of RF into a 50-Ω load. The 120-pF capacitor across the load provides some filtering of high-frequency spurs, but most of the filtering is provided by the tuned circuits in the DX-40.

The LM6181N op-amp, U32, provides a voltage gain of roughly 5, giving about 5 V p-p output. This is enough drive for the DX-40 external VFO input.

I made three changes to the PC control software shown in Listing 1 to support the CW transceive mode. (The original program is shown in Note 2.) First, I eliminated the 1.75-kHz offset between the displayed receiver frequency and the center of the receiver passband. This offset was needed for SSB operation, to make the displayed receiver frequency agree with the conventional SSB carrier frequency. For CW, the incoming keyed signal is normally tuned to the center of the passband, so the displayed receiver frequency should equal the center of the receiver passband.

Second, I added a command to write the current displayed receiver frequency to the DDS. Executing this command sets the transmit carrier to the current displayed receiver frequency, which is the center of the receiver passband. After executing this command, the receiver may be tuned to a different frequency without affecting the DDS; this allows split-frequency operation.

Third, I replaced the "alias" frequency display with a "VFO" frequency display. The alias display is not used for signals in the 80- and 40-m bands. The VFO frequency is displayed since the displayed receive frequency may not be the same as the VFO frequency.

The software normally updates the DDC frequency setting in the receiver each time a command is executed. After initializing data areas, the screen and the receiver, the software waits for a single-character command. When a character comes in, the appropriate case statement is executed, and a serial word is sent to the receiver to update the DDC receiver frequency setting.

The serial word contains 41 bits, and each bit is sent as a single ASCII character: 0, 1, 2, or 4. Data bits are sent as 0 or 1 characters. A 2 character at the start of the word tells the DDC to accept the remainder of the word as a control word. A 4 character late in the word loads the preceding bits of the word into the DDS frequency register.

The word is concatenated from three parts: a five-character preamble, a thirty-character binary number and a six-character postamble. Normally, the preamble and postamble are set to strings that cause the DDC frequency register to be loaded with the binary number.

When the DDS-write command (W) is executed, the preamble and postamble are changed to strings that cause the DDS frequency register to be written instead of the DDC frequency register. The preamble and postamble are then reset to their normal values before the next command is received.

The control software obviously can be given many more features, such as memories, multiple VFOs and frequency offsets.

For example, the frequency setting of the Collins 32S-1 transmitter is derived from a mechanically-tuned VFO, a switch-selectable crystal oscillator and an SSB/CW signal generated at a fixed carrier frequency of 455 kHz. A pair of DDSs could be used: one to replace the VFO and the other to re-

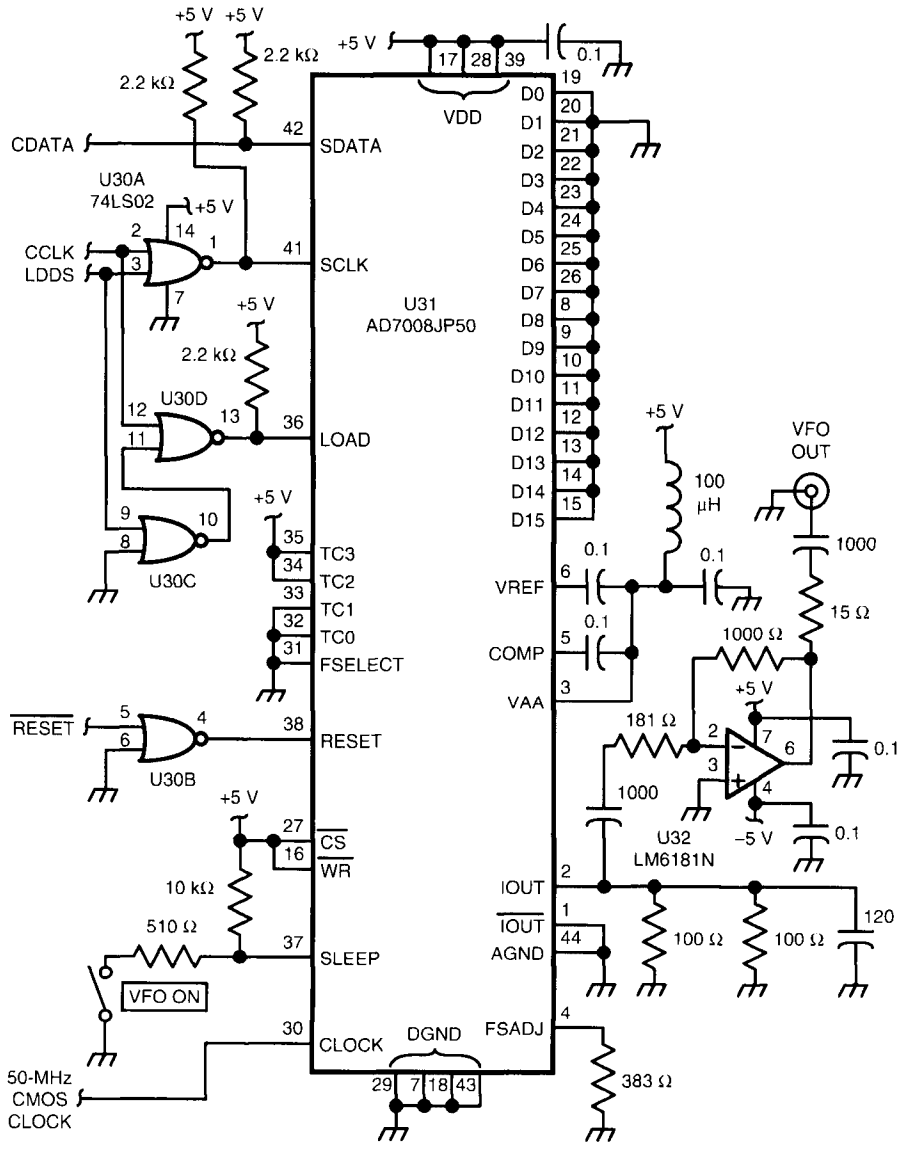


Fig 1—DDS with serial control interface and output amplifier.
U30—74LS02 quad NOR gate.
U31—Analog Devices AD7008JP50 direct digital synthesizer.
U32—LM6181N 100-MHz current-feedback amplifier.

place the crystal oscillator. Alternatively, a single DDS could replace the crystal oscillator, with a fixed-frequency source (counted down from the DDS clock) replacing the VFO. The arithmetic needed to program the DDSs is trivial to do in software.

To build a digital SSB generator would require a digital upconverter (DUC), which accepts a digital audio signal and outputs an SSB RF signal. A DUC converts the incoming audio to baseband I (in-phase) and Q (quadrature-phase) signals, increases the sample rate of the I and Q signals to an RF clock rate and then modulates an RF carrier with I and Q.

The receiver's DDC can be programmed to convert audio input to baseband I and Q. The AD7008 has the circuitry needed (but not used in this CW VFO application) to modulate the RF signal with I and Q. The missing piece is the sampling-rate increaser. This is not yet available in monolithic form but can be built from a dozen or so standard digital parts.

One advantage of a digital SSB generator is that it is easy to digitally predistort the generated SSB signal to compensate for the nonlinearities common in modern solid-state RF power amplifiers.

With the addition of a DDS, the digital receiver moves one step closer to being a digital transceiver. As always, this is a beginning for others to improve upon.

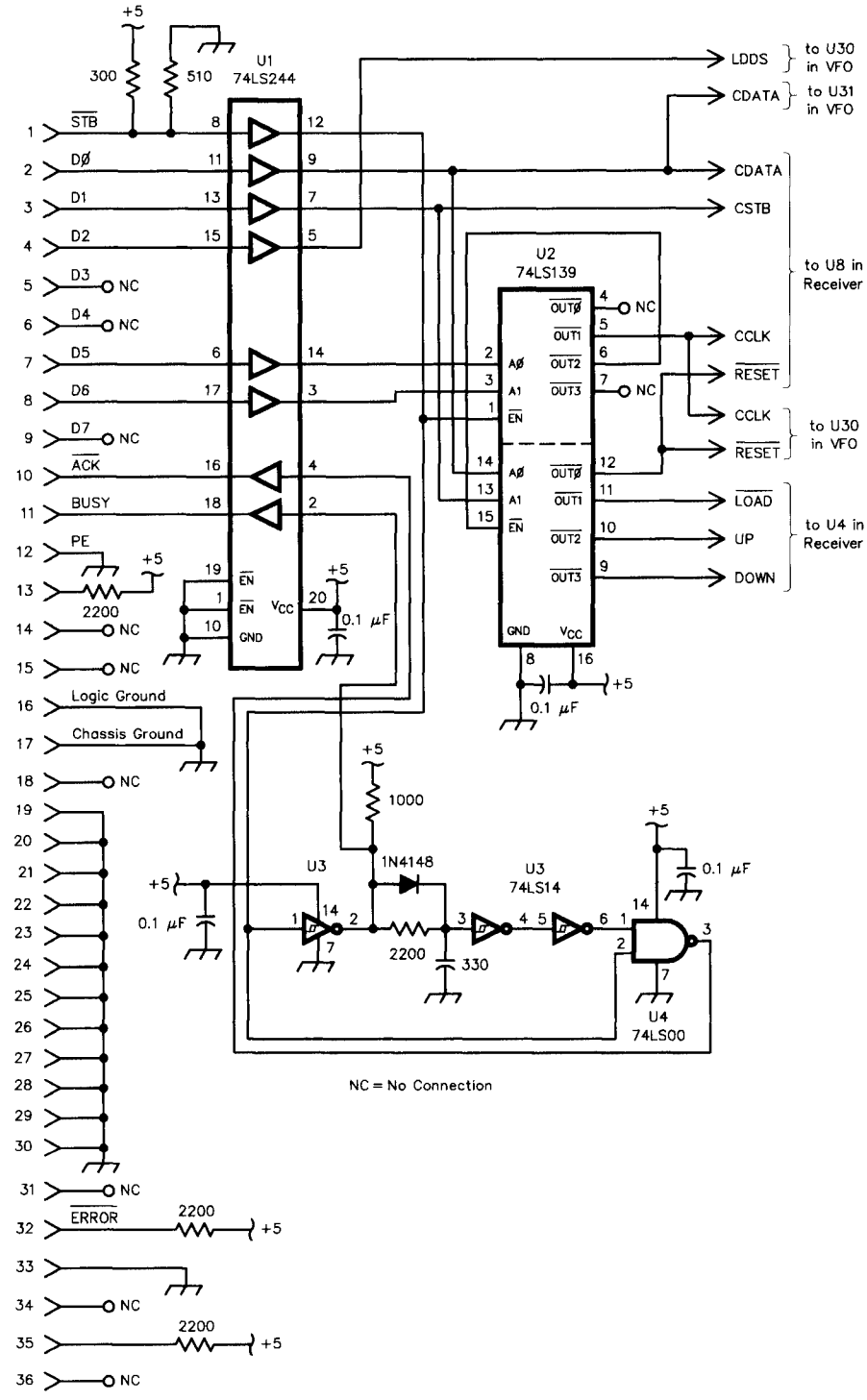


Fig 2—PC printer-port interface. This is the same as Fig 3 in Note 1, except that outputs for the DDS have been added. Signals labeled as going to U8 and U4 control the receiver.

Notes

[1]Anderson, P. T., "A Simple SSB Receiver Using a Digital Down Converter," *QEX*, Mar 1994, pp 3-7.

[2]Anderson, P. T., "A Better A/D and Software for the DDC-Based Receiver," *QEX*, Nov 1994, pp 11-15.

[3]Ahrens, T. and Gray, R., "A Microprocessor-Controlled Multiband Transceiver," *QEX*, Dec 1995, pp 3-13.

[4]Craswell, J., "Weekend DigiVFO," *QST*, May 1995, pp 30-32.

Listing 1—Control Software for the Receiver and VFO

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
/* PC control program for 50016 receiver @ 25 MHz and 7008 VFO @ 50 MHz*/
/* 16jul95 PTAnderson KC1HR */

main()
{
   int tempint = 0;
   int gain_state = 0;
   int j = 0;
   int c = 'b';
   int exit_char = '!';
   char preamble[] = "20011";
   char postamble[] = "000001";
   char ph_inc_string[] = "20011011001100101110101010001011110010001";
   long maxfreq = 12000000;
   long freq = 3700000;
   float tempfloat = 0.0;
   double dfreq = 0.0;
   double rxkhz = 0.0;
   double txkhz = 0.0;
   double dph_inc = 0.0;
   long ph_inc = 0;
   double fclock = 25000000.0;
   double two_up32_over_fclock = 0.0;
   two_up32_over_fclock = 4294967296.0 / fclock;

/* initialize receiver to minimum gain */
   fprintf(stdprn,"@A\n");

/* initialize control registers 2 and 3 (same data for SSB or CW) */
   fprintf(stdprn,"2010000000000000000000000000000000000000\n");
   fprintf(stdprn,"2011000000000000000000000000000000000000\n");

/* print help table */
   printf("\nSimple CW transeceiver for 160/80/40/30 meter bands");
   printf("\n\ncenter of RX passband = RX frequency");
   printf("\n\nexit: !      set DDC clk kHz: #      set RX frequency kHz: =");
   printf("\n\n    set TX vfo = RX frequency: W");
   printf("\n\nup RX frequency:");
   printf("\n\n q=1MHz w=100kHz e=10kHz r=5kHz t=1kHz y=100Hz u=10Hz i=1Hz");
   printf("\n\ndown RX frequency:");
   printf("\n\n  a=1MHz s=100kHz d=10kHz f=5kHz g=1kHz h=100Hz j=10Hz k=1Hz");
   printf("\n\ngain:                                    bandwidth Hz:");
   printf("\n\n   z=up   x=dn     c=min           b=2000 n=400  m=110");
   printf("\n\nlast command, gain, TX kHz, RX kHz now are:\n\n");

/* while not exit char */
   while(c-exit_char){

     /* set preamble and postamble to update 50016 phase increment */
     strcpy( preamble,"20011" );
     strcpy( postamble,"000001" );
     switch (c)
     {
       case 'z':            /* up gain */
  if( gain_state > 14 )
    gain_state = 15;
  else
  {
    gain_state = gain_state + 1;
    fprintf(stdprn,"C");
  }
  break;
       case 'x':            /* dn gain */
  if ( gain_state < 2 )
  {
```

```
  gain_state = 0;
  fprintf(stdprn,"A");
}
else
{
  gain_state = gain_state - 1;
  fprintf(stdprn,"B");
}
break;
     case 'c':            /* min gain */
gain_state = 0;
fprintf(stdprn,"A");
break;
     case '=':            /* set freq in kHz*/
printf("\rfreq kHz ");
cscanf( "%f", &tempfloat );
freq = 1000.0 * tempfloat;
tempint = getch();
break;
     case '#':            /* set fclock in kHz */
printf("\rclock kHz ");
cscanf( "%f", &tempfloat );
fclock = 1000.0 * tempfloat;
two_up32_over_fclock = 4294967296.0 / fclock;
tempint = getch();
break;
     case 'b':            /* 2000 Hz bandwidth */
       fprintf(stdprn,"2100000001000000000000000000000000101001\n");
       fprintf(stdprn,"2101000001101111100111111101110001100101\n");
       fprintf(stdprn,"2110000000001110101010010010000001011010\n");
break;
     case 'n':            /* 400 Hz bandwidth */
fprintf(stdprn,"2100000001000000000000000000000000010011\n");
fprintf(stdprn,"2101001000101110000010100110011010010101\n");
fprintf(stdprn,"2110000000001110101010010010000011001000\n");
break;
     case 'm':            /* 110 Hz bandwidth */
fprintf(stdprn,"2100000001000000000000000000000000000001\n");
fprintf(stdprn,"2101011111111111111111000000000000000101\n");
fprintf(stdprn,"2110000000001110101010010010001101000111\n");
break;
     case 'W':            /* TX frequency = RX frequency */
txkhz = rxkhz;
/* set preamble and postamble to update 7008 DDS phase increment */
strcpy( preamble,"00000" );
strcpy( postamble,"400000" );
break;
     case 'i':            /* up 1Hz */
freq = freq + 1 ;
break;
     case 'k':            /* dn 1Hz */
freq = freq - 1 ;
break;
     case 'u':            /* up 10Hz */
freq = freq + 10 ;
break;
     case 'j':            /* dn 10Hz */
freq = freq - 10 ;
break;
     case 'y':            /* up 100Hz */
freq = freq + 100 ;
break;
     case 'h':            /* dn 100Hz */
freq = freq - 100 ;
break;
     case 't':            /* up 1kHz */
freq = freq + 1000 ;
break;
     case 'g':            /* dn 1kHz */
```

```
freq = freq - 1000 ;
break;
     case 'r':              /* up 5kHz */
freq = freq + 5000 ;
break;
     case 'f':              /* dn 5kHz */
freq = freq - 5000 ;
break;
     case 'e':              /* up 10kHz */
freq = freq + 10000 ;
break;
     case 'd':              /* dn 10kHz */
freq = freq - 10000 ;
break;
     case 'w':              /* up 100kHz */
freq = freq + 100000 ;
break;
     case 's':              /* dn 100kHz */
freq = freq - 100000 ;
break;
     case 'q':              /* up 1MHz */
freq = freq + 1000000 ;
break;
     case 'a':              /* dn 1MHz */
freq = freq - 1000000 ;
break;
     default:
       break;
   }
   if ( freq > maxfreq )
     freq = maxfreq;
   if ( freq < 10000 )
     freq = 10000;
   dfreq = freq;
   rxkhz = 0.001*dfreq;
   dph_inc = dfreq * two_up32_over_fclock;
   ph_inc = dph_inc;
   dph_inc = ph_inc;
   printf("\r%c %2.2u %9.3f %9.3f          ", c, gain_state, txkhz, rxkhz);
   strcpy( ph_inc_string, preamble );
   /* convert thirty bits of ph_inc to ASCII string */
   for( j = 0; j < 30; j = j + 1 )
   {
     ph_inc = ph_inc << 1;
     if( ph_inc < 0 )
strcat( ph_inc_string, "1" );
     else
strcat( ph_inc_string, "0" );
   }
   strcat( ph_inc_string, postamble );
   /* send forty-one-character string to transceiver on printer port */
   for( j = 0; j < 41; j = j + 1 )
   {
     fprintf(stdprn,"%c", ph_inc_string[j]);
   }
   /* send newline characters to keep printer happy */
   fprintf(stdprn,"\n");
   c = getch();
  }
}
```

RF

By Zack Lau, KH6CP/1

Figuring Out the Range of Matching Networks

It's relatively easy to design a matching network that matches two known impedances, and much has been written to show equations that solve particular matching networks. It's more difficult to solve the inverse problem: what *range* of impedances can a particular circuit match. You'll find little in the literature that addresses this problem. In this column, I'll show you how to go about solving the problem and give a couple of graphs that indicate what the popular T and L matching networks will do.

Step one is to describe the situation mathematically. Fortunately, this is rather straightforward and easy, almost to the point of making mistakes embarrassing rather than inevitable. From the simple L-network of Fig 1, solve for the effective source impedance (the impedance Z_S, as transformed by the network) by merely combining the source, inductor and capacitor impedances in series and parallel. And assume that everything is conjugately matched. That is, looking back into the network from the load, we expect to see the complex conjugate of the load impedance. Since the complex conjugate is just a complicated way of saying the reactances cancel out, merely changing the calculated R+jX of the effective source impedance to R–jX gives us our answer. For this simple network, with a resistive source impedance, R_S, the effective source impedance is:

$$Z = (R_S + jX_L) \parallel -jX_C$$

$$= \frac{X_L X_C - jX_C R_S}{R_S + j(X_L - X_C)}$$

Using this equation, you can find the impedance matched by any combination of R_S, X_L and X_C. The tough part is converting the equation into something more useful, like a graph that shows the matched impedance across ranges of L and C values. That will let you show the tuning range of the network. I don't know about you, but I find graphing 500 points by hand awfully tedious. Fortunately, there are tools like *Mathcad* that ease the task. Instead of the tedium of hand graphing, you only have to figure out precisely what your graphing program does, lest you generate graphs that don't make any sense.

Graphing lines is pretty easy—you know what a line looks like so you just calculate two points and draw a straight line. Any extra points are just insurance against mistakes. But what if you don't know what the graph is supposed to look like? How many points do you need? If the points are too far apart, you might miss important details altogether, and if you try to draw the points too close together, you might run into program or computer limitations. If you studied calculus—and actually remember what you studied—the answer is obvious. Calculus lets you figure out where the important points of the graph are, so you don't have to blindly run numbers through equations, hoping you don't miss something important. But we'll skip the calculus and use our computer tools instead.

225 Main Street
Newington, CT 06111
email: zlau@arrl.org

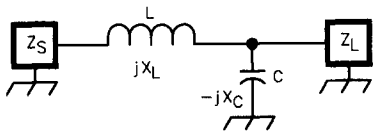


Fig 1—Schematic diagram of an L-network.

There is another "gotcha" with computer graphing. How is the program supposed to know which points to draw lines between? You want to vary more than one parameter, which means you'll need multiple lines. In the circuit of Fig 1, you might want to vary C across its range at an initial value of L, then change the value of L and vary C across its range again. Can you tell your software to draw separate curved lines, or will it try to draw one curved line through all the points? You can get radically different graphs, depending on how you connect the points! One solution is to plot just the points, instead of connecting them together to form lines.

Fig 2 shows a *Mathcad* plot for an L-network using a tapped coil, with the x axis being the effective source resistance and the y axis being the effective source reactance. The stepped inductor action, coupled with lots of points for the capacitor, makes it reasonably obvious how the impedance changes as you tune the matching network. In this example, the inductor varies between 90 nH and 13 µH, with tap points every 3 turns, while the capacitor covers 20 to 930 pF in 2-pF steps. The source resistance is set to 200 Ω.

You can extend this approach to any RLC network. Fig 3 shows a plot of points for a T-network. Here, the equation for the effective source impedance is:

$$Z = (R_S - jX_{C1}) \parallel jX_{L1} - jX_{C2}$$

$$= \frac{jX_{L1}(R_S - jX_{C1})}{R_S + j(X_{L1} - X_{C1})} - jX_{C2}$$

Again, the 13-µH inductor is tapped every 3 turns, but I changed the capacitors to accommodate the program's limitations. C1 covers 15 to 200 pF in 1-pF steps, while C2 covers 15, 70, 125 and 180 pF. You get a good idea of the matching range from the plot, but the effect of the individual components isn't so obvious.

Fig 4 more clearly shows the effect of capacitor C1 on the T-network—it can vary the effective resistance a great deal. Here, L1 and C2 are fixed at 2.25 µH and 100 pF, while C1 is varied between 15 and 200 pF in 1-pF steps.

Fig 5 shows the effect of capacitor C2 on the T-network—it just varies the reactance, as you might expect. You can also infer the action of the inductor via the discrete steps. Here, C1 is fixed at 15 pF, while C2 is varied in 1-pF steps.

You can extend these techniques to any RLC matching network. Modern software tools that calculate and graph equations with complex numbers make finding the matching range of a network possible.

For your convenience, I've placed a number of these *Mathcad* worksheet files into a ZIP file (netrange.zip) that you can retrieve from the ARRL BBS (860-594-0306) or via the Internet from **http://www.arrl.org/files/qex** or **ftp://ftp.arrl.org/pub/qex**. □□

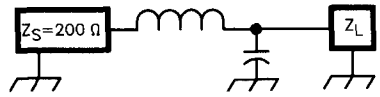
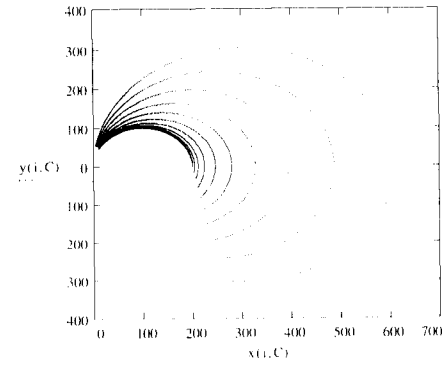


Fig 2—An L-network operating at 3.5 MHz, with a *Mathcad* graph that shows the matching range of the network.



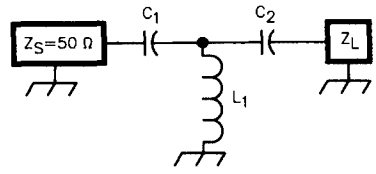
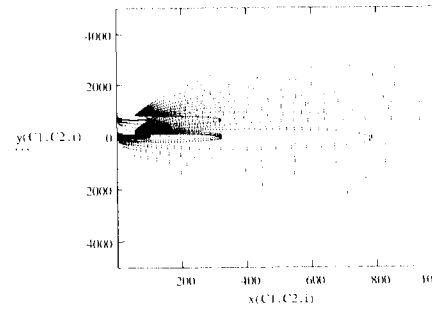


Fig 3—A T-network operating at 14 MHz, with a *Mathcad* graph that shows the matching range of the network.



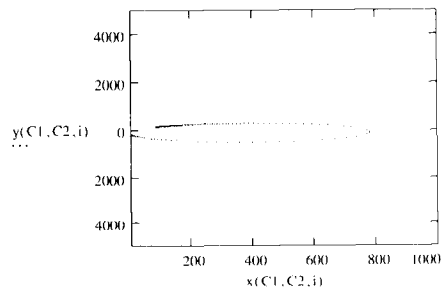


Fig 4—The effect of C1 on the T-network of Fig 3 can be seen by holding L1 and C2 at fixed values and varying C1.

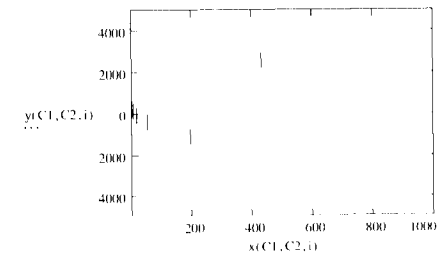


Fig 5—The effect of C2 and L1 on the T-network of Fig 3 is shown by holding C1 at a fixed value.