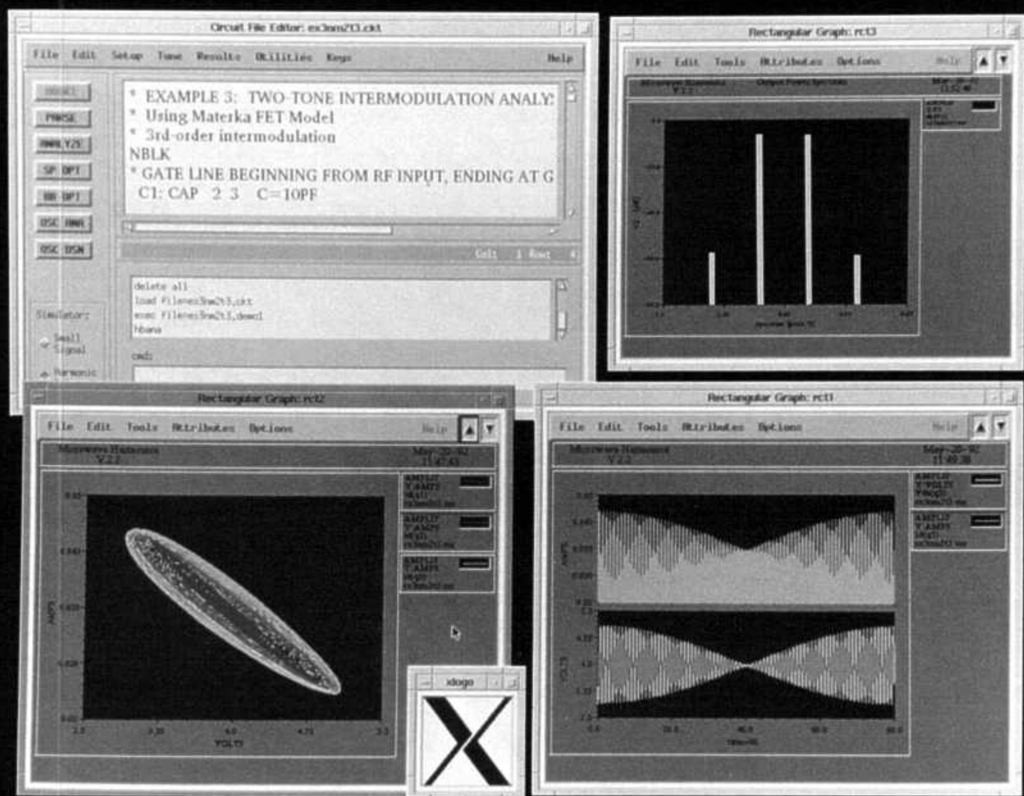
QEX

$1.75

ARRL Experimenter's Exchange

April 1993



Simulating Receiver Performance With CAD Tools

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

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

Offices

225 Main St, Newington, CT 06111 USA
Telephone: 203-666-1541
Telex: 650215-5052 MCI
FAX: 203-665-7531 (24 hour direct line)
Electronic Mail: MCI MAIL ID: 215-5052
Internet:qex@arrl.org

Subscription rate for 12 issues:
In the US by Third Class Mail: ARRL Member $12, nonmember $24;

US, Canada and Mexico by First Class Mail:
ARRL Member $25, nonmember $37;

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.

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

About the Cover:

Simulation of systems is the modern approach to design. Both new and old receiver circuits were simulated by Ulrich Rohde, KA2WEU/DJ2LR/HB9AWE

134

TABLE OF CONTENTS

COLUMNS

APRIL 1993 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: 203-666-1541 Telex: 650215-5052 MCI.
MCI MAIL (electronic mail system) ID: 215-5052
FAX: 203-665-7531 (24-hour direct line)

Canadian membership inquiries and correspondence should be directed to CRRL Headquarters, Box 56, Arva, Ontario NØM 1CO, tel 519-660-1200.

Officers

President: GEORGE S. WILSON III, W4OYI
1649 Griffith Ave, Owensboro, KY 42301

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...

The Right Tool For The Job

In this month's "Digital Communications" column, Harold Price, NK6K, points out that the AX.25 packet radio protocol, for all its pervasiveness, is not the best solution to all amateur digital communications needs. That's not exactly news, even if it does bear repeating. And recently, we've seen some progress in the development of techniques that perform better than AX.25.

Most of the recent work has occurred at HF, where AX.25 is an exceptionally weak protocol. Using AX.25 on a typical HF link is like watching a dog walking on its hind legs: you don't expect it to do it well; you're amazed it can do it at all.

Two-legged creatures like Clover and PACTOR are clearly going to push AX.25 aside for any serious HF data communications. They were designed for the HF environment and it shows. Yet, as NK6K points out, the fact that AX.25 isn't a good choice for some applications—HF foremost among them—doesn't mean it doesn't have its place. It's fine for VHF, for instance... or is it?

At the Tucson Amateur Packet Radio annual meeting on March 7, Phil Karn, KA9Q, gave a brief report on some experimentation he has been doing. He got to thinking about the interference problems encountered on the 70-cm band from pulsed radar systems. The pulses from these systems are relatively infrequent—every several milliseconds, and short, but devastatingly large in amplitude—so much so that overcoming radar interference by "turning up the wick" simply isn't practical in most cases. At fast digital speeds, like 56 kbit/s, these pulses can completely destroy one or more bits of the transmission. That's deadly to an AX.25 system. And the radar signals may continue for minutes—or hours—so retransmissions of AX.25 frames may not solve the problem.

What KA9Q is experimenting with is the use of convolutional codes to perform forward-error correction. The implementation he is testing is most efficient when the error rate is only a percent or two—precisely as is typical of pulsed radar interference. His program, running on a '486 computer, can keep up with a 56-kbit/s data stream under these conditions. This means that systems using no special-purpose hardware are feasible.

One likely application for this kind of system is for use with the Phase 3D satellite under construction by AMSAT. When 70-cm digital downlinks are implemented on this satellite, a technique like that being investigated by KA9Q may allow viable sharing of the band between satellite users and radars. For this application AX.25 need not apply.

Which brings us full circle. As NK6K points out, AX.25 was originally designed with AMSAT Phase 3 satellite use in mind. And it probably will be used with Phase 3—to link users to the satellite gateways, not to link to the satellite itself.

This month in *QEX*

Ulrich Rohde, KA2WEU/DJ2LR/HB9AWE, describes some simulations he performed to analyze several generations of receivers, including tube, bipolar transistor and FET designs. The results may surprise you!

Glitches. Any digital designer knows how annoying they can be. Sure, a $20,000 logic analyzer will help find them, but who has one of those in the basement? As a substitute, try the "Glitch Catcher," by Gary Sutcliffe, W9XT.

From time to time we like to present useful circuits to add to your files. If you "do radio," sooner or later you may want a circuit for a crystal oven. Rather than design one from scratch, just grab this issue of *QEX* from your bookshelf and build Ben Spencer's "Shrike Crystal Oven."

Finally, along with his critique of AX.25, Harold Price, NK6K, complains about some common problems seen in programs today. Even better, he suggests some fixes. Harold takes a look at Ian Wade's new book about KA9Q's NOS TCP/IP program, too.—*KE3Z, email: jbloom@arrl.org (Internet)*

Differences Between Tube-Based and Solid State Based Receiver Systems and Their Evaluation Using CAD

By Dr. Ulrich L. Rohde, KA2WEU/DJ2LR/HB9AWE
52 Hillcrest Road
Upper Saddle River, New Jersey 07458

Introduction

This paper is a summary and excerpt of a presentation of "Overview of State of the Art of Modeling the Dynamic Range of VHF, Microwave and Millimeter Receiver Systems using Computer-Aided Design" given at the VHF Conference, September 19-20, 1992 at Weinheim, Germany. The purpose is to show the advantages which modern solid-state design has over tube design, including some of the weaknesses more inherent in the design than the technology. A good example is that even 30 years ago one was able to build better mixers than usually were used, as seen in some rare examples of receivers.

The rapid advances in the development of semiconductor devices have allowed us to develop transistors which work well into the millimeter wave area. Looking back to 1960, when tubes like the 6CW4 and 417A were the dominant devices for building VHF and UHF receivers, the improvement of noise figure and sensitivity has been dramatic. Many people have wondered whether or not the actual dynamic range, which is the ratio of the maximum input level (close to the 1dB compression point) to the noise floor, really has been improved. In addition, in the "old days," it was necessary to build all circuits prior to being able to evaluate them. Today's technologies allow us to do feasibility studies. As preamplifiers and mixers are part of the chain of the system which largely effects the overall performance, the following modeling capabilities are mandatory:

1. Modeling of noise figure of low-noise amplifiers.
2. Modeling the 3rd-order intercept point of amplifiers.
3. Modeling the insertion gain/loss of mixers, including noise figure.
4. Predicting the phase noise of an oscillator.

I have decided, for historical reasons, to even revisit vacuum tubes to have the same point of reference as to the simulation of the noise figure.

Table 1 is a set of equations for bipolar transistors, tubes, and FETs to calculate the necessary noise parameters.

Modeling of Noise

Fig 1 shows the equivalent noise circuit of a two-port device. The set of equations for this linear system is device independent, but the actual coefficients in the equation reflect the device parameters. Over the last few years, these noise equations have been expressed in z-parameters, y-parameters and, today, s-parameters, for which today's modern measuring equipment uses 50 ohms as a reference using precise terminations. Initially, like in 1960, z-parameters were used, but at higher frequencies there are no "open" circuits because of stray capacitance. From 1965-1970, the so-called y-parameters were considered, which required a short circuit at the output. At high frequencies, short circuits do not exist. Modern transmission lines in micro strip technology with precision 50 ohm terminations allow s-parameters to work up to 100 GHz. A detailed mathematical derivation of this can be found in *Microwave Circuit Design-Using Linear and Nonlinear Techniques*[1].

Modern CAD programs, like SuperCompact/ Microwave Harmonica, take measured values of F_{min},

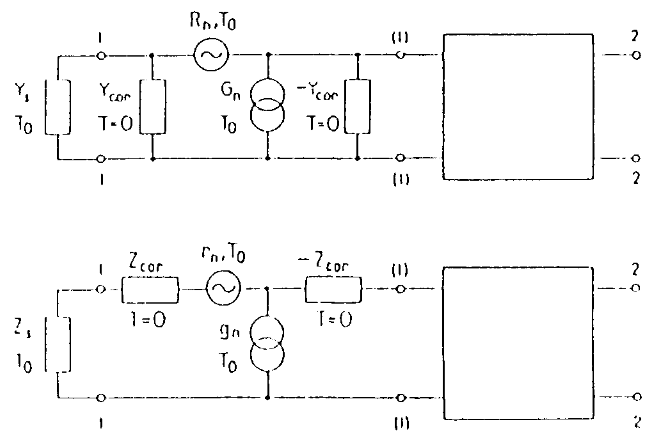


Fig 1—Noisy two port representation in Y and Z matrix form.

G_{opt}, and R_n at one frequency, like 10 GHz, and then by using a de-embedding technique allow accurate noise prediction over a wider frequency range. This is done by solving a set of linear equations and expanding the frequency range from a few hundred MHz to an upper cut-off frequency determined by the validity of the equivalent circuits of the device.

Typical values for R, P and C are:
R = 0.1 1
P = 0.2 3
C = 0.0 1

In many instances, you can assume C to be 0, R to be 0.1 and P proportional to $I_d / I_{dss}{}^2$.

The actual calculation of the noise figure is done by

Table 1

A. Tube Parameters

$$R_n = \frac{3.2}{S} \quad [1]$$

$$R_e(Y_{cor}) = \sqrt{R_e(Y_{11})^2 + 2p \cdot \omega^2 Cg^* k} \quad [2]$$

$$I_m(Y_{cor}) = \frac{2p \cdot f_0 \cdot Cg^* k}{10} \quad [3]$$

$$F_{min} = 1 + 2\sqrt{2} \cdot p \cdot R_n \cdot f_0 \cdot Cg^* k \quad [4]$$

$Cg^* k = \Delta C_{gk}$ from cut-off to bias for normal operation.

C. GaAs FETS

$$F_{min} = 1 + 2\left(A\sqrt{A^2 + A + B}\right) \quad [9]$$

$$R_n = R_g + R_s + \frac{R}{g_m} + \frac{P}{g_m}\left[1 + \left(\omega C_{gs} R_T\right)^2\right] \quad [10]$$

$$R_{s,\,opt} = \frac{1}{\omega C_{gs}}\sqrt{\frac{g_m\left(R_s + R_g\right) + R\left(1 - C^2\right)}{P} + (\omega C R_T)^2} \quad [11]$$

$$X_{s,\,opt} = \frac{1}{\omega C_{gs}}\left(1 + \sqrt{\frac{R}{P}}\right) \quad [12]$$

where

$$R_T = R_g + R_s + R_i \quad [13]$$

and

$$A = \left(\frac{\omega C_{gs}}{g_m}\right)^2 R_T P g_m \quad [14]$$

$$B = \left(\frac{\omega C_{gs}}{g_m}\right)^2 \left[PR\left(1 - C^2\right) - P g_m R_i\right] \quad [15]$$

The R, P and C values are device parameters for noise modeling.

B. Bipolar Parameters

$$F_{min} = \alpha \frac{R_B + R_{OPT}}{R_e} + \left(1 + f\frac{f^2}{f_b^2}\right)\frac{1}{\alpha_0} \quad [5]$$

The optimum source resistance is

$$R_{OPT} = \left\{R_B{}^2 - X^2{}_{OPT} + \left(1 + f\frac{f^2}{f_b^2}\right)\frac{R_e(2R_B + R_e)}{\alpha_0 \alpha}\right\}^{1/2} \quad [6]$$

and optimum source reactance is

$$X_{OPT} = \left(1 + f\frac{f^2}{f_b^2}\right)\frac{2\pi f C_{Te} R_e^2}{\alpha_0 \alpha}$$

where

$$\alpha = \left\{\left(1 + f\frac{f^2}{f_b^2}\right)\left(1 + f\frac{f^2}{f_e^2}\right) - \alpha_0\right\}\frac{1}{\alpha_0} \quad [7]$$

$$R_n = R_b\left(A - \frac{1}{\beta_0}\right) + \frac{R_e}{2}\left[A + \left(\frac{R_b}{R_e}\right)^2\left[1 - \alpha_0 + \left(\frac{f}{f_b}\right)^2 + \left(\frac{1}{\beta_0} - \left(\frac{f}{f_b}\right)\left(\frac{f}{f_e}\right)\right)^2\right]\right] \quad [8]$$

where

$$A = \frac{1 + \left(\frac{f}{f_b}\right)^2}{\alpha_0^2}$$

$Re \approx 26mv/Ie$
Ie=emitter DC current

provides a convenient set of equations for representing the low frequency noise performance of a bipolar transistor. Unlike Fukui's formula, the new expression does not involve the unity current gain frequency f_T;

and f_b denotes the "cutoff" frequency of the common base current gain α\f.

using F_{min}, G_{opt}, and R_n.

The noise factor can now be determined:

$$F = 1 + \frac{G_u}{G_g} + \frac{R_n}{G_g}\left[(G_G + G_{cor})^2 + (B_G + B_{cor})^2\right] \quad [16]$$

$$F = 1 + \frac{R_u}{R_g} + \frac{G_n}{R_g}\left[(R_G + R_{cor})^2 + (X_G + X_{cor})^2\right] \quad [17]$$

The noise factor is a function of various elements, and the optimum impedances for best noise figure can be determined by minimizing F with respect to generator reactance and resistance.

This gives:

$$R_{On} = \sqrt{\frac{R_n}{G_n} + R^2{}_{cor}} \quad [18]$$

$$X_{On} = -X_{cor} \quad [19]$$

and

$$F_{min} = 1 + 2G_nR_{cor} + 2\sqrt{R_uG_n + (G_nR_{cor})^2} \quad [20]$$

In order to distinguish between optimum noise and optimum power, we have introduced the convention On instead of the more familiar abbreviation opt.

At this point we see that the optimum condition for minimum noise figure is not a conjugate power match at the input port. We can explain this by recognizing all noise sources present at the input, not just the thermal noise of the input port. We should observe that the optimum generator susceptance, $-X_{cor}$, will minimize the noise contribution of the two noise generators.

In rearranging for conversion to s-parameters, we write:

$$F = F_{min} + \frac{G_n}{R_G}\left|Z_G - Z_{On}\right|^2 \quad [21]$$

$$F = F_{min} + \frac{R_n}{G_G}\left|Y_G - Y_{On}\right|^2 \quad [22]$$

from the definition of the reflection coefficient,

$$\Gamma_G = \frac{Y_O - Y_G}{Y_O + Y_G} \quad [23]$$

$$r_n = \frac{R_n}{Z_0} \quad [24]$$

the normalized equivalent resistance

$$F = F_{min} + \frac{4r_n\left|\Gamma_G - \Gamma_{on}\right|^2}{\left(1 - \left|\Gamma_G\right|^2\right)\left|1 + \Gamma_{on}\right|^2} \quad [25]$$

$$r_n = (F_{50} - F_{min})\frac{\left|1 + \Gamma_{on}\right|^2}{4\left|\Gamma_{on}\right|^2} \quad [26]$$

$$\Gamma_{on} = \frac{Z_{On} - Z_0}{Z_{On} + Z_0} \quad [27]$$

The noise performance of any linear two-port system can now be determined if the values of the four noise parameters, F_{min}, $r_n = R_n/50$, and T_{On} are known.

The factor of 3.2 in the equation for R_n for tubes (the equivalent noise resistor) has to do with cathode temperature. This factor applies to triodes only. For pentodes and tubes with larger numbers of grids, the value varies between 5 and 7. It can be seen that the source for the noise in the high frequency range is due to effects either from thermal contribution or Schottky noise. The so-

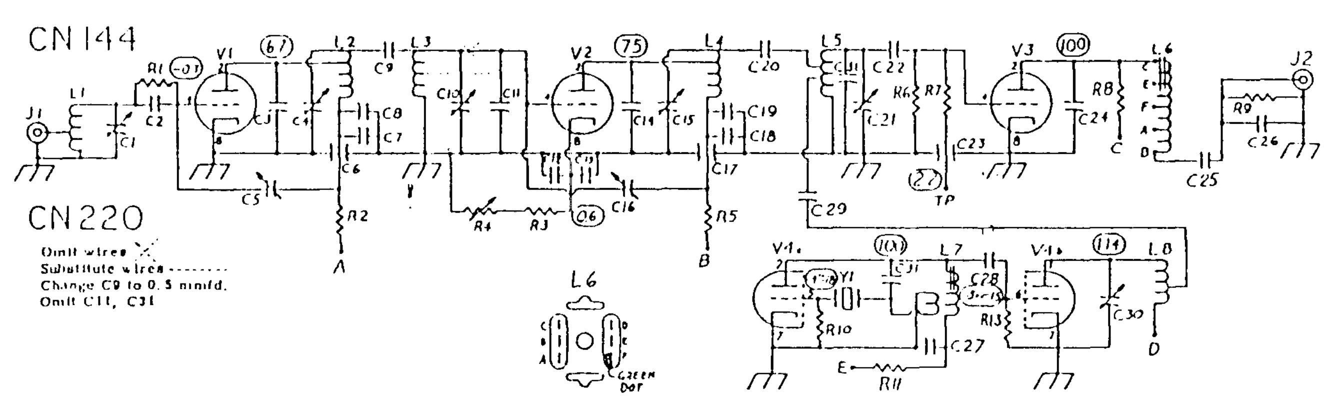


Fig 2—Schematic of Ameco Converters

called equivalent noise resistor determines the minimum noise figure at low frequencies—those above the frequency of flicker noise or other surface-related noises. In its simplest form, the noise factor $F=1+R_n/R_g$. R_n stands for the equivalent noise resistor, and R_g is the value of the generator impedance. The lower the R_n, the lower the minimum noise figure becomes. At higher frequencies, parasitics begin to contribute, particularly the input capacitance and feedback capacitance. The feedback and input reactance determines the correlation

Table 2
Converter Test Data

Ameco		*417A*		*Bipolar-Cascode*		*MOS-FET*		*GaAS-FET*	
Measured	*Calculated*	*Measured*	*Calculated*	*Measured*	*Calculated*	*Measured*	*Calculated*	*Measured*	*Calculated*
P_G = 20 dB	21 dB	P_G = 20 dB	21 dB	P_G = 20 dBm	22 dB	P_G = 20 dB	21 dB	P_G = 20 dB	21 dB
NF = 4 dB	3.8 dB	NF = 1.6 dB	1.7 dB	NF = 0.8 dB	0.79 dB	NF = 0.7 dB	0.65 dB	NF = 0.4 dB	0.42 dB
IP_3 = 0 dBm	1 dBm	IP_3 = 5 dBm	+5.5 dB	IP_3 = 7 dBm	6.5 dBm	IP_3 = 10 dBm	11 dBm	IP_3 = 10 dBm	10.5 dBm
Tube-Mixer		*Tube-Mixer*		*Diode-Ring* + 20 dBm		*Diode-Ring* +20 dBm		*Diode-Ring* +20 dBm	
Q = −4	Q = −2.8	Q = 3.4	Q = 3.8	Q = 6.2	Q = 5.71	Q = 9.3	Q = 10.35	Q = 9.6	Q = 9.58

Q = Figure of Merit = IP_3-NF

In the case of the Bipolar and FET versions, a high-level double balanced mixer has been used. Indications are that the limitations in this design are given by two factors: 1) the IP_3 of the double-balanced mixer and 2) the gain distribution of the system, including the matching. For stability reasons at VHF, MOS FETs can be handled better than GaAs FETs.

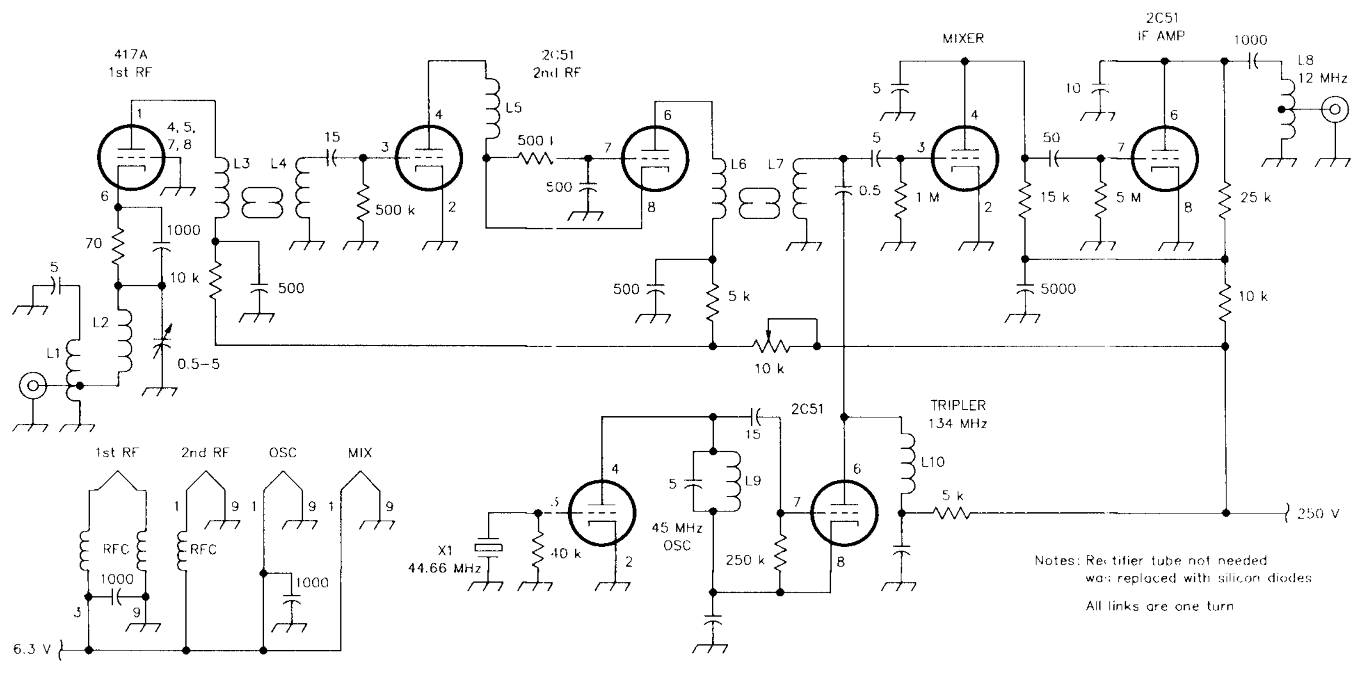


Fig 3—Schematic of 417A Two-Meter Converter.

L1—5 turns, tap at 1 turn, #24 enam, 0.2" L × 0.1" D, brass slug.
L3—3⅔ #24 enam, 0.2" L × 0.1" D, brass slug.
L4—4 #24 enam, 0.2" L × 0.1" D, brass slug.
L6—4⅔ #24 enam, 0.2" L × 0.1" D, brass slug.
L7—4 turns #24 enam, 0.2" L × 0.1" D, brass slug.
L9—9 turns #24 enam, 0.2" L × 0.1" D, brass slug.
L10—5 turns #24 enam, 0.2" L × 0.1" D, brass slug.
L8—37 turns #30 enam, 1⁄16" L × ½" D, tap at 6 turns, ironslug
L2—6 turns #20 enam, ½" × ⅛" D, brass slug
L5—6 turns #20 enam, ¼" × ⅛" D, brass slug
C9—2-3 twists of small-guage hookup wiree.
RFC—1-2 µH RF choke

between the Johnson noise (thermal noise) and the Schottky noise. The Schottky noise is determined by the emission of the device and is equal to 2IQ with I being saturation current and Q the charge of an electron.

The correlation coefficients, described as magnitude and phase, are the combination of the fluctuation of both noise sources. From this introduction, two immediate deductions are possible:

1. Devices with higher gain or higher transconductance have lower noise. Also, devices with smaller input capacitance have lower noise.
2. Any parasitics which cause unwanted feedback, either capacitance or inductance, will change the noise.

As a result, one can compensate the noise at particular frequencies and apply "noise" feedback. This will result in almost simultaneous matching of the operation point for best gain and best noise operation. Low-noise

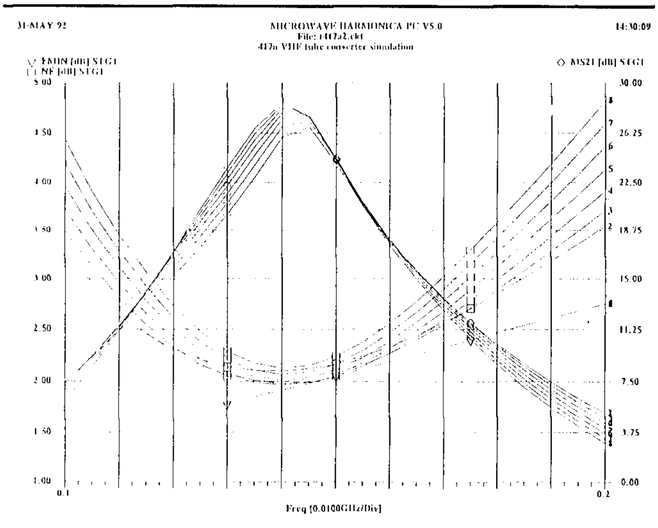


Fig 4—Simulated minimum noise figure. Noise figure and gain as a function of input matching. For the 417A tube, note that F_{min} as per definition is matching independent.

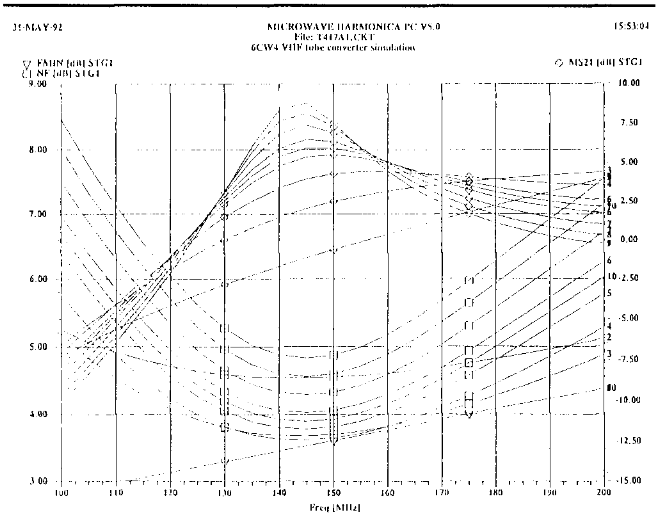


Fig 5—Simulated minimum noise figure. Noise figure and gain for the 6CW4 VHF tube. Due to the significant feedback without neutralization, there is a lot of interaction of noise figure and gain as the input matching changes.

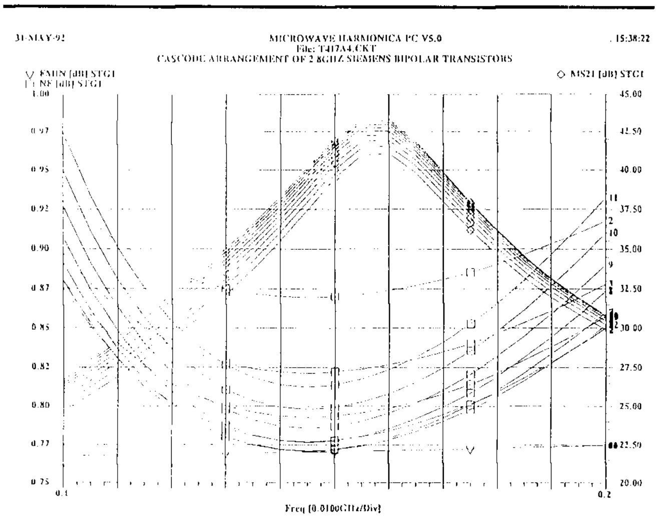


Fig 6—Simulated minimum noise figure. Actual noise figure and gain for bipolar cascode stage. Note that changes of the noise figure in percentage are very small because of the high reverse isolation which is common to all cascode arrangements.

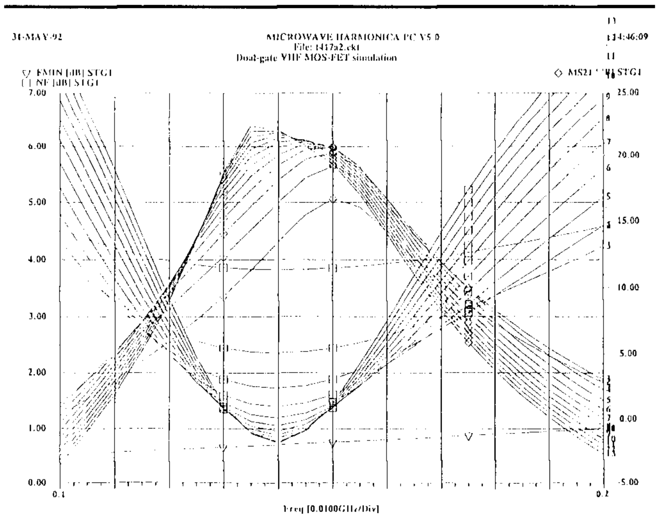


Fig 7—Simulated minimum noise figure. Noise figure and gain of a dual-gate MOS stage. This type of arrangement is much more sensitive to changes in the input match which equally effects noise figure and gain.

operation assumes an operating point in the linear region, but does not address the issue of large signal handling capabilities. An oscillator or mixer is a hybrid in the sense that an active device is being "pumped" with a large ac current or ac voltage swing. As a result, some of the nonlinear parameters are changed as a function of drive level. In the case of a mixer, an external oscillator "pumps" and/or switches the nonlinear devices between two states (on and off) and a mixing between several tones occurs. This type of mixing is referred to as linear mixing, and the basic goal is that the active device is either on or off. Things not being ideal, there are still interactions during both the on and off conditions, particularly during the cross-over time. In the on mode, modulations due to a change in currents can occur, specifically modulating G_m, and in the off condition, due to large voltage swings, the dynamic capacitance will vary significantly. In the case of an oscillator, the active device is operated in the region of negative resistance and will start to oscillate into a resonant circuit which determines the resonant frequency. The purity of the signal depends on the flicker noise contribution, specifically due to ampli-

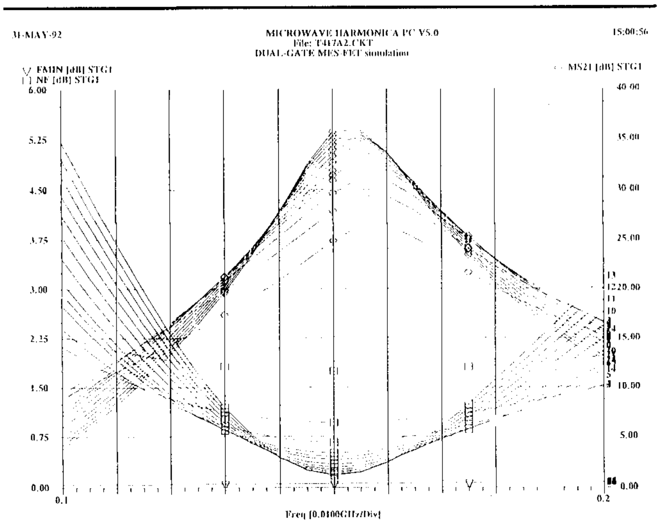


Fig 8—Simulated minimum noise figure. Noise figure and gain as a function of input matching. Dual gate MES FETs are much less sensitive to matching changes.

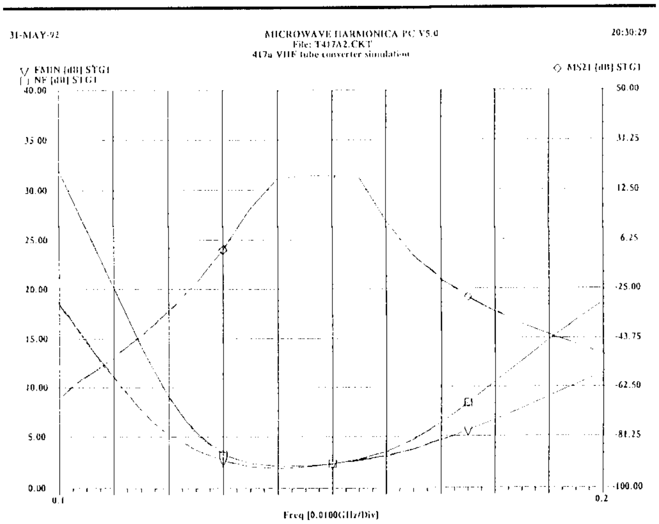


Fig 9—Overall noise figure and gain of the 417A converter. Note that the minimum noise figure and predicted noise figure agree quite well.

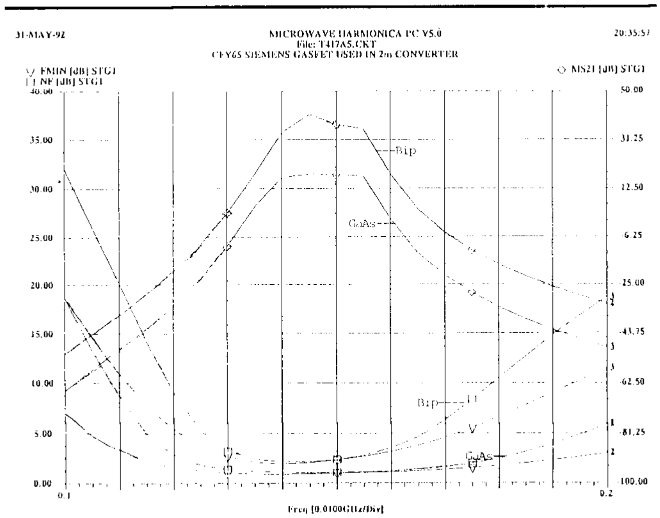


Fig 10—Overall, gain and noise figure of both the bipolar and GaAs FET version of the converter. Due to some feedback phenomena, the bipolar version does not show the same flat performance as the GaAs FET.

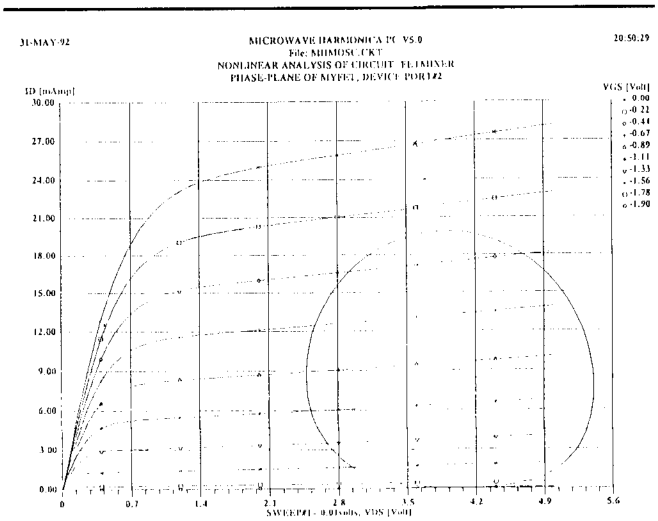


Fig 11—Dynamic load line of a tube mixer where a modified FET model was used to simulate the tube mixer.

tude or phase variations.

Verification Circuits

As promised, we first are taking a step back into history to look at a set of tube converters. These tube converters have been built with either 6CW4 tubes or 417As. Fig 2 shows the schematics of very popular Ameco converters and Fig 3 shows an equally popular 417A two-meter converter.

The reason these tube converters were considered is

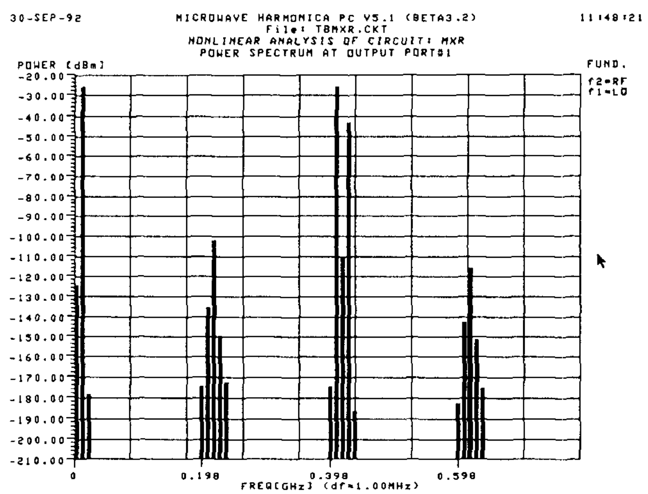


Fig 12—Simulated output of the double balanced mixer shown in Fig 14. Please note the extraordinary suppression of the harmonic frequencies due to this clever symmetrical circuit.

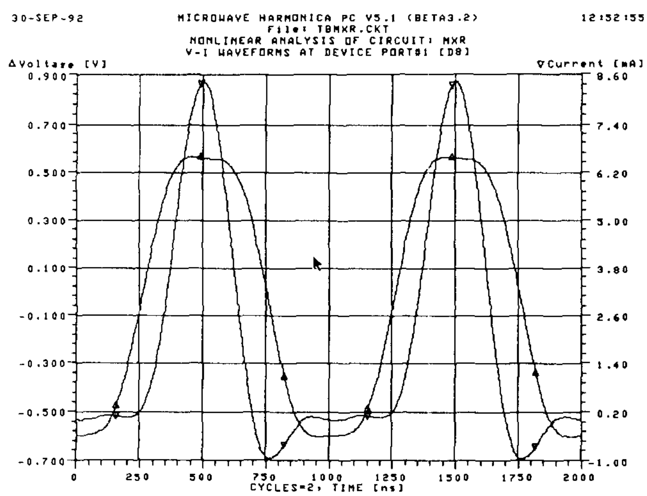


Fig 13—Simulation of currents and voltages inside the double-balanced mixer at one particular diode. The high harmonic content gets canceled because of symmetrical circuit as can be seen from simulation in Fig 12.

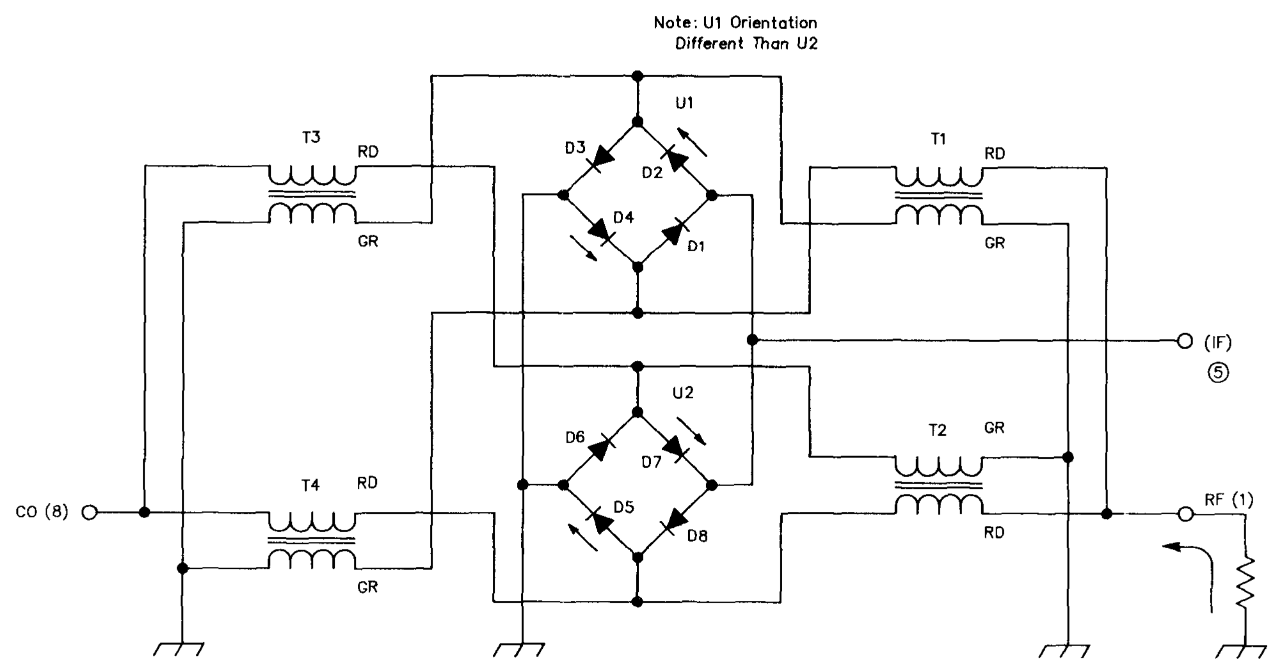


Fig 14—Circuit diagram of a double/doubly balanced mixer. Because of the simple construction which its transformers, this type of mixer provides much higher isolation and only a slight increase of component cost. This mixer shows the best performance in dynamic range.

that I wanted to be able to look at the relationship between modeling and measurement. Table 2 provides a listing of the small- and large-signal performance and indicates the measured performance versus predictions using Compact Software's Microwave Harmonica program for simulation. I then took the basic design shown for the 6CW4 converter and replaced the tubes with bipolar transistors, then MOS transistors, and finally GaAs FETs.

The result of this simulation is also shown here, and the following Figs 4-8 are a set of simulations for the various devices at different current settings for the neutralization.

In the case of the 6CW4, the differences in the noise matching are dramatic, as is the change in gain.

By using a cascode arrangement of Siemens bipolar transistors, I obtained the best noise figure thus far (approximately 0.76 dB), but also obtained the highest gain and the least feedback. This is directly attributable

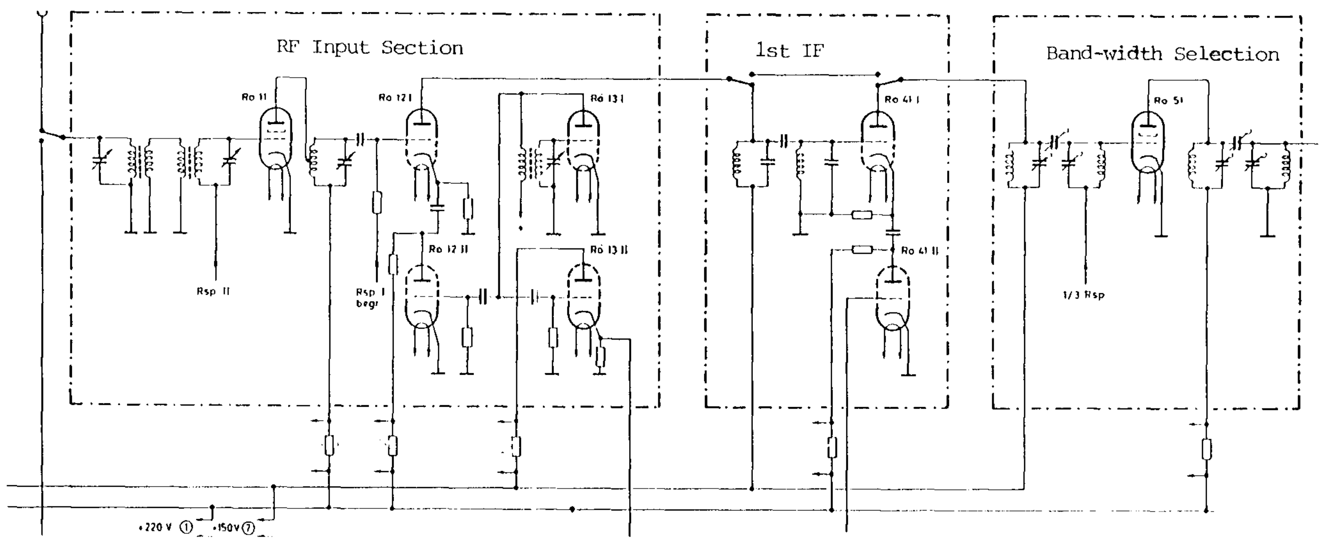


Fig 15—Simplified circuit diagram of the vintage 1960 Rohde & Schwarz EK07 Receiver. It uses a double-tuned input stage, pentode pre-amplifier synthesized local oscillator with a mixer similar to modern dual-gate mixer applications. Note that AGC voltage is even applied to the mixer for high intercept performance.

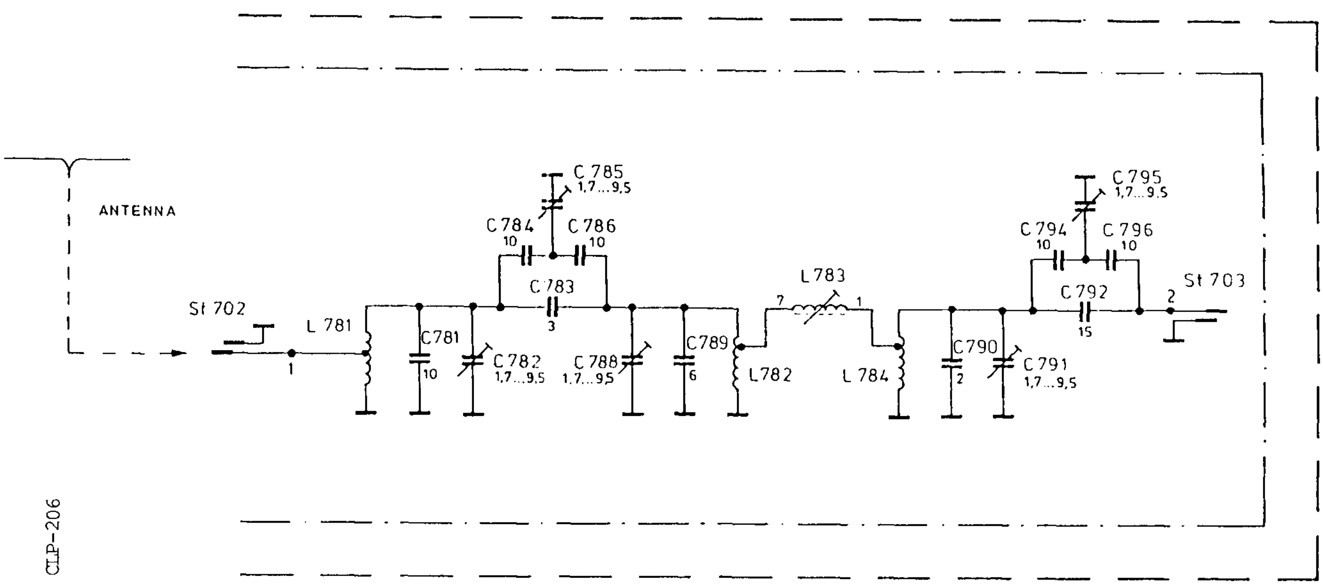
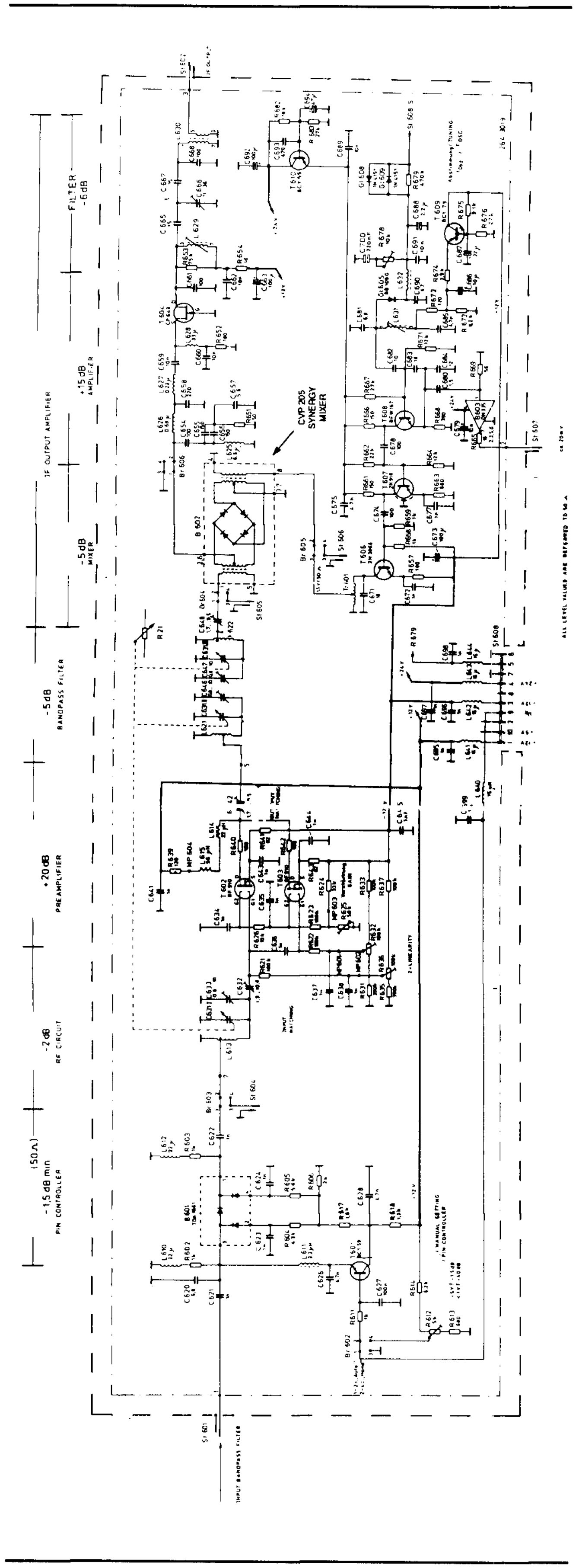


Fig 16—Input band-pass filter for relay receiver.



to the high reverse isolation. If I substituted the bipolar cascode with a dual-gate MOS FET, the resulting noise figure is about the same, but it can be seen from the curves that the selectivity is vastly improved. This means for the same noise figure, more suppression of interference is possible. Finally, if we look at the GaAs FET, the minimum noise figure available is approximately 0.2dB. But this is not quite achieved in the circuit due to input matching losses.

It is also useful to compare the best tube converter with the best GaAs FET converter. By substituting and replacing the tubes with GaAs FETs, a system noise figure of approximately 1 dB is achieved versus 1.6 dB with the tubes, as shown in Figs 9 and 10.

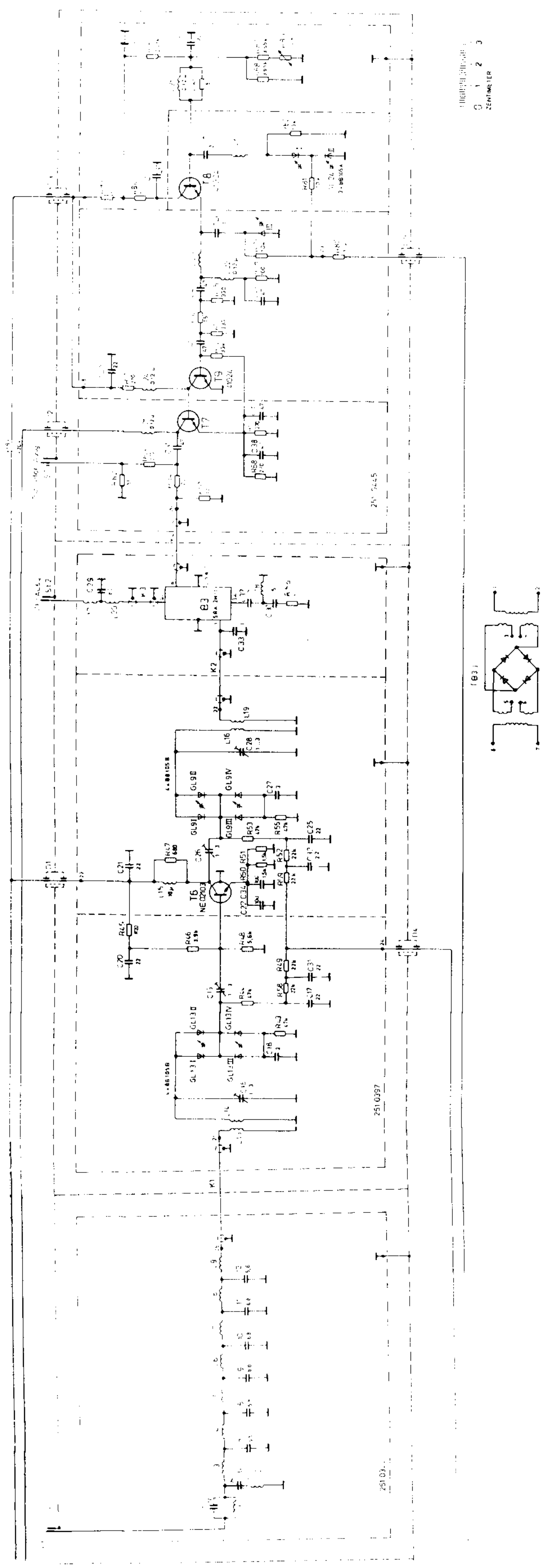
Since the converter typically consists of a pre-amplifier and mixer, we should look into the nonlinearities of the mixer. Fig 11 shows the phase plan of the active (tube) mixer at the output. The plum-shaped curve is due to the LC tuned circuit at the output. The diode mixer (4 high level diodes) is simulated in Fig 12 showing the spurious output, which is lower compared to the single-ended tube mixer of Fig 11.

Fig 13 shows how powerful the diode gating is by analyzing current and voltage of distributed wave forms inside the diode mixer. This high level double/double-balanced mixer, type CPL206 made by Synergy Microwave, is shown in Fig 14. Needless to say, the intermodulation distributed characteristics depend highly on the ferrite material, the exact matching of the transformers and, of course, selection of high-level diode mixers. These mixers typically require +17 to +23 dBm of LO power.

Some practical circuits

Fig 15, shows a simplified schematic of the input stage of a tube circuit based RF input stage, with a double-tuned input stage, the pre-amplifier and triode mixer. The other two triodes are the actual oscillator for the first LO and its driver for the mixer and output stage. The second chamber contains the second mixer and second LO driver, and the last chamber contains a system of switchback filters for different bandwidths. This 1960 design (Rohde & Schwarz EK07) had an intercept point of +30 dBm and used a combination of RF feedback and distributed AGC. The novelty in this design at the time was that even the mixer received AGC voltage and that the triode mixer was linearized by proper

Fig 17—Schematic of input stage of an ultra-high dynamic relay receiver. Following the input filter shown in Fig 16, a PIN diode attenuator is responsible for extending the AGC range. Two dual-gate MOSFETs in parallel are chosen for better simultaneous matching for noise and gain. Each transistor, therefore, contributes to the output current which increases the dynamic range by at least 3 dB and the gain is up by about 6 dB. Sufficient gain is provided to offset the loss in the following band-pass filter and the double-balanced mixer.



RF feedback. An equivalent VHF/UHF design approach is shown in Figs 16 and 17.

Fig 16 shows a band-pass filter for the FM broadcast band, which precedes the high dynamic range tuner. This tuner is part of an FM relay receiver, which is used to receive FM broadcast stations on mountain tops and then modulates an on-site FM transmitter. This FM transmitter frequency has 10 KW output power and is only a few hundred yards away from the receiving station. These relay receivers, therefore, must have a huge dynamic range. The relay receiver tuner shown in Fig 17 uses a PIN diode attenuator in the front end for automatic gain control and several tuned stages prior to the mixer. Two FETs are used in parallel to increase the dynamic range of the pre-amplifier, which is necessary to compensate for the loss of the band-pass filter between the pre-amplifier and the mixer. Such high selectivity is required in this "hostile" environment where so many large signals are present. The double-balanced mixer is one of the CLP family made by Synergy Microwave.

At the output of the mixer one finds a diplexer and a grounded gate amplifier for a precise 50-ohm termination. In this environment, you can imagine the need for high performance of the receiver, which has an intercept point of +23 dBm and a noise figure of 2 dB.

Modern CAD tools have been used to perform feasibility studies. On the cover of this magazine is a photograph of the display of a CAD tool on which linear and nonlinear circuits can be simulated simultaneously. It can show the output spectrum like on the spectrum analyzer, the beat note for the two tones at the output and even show the increased level of voltage and current at different points of a three-stage distributed amplifier. Because of the beat note phenomena, the output load line shows a wider color band than one would obtain from a single tone.

Conclusion

It has been shown that the weakest point in past designs has been the understanding of how to design good mixers. Ever since the introduction of the 417A, low-noise input stages up to 150 MHz were available. The microwave area requires modern GaAs FETs. Measurements and simulation agree quite well, which indicates that modern CAD tools are quite capable of making good predictions. The major differences between the tube designs and bipolar transistor/GaAs designs lie in the matching techniques between the stages and compromises between selectivity, noise figure, and losses.

I would like to thank all the radio amateurs who supplied me with converters, either gratis or at the advertised purchase price, for this project, and they are:

John Abbruscato	KC5GB
Martin Dew	K3RGH

Fig 18—An added circuit showing an all bipolar solution for high performance.

Martin Feeney	K7OYB
David Knepper	W3BJZ
Charles Lustick	K3HSS
Jacob Makhinson	N6NWP
John Pivnichny	N2DCH
Earl Shinn	K5KAC

I would additionally like to thank those radio amateurs who responded to my project, but whose converters I had to refuse due to the overwhelming response.

References

1. Vendelin, G., Pavio, A.M., Rohde, U.L., *Microwave Circuit Design Using Linear and Nonlinear Techniques*, John Wiley & Sons, New York, New York, January, 1990.
2. U.L. Rohde, "Improved Noise Modeling of GaAs FETs, Part I and II: Using an Enhanced Equivalent Circuit Technique", *Microwave Journal*, November, 1991 and December, 1991, pp. 87-101 and 87-95 respectively.

Recommended Reading

Rohde, U.L., Bucher, T.N.N., *Communications Receivers: Principles and Design*, McGraw-Hill Book Company, New York, New York, 1987.

Troubleshooting Digital Circuits With the "Glitch Catcher"

By Gary C. Sutcliffe, W9XT
3310 Bonnie Lane
Slinger, WI 53086
email: ppvvpp@mixcom.com (Internet)

Anyone who troubleshoots electronic equipment will tell you that the toughest problems to fix are the ones that occur infrequently. "You can't fix it if it ain't broke" goes the old saying. In working with digital circuits one problem that sometimes occurs is that the system will run fine for hours or days at a time and then suddenly the whole thing locks up or otherwise fails. The problem might have been caused by a glitch lasting only a few nanoseconds. Finding the right nanosecond in a period of several hours can make looking for needles in haystacks seem like child's play in comparison.

One method of tracking these types of problems is to use the scientific method. First you look at all the evidence, then formulate a theory that explains it. Next you devise experiments that support or disprove your theory. The results of your experiments may require you to modify your theory and run further experiments. Eventually you narrow everything down to a theory that explains what is happening.

Suppose you have a digital circuit for controlling some widget. Everything works great, but every few hours something happens and the whole thing goes wild or just hangs up. Keeping in mind any symptoms of the problem you have noticed, try to think of situations where something abnormal would cause the behavior you see. Don't get too hung up on explaining *everything* though. You might have more than one problem with your circuit.

Once you have your theory, it is time to design your experiment. You want to try to get this problem to occur more frequently since you will probably have to go through several experiments and theories as you zero in on the problem.

What can you do to make it fail more often? If you suspect that the problem has something to do with changes from a sensor or switches, set things up so those changes occur as often as possible. If it seems to occur when some special but infrequent activity is going on, how can you make this activity happen more often? Hopefully you can make your circuit fail in seconds to minutes instead of the minutes to hours it used to.

The next step is to use your test equipment to see if what you suspect is happening really is. Traditional digital test equipment includes oscilloscopes, logic probes and logic analyzers.

Scopes are great for tracking down many problems, but poor for finding single-shot events. You just can't see signals with low repetition rates unless you have a storage or digital oscilloscope. Even then, you may not have the triggering capability to capture just what you need to.

Logic probes are great for finding short, infrequent pulses on an otherwise steady state signal. Not only that, they are extremely inexpensive. Where they fail is in trying to correlate those pulses with other signals.

Logic analyzers are often the best tool for finding single shot events because of their excellent triggering abilities. The disadvantage is they are rather expensive, and not everyone has one at his disposal. Even if one is available, you may not be in a position to dedicate it for days at a time waiting for the next failure.

What is needed is something that is simple to use and inexpensive, like the logic probe, but with some of the triggering ability of a logic analyzer.

When facing problems like this over the years, I have often built little circuits to aid trouble-shooting. Usually they were wire-wrapped kludges that included a few gates, a flip-flop, and an LED. These have progressed to a more flexible piece of test equipment I call the Glitch Catcher. You might think of it as a logic probe with external triggering. Alternatively it could be called a logic analyzer with a one bit wide and one bit deep memory.

Before describing the Glitch Catcher in greater detail, let's continue on in our quest to find the problem with the widget control circuit.

We now have a theory, and hopefully a way to make the failure more repeatable. From your available arsenal of test gear, be it scope, probe, analyzer, or Glitch Catcher, select the best tool for this situation. Set up your test equipment to detect the condition you think may be causing the failure, and run the circuit in the manner that reproduces the problem the most often.

At this point several things can occur. Ideally the condition you are checking for will occur only once, and at the time of the failure. In this case you probably have found the problem. You can now dig deeper to find the cause or work on a solution.

Another possible outcome is that the condition tested for occurs many times before the error occurs. In that case you may have to refine your theory, refine how you are detecting the situation, or start with a fresh theory and approach.

The final possibility is that you did not detect the condition you were looking for, but the failure occurred anyway. Assuming there is only one failure mode, you can probably rule out your initial theory as a possible cause

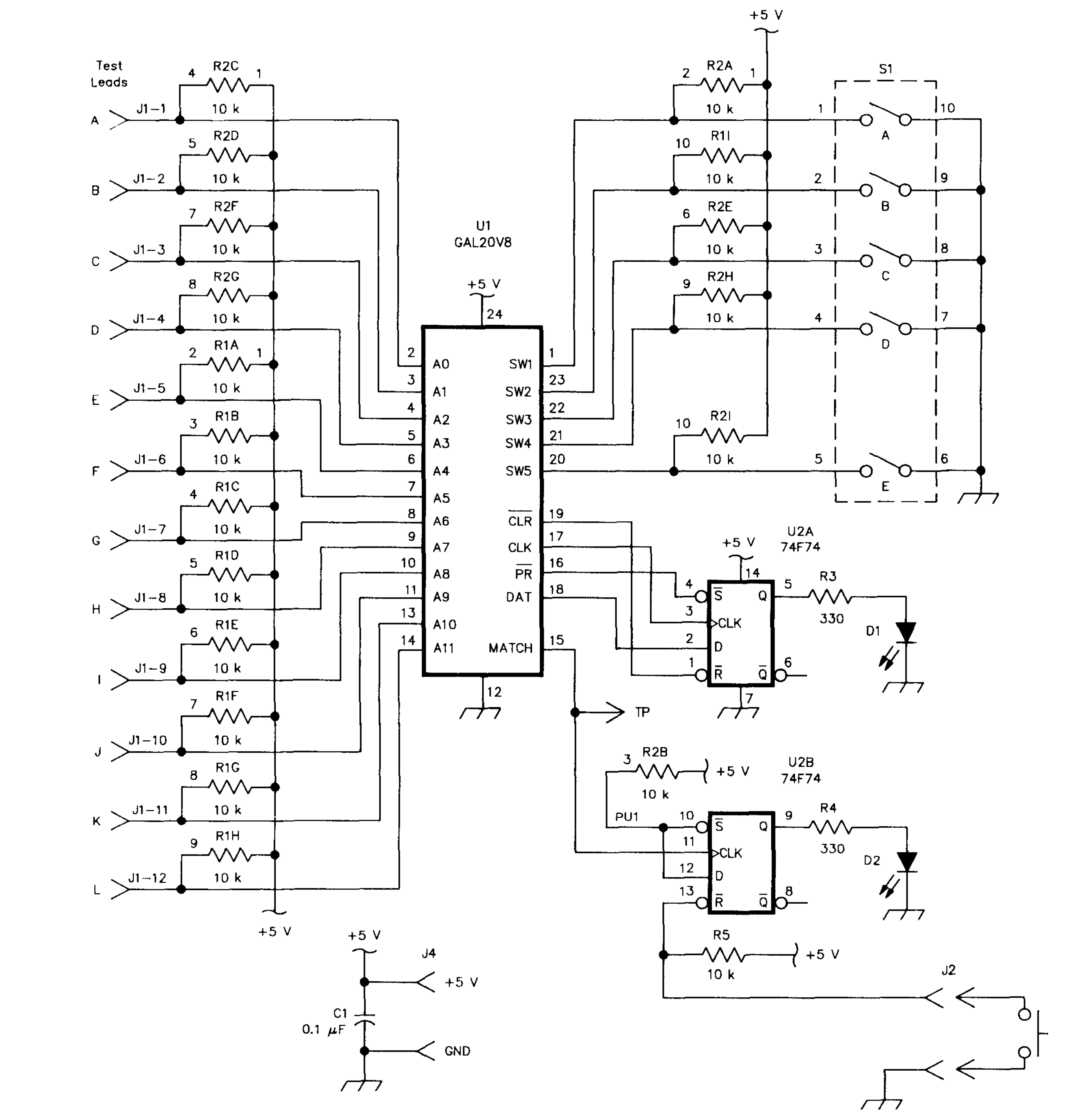


Fig 1—Schematic of the Glitch Catcher.

U1—GAL20V8-15 programmable logic chip. Must be programmed according to Fig 5.
U2—74F74 (74LS74 may be used for testing slower circuits)
R1, R2—10-pin resistor networks, common pin 1 (9 resistors), 10 kΩ.
R3, R4—330 Ω, ¼ W, 5%.
R5—10 kΩ, ¼ W, 5%
C1—0.1 µf, 50 V
D1, D2—LEDs
TC1-TC12—Test clips (JDR MH-1.5R, MH-1.5B; Digi-Key 923830-RD, 923830-BK, or similar)

Table 1—Transient Mode Switch Settings (C = closed, O = open)

DAT (U2-2)	*S1B*	*S1A*	*CLK (U2-3)*	*S1C*
ABC	C	C	DE	C
$AB\overline{C}$	C	O	$D\overline{E}$	O
$\overline{A}\overline{B}C$	O	C		
$\overline{A}\overline{B}\overline{C}$	O	O		
PR (U2-4)	*S1D*		*CLR*	*S1E*
F	O		G	O
$\overline{F}$	C		$\overline{G}$	C

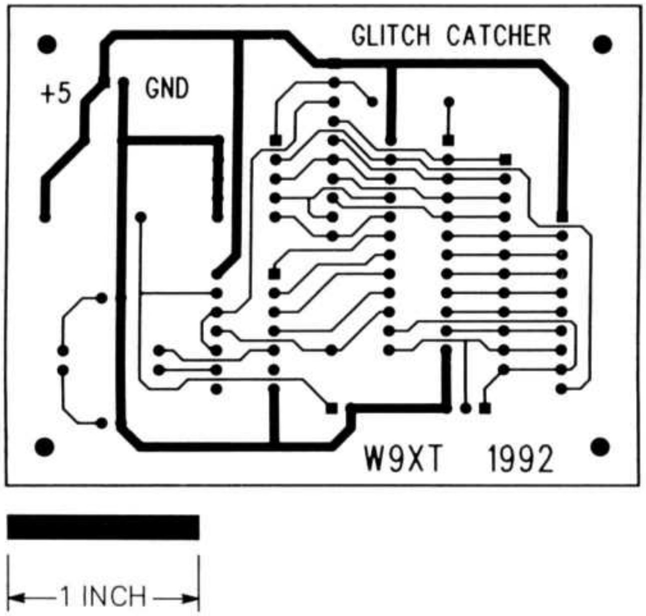


Fig 2—Circuit board layout, solder side, foil side view.

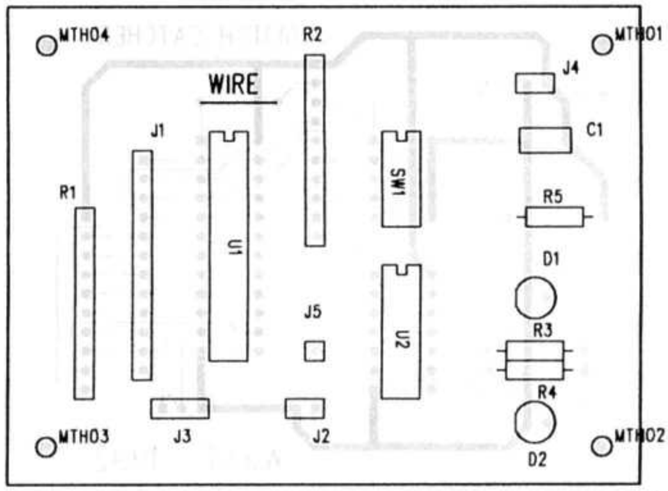


Fig 3—Parts-placement diagram.

of your trouble. In fact, it is sometimes better to come up with a theory and then test to disprove the theory rather than trying to prove it. This is an especially good method if you come up with several plausible theories.

Whatever the outcome of your tests, repeat them several times before trying to fix the problem or changing your theory. It's possible that the condition you detected when the failure occurred was only a coincidence. The type of problems we are trying to fix are often based on probabilities. You wouldn't flip a coin once and if, it comes up heads, assume that it will always come up heads in the future.

Similarly, once you think you have solved the problem, give it plenty of time to fail again before declaring it fixed. Getting back to our coin flipping example, it is unlikely to get several heads in a row, but it can happen. The longer the string of heads in a row (or any specific pattern of heads and tails for that matter) you try to get, the less likely it is to occur. As a rule of thumb, I like to see the circuit work for at least 8-10 times the mean time between failures before I'm confident of my fix. That means that if your circuit used to fail approximately once every five minutes, see if it will run for at least an hour before you pronounce it fixed.

The Glitch Catcher

The Glitch Catcher is a simple two-IC piece of test equipment for digital circuits using TTL logic levels. The heart of the circuit is U1, a GAL20V8 programmable logic device. This drives U2A, a 74F74 (a 74LS74 may be used if the circuit under test is not too fast) flip-flop which acts as the one bit wide, one bit deep memory described earlier. LED D1 is used as an indicator.

The basic glitch-catching design leaves portions of U1 and U2 unused. These are used to provide an additional debugging tool, an address match detector. This part of the Glitch Catcher uses U2B and D2.

The GAL is set up to provide a variety of logic combinations to the flip-flop's inputs. For example, there are three signal lines that can be used to feed the flip-flop D input. The switches select between four different logic combinations of these lines. See Table 1 for a listing of the possible combinations.

Note that all the inputs have 10-kΩ pull-up resistors. These are provided so that you don't have to connect unneeded inputs. Just be sure that the switches are set so that a high state on the unconnected lines permits the operation you want.

The Glitch Catcher uses a GAL for two reasons. The first is that the GAL will provide a more constant signal delay than the equivalent circuit implemented in inverters and gates. This will minimize signal skewing which could mask the condition you are looking for or create false error conditions. The other reason that a GAL was

Table 2—Address Match Mode Switch Settings (C = closed, O = open)

Match if input is

0	*1*	*S1C*	*S1B*	*S1A*	
—	A-L	C	C	C	(All high)
A	B-L	C	C	O	(One low)
A-B	C-L	C	O	C	(Two low)
A-C	D-L	C	O	O	(Three low)
A-D	E-L	O	C	C	(Four low)
A-E	F-L	O	C	O	(Five low)
A-F	G-L	O	O	C	(Six low)
A-G	H-L	O	O	O	(Seven low)

Table 3—Parts Sources

Atlas Circuits Company
PO Box 892
Lincolnton, NC 28092
Tel: 704 735-3943
(A blank circuit board is available from Atlas circuits for $10 each, plus $3 shipping. Specify Unified Microsystems P/N 92-1).

Digi-Key Corp
701 Brooks Ave S
PO Box 677
Thief River Falls, MN 56701
Tel: 800 344-4539
(GALS, misc electronic parts)

EasyTech, Inc
2917 Bayview Dr
Fremont, CA 94538
Tel: 800 582-4044
(GALS, misc electronic parts)

JDR Microdevices
2233 Samaritan Dr
San Jose, CA 95124
Tel: 800 538-5000
(GALS, misc electronic parts)

Radio Shack
Your local Radio Shack outlet
(Misc electronic parts)

Unified Microsystems
PO Box 133
Slinger, WI 53086
Tel: 414 644-9036
(Pre-programmed GAL, $11 postage paid US & Canada. Check or money order only.)

Other than the programmed GAL available from Unified Microsystems, the author does not warrant or guarantee any of the products of the companies listed above.

used is that if you need a particularly complex triggering equation, you can reprogram the GAL to suit your needs.

The Glitch Catcher can be built using wire wrap or point-to-point wiring. Alternatively, you can make a single-sided circuit board using the design shown in Fig 2. The author has also made arrangements for boards to be supplied by Atlas Circuits for $10 each plus $3 shipping. (See Table 3 for a listing of part sources.)

Since the circuit uses a lot of pull-up resistors, 10-pin resistor networks with a common pin 1 were used. You could use discrete resistors instead. If you are using a circuit board, mount the individual resistors upright in pins 2-10. Put a piece of wire in pin 1, bend it over, and solder it to the free end of each resistor.

Several vendors sell small test clips that work well for grabbing onto IC leads. Use the kind you like the most. The logic equations for U1 are shown in Fig 5. This listing is in the format for the National Semiconductor boolean-logic-to-JEDEC-file assembler, EQN2JED. You may need to make some changes if you use a different logic compiler. A source listing and an assembled JEDEC file are available off the ARRL BBS (203 666-0578). The file names are GCGAL.EQN and GCGAL.JED.

GALs can be programmed on most logic programmers capable of handling PALs. Readers without access to programmers can order a programmed GAL from the author's electronic and computer consulting firm, Unified Microsystems (see Table 3).

Using the Glitch Catcher

There are two modes of operation. One is the Transient mode, and the other is the Address Match mode. They both operate at the same time. You just ignore the part of the circuit corresponding to the function you are not interested in. The same switches have different meanings for the GAL outputs for the two modes.

Using the Transient Mode requires a bit of creativity on the part of the user. Appropriate signals for triggering must be found. Suppose you suspect the problem might be occurring because of a pulse on a signal at the wrong time, as shown in the example in Figure 4. Suppose EOC should be low when ACTIVE is high and $\overline{\text{STOP}}$ is low. If EOC were to somehow pulse high at the wrong time, the circuit would lock up.

In this case, you might want to put the Glitch Catcher test lead A on the ACTIVE signal, and test lead C on $\overline{\text{STOP}}$. Lead B is left unattached. Setting switches S1A open, and S1B closed will select the logic for the signal applied to the D input of U2A to be: $AB\overline{C}$. Since the B test lead is pulled high, U2 pin 2 will be high when ACTIVE is high and $\overline{\text{STOP}}$ is low.

Test leads F and G control the flip-flop preset and clear. In this example, we did not use these signals. Leaving S1D and S1E closed will put U2A's clear and preset inputs to the inactive state. Test lead D should connect to EOC.

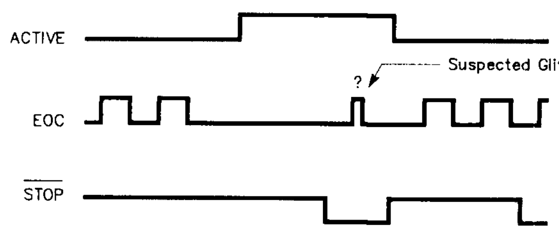


Fig 4—Example glitch detection waveform.

```
title    Glitch Catcher GAL
pattern
revision A
author   Gary C. Sutcliffe, W9XT
company  Unified Microsystems
date     4 JUN 92

;*******************************************************************
; This GAL is used to implement the combinational logic for the Glitch
; Catcher.  It also has a large combinational  decoder that can be used
; for scope triggering.
;*******************************************************************

chip qbox gal20v8

; pin 1  2    3    4    5   6   7    8   9   10  11  12
     sw1 a    b    c    d   e   f    g   h   i   j   gnd

; pin  13    14    15    16     17   18    19   20   21   22   23  24
       k     l    match  /pr    clk  dat   /clr sw5  sw4  sw3  sw2 vcc

; SWn = 1 if switch is open, = 0 if switch closed

equations

dat = a * b * c * !sw2 * !sw1    ;to flip flop D input
    + a * b * !c * !sw2 * sw1
    + a * !b * !c * sw2 * !sw1
    + !a * !b * !c * sw2 * sw1

clk  = d * e * !sw3              ; to flip flop CLK
     + d * !e * sw3

pr   = !f * !sw4                 ;to flip flop preset
    + f * sw4

clr  = !g * !sw5                 ;to flip flop clear
     + g * sw5

match =  l * k * j * i * h * g * f * e * d * c * b * a * !sw3 * !sw2 * !sw1
      +  l * k * j * i * h * g * f * e * d * c * b * !a * !sw3 * !sw2 * sw1
      +  l * k * j * i * h * g * f * e * d * c * !b * !a * !sw3 * sw2 * !sw1
      +  l * k * j * i * h * g * f * e * d * !c * !b * !a * !sw3 * sw2 * sw1
      +  l * k * j * i * h * g * f * e * !d * !c * !b * !a * sw3 * !sw2 * !sw1
      +  l * k * j * i * h * g * f * !e * !d * !c * !b * !a * sw3 * !sw2 * sw1
      +  l * k * j * i * h * g * !f * !e * !d * !c * !b * !a * sw3 * sw2 * !sw1
      +  l * k * j * i * h * !g * !f * !e * !d * !c * !b * !a * sw3 * sw2 * sw1


; end of  qbox
```

Fig 5

Setting S1C closed will clock the flip-flop when the signals on test leads D and E are both high. Since test lead E is pulled high, and not connected to anything else, only signal D controls clocking the flip-flop. In our example test circuit, that will be fine if EOC does not change states any more after the error occurs. If EOC will continue to change after an error the LED might light up too briefly to notice, you might want to disable further clocking of the flip-flop. This is easily done by connecting test lead E to U2 pin 6. Soldering a test point to U2 pin 6 will make clipping test leads to this pin easier.

The setup just described will light up the LED D1 if the suspected error condition occurs. If the LED is already lit up before you start up the test, briefly opening S1E will clear the flip-flop. You are now ready to run the circuit and wait for the next failure.

You will have to find the best switch and test lead configuration for the Glitch Catcher to find the conditions for your particular circuit. Keeping a few things in mind may make it easier to find the best configuration.

First, remember DeMorgan's theorems for selecting the proper logic equation. The switch setting used in our example gave $\mathrm{DATA} = \mathrm{AB}\overline{\mathrm{C}}$. The DeMorgan equivalent is $\overline{\mathrm{DATA}} = \overline{\mathrm{A}}+\overline{\mathrm{B}}+\mathrm{C}$. We could preset the flip-flop before running the test. In this case, if either A or B were low, or C was high when the clock condition occurred, the flip-flop would be clocked low (and the LED turned off) on an error.

You can often use the preset and clear test leads (F and G respectively) to your advantage. Suppose you want to know if the system hangs while a specific operation is going on. You might find a signal that indicates the start of the operation. You could use this signal on the preset test lead (F) to turn on the LED. Some other signal(s) could be used with the clear (test lead G) or the data and clock test leads to indicate successful completion. In this case, if the LED was lit when the system hung, you know it was in the middle of the special operation.

Many intermittent problems are caused by improper set-up and hold times. If you suspect this might be causing problems in your circuit, make sure the set-up and hold times of flip-flop U2, along with the delays and skews in the GAL, are up to the task.

The other mode of operation is Address Match. In this mode you use the GAL to decode a specific address of a microprocessor or microcontroller. First you have to figure out how many zeros (lows) there are in the address of interest. Then you set the switches to correspond with the number of zeros. (See Table 2.) Starting with test lead A, connect the test leads, in order, to the address lines which will be zero for the desired address. Connect the remaining test leads to the remaining address lines which will be logical 1. Leave any extra test leads unconnected; they will be pulled high by the pull-up resistors.

You will want to connect one of the test leads to a signal that indicates that the address is valid to prevent switching glitches from falsely creating a match condition. In some cases, the chip enable to a memory chip may work.

Next, you have two ways to use the Glitch Catcher. The first is to have a match condition set flip-flop U2B and light up D2. This will tell you that the address in question has been reached. A normally open momentary switch attached to J2 can be used to reset U2B.

The address you choose can correspond to an instruction in a program loop. Hook the external trigger probe of your oscilloscope to test point TP. This will make it much easier to look at the signals occurring around the time that instruction is being executed by the microcontroller.

I don't claim that using the Glitch Catcher in the Address Match mode is particularly convenient, especially if you have to change the address often. Address matching is included primarily because it only cost a couple of resistors and an LED to implement. Still, it can be very helpful when nothing better is available.

Use caution in hooking up the Glitch Catcher to the circuit under test. Turn the power off while making connections so you won't damage anything if you accidentally short out something while connecting the test clips. The Glitch Catcher gets it power from the circuit being tested via the +5 V and GND leads.

Summary

Having access to sophisticated test equipment certainly makes digital circuit debugging much easier and faster. Unfortunately such tools are often well beyond the means of the average ham. Combined with effective debugging strategies, the Glitch Catcher is a device that can help fill the gap between simple tools like the logic probe and expensive ones like logic analyzers.
□□

The Shrike Crystal Oven

By Ben Spencer, G4YNM
100 Linslade Street
Swindon SN2 2BN England

Introduction

Crystal ovens generally comprise two distinct sections, a heat source and a crystal oscillator. The heat source is required to raise the temperature of the crystal from ambient to well above that which might ordinarily be found in electronic instrumentation. Usually this is taken to be +70° C or thereabouts. At power-up the heat source must rapidly reach the operating temperature without overshooting and then accurately maintain that temperature.

Most commercially available ovens are unjustifiably expensive and use a bimetallic switch to control the temperature, this latter fact means that the crystal is not accurately maintained at a stable temperature but drifts between two temperature limits, hardly desirable!

The Shrike uses a negative feedback loop to provide a continuously variable dc energy input which takes account of any slight fluctuations in the environment temperature and thus maintains a truly constant crystal temperature.

Construction, testing and calibration are straight-forward, requiring no special skills or equipment. (A 10-MHz SSB/CW receiver and a thermometer covering +40° C to 100° C are required.) There are no rare or hard-to-find components in the design.

Circuit Description

The full circuit diagram for the Shrike is shown in Fig 1.

Heat Source

A Darlington pair formed by Q1 and Q2 converts dc energy into thermal energy. The thermal energy dissipated in Q1 is negligible, that in Q2 is not. Transistor Q1 is biased via potential divider R1, RV1 and R2. Thermistor R2 has a negative temperature coefficient. That is, it provides a high resistance when cold and a low resistance when hot.

When the oven is first powered up, Q1 is biased on quite heavily and so Q2 is driven to saturation. This would normally short out the 8-V supply so a current limiting circuit formed by R3 and Q3 limits flow to a maximum of 650 mA. Hence initially the energy conversion in Q2 is about 5.2 W (or 5.2 Js^{-1}) and the transistor heats up fairly rapidly. Transistor Q2 and thermistor R2 are thermally coupled so as Q2 heats up so does R2, and the value of R2 begins to

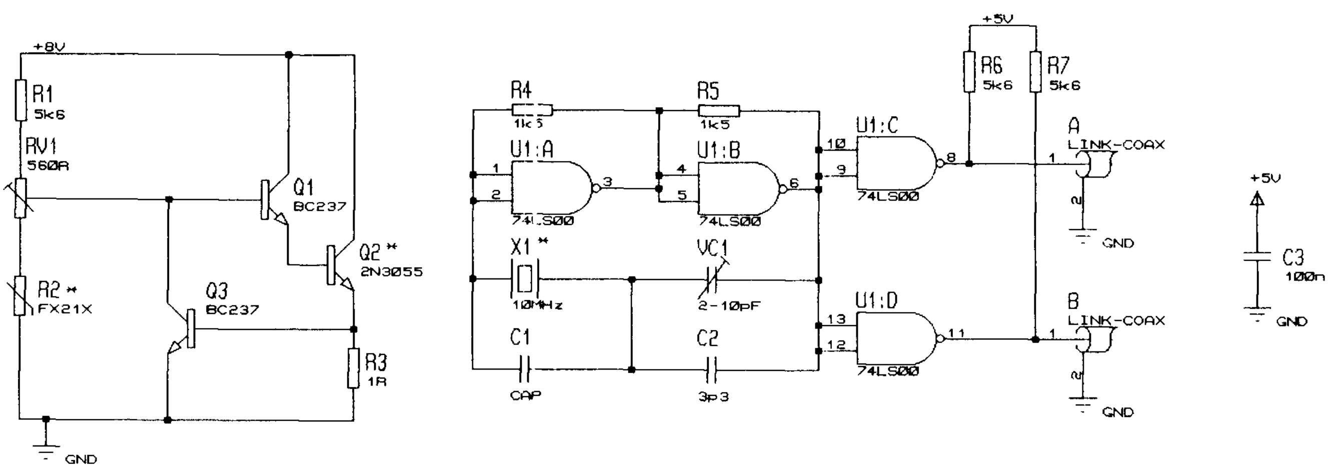


Fig 1—Schematic diagram of the Shrike crystal oven.

C1—Value as required to resonate crystal.
Q1, Q3—BC237 NPN transistor or equivalent.
Q2—2N3055 power transistor.
R2—4.7 kΩ @ 25° C, 318 Ω @ 100° C thermistor, Maplin FX21X or equivalent.
R3—1 Ω, 0.6 W (or 1 watt).
RV1—560-Ω horizontal PCB-mount potentiometer.
U1—74LSOO quad 2-input NAND gate.
X1—10-MHz crystal in HC-43/W can.

fall, reducing the drive to Q2 until a state of equilibrium is reached (at +70° C) where the energy converted in Q2 is the same as the thermal energy transferred to the crystal.

On a cold day, the energy conversion increases and on a summer's day the energy conversion decreases, but the crystal will still be maintained at +70° C. In short, this means that once calibrated, the instrument will have the same accuracy summer or winter.

Crystal Oscillator

The crystal oscillator is formed around NAND gates U1A and U1B with U1C and U1D acting as buffers with pull up resistors (R6 and R7). Identical outputs are provided at A and B. Adjustment of frequency is provided by trimmer capacitor VC1. Capacitors C1 and C2 are provided in case the user wishes to use the oven at other crystal frequencies, with capacitance values to suit.

Note that two separate supply voltages are needed, +8 volts for the heat source and +5 volts for the oscillator. The regulators for these are not included on the PCB as the heat dissipation from the regulators might affect the feedback loop of the heat source. The heat source regulator must be stable; if the 8-V supply drifts then so will the temperature of the source. A 7808-type regulator is adequate.

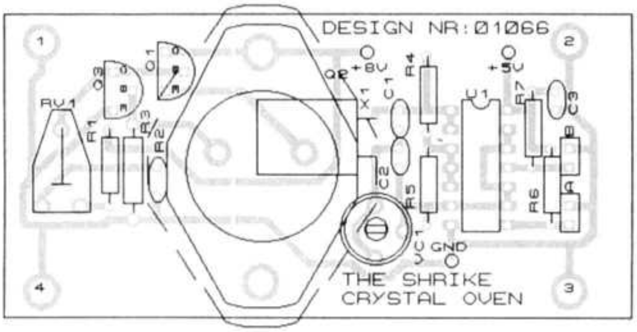


Fig 2a—Component layout.

Construction

A single-sided PCB has been designed for the Shrike. Component layout, drilling guide and foil pattern are shown in Figs 2a, 2b and 2c respectively.

Transistor Q2 is mounted just above the PCB by using two isolation spigots from a TO3 mounting kit on the component side of the PCB, this reduces incidental thermal transfer to the PCB substrate. Thermistor R2 and crystal X1 are mounted horizontally on top of Q2 and thermally coupled to it with heatsink paste—use lots of it!

The completed PCB is fitted into a suitable metal box, a small piece of expanded polystyrene ceiling tile (which has excellent thermal insulation properties) is placed beneath the PCB and another on top. Ensure that two holes are left in the upper polystyrene to allow a trimming tool to reach RV1 and VC1. A hot piece of metal from a coat hanger will melt the polystyrene, leaving a reasonably neat hole.

Testing & Calibration

Set RV1 fully clockwise (maximum temperature) and couple the thermocouple (or thermometer) to the crystal. Switch on the 8-V and 5-V sources and monitor the 8-V current. Initially this should be about 650 mA and should begin to fall within 90 seconds. If it does not then the thermistor is probably not thermally coupled to Q2. After about 7 minutes the current should have reached a stable value. This value will depend upon ambient temperature, thermal losses to the environment etc, but in general should be less than about 250 mA.

Reset RV1 fully counter-clockwise (minimum temperature) and allow the oven to cool completely. This will take some time. The user then will have to adjust RV1 clockwise *very slightly*, wait ten minutes minimum, recheck and readjust as necessary until +70° C is reached and maintained, the equilibrium temperature may wander slightly but the worst case measured the prototype was ±0.1° C when set to a nominal +70° C.

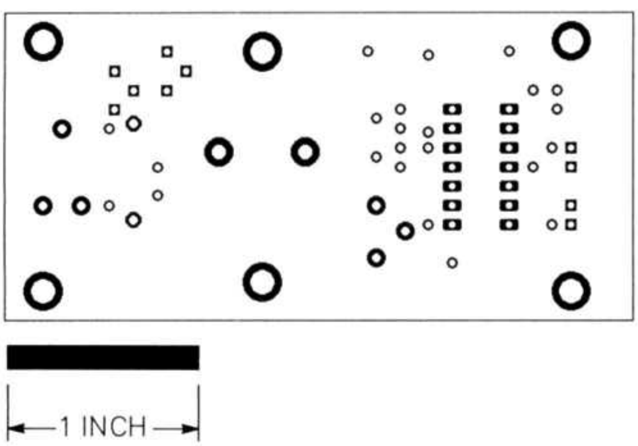


Fig 2b—Drilling guide.

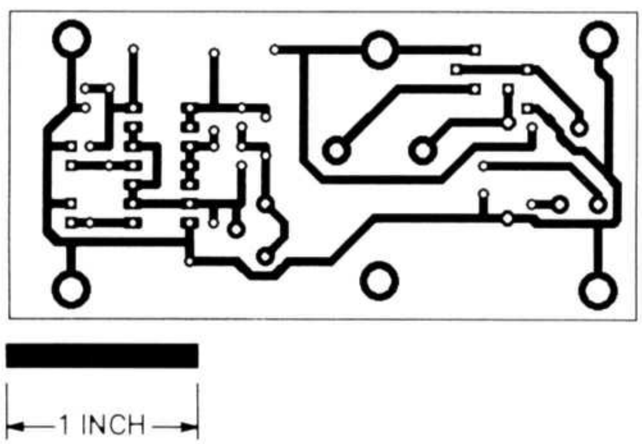


Fig 2c—Solder side foil pattern.

Crystal Calibration

This is a straightforward process. First complete the thermal calibration as described above and allow the unit to reach thermal equilibrium.

Tune an HF radio to WWV at 10 MHz and adjust for zero beat on SSB or CW. Connect a short wire to the 10-MHz output from A or B. The oscillator should be heard on the HF radio beating against WWV. Using a trimming tool, adjust VC1 for zero beat.

The Shrike is now set to 10 MHz and ready for use.

Note: the completed unit should be placed sensibly within any instrument, that is, not close to any localized source of heat such as a power supply or heatsink. To do otherwise is asking for trouble!

[One source for Maplin parts is: Maplin Supplies, PO Box 777, Raleigh, Essex, SS6 8LU England—G4YNM]. □□

Digital Communications

By Harold Price, NK6K
5949 Pudding Stone Lane
Bethel Park, PA 15102
email: nk6k@amsat.org (Internet)
or 71635,1174 (CompuServe)

Jihad!

Nothing gets a Holy War in amateur packet circles started faster than the suggestion that AX.25 needs to be replaced. Some argue to keep it, but there is a plethora of ideas on what a new standard bearer should be. This topic has recently been brought up again. Here's my two cents.

In 1982, AMSAT commissioned AX.25 as a method to link small groups of VHF users through AO-10. The protocol was built and tested and used in small VHF LANs, but it was never used for any real AO-10 or AO-13 gateways. The first AX.25 implementations were meant for user-to-user or user-to-dumb terminal BBS interface service. Since that time, many applications have been built, but we seldom stopped to think about the viability of the AO-13 gateway protocol for the other media it was being put on. My December 1992 column about AX.25 BBS forwarding on HF with no file checkpoint restart gives an example of when naked AX.25 is not the best choice.

AX.25 TNCs were caught in a backward compatibility trap. They used Bell 202 modem tones because that was what the predecessor Vancouver VADCG TNC had commonly used. Even though no more than 200 VADCG TNCs were ever on the air, the 100,000+ TNC1 and TNC2 derived systems are still compatible with 1979's physical layer. The reason that this modem standard was picked in the first place is lost in the mists of time, but I suspect it was because the VADCG group had gotten some 202 modems for free. After all, 9600-baud packet had been done even earlier in Ottawa. The name Terminal Node Controller was backward compatible with the VADCG effort, even though the original TNC was just meant as a device to talk to VADCG's Network Node Controller (now also lost in the mist). Now, even NOS (TCP/IP) carries excess baggage caused by 1979 compatibility.

Still, the thought of how to upgrade 100,000 users is chilling. The easiest way to free ourselves from the past is by realizing that we don't *need* to upgrade these users. Most of them are getting what they want from the current equipment, that is, a low-speed point-to-point link to a service access point. That was AX.25's original intent, and it still serves the purpose. The new work is at the network end. As Mike Gallaher, WA2HEE, said on the TCP-GROUP mailing list, TCP/IP is a network building tool. Much of the new work is inside the network cloud. Many of the 100,000 users of packet don't want to be inside the network—and they don't need to be.

This frees us, hopefully, to use the right protocols and the right physical layer for the right job. Yes, the Clover HF physical layer doesn't use Bell 103 tones, doesn't use AX.25, and costs $1000, but not everyone needs to have one. They would make an excellent HF network building tool. AX.25, with its 256-byte frame size, isn't optimum for high-speed links. But not everyone will have a 10-Mbit interface, so don't worry about the 100,000. Pick the right protocol for the job. More importantly, don't try to pick a single media-access protocol that will suit the HF problem and the satellite problem and the wormhole problem and the 10-Mbit problem. "With what should we replace AX.25" is the wrong question, Grasshopper. AX.25 doesn't need to be replaced, it just shouldn't be used for everything. New things need to be implemented, each with a low-level access protocol that suits its needs, and each with the ability to encapsulate IP frames. And the FCC will still be happier if call signs are used. It's always something.

We're Still Doing it Wrong

Hey programmers, wake up and smell the coffee. It's 1993. Almost no one has just a dumb terminal anymore. Yet, we're still making the same mistake on device control, and we're doing it over and over. The mistake is that we're writing the device control software assuming that what's connected to the device is a human typing at a keyboard. I made this mistake in 1982 with the TNC1. Back then, of course, most users *did* have a dumb terminal, but the TNC1 should have had a computer interface first and a terminal interface second. We released it with just the terminal interface though, and 11 years later we're still suffering from the kinds of problems you get from computers pretending they're people so they can talk to other computers!

In people-type interfaces, you're relying on the user's intuition and real-time adaptive cognitive abilities to tell when something is wrong. Computers aren't very good at this sort of thing; they need a much more controlled environment to function well. For example, you and I can assume the TNC is no longer connected when we realize we haven't heard the PTT relay click in a "while," or we see "*** Disconnected" interspersed with other characters, or the room lights flicker, or some infinitely variable timer sense tells us something isn't right. Computers are left in the dust at this sort of thing.

Here are some other recent examples I've struggled with in the past three weeks. These are not meant to call undue attention to the specific vendors involved; the problems are common to many devices. First, the Kenwood TS-790. This has computer control, but when you hit almost any key on the radio's keypad, the computer's view and reality start to diverge. There are modes you can put the radio in that the computer control can't get you out of, or even tell that you've gotten in to. I'm writing software to remotely control the radio but a tech accidentally brushing against the front panel can lock me out in any number of ways. These aren't mechanical buttons, mind you, just options that the command set has no control over. This makes it difficult to do reliable computer control, which is the sort of thing that makes humans leery about computer control in the first place.

Plea number one: Give the computer interface some global reset command that puts the device in a known state. If the human users don't want this to happen, they are free to disconnect the wire labeled "remote."

The second problem with the '790 (and many other devices) is the lack of positive command acknowledgments. If you send a proper command to the '790, it beeps for the human, but doesn't send anything back to the computer. If you can't tell if a command got in, you've lost positive control. The '790 has a main transceiver and a sub-receiver, and is dual band on 2 meters and 70 cm. To change frequencies, you send one command saying which receiver you want to address, and a second command to set the frequency. In my case, the two-meter RF jack goes to a transmit antenna, the 70-cm RF jack goes to a preamp. It's not a good idea to transmit to a preamp, and it would be nice if we could trust the computer to not command the main transceiver to a 70-cm frequency. The problem: since there is no acknowledgment for the destination command, you can send it, have it be missed, and then send the frequency command to the wrong receiver. Sizzle. I wouldn't mention this if it hadn't happened in testing. The destination command was missed due to a power glitch, and suddenly the main receiver was on 70 cm. Fortunately, no preamp was in use at the time.

Plea number two: Please acknowledge all input commands. If it is important to beep for the user, it is more important to beep for the computer.

The next example is the *TrakBox*. Sold by a variety of sources, the *TrakBox* takes as input your location, a list of satellite frequencies and orbital elements, and the time, and automatically tracks antennas and corrects for Doppler shift. You load this information though a serial port. The serial-port interface leads you through a series of menus. This is very difficult to automate in any reasonable way, especially when the menus and the prompts change from release to release. The desire to write a menu interface for these devices is strong; No matter what kind of host computer the user has, it will have a dumb-terminal emulator program. That program can be used to talk to the device, eliminating the need for the device's programmer to supply an interface program for each of *n+1* host computers. The problem is that by providing only a menu interface, it becomes difficult or impossible for anyone *else* to supply a software-driven interface.

Plea number three: Provide a host interface first, then do the menu interface.

The final example is the TNC KISS protocol. KISS was invented partly to solve the TNC's original dumb-terminal interface. Almost all of the command functions were stripped away, and the data interface was replaced with a protocol similar to the TCP/IP-world's SLIP protocol. The SLIP protocol is a way to move IP datagrams over a dumb serial interface, and it sounds like the perfect thing to use to move AX.25 frames over a serial interface. The problem was that KISS neglected one detail. In real SLIP, the entire IP datagram is transferred, including the checksums. In KISS, the AX.25 frame is transferred except for the checksum (CRC). In a perfect world, the few inches between the TNC and the host computer could be traversed without loss. In our world, RF on the wires can cause problems. Sometimes the few inches becomes fifty feet. More commonly, at high speeds, lack of compute power in the TNC or the host computer causes bytes to be dropped. With no checksum or length information in the KISS frame, missing bytes are undetected. This was so common in ground stations for the 9600-baud satellites that we had to add checksums in the data in addition to the 16-bit CRC in the AX.25 frame. This adds overhead on the "expensive" satellite links to avoid a problem in the "free" KISS link.

Final plea: Use checksums on any computer-to-computer link, no matter how small you think the chance of error is. Someone or something will find a way to induce errors on that path.

Mail Bag

I received a copy of *NOSintro - TCP/IP over Packet Radio, An introduction to the KA9Q Network Operating System.* (Even naming a NOS book requires many words!) I got the book from the author, Ian Wade, G3NRW. This was a not-so-subtle try at a free plug, which has apparently succeeded. The book is professionally printed, and well written with plenty of charts. It devotes its 300+ pages to giving practical information on the mechanics of configuring and using NOS, aimed at the beginner. Because NOS covers so much ground as a gateway, a router, a BBS, a mailer, a terminal interface, and a file server, 300 pages is about the minimum for getting it all in. Not every potential NOS user needs to read all of it, of course. The author also provides examples of all the setup files you'll need. *NOSintro* uses the mythical inhabitants of Nosland as examples of various types of stations. The Noslanders have NETROM, regular AX.25 users, BBS systems, and an IP network; they have plenty of problems to solve. As an aside, the *Star Trek* universe has recently introduced a new group of generic bad guys called the Nosicans, but this is probably just a coincidence.

NOSintro is available from the author, g3nrw@dircon.co.uk. [As we went to press we learned that *NOSintro* will also be available from ARRL. Pricing to be announced.—Ed.] The author's latest *NOSview* on-line documentation is also available. Along with the NOS documentation and help files is a pre-configured version of NOS for the IBM PC. This documentation is a bit more advanced than the *NOSintro* book. At this writing *NOSview* is available with FTP from ucsd.edu in/hamradio/packet/tcpip/nosview as nosview.zip and nosvw304.zip. [nosvw304.zip is also available from the ARRL telephone BBS at 203 666-0578.—Ed.] □□