THALES COMMUNICATIONS, INC.
This information represents intellectual property which shall not be disclosed or released, except to comply with contractural requirements. This information is provided on a limited basis and does not include any rights to manufacture, or have manufactured, any equipment depicted therein.

Datron, DSP Software Document

Prepared by: Thales Communications, Inc.

©Copyright 2001, Thales communications, Inc.
All rights reserved.

November 19, 2001

Chapter 4

Transmit (TX) Mode

4.1 Transmit Chain Structure

Most of the signal processing performed while the radio is transmitting depends on its operational mode. The four possible modes are APCO Project 25 voice mode, APCO Project 25 Data mode, CVSD DES mode, and analog FM mode. The software modules that make up these modes is shown in Figure 4.1. This figure shows a block diagram containing the major signal processing functions and buffers. The diagram covers the software components and buffers from the codec input to the digital to analog converter (DAC) output.

APCO Project 25 Data mode is currently under development. Information contained in this document that refers to the data mode should be considered preliminary design specifications.

4.2 TX Circular Buffers

Typically, circular buffers that are accessed using C code rely on the circular buffer structure described in Chapter 3. The transmit buffers, and their associated pointer structures, are listed in Table 4.1. The setup code for these buffers in located in the file gt_cbuf.c. Notice that a physical buffer may be accessed using multiple pointer structures.

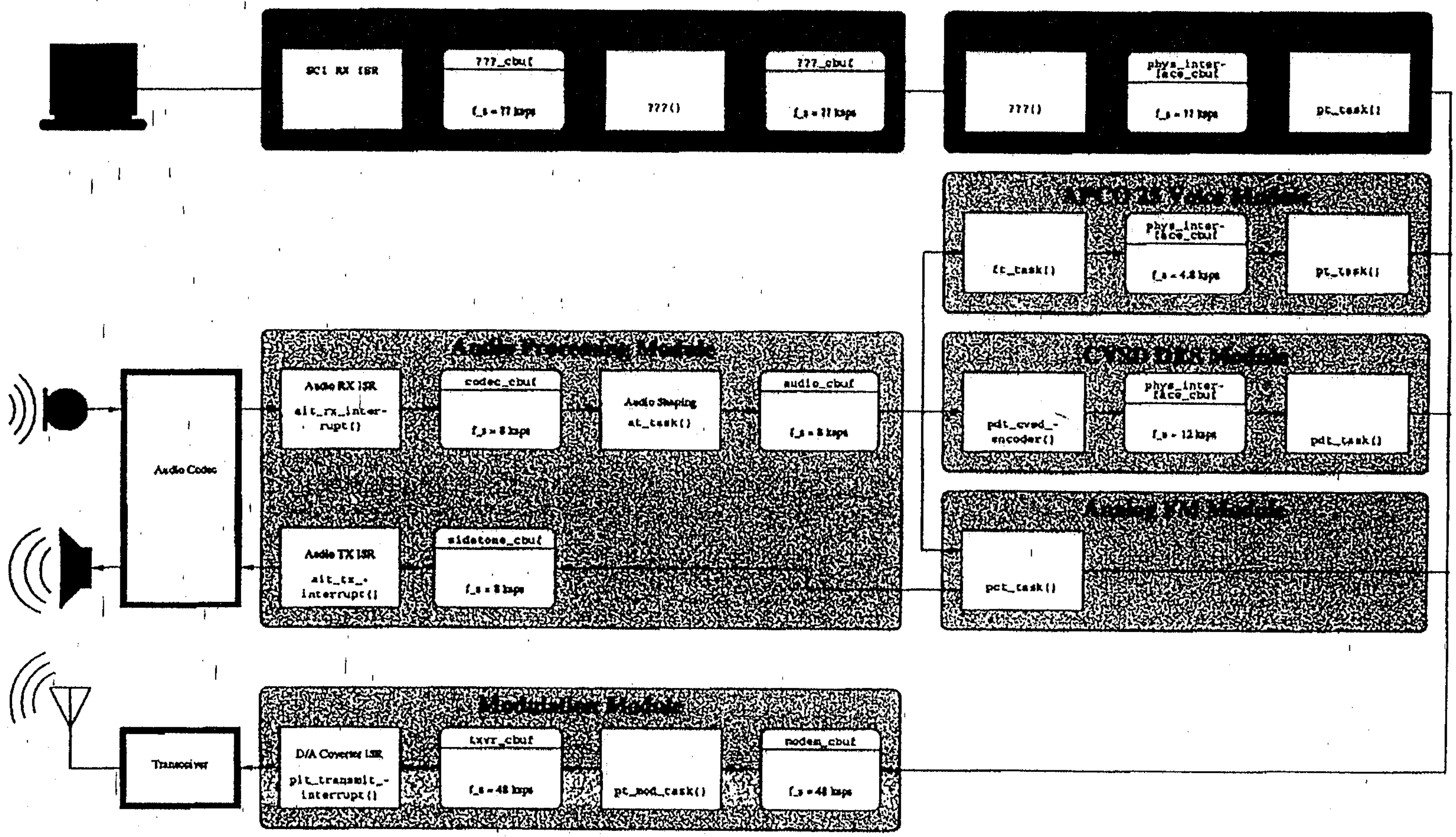

Figure 4.1: Transmit Chain Block Diagram.

Buffer Access, Pointer Structure	Physical Buffer	Description
Front-end buffers (Audio):		
`X:codec_cbuf`	`X:codec_rx_buffer[256]`	Holds samples received from the CODEC.
`Y:audio_cbuf`	`Y:audio_buffer[256]`	
`X:sidetone_cbuf`	`X:codec_tx_buffer[256]`	Used to output DTMF tones to speaker.
Intermediate processor buffers:		
`X:phys_interface_cbuf`	`X:phys_interface_buffer[140]`	
CVSD DES:		
`X:tx_upsampled_audio_cbuf`	`X:modulation_buffer[256]`	
`X:tx_cvsd_encoded_audio_cbuf`	`X:tx_cvsd_encoded_audio_buffer[256]`	
`X:tx_cvsd_des_symbol_cbuf`	`X:symbol_buffer[256]`	
APCO:		
`X:tx_symbol_cbuf`	`X:symbol_buffer[256]`	
`X:tx_filtered_symbol_cbuf`	`X:modulation_buffer[256]`	
Analog FM:		
`X:tx_filtered_audio_cbuf`	`X:modulation_buffer[256]`	
`X:tx_limited_audio_cbuf`	`X:limited_audio_buffer[256]`	
`X:tx_shaped_audio_cbuf`	`X:tx_cvsd_encoded_audio_buffer[256]`	
Back-end buffers (Transceiver interface):		
`X:modem_cbuf`	`X:modem_buffer[128]`	
`Y:txvr_cbuf`	`Y:txvr_buffer[740]`	

Table 4.1: Transmit Chain Circular Buffers

4.5.2 APCO 25 Physical Layer Task

The physical layer task (pt_task()) is responsible for converting the binary dibit data stream into a sampled waveform, which is then fed to the Modulation Module of Figure 4.1. The 4.8 ksps, dibit data stream is converted to a 48 ksps, real valued waveform. The code for the physical layer task is located in the file pt_task.c. Figure 4.2 shows the APCO physical layer processing in the form of a block diagram.

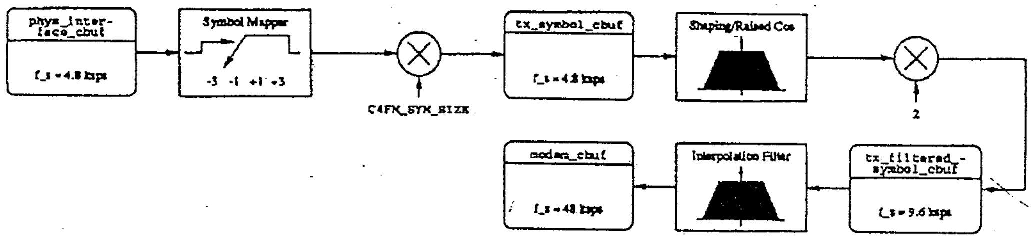

Figure 4.2: APCO Physical Layer Block Diagram

The physical layer task starts by calculating how many dibits have been written to the phys_-interface_cbuf[] buffer. It then checks to see how much space, in the output buffer modem_-cbuf[], is available for processed data. This is calculated by dividing the number of available words in the output buffer by the overall interpolation value of 10. This value is then compared to the number of dibits in the input buffer, and the smaller of the two values is selected. This gives the number of dibits that can be processed. The number of dibits to process must be greater than SYMBOL_SINC_FILTER_UPSAMPLE_RATE = 2, or the function exits.

Symbol Mapper

The symbol mapper converts the 4.8 kHz dibit data stream into a 4.8 kHz real valued data stream. The dibit to symbol mapping is given in Table 4.3. This table corresponds to a value of 0.0375 for the scaling factor C4FM_SYM_SIZE, shown in Figure 4.2.

Dibit Value	Symbol Value	Scaled Symbol Value
01	3.0	0.1125
00	1.0	0.0375
10	-1.0	-0.0375
11	-3.0	-0.1125

Table 4.3: Dibit to Symbol Mapping

The dibit to symbol mapping is performed by using the dibit value as an index into the symbol array c4fm_symbol[], which holds the scaled symbol values given in Table 4.3. The dibit sample stream is then written to the tx_symbol_cbuf[] buffer.

Raised Cosine and Shaping Filter

The shaping and sinc compensation filter is applied to the 4.8 kHz real valued data stream. This filter up samples by two to produce a 9.6 kHz sampled waveform. The magnitude response for this filter is shown in Figure 4.3. This figure shows three traces. The *shaping* trace is the shaping filter frequency response specified in Section 9.3 of Reference [8]. The *raised cos* trace is the frequency response of the raised cosine filter, also specified in Reference [8]. The *combined* trace is the frequency response of the filter used in the DSP code. It shows that the DSP filter is indeed the combination of the shaping and raised cosine filters specified in the APCO standards. Figure 4.4 shows the impulse response for the combined filter.

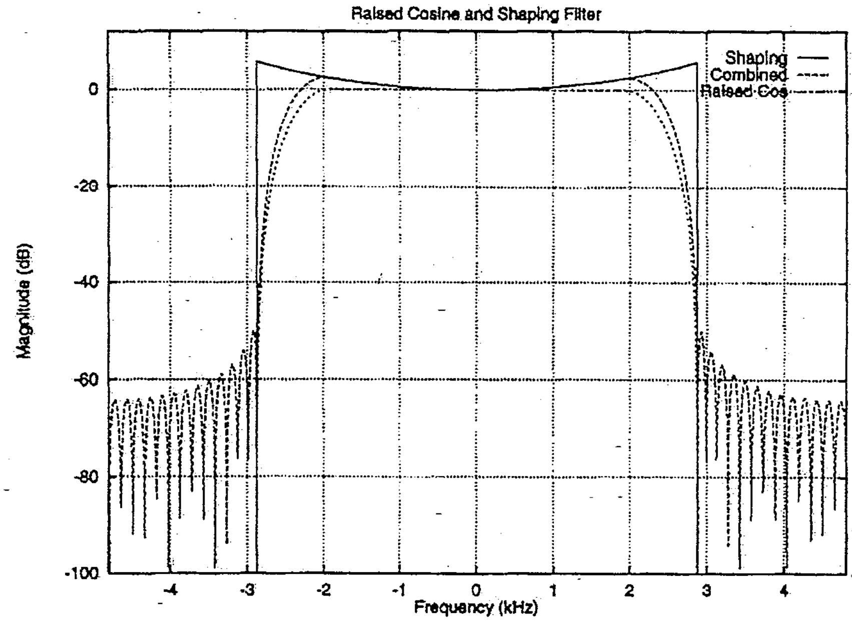

Figure 4.3: Raised Cosine & Shaping Filter Frequency Response

Because the combined filter interpolates by two, the output data is also scaled by a factor of two. The 62 coefficients, of the combined filter, are split into two sets. The first set consists of the odd numbered coefficients c_i ; $i \in \{1, 3, 5, \ldots, 61\}$, and the second set consists of the even numbered coefficients c_i ; $i \in \{2, 4, 6, \ldots, 62\}$. This interleaving of coefficients allows the filter to be implemented in C, using Equation A.11, page 100. For each dibit, two output samples are produced. The first output sample is the result of applying the first 31 filter coefficients to the input data located in the `tx_symbol_cbuf[]` buffer. The second output sample is the result of applying the second 31 coefficients to the input data.

Interpolation by 5

Figure 4.5 shows the frequency response of the interpolate by five filter. This filter is used to bring the sample rate of the output from the combined filter up to 48 ksps. Notice that a gain factor of five is built into the coefficients. This accounts for the attenuation caused by the interpolation process. This filter also has its 50 coefficients interleaved according to Equation A.12.

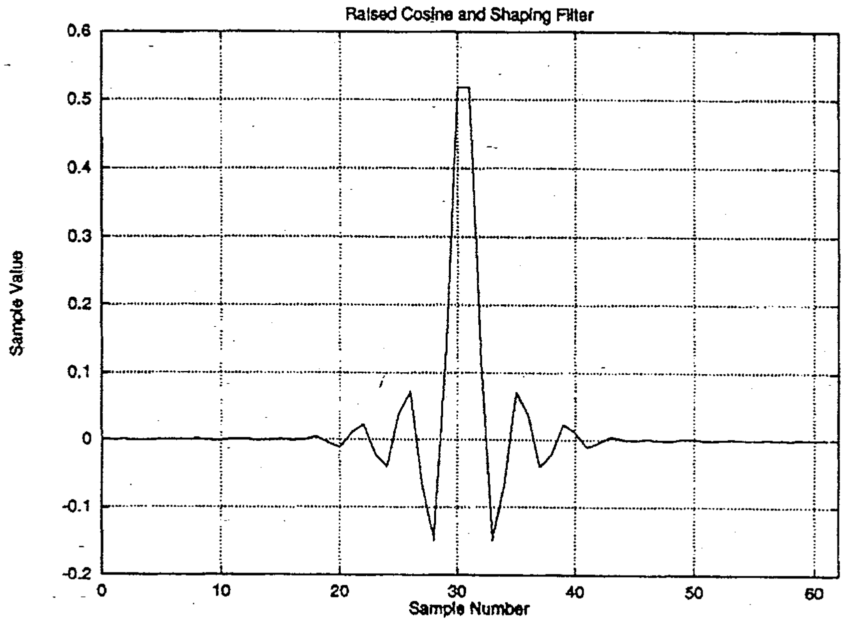

Figure 4.4: Raised Cosine & Shaping Filter Impulse Response

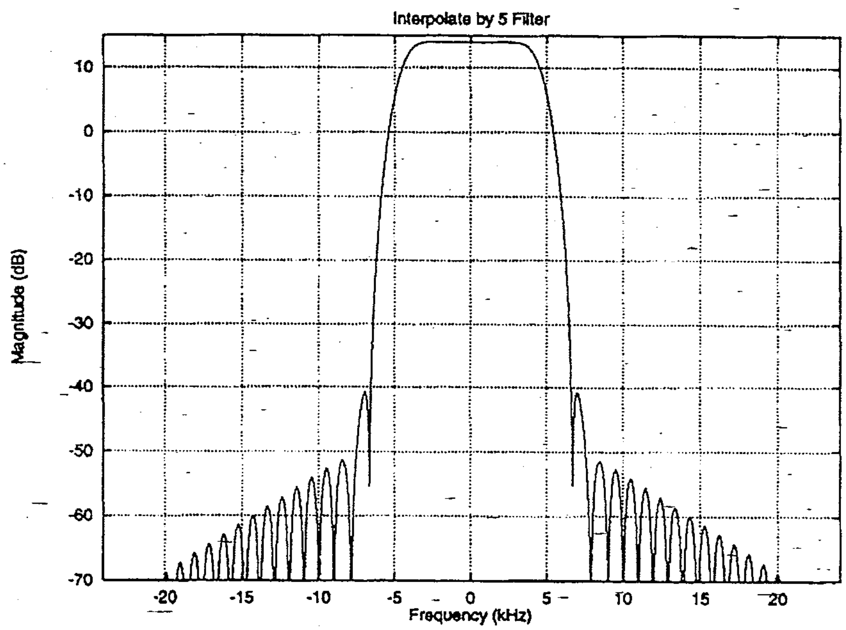

Figure 4.5: Interpolate by 5 Filter Frequency Response

4.6 CVSD DES Module

4.6.1 CVSD Encoder and DES encryption

4.6.2 CVSD Physical Layer

The CVSD physical layer converts CVSD encoded and DES encrypted bit stream, described in the previous section, to sampled waveform that is suitable for processing by the Modulation Module of Figure 4.1. This conversion is shown in the block diagram of Figure 4.6.

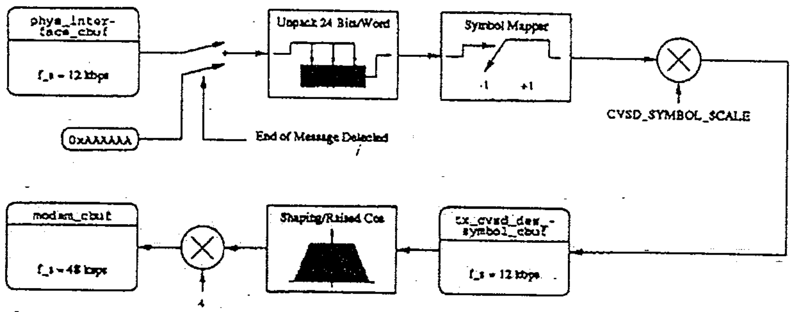

Figure 4.6: CVSD Physical Layer Block Diagram

Data Unpacking

When the radio is in CVSD DES mode, the buffer phys_interface_cbuf[] contains packed, DES encrypted, CVSD data bits. Each buffer location holds 24 of these packed bits. The most recently received data bit is located in the LSB of the buffer word. Therefore, the bits need to be processed starting with the MSB. The bit unpacking block in Figure 4.6 illustrates the unpacking procedure for the case of three bits per word.

Symbol Mapper

The symbol mapper converts the binary data stream into a real valued data stream. The bit to symbol mapping is given in Table 4.4. This table corresponds to a value of 0.2375 for the scaling factor CVSD_SYMBOL_SCALE, shown in Figure 4.6.

Bit Value	Symbol Value	Scaled Symbol Value
0	-1.0	-0.2375
1	1.0	0.2375

Table 4.4: CVSD Bit to Symbol Mapping

Raised Cosine and Shaping Filter

The raised cosine and shaping filter is a 64-tap linear phase, FIR filter. The frequency response for this filter is shown in Figure 4.7, and the filters impulse response is shown in Figure 4.8. This filter also acts as an interpolation filter and has its coefficients interleaved according to Equation A.11.

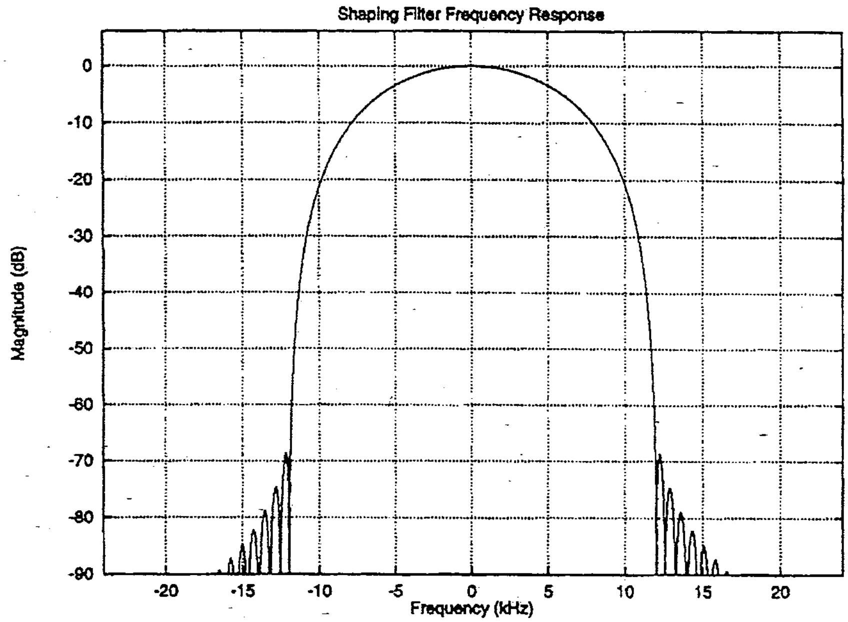

Figure 4.7: Raised Cosine & Shaping Filter Frequency Response

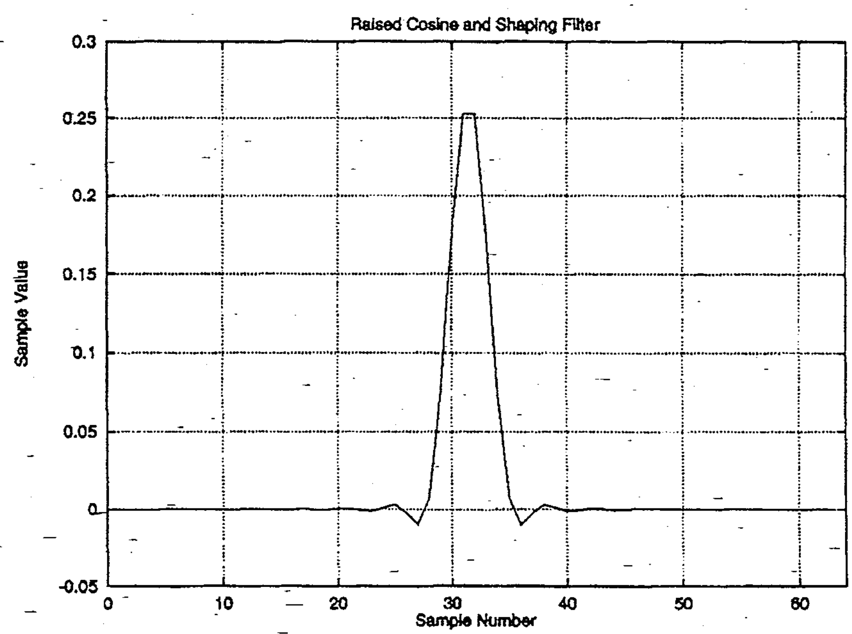

Figure 4.8: Raised Cosine & Shaping Filter Impulse Response

End of Message (EOM) Indicator

The end of a transmission is signaled by transmitting 160 ms of alternating ones and zeros. This allows the receiving radio to squelch the audio output, before the transmitting radio stops transmitting.

The physical layer produces the EOM indicator by replacing the packed input data with the 24-bit value `0xAAAAAA`. This is shown in Figure 4.6 by the switch preceding the data unpacking block. Transmission is disabled after 80 EOM words are sent. This gives

$$(80 \text{ words}) \left(\frac{24 \text{ bits}}{\text{word}}\right) \left(\frac{4 \text{ samples}}{\text{bit}}\right) \left(\frac{48 \times 10^3 \text{ samples}}{\text{sec}}\right)^{-1} = 160 \text{ ms.} \tag{4.1}$$

4.8 Modulation Module

This module prepares the signal for transmission by the transceiver. Section 4.9 gives an overview of how the DSP baseband outputs are converted into a FM modulated waveform.

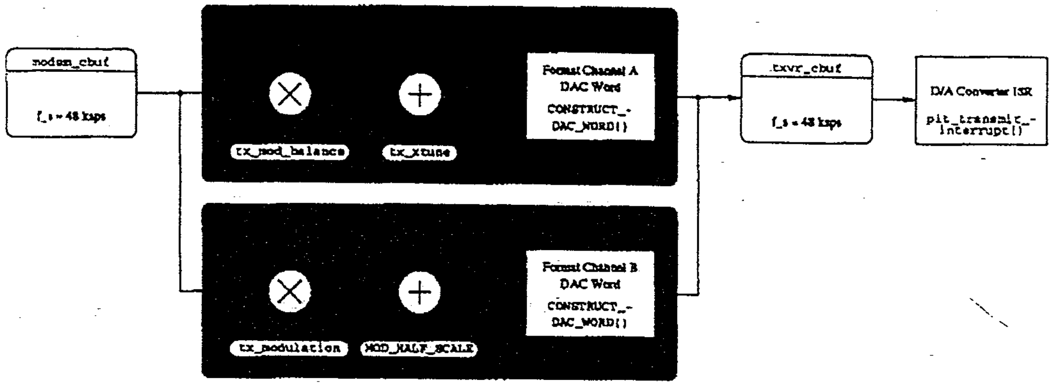

Figure 4.11: Modulation Module Block Diagram

The operation of this module is shown in the block diagram of Figure 4.11. The baseband signal is split into a reference oscillator signal and a VCO signal. This allows independent scale and offset values for each signal. The variable `tx_mod_balance` scales the reference oscillator voltage, so that the maximum frequency deviation is constant for all RF channels. The variable `tx_modulation` does the same thing for the VCO signal. Temperature compensation of the reference oscillator is accomplished by varying the value of `tx_tune`. The value of this variable is read from the serial EEPROM and passed to the DSP by the host processor. Adding `MOD_HALF_SCALE` to the VCO signal moves the DC bias point of the varactor diode to one half the DAC reference voltage. Table 4.5 gives the correlation between the variables in Figure 4.11 and those of Equation 4.4.

Code Variable	Parameter
`tx_mod_balance`	k_a
`tx_modulation`	k_b
`tx_tune`	c_a
`MOD_HALF_SCALE`	c_b

Table 4.5: Modulation Calibration Variables

4.8.1 DAC Data Formatting

4.9 FM Modulation

The Racal 25 uses a two port modulation technique, shown in Figure 4.12, to convert the base band signal $m(t)$ into a FM modulated signal $s(t)$.

The DSP outputs separate signals to channel A and channel B of the DAC. The DAC's outputs produce the analog versions of these signals. These signals are defined by

$$\begin{aligned} y_a(t) &= k_a m(t) + c_a \\ y_b(t) &= k_b m(t) + c_b. \end{aligned} \tag{4.4}$$

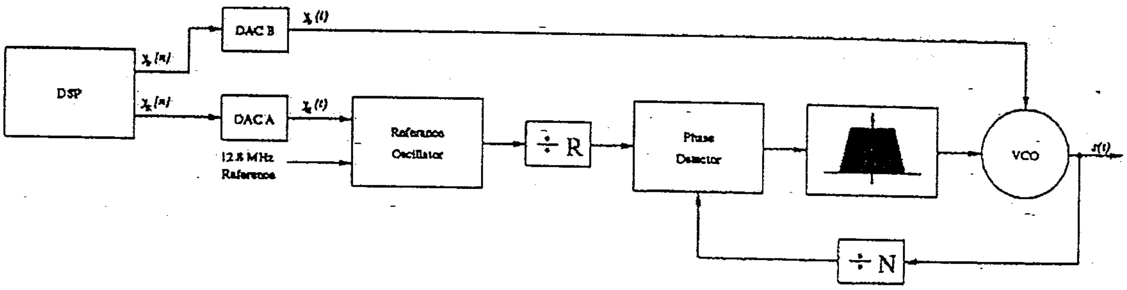

Figure 4.12: Two Port, FM Modulation Block Diagram

The constant c_a equals the DC constant from the temperature compensation table, and c_b equals a DC value selected so that the VCO operates in the linear region for the varactor diode. Figure 4.13 illustrates the form of the varactor diode capacitance vs. voltage plot. For the Racal 25 transceiver, it was found that a bias voltage of 2.5 volts put the average operating point of the varactor in the center of its linear region.

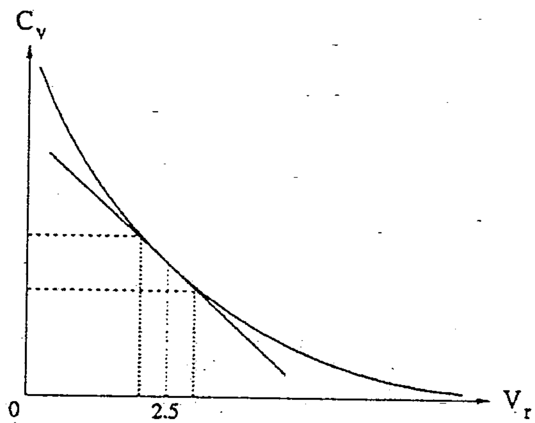

Figure 4.13: Plot Illustrating the VCO Linear Region

The values of k_a and k_b are a function of the carrier frequency. A higher carrier frequency requires smaller values of k_a and k_b for a given frequency deviation. For each individual transceiver, these values are determined and stored in the transceiver's serial EEPROM. Once a transmit channel has been selected, the control processor supplies the DSP with the appropriate values for the tuned channel. See Section 4.8.1 for the formatting and use of the variables described in Equation 4.4.

4.9.1 Modulation Calibration