

Efficient Implementation of an I-Q GMSK Modulator

Alfredo Linz, *Member, IEEE*, and Alan Hendrickson

Abstract—An efficient implementation of GMSK modulators achieved taking advantage of symmetry properties of the premodulation filter is described. The design bypasses typical processing steps such as modulating waveform ROM lookup, integration, and sine/cosine ROM lookup, and therefore does not introduce the related errors. A design for a cordless telephony IC with a bit rate of 72 Kbps and 64 samples per symbol was implemented using only simple logic and 128×6 bits of ROM plus 2 simple 6 b + sign DAC's.

I. INTRODUCTION

GAUSSIAN MINIMUM SHIFT KEYING (GMSK) is a frequency modulation scheme that has gained increased popularity in mobile radio communication systems, such as cordless telephony, where it is desirable to achieve narrow spectra to maximize the number of available channels, and at the same maintain low adjacent channel interference. Since the modulated signal has constant amplitude, efficient RF amplifier configurations such as class C can be used to minimize power consumption, an important consideration for battery-powered units.

Typically, the carrier frequencies for this kind of system may range from tens to thousands of MHz. This precludes on-chip generation of the final modulated signal in most presently available technologies such as a typical digital CMOS process. The method of choice is to generate equivalent baseband signals, which are then utilized by external RF components. This paper describes an efficient method for generating such signals, in which errors are minimized. Part 2 briefly discusses the fundamentals of GMSK modulation. Part 3 reviews some relevant properties of Gaussian filters. Part 4 deals with the mathematical description of the frequency and phase trajectories. Part 5 describes the synthesis method of the *I* and *Q* signals. Part 6 describes the actual implementation and compares it with a conventional design. Some experimental and simulation results are given in Part 7.

II. FUNDAMENTALS OF GMSK MODULATION

GMSK is derived from continuous-phase FSK (CPFSK) by selecting the frequency deviation to be the minimum possible (MSK) and filtering the baseband modulating signal with a Gaussian filter (GMSK). All this is done to minimize the spectral width of the signal. Fig. 1 shows the basic system.

Manuscript received April 24, 1994; revised September 12, 1994. This paper was recommended by Associate Editor M. Steyaert.

The authors are with the Advanced Micro Devices Inc., Communications Products Division, Austin, TX 78741 USA.

Publisher Item Identifier S 1057-7130(96)00883-X.

The first step is conversion of a binary data stream $a(n)$, $a \in (0, 1)$, to an antipodal sequence $b(n)$, $b \in (-1, 1)$, and to a stream of rectangular pulses $r(t)$

$$r(t) = \sum_n b(n)P(t - nT_b) \quad (1)$$

where

$$\begin{aligned} P(t) &= 1 & t \in (0, T_b) \\ P(t) &= 0 & \text{otherwise} \\ T_b &= 1/f_b & \text{is the symbol interval.} \end{aligned}$$

The next operation is Gaussian filtering of $r(t)$

$$g(t) = r(t) * h(t) \quad (2)$$

where

$$\begin{aligned} g(t) &= \text{filtered pulse stream} \\ h(t) &= \text{impulse response of Gaussian filter.} \end{aligned}$$

Modulation of the carrier wave frequency by $g(t)$ around a center frequency f_c implies modulating its phase Φ by $\int g(t) dt$

$$\Phi(t) = \omega_c \cdot t + 2\pi \cdot f_m \cdot \int_0^t g(\tau) d\tau \quad (3)$$

where the phase has been normalized to 0 at $t = 0$. The instantaneous frequency of the modulated signal will be

$$f(t) = \frac{1}{2\pi} \frac{d\Phi}{dt} = f_c + f_m \cdot g(t) \quad (4)$$

where

$$\begin{aligned} f_c &= \text{carrier frequency} = \omega_c/2\pi \\ f_m &= \text{peak frequency deviation.} \end{aligned}$$

The peak frequency deviation is determined by the bit rate. MSK requires that one bit interval of duration T_b contain N cycles of frequency $f_c - f_m$ and $N + 1/2$ cycles of $f_c + f_m$. This leads to the relation [6]

$$f_m = \frac{1}{4T_b} = \frac{f_b}{4} \quad (5)$$

The modulated carrier is then generated. A voltage controlled oscillator (VCO) or equivalent digital synthesis method can produce directly

$$y(t) = \cos(\Phi[t]). \quad (6)$$

For high carrier frequencies, however, maintaining an adequate sampling rate with a digital approach requires an

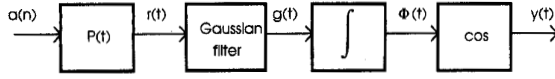


Fig. 1. Basic GMSK modulator.

extremely high operating frequency, making this "direct synthesis" method impractical. An alternative is the quadrature implementation. The desired signal $y(t)$ can be written as

$$y(t) = \cos(\omega_c t) \cos\left(2\pi f_m \int g[\tau] d\tau\right) - \sin(\omega_c t) \sin\left(2\pi f_m \int g[\tau] d\tau\right) \quad (7)$$

or,

$$y(t) = I(t) \cdot \cos(\omega_c \cdot t) - Q(t) \cdot \sin(\omega_c \cdot t) \quad (8)$$

$I(t)$ and $Q(t)$ can be written as

$$I(t) = \text{Re}\left\{\exp\left[j2\pi f_m \int_0^t g(\tau) d\tau\right]\right\} = \text{Re}\{u(t)\} \\ Q(t) = \text{Im}\{u(t)\} \quad (9)$$

where $u(t)$ is a complex phase function.

The argument of the $I(t)$ and $Q(t)$ signals varies much more slowly than $\Phi(t)$, and this makes it feasible to generate I and Q digitally. After conversion to the analog domain, the final signal $y(t)$ can be generated using a suitable mixer.

In a conventional implementation, the steps outlined in Fig. 1 are followed literally; the filtered data $g(t)$ are generated first, either using an actual filter that approximates the Gaussian characteristic or, more efficiently, using ROM table lookup methods. Integration of $g(t)$ is done in a "phase accumulator," and the cosine and sine are obtained from a sine ROM, using the accumulator value as an address.

In our implementation, these steps are skipped, and the I and Q signals are generated directly from the binary data, thereby eliminating the errors in filtering, phase truncation and sine/cosine computation inherent in the conventional architecture.

To develop the method, we use some properties of Gaussian filters described in the next section.

III. PROPERTIES OF THE GAUSSIAN FILTER

In this section we review some important properties and parameters of a Gaussian filter. For more detail, please see [1].

Transfer Function:

$$H(\omega) = A_0 e^{-\alpha \omega^2} e^{-j\omega t_0} \quad (10)$$

The system has linear phase, with group delay t_0 . The parameter α defines the bandwidth.

3 dB Bandwidth:

$$B_{3dB} = \sqrt{\frac{\ln(2)}{2\alpha}} \quad (11)$$

This is the frequency at which $|H(\omega)|$ is $0.707A_0$.

Noise Bandwidth: With white noise exciting the system, the total power at the output of the filter can be found from $H(\omega)$. The same power is obtained from a brickwall filter of gain $H(0) = A_0$ and bandwidth B_n , which is defined as the noise bandwidth

$$B_n = \frac{\int_0^\infty |H(\omega)|^2 d\omega}{A_0^2} \quad (12)$$

For the Gaussian case, this reduces to

$$B_n = \frac{1}{2} \sqrt{\frac{\pi}{2\alpha}} \quad (13)$$

The ratio of the 3 dB bandwidth to the noise bandwidth is

$$\frac{B_{3dB}}{B_n} = 2\sqrt{\frac{\ln(2)}{\pi}} = 0.93944. \quad (14)$$

Normalized 3 dB Bandwidth: This is defined as

$$B_b T = \frac{B_{3dB}}{2\pi f_b} \quad (15)$$

where $B_b = B_{3dB}/2\pi$ and $T = T_b = 1/f_b$. $B_b T$ is a common design parameter. The parameter α can be found from $B_b T$ from the following relationship

$$\alpha = \frac{\ln(2)}{2(B_b T \cdot 2\pi f_b)^2} \quad (15a)$$

Normalized Noise Bandwidth: This is defined as the noise bandwidth divided by the bit rate f_b of the system

$$B_N T = \frac{B_n}{2\pi f_b} \quad (16)$$

Some workers use $B_N T$ instead of $B_b T$ as a design parameter. The two are related by (14). The parameter α can be found from $B_N T$ from the following relationship

$$\alpha = \frac{\pi}{2(B_N T \cdot 4\pi f_b)^2} \quad (16a)$$

Impulse Response: The following functions form a Fourier Transform pair

$$e^{-\alpha t^2} \leftrightarrow \sqrt{\frac{\pi}{\alpha}} e^{-\frac{\omega^2}{4\alpha}} \quad (17)$$

Comparing the RHS with (10) and recalling that the complex exponential factor is equivalent to a time shift, one can obtain

$$h(t) = \mathcal{F}^{-1}\{H(\omega)\} = \frac{A_0}{\sqrt{\pi}} \beta e^{-[(t-t_0)\beta]^2} \quad (18)$$

where

$$\beta = \sqrt{\frac{2}{\pi}} B_n = 2\sqrt{2\pi} f_b (B_N T) = \sqrt{\frac{2}{\ln(2)}} \pi f_b (B_b T). \quad (19)$$

Note that $h(t)$ is symmetric around $t = t_0$.

Step Response: For a linear phase system in general, the step response $a(t)$ is related to the impulse response $h(t)$ by

$$a(t) = \frac{A_0}{2} + \int_{t_0}^{\infty} h(\tau) d\tau. \quad (20)$$

Here, A_0 is the value of the magnitude of $H(\omega)$ at $\omega = 0$. Substitution of (18) into (20) yields

$$a(t) = \frac{A_0}{2} [1 + \text{erf}(\beta[t - t_0])] \quad (21)$$

where $\text{erf}()$ is the *error function*, defined as [2]

$$\text{erf}(x) = \int_0^x \frac{2}{\sqrt{\pi}} e^{-x^2} dx = -\text{erf}(-x). \quad (22)$$

Note that $a(t)$ is symmetric in the following sense

$$a(t_0 + \tau) + a(t_0 - \tau) = A_0 \quad (23)$$

This follows from (21) and (22).

Response to a Pulse: Since a pulse can be expressed as the difference of two step functions and the filter is a linear system, the response $p(t)$ to a pulse $P(t)$ of duration T_b can be found to be

$$p(t) = \frac{A_0}{2} \{ \text{erf}[\beta(t - t_0)] - \text{erf}[\beta(t - t_0 - T_b)] \}. \quad (24)$$

Response to a Pulse Train: Linearity also allows us to apply superposition to find the response to a train of NRZ pulses of the form of (1). We express the response to a pulse train varying from $-A_0$ to A_0 as the response to a pulse train varying from 0 to $2A_0$ plus a constant of value $-A_0$

$$g(t) = A_0 \sum_n a(n) \{ \text{erf}[\beta(t - t_0 - nT_b)] - \text{erf}[\beta(t - t_0 - [n+1]T_b)] \} - A_0 \quad (25)$$

From this point on, we make $A_0 = 1$ without loss of generality. This means $g(t)$ has a maximum peak value of 1, which in turn means that $f(t)$ varies from $f_c - f_m$ to $f_c + f_m$ according to (4).

IV. FREQUENCY AND PHASE TRAJECTORIES

Equation (24) shows that the pulse response extends theoretically from $-\infty$ to $+\infty$, so the response to a pulse train inside a given symbol interval has interference from all past and future symbols. In practice, however, the extent of the pulse response is usually considered limited to only the two nearest neighbors, and this is reflected in the fact that only three consecutive bits are used at a time. This approximation has been used even for $B_b T = 0.3$ [8] and it is equivalent to multiplying the actual pulses by a rectangular one of duration $3T_b$. Values of $B_b T$ significantly lower than this limit result in more appreciable interference from the second-nearest neighbors and more difficulty in achieving reliable demodulation due to eye closure, so practical systems tend to use values of $B_b T$ as large as possible.

If we impose the condition that the response within a symbol interval ($mT_b, mT_b + T_b$) be dependent only on

TABLE I

Index	Equation	Comments
000	$f_0(t) = -1$	$f_0(t) = -f_7(t)$
001	$f_1(t) = \text{erf}(\beta[t - T_b])$	$f_1(t) = -f_6(t)$
010	$f_2(t) = \text{erf}(\beta[t]) - \text{erf}(\beta[t - T_b]) - 1$	
011	$f_3(t) = \text{erf}(\beta[t])$	
100	$f_4(t) = -\text{erf}(\beta[t])$	$f_4(t) = -f_5(t)$
101	$f_5(t) = 1 - \text{erf}(\beta[t]) + \text{erf}(\beta[t - T_b])$	$f_5(t) = -f_2(t)$
110	$f_6(t) = -\text{erf}(\beta[t - T_b])$	$f_6(t) = -f_1(t)$
111	$f_7(t) = 1$	

(a_{m-1}, a_m, a_{m+1}), it can be shown that this is equivalent to requiring that

$$\text{erf}(\beta x) \approx 1 \quad \text{for } |x| \geq T_b. \quad (26)$$

In particular, this implies

$$\text{erf}(\beta T_b) \cong 1 \quad (27)$$

where

$$\beta T_b = 2\sqrt{2\pi}(B_b T) \quad (28)$$

For example, $B_b T = 0.3$ results in a 2.4% error in (27), while $B_b T = 0.337$ gives a 1% error and $B_b T = 0.452$ results in 0.1% error. A value of $B_b T = 0.7$, which is used in our application, gives negligible deviation.

If (27) is true, then the response in the interval considered can be expressed as

$$g_m(t) \cong a_{m-1}[1 - \text{erf}(\beta t)] + a_m[\text{erf}(\beta t) - \text{erf}(\beta[t - T_b])] + a_{m+1}[1 + \text{erf}(\beta[t - T_b])] - 1. \quad (29)$$

A. Frequency Trajectories

As the pulses can have one of two possible polarities ($a_m = 0$ or 1), we can have $2^3 = 8$ possible curves. Since the frequency of the modulated signal has to be proportional to $g(t)$ [see (4)], the curves are called "frequency trajectories".

Equations for the trajectories can be obtained from (29) substituting particular values for the a_m .

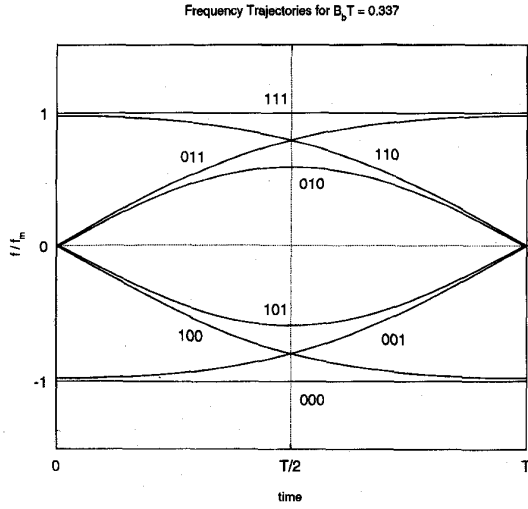
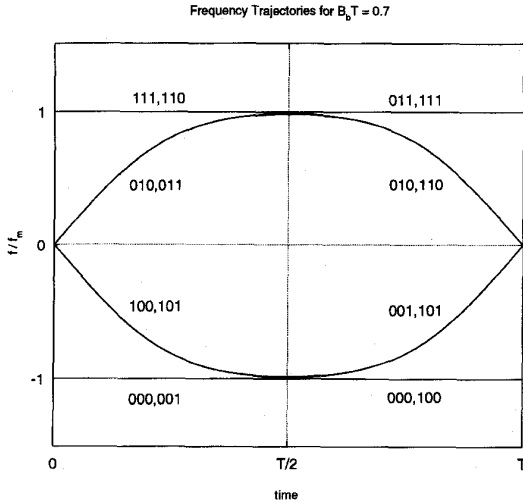
We can show that there are only three independent frequency trajectories, from which the remaining 5 can be obtained by means of simple operations, provided (27) is met.

The concept of frequency trajectory allows us to express the modulating signal $g(t)$ as a train of baseband pulses, as follows

$$g(t) = \sum_n l(n)(t - nT)P(t - nT) \quad (30)$$

where $l(n) = 4a_{n-1} + 2a_n + a_{n+1}$ is the index of the trajectory. The equations for the 8 paths are given in Table I

Trajectories 2, 3, and 7 are calculated first. The remaining can be found from f_2 , f_3 , and f_7 as follows, provided (27) holds. First, trajectories corresponding to complemented triplets are just the negative of the trajectory corresponding to the original triplet (a_{n-1}, a_n, a_{n+1}). Trajectories 0, 5, and 4

Fig. 2. Frequency trajectories for $B_b T = 0.337$.Fig. 3. Frequency trajectories for $B_b T = 0.7$.

are obtained in this way. Trajectories corresponding to time-reversed triplets are also time-reversed in the sense explained in the table comments. This applies to trajectories 6 and 1.

All frequency trajectories can be obtained from f_2, f_3 , and f_7 by sign change and/or time reflection. The frequency trajectories obtained from these equations are shown in Figs. 2 and 3 for two different values of $B_b T$.

B. Phase Trajectories

The complex phase function $u(t)$ can be expressed as a train of baseband pulses, in much the same way as is done with a simple FSK signal [3].

$$u(t) = \sum_n u_n(t - nT)P(t - nT) \quad (31)$$

where $u(t)$ is defined in (9).

To find the value of $u(t)$ in the interval $(nT, [n+1]T)$ we first separate the integral

$$u(t) = \exp \left\{ j2\pi f_m \left[\int_0^{nT} g(\tau) d\tau + \int_{nT}^{nT+t} g(\tau) d\tau \right] \right\} \\ = u(nT) \exp \left\{ j2\pi f_m \int_{nT}^{nT+t} g(\tau) d\tau \right\}. \quad (32)$$

The first integral in the exponent is evaluated over an integer number of symbol intervals, and is just the cumulative sum of phase changes up to time nT

$$2\pi f_m \int_0^{nT} g(\tau) d\tau = \sum_{k=1}^n \Phi_{l(k)} = \Theta_n \quad (33)$$

where $\Phi_{l(k)}$ is the incremental change in phase over the interval $(k-1)T \leq t < kT$, and depends on the particular frequency trajectory in that interval, $f_{l(k)}$, where

$$l(k) \in \{0, 1, \dots, 7\}. \quad (34)$$

The phase $u(t)$ in the interval $(nT, [n+1]T)$, which we denote as $u_n(t)$, can then be expressed as

$$u_n(t) = \exp\{j\Theta_n\} \exp \left\{ j2\pi f_m \int_{nT}^{nT+t} g(\tau) d\tau \right\} \quad (35)$$

and the total complex phase function can be written as

$$u(t) = \sum_{n=-\infty}^{\infty} \exp \left\{ j \left[\Theta_n + 2\pi f_m \int_{nT}^t g(\tau) d\tau \right] \right\} P(t - nT). \quad (36)$$

From (31), (36) it follows that

$$u(t) = \sum_{n=-\infty}^{\infty} \exp \{ j[\phi_{l(n)}(t - nT)] \} P(t - nT) \quad (37)$$

where $\phi_{l(n)}(t)$ will be called a "phase trajectory"

$$\phi_i(t) = 2\pi f_m \int_0^t f_i(\tau) d\tau + \phi_i(0) = 2\pi f_m \phi_i^v(t) + \phi_i(0). \quad (38)$$

Here $\phi_i(0)$ is the initial phase value Θ_n , which depends on the position of the trajectory in time, or equivalently, on the history of the sequence $a(m)$. The time-varying component $\phi_i^v(t)$ depends only on the frequency trajectory during the interval under consideration and is not otherwise dependent on the history of $a(m)$.

To determine the $\phi_i(0)$, we need the phase change over one complete symbol interval, defined to be consistent with (33) as

$$\Phi_i = \phi_i(T_b) - \phi_i(0) = 2\pi f_m \phi_i^v(T_b). \quad (39)$$

Since all frequency trajectories can be expressed in terms of

TABLE II

trajectory	Equation	Final value
000	$\phi''_0(t) = -\phi''_7(t) = -t$	$\Phi_0 = 2\pi f_m \phi''_0(T_b) = -\frac{\pi}{2}$
001	$\phi''_1(t) \equiv I_1(\beta, t - T_b) - T_b = -\phi''_6(t)$	$\Phi_1 \equiv 2\pi f_m [I_1(\beta, 0) - T_b]$ $= -\frac{\pi}{2} + 2\frac{f_m \sqrt{\pi}}{\beta} = -\Phi_6$
010	$\phi''_2(t) \equiv I_1(\beta, t) - I_1(\beta, 0) - I_1(\beta, t - T_b) + T_b - t$	$\Phi_2 \equiv \frac{\pi}{2} - 4\frac{f_m \sqrt{\pi}}{\beta} = \frac{\pi}{2} - 4\Delta$
011	$\phi''_3(t) \equiv I_1(\beta, t) - I_1(\beta, 0)$	$\Phi_3 \equiv 2\pi f_m [-\frac{1}{\beta\sqrt{\pi}} + T_b]$ $= \frac{\pi}{2} - 2\Delta$
100	$\phi''_4(t) \equiv -I_1(\beta, t) + I_1(\beta, 0) = -\phi''_5(t)$	$\Phi_4 \equiv 2\pi f_m [I_1(\beta, 0) - T_b]$ $= 2\frac{f_m \sqrt{\pi}}{\beta} - \frac{\pi}{2} = -\Phi_5$
101	$\phi''_5(t) \equiv t - I_1(\beta, t) + I_1(\beta, 0) + I_1(\beta, t - T_b) - T_b = -\phi''_2(t)$	$\Phi_5 \equiv 2\pi f_m [2I_1(\beta, 0) - T_b]$ $= 4\frac{f_m \sqrt{\pi}}{\beta} - \frac{\pi}{2} = -\Phi_2$
110	$\phi''_6(t) \equiv -I_1(\beta, t - T_b) + T_b$	$\Phi_6 \equiv 2\pi f_m [T_b - I_1(\beta, 0)]$ $= \frac{\pi}{2} - 2\frac{f_m \sqrt{\pi}}{\beta} = \Phi_3$
111	$\phi''_7(t) \equiv t$	$\Phi_7 = 2\pi f_m T_b = \frac{\pi}{2}$

$\text{erf}(x)$, the following integral is useful ([4])

$$I_1(\beta, x) = \int \text{erf}(\beta x) dx = x \text{erf}(\beta x) + \frac{e^{-\beta^2 x^2}}{\beta\sqrt{\pi}} \quad (40)$$

special cases of interest are

$$I_1(\beta, 0) = \frac{1}{\beta\sqrt{\pi}} \quad (41a)$$

$$I_1(\beta, -x) = I_1(\beta, x). \quad (41b)$$

We are concerned with the case where $\text{erf}(\beta x) \sim 1$ (26). One can make use of the following asymptotic expansion ([5]) for positive x

$$\sqrt{\pi}\beta x e^{(\beta x)^2} (1 - \text{erf}[\beta x]) = 1 + \sum_{m=1}^{\infty} (-1)^m \frac{1 \cdot 3 \cdots (2m-1)}{[2(\beta x)^2]^m} \quad (42)$$

to show that in that case (33) is approximately equal to

$$I_1(\beta, x) \cong x + \frac{e^{-(\beta x)^2}}{2\beta\sqrt{\pi}(\beta x)^2} \cong x. \quad (43)$$

With these definitions, we proceed to find the time-varying components of the phase trajectories and the total phase change over a symbol interval. The validity of (27) is assumed throughout. After considering one trajectory in some detail, we summarize them in Table II.

Trajectory for (010):

$$\phi''_2(t) \equiv I_1(\beta, t) - I_1(\beta, 0) - I_1(\beta, t - T_b) + T_b - t. \quad (44)$$

Final Value

$$\Phi_2 \cong 2\pi f_m [T_b - 2I_1(\beta, 0)]. \quad (45)$$

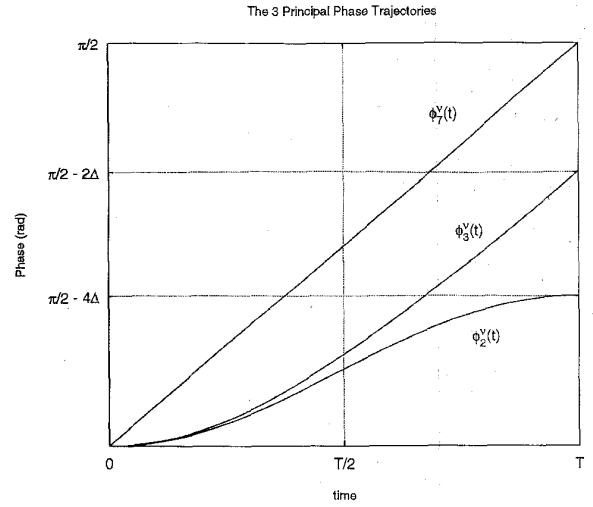


Fig. 4. The three principal phase trajectories for $B_b T = 0.337$.

Using the fact that $f_m = f_b/4$ in GMSK, we have

$$\Phi_2 \cong \frac{\pi}{2} - 4\frac{f_m \sqrt{\pi}}{\beta} = \frac{\pi}{2} - 4\Delta \quad (46)$$

where

$$\Delta = \frac{f_m \sqrt{\pi}}{\beta} = \frac{1}{2\sqrt{2}} \frac{1}{4(B_b T)} = \sqrt{\frac{\ln(2)}{2\pi}} \frac{1}{4(B_b T)}. \quad (47)$$

This quantity also appears in other final value expressions.

The approximate trajectory for (110) can be obtained from ϕ_3 using the relationship connecting both trajectories, as shown in Table I under comments. We have

$$\phi_6^v(t) = \int_0^t f_6(\tau) d\tau = - \int_0^t f_3(\tau - T_b) d\tau \quad (48a)$$

$$\phi_6^v(t) = -\phi_3^v(t - T_b) + \phi_3^v(T_b). \quad (48b)$$

Similarly, the trajectory for (001) can be obtained from $\phi_3(t)$ or $\phi_6(t)$, through the connections given in that table.

From Table I, it can be seen that all phase trajectories can be obtained from $\phi_2(t)$, $\phi_3(t)$ and $\phi_7(t)$ by sign change, time reflection and shifting.

The three principal phase trajectories are shown in Fig. 4 for $B_b T = 0.337$, $f_b = 72$ kHz.

C. Phase Trellis

The sequence of possible phase trajectories is constrained by the fact that only one of the three bits that determines them changes at any one symbol interval. For example, the sequence 001–100 is not permitted; only 001–010 and 001–011 are.

Since we are interested in the sine and cosine of the phase, its values can be restricted to $(0, 2\pi)$ or $(-\pi, \pi)$. If the phase trajectories (modulo 2π) for all possible bit sequences are superimposed, a discrete set of paths or “trellis” emerges. The phase values at the symbol interval boundaries, taken modulo 2π , are members of a discrete, finite set. There is one set for

TABLE III

(a_{m+1}, a_m, a_{m+1})	can occur after	trajectory	phase change	initial condition for T_0	initial condition for T_1
000	000,100	$\phi_0(t)$	$-\frac{\pi}{2}$	0	$\frac{\pi}{2}$
001	000,100	$\phi_1(t)$	$-\frac{\pi}{2} + 2\Delta$	0	$\frac{\pi}{2}$
010	001,101	$\phi_2(t)$	$\frac{\pi}{2} - 4\Delta$	$\frac{\pi}{2} + 2\Delta$	$-\frac{\pi}{2} + 2\Delta$
011	001,101	$\phi_3(t)$	$\frac{\pi}{2} - 2\Delta$	2Δ	$-\frac{\pi}{2} + 2\Delta$
100	010,110	$\phi_4(t)$	$-\frac{\pi}{2} + 2\Delta$	-2Δ	$\frac{\pi}{2} - 2\Delta$
101	010,110	$\phi_5(t)$	$-\frac{\pi}{2} + 4\Delta$	-2Δ	$\frac{\pi}{2} - 2\Delta$
110	011,111	$\phi_6(t)$	$\frac{\pi}{2} - 2\Delta$	0	$-\frac{\pi}{2}$
111	011,111	$\phi_7(t)$	$\frac{\pi}{2}$	0	$-\frac{\pi}{2}$

even intervals and another for odd intervals, hence the trellis is periodic with a two interval period.

To construct the phase trellis one can start at a symbol boundary with some arbitrary initial phase value (say 0) and assume that the two previous bits have been 0's or 1's. The first interval will be labeled T_0 ; T_0 , T_2 , etc. will be called "even" intervals, and T_1 , T_3 , etc. "odd" intervals.

As an aid, we list in Table III the phase change over one symbol interval and the initial conditions for intervals T_0 and T_1 .

With this information the phase trellis can be constructed as shown in Fig. 4. To verify that the phase values at the boundaries are indeed discrete, an arbitrary valid bit sequence can be constructed. Using the expressions for the final values of the trajectories (which assume (27) is valid), it can then be shown that the accumulated phase values fall inside the original set.

For example,

- a) The sequence 000-001-010-100 starting at 0 phase results in a final value

$$\Phi_0 + \Phi_1 + \Phi_2 + \Phi_4 = -\pi.$$

- b) The sequence 110-101-011-111 yields

$$\Phi_6 + \Phi_5 + \Phi_3 + \Phi_7 = \pi.$$

and so on. It can be shown that the possible phase values at the boundaries are always in the sets

$$S_1 = \{-\pi, -\pi + 2\Delta, -2\Delta, 0, 2\Delta, \pi - 2\Delta, \pi\} \quad (49a)$$

for even intervals, and,

$$S_2 = \{-\pi/2 - 2\Delta, -\pi/2, -\pi/2 + 2\Delta, \pi/2 - 2\Delta, \pi/2, \pi/2 + 2\Delta\} \quad (49b)$$

for odd intervals.

V. SYNTHESIS OF THE I AND Q SIGNALS

From the phase function, the sine and cosine must be obtained to generate the I and Q signals. We describe in detail

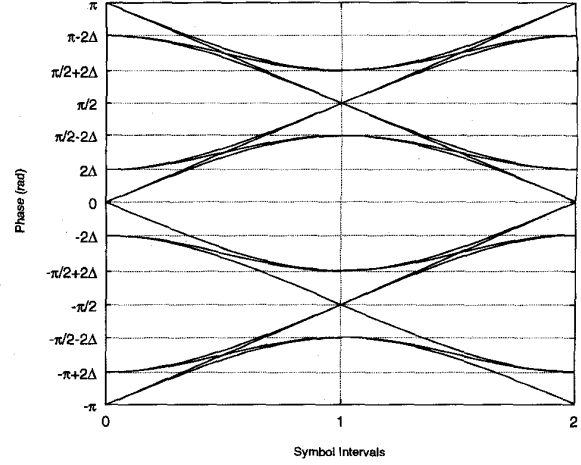
Phase Trellis for GMSK ($B_b T = 0.337$)Fig. 5. Phase trellis for GMSK ($B_b T = 0.337$).

TABLE IV

original trajectory	Θ_n for even intervals	Θ_n for odd intervals	redefined trajectory	Ψ_n for even intervals	Ψ_n for odd intervals
ϕ_0	0, π	$\pi/2, -\pi/2$	ϕ_0	0, π	$\pm \pi/2$
ϕ_1	0, π	$\pi/2, -\pi/2$	ϕ_1	0, π	$\pm \pi/2$
ϕ_2	$-\pi + 2\Delta, 2\Delta$	$\pi/2 + 2\Delta, -\pi/2 + 2\Delta$	ϕ_2	$-\pi, 0$	$\pm \pi/2$
ϕ_3	$-\pi + 2\Delta, 2\Delta$	$\pi/2 + 2\Delta, -\pi/2 + 2\Delta$	ϕ_3	$-\pi, 0$	$\pm \pi/2$
ϕ_4	$\pi - 2\Delta, -2\Delta$	$-\pi/2 - 2\Delta, \pi/2 - 2\Delta$	ϕ_4	$\pi, 0$	$\pm \pi/2$
ϕ_5	$\pi - 2\Delta, -2\Delta$	$-\pi/2 - 2\Delta, \pi/2 - 2\Delta$	ϕ_5	$\pi, 0$	$\pm \pi/2$
ϕ_6	0, $-\pi$	$-\pi/2, \pi/2$	ϕ_6	0, $-\pi$	$\pm \pi/2$
ϕ_7	0, $-\pi$	$-\pi/2, \pi/2$	ϕ_7	0, $-\pi$	$\pm \pi/2$

the I signal, which can be written as in (9) and (37)

$$\begin{aligned} I(t) &= \text{Re}\{u(t)\} \\ &= \sum_{n=-\infty}^{\infty} \cos[\Theta_n + 2\pi f_m \phi_{l(n)}^v(t - nT)] P(t - nT) \end{aligned} \quad (50)$$

As explained before, the Θ_n are always elements of the two discrete sets of boundary phase values S_1 and S_2 . Not only that; from that discussion it can be shown that each particular phase trajectory is associated with only 4 possible initial conditions, as shown in the first 3 columns of the following Table IV. It is convenient at this point to redefine the trajectories 2, 3, 4, and 5 so as to include 2Δ , and leave this quantity out of the initial conditions

$$I(t) = \sum_{n=-\infty}^{\infty} \cos[\Psi_n + \varphi_{l(n)}(t - nT)] P(t - nT) \quad (51)$$

where

$$\begin{aligned} \varphi_{l(n)}(t) &= 2\pi f_m \phi_{l(n)}^v(t) & l(n) &= 0, 1, 6, 7 \\ \varphi_{l(n)}(t) &= 2\pi f_m \phi_{l(n)}^v(t) + 2\Delta & l(n) &= 2, 3 \\ \varphi_{l(n)}(t) &= 2\pi f_m \phi_{l(n)}^v(t) - 2\Delta & l(n) &= 4, 5 \end{aligned} \quad (52)$$

TABLE V
GENERATION OF $I(t)$

curve	sign change	time reversal	obtain	curve	sign change	time reversal	obtain
$\sin(\varphi_7[t])$	-	-	$\cos(\varphi_6[t] + \frac{\pi}{2}), \cos(\varphi_7[t] + \frac{\pi}{2})$	$\sin(\varphi_3[t])$	-	-	$\cos(\varphi_3[t] - \frac{\pi}{2}), \cos(\varphi_4[t] + \frac{\pi}{2})$
	γ	-	$\cos(\varphi_6[t] - \frac{\pi}{2}), \cos(\varphi_7[t] - \frac{\pi}{2})$		γ	-	$\cos(\varphi_3[t] + \frac{\pi}{2}), \cos(\varphi_4[t] - \frac{\pi}{2})$
	-	γ	$\cos(\varphi_6[t] + 0), \cos(\varphi_7[t] + 0)$		-	γ	$\cos(\varphi_3[t] + 0), \cos(\varphi_4[t] + 0)$
	γ	γ	$\cos(\varphi_6[t] + \pi), \cos(\varphi_7[t] + \pi)$		γ	γ	$\cos(\varphi_3[t] + \pi), \cos(\varphi_4[t] - \pi)$
$\sin(\varphi_2[t])$	-	-	$\cos(\varphi_2[t] - \frac{\pi}{2}), \cos(\varphi_5[t] + \frac{\pi}{2})$	$\sin(-\varphi_1[t])$	-	-	$\cos(\varphi_1[t] + \frac{\pi}{2}), \cos(\varphi_6[t] - \frac{\pi}{2})$
	γ	-	$\cos(\varphi_2[t] + \frac{\pi}{2}), \cos(\varphi_5[t] - \frac{\pi}{2})$		γ	-	$\cos(\varphi_1[t] - \frac{\pi}{2}), \cos(\varphi_6[t] + \frac{\pi}{2})$
	-	γ	$\cos(\varphi_2[t] + 0), \cos(\varphi_5[t] + 0)$		-	γ	$\cos(\varphi_1[t] + 0), \cos(\varphi_6[t] + 0)$
	γ	γ	$\cos(\varphi_2[t] + \pi), \cos(\varphi_5[t] - \pi)$		γ	γ	$\cos(\varphi_1[t] - \pi), \cos(\varphi_6[t] + \pi)$

and

$$\begin{aligned} \Psi_n &= \Theta_n & l(n) &= 0, 1, 6, 7 \\ \Psi_n &= \Theta_n - 2\Delta & l(n) &= 2, 3 \\ \Psi_n &= \Theta_n + 2\Delta & l(n) &= 4, 5 \end{aligned} \quad (53)$$

as shown in the last 3 columns of Table IV

To generate $I(t)$ and $Q(t)$, the cosine and sine of the above (redefined) trajectories must be obtained. At first sight, it would appear that a large number of curves must be generated. This is not so, however. First, the bottom 4 curves and their initial conditions are just the negatives of the upper 4. This cuts the number of curves by half. We choose 1, 2, 3, and 7.

Second, we saw that trajectory 1 can be obtained from trajectory 3. In terms of redefined trajectories we get

$$\varphi_1(t) = \varphi_3(t - T_b) - 2\Delta + 2\pi f_m \phi_3^v(T_b) = \varphi_3(T_b - t) - \frac{\pi}{2} \quad (54)$$

or

$$\varphi_3(t) = \varphi_1(T_b - t) + \frac{\pi}{2}. \quad (54a)$$

For the cosine of $\phi_3(t)$ we have, for example,

$$\begin{aligned} \cos[\varphi_3(t)] &= \cos\left[\varphi_1(T_b - t) + \frac{\pi}{2}\right] = -\sin[\varphi_1(T_b - t)] \\ &= -\sin[\varphi_1(t)]_{t \rightarrow T_b - t}. \end{aligned} \quad (55)$$

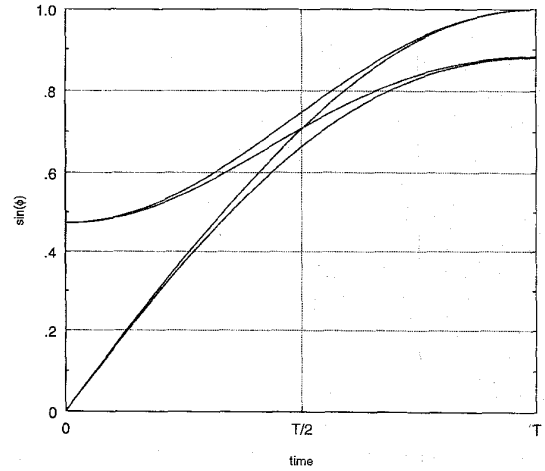
This is the time reversed and inverted version of the sine of trajectory 1. A similar relationship holds for the sine. Thus, only the sine and cosine of trajectory 1 (or 3) are needed.

Further reductions come about in virtue of symmetries in the redefined trajectories, of which 2 and 7 satisfy the following relationship

$$\pm \varphi_i(t) + \frac{\pi}{2} = \varphi_i(T_b - t) \quad i = 2, 7 \quad (56)$$

for some choice of sign. This condition guarantees that

$$\cos(\varphi_i[t]) = \sin(\varphi_i[T_b - t]) \quad i = 2, 7. \quad (57)$$

Basic Curves for $B_b T = 0.337$ Fig. 6. Basic curves for $B_b T = 0.337$.

So, only 4 curves are needed. We select sine of trajectories 2, 7, and 3 and sine of the negative of trajectory 1. These functions are shown in Figs. 6 and 7 plotted as a function of time as fraction of the symbol interval T_b . The values of $B_b T$ used were 0.337 and 0.7.

Table V shows how the four curves can be used to generate the cosines (for the $I(t)$ signal). A similar table can be constructed for $Q(t)$.

VI. CMOS IMPLEMENTATION

A. General Architecture

The block diagram of the proposed implementation is shown in Fig. 8. The core is a ROM that stores the 4 basic curves of Figs. 6 or 7.

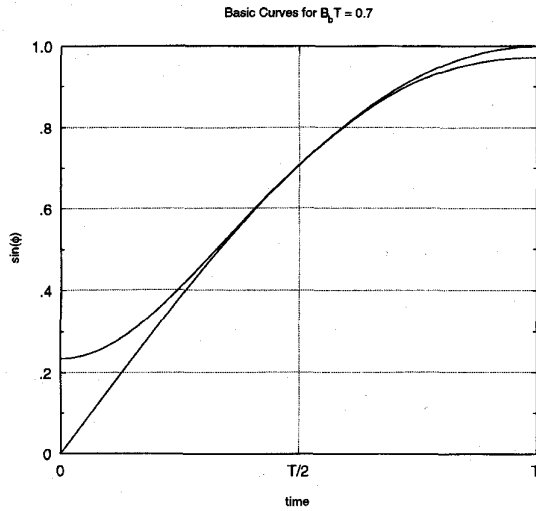
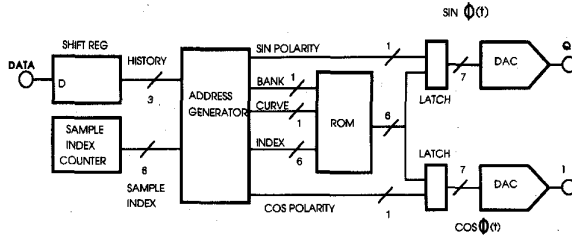
Fig. 7. Basic curves for $B_b T = 0.7$.

Fig. 8. New synthesizer architecture.

B. Look-Up ROM

The ROM size is determined by the DAC sampling rate and resolution, in our case, 64 times the bit rate, and 7 b, respectively. This might indicate that the ROM should store 4 tables of 64 7-b words. However, certain reductions can be made due to symmetries. First, the sign bit is calculated using a simple state machine, reducing the word width to 6. Second, as can be observed in Fig. 9 for the case $B_b T \geq \sim 0.7$, the four basic curves of Fig. 7 are composed of a common set of four half-curves, which can be divided into two banks with a smooth boundary at time $t = 0.5 T$. Each of the four entire curves can be constructed by concatenating one half-curve from each bank using a simple control mechanism. Therefore, the ROM includes 2 banks, each with 2 tables of 32 words of 6-b length. For this application the storage requirements are reduced by a factor of 2 at negligible cost in added logic.

C. Address Generator

The address generator block drives the look-up ROM address by determining from three consecutive data bits which one of the eight frequency trajectories is to be used and translating that information to a series of ROM addresses. The result is valid for one entire bit period and maps directly to a phase trajectory $\Phi(t)$, as previously shown in Section IV. The frequency and phase trajectories themselves never appear in

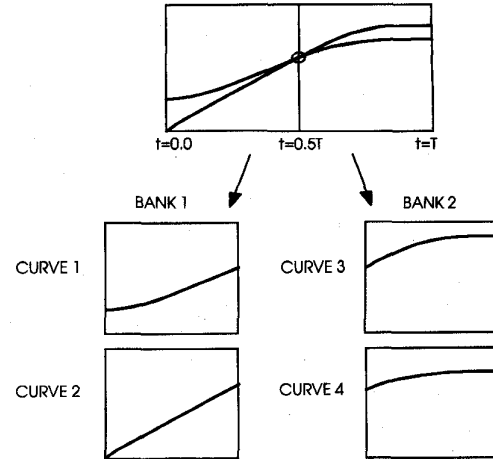


Fig. 9. Half-curves used in synthesis.

the hardware implementation, since only the sine of the phase is needed; however, a state machine tracks the phase quadrant, which either increments or decrements by 1 at the end of each bit interval, in order to provide arithmetic sign information.

The ROM tables store successive samples of the waveform $\sin(\Phi(t))$; therefore, the address generator need only increment the ROM index at the intended sample rate to recreate the waveform. A new ROM bank is selected for each one-half bit, allowing waveform synthesis with the half-curves described in Fig. 9. The same ROM is used for both I (cosine) and Q (sine) synthesis and is addressed twice during each sampling interval, once for the I value and once for the Q value. Since only sine values are stored in ROM, the cosine value is generated by modifying the ROM lookup address by a number equivalent to a 90° phase shift. The phase shift entails choosing a curve from the opposite bank of half-curves and applying a sign inversion and/or a time reversal, depending on the retained phase quadrant. Time reversals are implemented as the one's complement of the address index, i.e., the curve is indexed from right to left in Fig. 9, rather than left to right. The latches at the ROM output synchronize the output samples so that data is presented to both DAC's simultaneously.

D. DAC Sampling Rate and Resolution

The conflicting parameters of sampling rate and resolution on one hand and power consumption and die size on the other were optimized so as to meet overall system performance and to be manufacturable in the available CMOS technology. A high sampling rate is convenient from the RF design point of view, since output spectral images from the D/A conversion will appear relatively distant from the fundamental, allowing better rejection in subsequent processing stages. From the IC design stand-point, however, a high sampling rate implies a larger ROM. In order to guarantee that periodic access to the ROM allows the curves to be synthesized faithfully, the selected sampling rate must be an integral multiple of the data bit rate, in this case 72 kbps. The optimum sampling rate based on silicon and system parameters was determined to

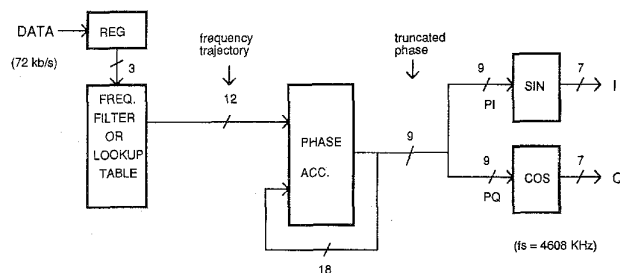


Fig. 10. Typical conventional synthesizer architecture.

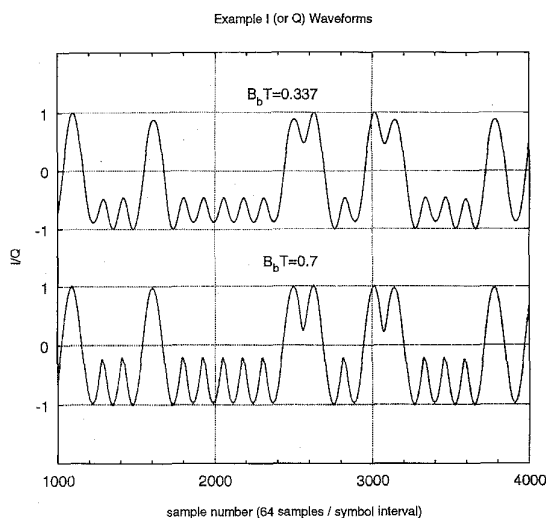


Fig. 11. Sample waveforms.

be 4.608 MHz. The DAC resolution determines the distortion performance, and 6-b plus sign was adequate for the target application.

E. Comparison with Conventional Designs

A conventional synthesizer implementation [8] includes a filter or look-up table to calculate the frequency trajectory, an integrator to calculate the phase $\Phi(t)$ of the modulating signal, and sine and cosine look-up tables to generate the I and Q signals. Fig. 10 represents a conventional digital GMSK modulation technique applied to the following requirements: 72 Kb/s data rate, 4.608 MHz sampling rate, $B_b T = 0.7$, and 7-b DAC resolution. The resolution of the truncated phase and the frequency trajectory shown are for example purposes only, with consideration for the desired phase and quantization error of the I and Q signals. The approach shown has the disadvantages of complexity and inherent errors of the frequency filter/lookup table due to time and amplitude quantization, phase truncation after the integrator, and the relatively large SIN and COS ROM requirements.

The frequency filter or look-up table translates a data sequence to a frequency trajectory, either by calculation or simply by addressing ROM sequentially. Assuming a look-up ROM implementation with 64 samples per trajectory and

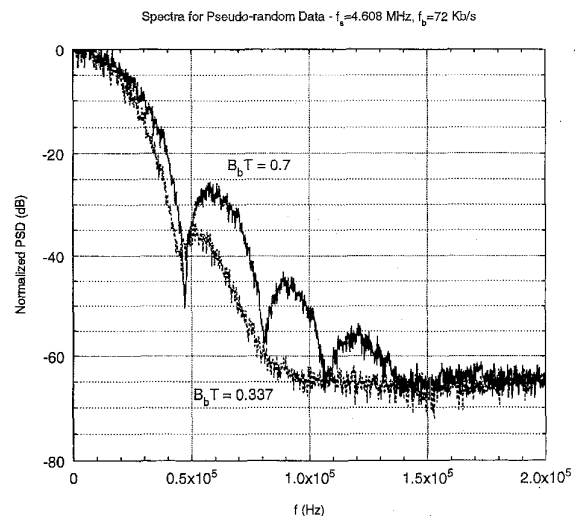


Fig. 12. Spectra for pseudorandom data.

addressing logic to select a trajectory, read it forward or backward, and control the sign, the storage requirements are $2 \times 64 \times 12 = 768$ b.

The frequency trajectory information feeds the phase accumulator. First, we point out that both implementations make the same assumption of insignificant second-nearest-neighbor symbol interference, as evidenced by usage of only three consecutive data bits. However, if the frequency trajectory filter/look-up table is not carefully balanced, the conventional implementation may leave an undesirable residual in the accumulator which will in fact influence further calculations, whereas the method proposed cannot under any circumstance interfere beyond the nearest neighboring symbol.

The conventional technique introduces phase truncation at the accumulator output in order to minimize the SIN and COS look-up table size and complexity. Using a 9-b phase value, the final ROM requirements for the conventional technique are $2 \times 7 \times 512 = 7168$ b. The proposed implementation introduces only quantization error in sine/cosine computation with no error related to phase truncation.

The complexity of the entire control structure of the proposed method is comparable to that of the frequency look-up table addressing mechanism in the conventional design. Without taking advantage of the 50% memory reduction possible with large values of $B_b T$, the ROM requirements of our design are $4 \times 64 \times 6 = 1536$ b, whereas the total memory requirement of the conventional design is roughly five times as large. In conclusion, the hardware advantage of the proposed method is the elimination of the accumulator and a substantial reduction in memory requirements. The principal performance advantage is the elimination of phase errors.

VII. EXPERIMENTAL RESULTS

Experimental results agreed strongly with predictions, with no significant deviations for random input sequences. When the input was held static, causing generation of uninterrupted sinusoidal waveforms at both the I and Q outputs, spectral

analysis revealed power at odd harmonics of the data rate due to the repetitive look-up operation, wherein any quantization error appears at the output as a periodic signal. The ROM table look-up values were designed such that the level of the most powerful harmonic was approximately 50 dB below the intended fundamental. Fig. 11 shows a segment of the I (or Q) output for $B_bT = 0.337$ and $B_bT = 0.7$. Fig. 12 shows the corresponding spectra, estimated from simulation data. The modulator also included additional logic to guarantee smooth ramp-up and ramp-down at the beginning and end of a burst to minimize spectral "splash". The area required was around 3000 mils² (~ 2 mm²) in 0.8 mm CMOS technology, with 1120 mils² (0.72 mm²) for the digital portion and the rest for DAC's, opamps, line drivers, reference, and pads.

REFERENCES

- [1] A. Papoulis, *The Fourier Integral and its Applications*. New York: McGraw-Hill, 1962, ch. 6.
- [2] M. Abramowitz and I. Stegun, Eds.: *Handbook of Mathematical Functions*. New York: Dover, ch. 7, 7.1.1, 7.1.9.
- [3] J. Proakis, *Digital Communications*. New York: McGraw-Hill, 1983, sec. 3-5.
- [4] I. S. Gradshteyn and I. M. Ryzhik *Table of Integrals, Series and Products*. New York: Academic, 1980, p. 633, 5.41.
- [5] See Ref. [2], Eq. 7.1.23.
- [6] S. Pasupathy, "Minimum shift keying: A spectrally efficient modulation," *IEEE Commun. Mag.*, July 1979.

- [7] K. Murota and K. Hirade, "GMSK modulation for digital mobile radio telephony," *IEEE Trans. Commun.*, vol. COM-29, pp. 1044-1050, July 1981.
- [8] J. Haspeslagh *et al.*, "A 270 Kb/s 35-mW modulator IC for GSM cellular radio hand held terminals," *IEEE J. Solid-State Circuits*, vol. 25, pp. 1450-1457, Dec. 1990.



Alfredo Linz (S'73-M'84) was born in Quito, Ecuador. He received the degree of Ingeniero en Electronica y Telecomunicaciones from the Escuela Politecnica Nacional (E.P.N.), Quito, in 1977 (*summa cum laude*). He also received the M.S.E.E. degree in 1981 and the Ph.D. degree in electrical engineering in 1984, both from Southern Methodist University, Dallas, TX.

He is a Senior Member of the Technical Staff of the Integrated Networking Division, Multimedia Group of Advanced Micro Devices, Austin, TX,

where he is engaged in the design of architectures and circuits for DSP based integrated products.

Dr. Linz is a member of Tau Beta Pi.

Alan Hendrickson received the B.S. degree in electrical engineering from Texas A & M University, College Station, in 1987.

He joined Advanced Micro Devices, Austin, TX, in 1989, where his focus has been on modeling and simulating semiconductor devices for digital telecommunications. Current development efforts involve large scale integration for digital cordless telephony using CMOS technology.