42nd Annual Frequency Control Symposium - 1988

The Optimization of Direct Digital Frequency Synthesizer Performance in the Presence of Finite Word Length Effects

Henry T. Nicholas, III
TRW Digital Applications Laboratory
One Space Park, M5/1017
Redondo Beach, CA 90278

Henry Samueli
Electrical Engineering Department
University of California, Los Angeles
Los Angeles, CA 90024

Bruce Kim
Intel Corp.

Abstract

Techniques for optimizing the performance of Direct Digital Frequency Synthesizers (DDFS) that use the Tierney, Rader, Gold architecture [1] are presented. These techniques permit the optimal partitioning of DDFS hardware to minimize the effects of finite phase resolution and finite output word-length. Using these results, given constraints on the phase resolution and output word-length, DDFS performance can be optimized under either a worst case spurious response criterion or under a total signal to noise ratio criterion. These optimization techniques are also used to analyze the errors introduced by the compression of the required look-up table storage size. Two look-up table compression techniques are examined: the Sunderland technique [3][4], and a proposed modification of the Sunderland technique. It is shown that by using a different optimization technique to choose the look-up table samples, the algorithmic error incurred by the Sunderland architecture can be improved by approximately 12 dB. Furthermore, a new compression technique is proposed, based upon an extension of the Sunderland technique, which provides additional improvements in the look-up table storage compression. Finally an actual DDFS design is presented that utilizes this new compression technique and which provides a simulated digital spurious rejection of 90.2 dB, a 14 bit digital output, and a frequency resolution of .023 Hz at the maximum simulated clock frequency of 100 Mhz. This design, which will be fabricated in a radiation hardened 1.25μM CMOS process is presently in the layout phase of design and will enter fabrication in November '88.

Introduction

Direct Digital Frequency Synthesis is an indispensable technique for generating reference frequencies whenever extremely precise frequency resolution and fast switching speed are required. The most common DDFS architecture is the Tierney, Rader, Gold, architecture [1]. This architecture, which has been well documented in the literature [5][6][7], synthesizes a sine wave by using a periodically overflowing 2's complement phase accumulator to generate and store phase information, and uses a Read-Only Memory (ROM) based look-up table to compute the sine function. There are three sources of noise which are inherent to all DDFS implementations, in addition to the noise generated in the D/A conversion process. The first source of noise is P(n), the distortion due to phase truncation at the input to the sine function computing hardware, which is usually a ROM. The second is g(•), which is a nonlinear distortion that is usually present when methods of compressing the storage requirements of the look-up table are employed, and the third is A(n), the noise introduced by the finite precision of the sine samples stored in the look-up table. These noise sources are depicted symbolically in Figure 1. This paper will be concerned with the characterization of these sources and their relation to the hardware requirements.

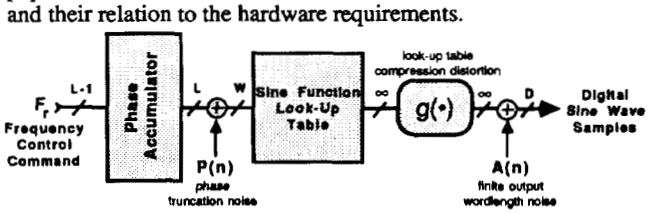

Figure 1. DDFS Finite Word-Length Effects.

Effects of Finite Phase Word-Length

There are two finite word-length effects in the phase accumulator that effect DDFS performance: the word-length of the phase accumulator register, denoted as L in Figure 1, and the truncation of the phase accumulator output, which is used to index the sine function look-up table, which is denoted by B. An understanding of these two finite word-length effects is extremely important because they have a profound impact upon the spectral characteristics of all other system nonlinearities. An exact spectral analysis of DDFS spurious performance was presented in [2], which provided an algorithm for determining the magnitude and position of all spurs generated by phase accumulator truncation. This algorithm is restated in Figure 2 to emphasize some of the subtleties of phase accumulator operation that impact the overall DDFS spurious spectrum.

Number of spurs = $2\Lambda - 1$

Magnitude of all spurs $\zeta_K = \dfrac{\pi\, 2^{-L}\,(F_r,2^B)}{\sin\left(\dfrac{\pi\, K\,(F_r,2^B)}{2^B}\right)}$

Position of the spurs in the spectrum (between 0 and $\dfrac{2^L}{(F_r,2^L)}$)

$K = \left\langle \dfrac{F_n - \Gamma}{2^{L-B}}\,\Gamma\Lambda\text{-}1 \right\rangle_{2\Lambda}$ for $F_n - \Lambda$ divisible by 2

$K = \left\langle \dfrac{-F_n - \Gamma}{2^{L-B}}\,\Gamma\Lambda\text{-}1 \right\rangle_{2\Lambda}$ for $F_n - \Lambda$ divisible by 2

$\zeta_K = 0$ otherwise

B = number of phase bits truncated
L = phase accumulator word length
(a,b) = greatest common divisor of a and b

$\Gamma = \dfrac{F_r}{(F_r,2^B)}$ $\quad \Lambda = \dfrac{2^B-1}{(F_r,2^B)}$

Figure 2. Exact Spectrum of Phase Truncation Spurs

This algorithm provides an expression for the number of spurs generated, their amplitude, and their position in the spectrum as a function of B, the number of bits truncated from the phase accumulator, L, the phase accumulator word-length, and the frequency control word, F_r. These results illuminate some important properties of phase accumulator operation that effect the overall DDFS error spectrum. The most important property to note is that the number of spurs and the magnitude of the spurs depend on F_r only through the greatest common divisor function of F_r and 2^B, which is denoted by $(F_r,2^B)$. As a result of this, for input values of F_r that have the same value of $(F_r,2^B)$, the number of spurs and their respective amplitudes remain the same. Only the position of each spur in the spectrum changes. This phenomenon is a result of certain number theoretic properties of the phase accumulator. These properties are most easily demonstrated if we view the sequence of numbers formed by one numerical period of the phase accumulator as a single time series vector, $\mathbf{X}$. In this representation the length of one numerical period of the phase accumulator is defined as the length of the sequence of numbers that may be output by the phase accumulator before the first number is repeated. As was shown in [2], the length of this

CH2588-2/88/0000-357 $1.00 C 1988 IEEE

sequence is $2^L/(F_r,2^L)$. We then define ${}_1\mathbf{X}$ as the vector of length $2^L/(F_r,2^L)$, which is formed by the consecutive output samples of the phase accumulator when the input is $F_r = 1$. The elements of this vector are then defined using the notation:

$$ {}_1\mathbf{X} = [\ {}_1X_0\ ,\ {}_1X_1\ ,\ {}_1X_2\ ,\ {}_1X_3\ ,\ \bullet\bullet\bullet\ ,\ {}_1X_n\ ]^T \qquad (1) $$

This vector is printed as the transpose of a row vector for typographical convenience. Using this notation, the vector ${}_{F_r}\mathbf{X}$ can be formed by using:

$$ {}_1X_n = \langle F_r n \rangle_{2^L} \qquad (2) $$

to denote the value of each element of the vector from n = 0 to $2^L/(F_r,2^L)$ - 1. To illustrate this concept, the example of a 3 bit accumulator is shown in figure 3. For this example, if F_r is odd then $(F_r,2^L)$ will be 1, thereby generating a time vector of length 8. The column vector on the left represents the time vector that results when F_r=1. For this simple example, the 3 bit accumulator starting with an initial phase accumulator register value of zero, simply adds "1" to the register contents every clock cycle until the register overflows, and returns to zero. This result agrees with equation (2) with the elements ${}_1X_0 = 0$, ${}_1X_1 = 1$, ${}_1X_2 = 2$, ect. The column vector to the right in figure 3 represents the time vector formed when $F_r = 3$. Note that although the phase accumulator accumulates and overflows three times faster for $F_r = 3$ than for $F_r = 1$, the numeric period of the two output sequences are the same.

An extremely important "side effect" of having the same numerical periodicity is that the column vector for $F_r = 3$ can be formed from a permutation of the values of the $F_r = 1$ vector. This permutation is represented by the arrows in Figure 3. It can be shown from the definition of the time vector in equation (2) that any phase accumulator output vector, ${}_a\mathbf{X}$, can be formed from the permutation of another output vector, ${}_b\mathbf{X}$, if $(a,2^L) = (b,2^L)$. The set of permutations that relate these output vectors belongs to a very special class of number theoretic relationships called affine permutations. Equation (2) thus defines the set of affine permutations that relate all time output vectors to the basic output vector, ${}_1\mathbf{X}$, when $(F_r,2^L) = 1$.

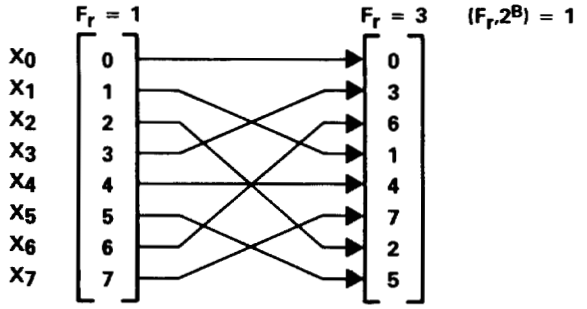

Figure 3. Time Series Vectors for a 3-Bit Accumulator for $F_r = 1$ and $F_r = 3$.

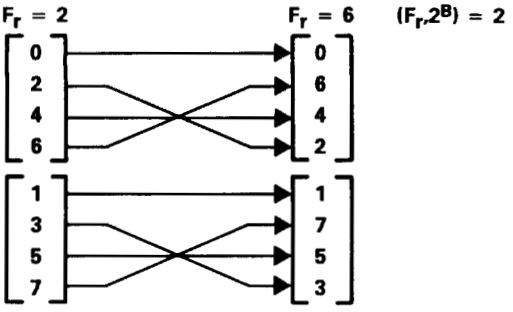

Figure 4. Time Series Vectors for a 3-Bit Accumulator for $F_r = 2$ and $F_r = 6$.

In Figure 4 we see the time vectors formed from values of $F_r = 2$ and $F_r = 6$. From the figure it is evident that the phase accumulator is now characterized by having two different sets of possible output vectors depending upon the initial contents of the phase accumulator. The top set of vectors in the figure represent the sequence of numbers output from the accumulator with an initial accumulator contents of 0. The bottom set of vectors represent the case for an initial accumulator contents of 1. In general, the phase accumulator will be characterized as having $(F_r,2^L)$ different "families" of output sequences to characterize its response, where output sequences belonging to the same "family" are defined as those having the same initial phase accumulator contents. As shown in Figure 4, time vectors belonging to the same family are related by a set of affine permutations of the same form as defined in equation (2).

There are several fundamental implications of these number theoretic properties of the phase accumulator. First, the worst case spurious response of the DDFS as a result of all memoryless system error effects will have exactly $(F_r,2^L)$ different values, one for each initial phase accumulator content from 0 to $(F_r,2^L)$ - 1. This means that simulations of the spectrum of DDFS nonlinearities are only valid for initial phase accumulator contents belonging to the same time series vector.

Another important implication is associated with the affine permutation relationship that links time output vectors of the same family. In equation (3) we see that the time output vector for any value of F_r can be formed from a permutation of the individual elements of the vector for $F_r = 1$ by permuting the vector indices using $i' = \langle F_r i \rangle_{2^L}$.

$$ {}_1\mathbf{X} = [\ {}_1X_0\ ,\ {}_1X_1\ ,\ {}_1X_2\ ,\ {}_1X_3\ ,\ \bullet\bullet\bullet\ ,\ {}_1X_n\ ]^T \quad \text{vector } F_r = 1 \qquad (3) $$

$$ {}_{F_r}\mathbf{X} = [\ {}_1X_{\langle F_r 0\rangle_{2^L}}\ ,\ {}_1X_{\langle F_r 1\rangle_{2^L}}\ ,\ {}_1X_{\langle F_r 2\rangle_{2^L}}\ ,\ \bullet\bullet\bullet\ ,\ {}_1X_{\langle F_r n\rangle_{2^L}}\ ]^T \quad \text{vector for arbitrary } F_r $$

It can be shown that the discrete Fourier transform operator is invariant under permutations of this class applied in both the time and frequency domain. If we assume that in general the DDFS operates by applying some memoryless nonlinear function $S\{\bullet\}$ to the phase accumulator output to produce the sine function, then the DDFS output at every point in time for a given F_r may be represented as function of one of the indices of ${}_{F_r}\mathbf{X}$, $S\{{}_{F_r}X_n\}$. Therefore the spectrum of the DDFS output may be represented as:

$$ N = \frac{2^L}{(F_r,2^L)} \qquad \mathbb{S}\{K\} = \sum_{n=0}^{N-1} S\{{}_{F_r}X_n\}\, e^{-j2\pi Kn/N} \qquad (4) $$

As was shown in equation (3), all input time vectors may be formed from a permutation of another time vector by permuting the indices. If this is applied to the definition of the DDFS output spectrum in (), it can be shown that the following identity holds for affine permutations of the class $\langle F_r i \rangle_{2^L}$:

$$ \mathbb{S}\{\langle F_r K\rangle_{2^L}\} = \sum_{n=0}^{N-1} S\{{}_{F_r}X_{\langle F_r n\rangle_{2^L}}\}\, e^{-j2\pi Kn/N} \qquad (5) $$

Therefore a permutation of the samples in the time domain results in an identical permutation of the DFT samples in the frequency domain. In practical terms this means that the spurious spectrum due to all system nonlinearities can be generated from a permutation of another spectrum of the same number theoretic class. Thus, if we restrict the allowable values of F_r such that $(F_r,2^L)$ is always the same (such as by only allowing F_r to be odd)

then all spurious spectrums may be generated from permutations of a single basic spectrum. If it is desired to optimize the worst case spurious response, then only one spectrum need be calculated for all values of F_r with common values of $(F_r, 2^L)$, since each spectrum will differ only in the position of the spurs and not in the magnitudes.

A simple modification to the basic phase accumulator structure was introduced in [2] that causes all phase accumulator output sequence to belong to the number theoretic class $(F_r, 2^{L+1}) = 1$, regardless of the value of F_r. By using this phase accumulator modification, which is depicted in Figure 5, only one simulation need be performed to determine the value of the worst case spurious response or the total signal to noise ratio due to memoryless system nonlinearities. In addition, this modification has the added advantages of eliminating any dependence of the output spectrum on the initial phase accumulator contents, and of reducing the magnitude of the worst case spurious response by 3.92 dB.

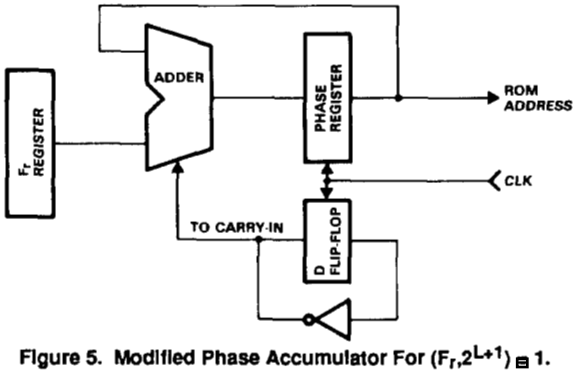

Figure 5. Modified Phase Accumulator For $(F_r, 2^{L+1}) \equiv 1$.

Finite Amplitude Word-Length Effects

The results relating the number theoretic behavior of the phase accumulator finite word-length effects to the DDFS output spectrum may now be applied to help bring about the optimization of the sine sample computing hardware. Two error effects are introduced in computing the sine samples from the phase argument. These are the error from the storage compression nonlinearity, and the error from the amplitude quantization of the sine samples, which are depicted in Figure 1. In the past these error effects have been treated as random phenomena. However, using the results from the previous section, it is now computationally feasible to treat them as a deterministic process.

Look-up Table Storage Compression

Exploitation of Sine Function Symmetry

The most elementary method of sine storage compression is to exploit the symmetry of the sine function about π and $\pi/2$. By properly inverting the phase and the amplitude of the sine function, look-up table samples need only be stored for phase values between 0 and $\pi/2$. The details of this method are shown in Figure 6. As shown in the figure, the sine function between 0 and π may be synthesized from the samples between 0 and $\pi/2$ by taking the phase modulo $\pi/2$ and then taking the absolute value of the phase. This is easily implemented in hardware by truncating the phase MSB and then using the second MSB to full wave rectify the magnitude of the phase. As shown in Figure 6, the sampled waveform at the output of the look-up table is a full wave rectified version of the desired sine wave. The final output sine wave is then generated by multiplying the full wave rectified version by -1 when the phase is between π and 2π. This is accomplished simply by multiplying by the negative of the phase accumulator MSB.

In most practical DDFS digital implementations, numbers are represented in 2's complement format. Therefore the 2's complementor must be used to take the absolute value of the phase and multiply the output of the look-up table by -1. However, it can be shown that if a 1/2 LSB offset is introduced into the number that is to be complemented, then a 1's complementor may be used in place of the 2's complementor without introducing an error. This provides a considerable savings in hardware since a 1's complementor may be implemented as a simple exclusive-or gate. This 1/2 LSB offset is provided by choosing the look-up table samples such that there is a 1/2 LSB offset in both the phase and the amplitude of the samples.

Compression of Sin(θ) for $0 < \theta < \pi/2$

Prior to the publication of this paper, the most effective method of compressing the look-up table storage requirements for $0 < \theta < \pi/2$ was introduced by Sunderland et. al. [3][4]. This architecture, which is shown in Figure 7, provides a reduction in the look-up table storage requirements by replacing the storage requirements of one large ROM of size 2^{A+B+C} words with two smaller ROM's of sizes 2^{A+B} words and 2^{A+C} words, whose outputs are added together to reconstruct the sine function. This provides a storage compression ratio of 45,056 bits to 3,840 bits, or 11.7:1. The assignment of look-up table samples in this architecture is based upon several trigonometric approximations. First, the bits representing the phase argument, θ, of one quarter period of the sine function are decomposed into the sum of three functions: $\alpha < (\pi/2)$, $\beta < (\pi/2)(2^{-A})$, and $\chi < (\pi/2)(2^{-(A+B)})$ such that

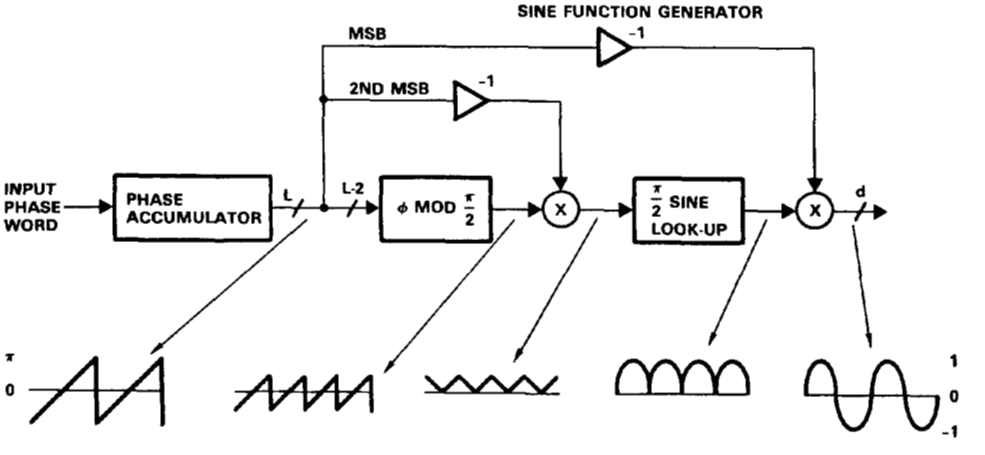

Figure 6. Logic to Exploit Quarter Wave Symmetry.

$\theta = \alpha + \beta + \chi$ [3]. Using trigonometric identities, the sine function is given by

$$\sin(\alpha+\beta+\chi) = \sin(\alpha+\beta)\cos(\chi) + \cos(\alpha)\cos(\beta)\sin(\chi) - \sin(\alpha)\sin(\beta)\sin(\chi) \quad (6)$$

using the relative magnitudes of α, β, and χ, (6) may be approximated by

$$\sin(\alpha+\beta+\chi) \approx \sin(\alpha+\beta) + \cos(\alpha)\sin(\chi) \quad (7)$$

The contents of the first ROM is then chosen to be $\sin(\alpha+\beta)$ and the second ROM stores the result of $\cos(\alpha)\sin(\chi)$. The error resulting from this approximation for a 12 bit output with a phase word segmentation of A = 4, B = 4, and C = 4, which will be denoted as a (4,4,4) segmentation, can be shown to be approximately -72.2 dB.

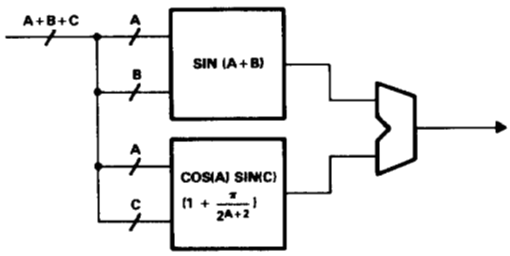

Figure 7. Sunderland Architecture.

However, significant improvements can be achieved, over the former results by using an alternate methodology for choosing the samples to be stored in the ROM's. Improvements in the spurious performance can be gained by modeling the segmentation of the phase arguments into the two ROM's as a low phase resolution course ROM, with additional phase resolution added by providing interpolation between the course phase samples using the fine ROM. We define as before, $\theta = \alpha + \beta + \chi$, with the word-length of the variable α to be A, the word-length of β to be B, and χ to be C. The sine function between 0 and $\pi/2$ is divided into 2^B different regions. In each region the same 2^B fine ROM interpolation samples are used for interpolation between course ROM samples given the same value for the variable χ. An example of this type of coarse-fine interpolation is shown in Figure 8 for the case of a 6 bit phase argument using a segmentation of A = 2, B = 2, and C = 2.

In this example, the course ROM samples are represented by the "heavy dots" along the dashed line, and the fine ROM samples are chosen to be the difference between the value of the sine function along the dashed line and the value value of the "error bars" directly below or above that point on the dashed line. In the example, the sine function is divided into 2^α quadrants, correspond to α = 00, 01, 10, and 11. Within each quadrant, only one correction value may be used between the error bars and the dashed line for all χ offsets from a coarse sample. The value of the offset used for each value of χ is chosen to minimize either the mean square or the maximum error of all offsets for each value of β within the quadrant. The actual algorithm for choosing the fine ROM samples to minimize the mean square error is shown in (8).

$$F_c(\alpha,\beta) = \sin\left(\frac{\pi}{2}\left(\frac{\alpha 2^{B}+\beta}{2^{A+B}} + \frac{1}{2^{A+B+C}}\right)\right) \quad (8)$$

$$F_f(\alpha,\chi) = \sum_{n=0}^{N-1} \frac{1}{2^B}\left[\sin\left(\frac{\pi}{2}\left(\frac{\alpha 2^{B+C}+\beta 2^{C}+\chi}{2^{A+B+C}} + \frac{1}{2^{A+B+C+I}}\right)\right) - F_c(\alpha,\beta)\right]$$

Where in (8) $F_c(\alpha,\beta)$ is defined to be the coarse ROM samples and $F_f(\alpha,\chi)$ is defined to be the fine ROM samples. The optimization criterion for choosing the samples to minimize the maximum absolute error is shown in (9).

$$F_f(\alpha,\chi) = \quad (9)$$
$$\frac{1}{2}\mathrm{MAX}\left\{\sin\left(\frac{\pi}{2}\left(\frac{\alpha 2^{B+C}+\beta 2^{C}+\chi}{2^{A+B+C}} + \frac{1}{2^{A+B+C+I}}\right)\right) - F_c(\alpha,\beta)\right\}$$
$$- \frac{1}{2}\mathrm{MIN}\left\{\sin\left(\frac{\pi}{2}\left(\frac{\alpha 2^{B+C}+\beta 2^{C}+\chi}{2^{A+B+C}} + \frac{1}{2^{A+B+C+I}}\right)\right) - F_c(\alpha,\beta)\right\}$$

Based upon simulation results, it was determined that the minimum-maximum error criterion tended to result in lower values of the maximum spur, although this result has not been explicitly proven. The simulations also showed that the mean square error criterion, provides the lowest total spur energy, in agreement with theory.

This methodology of choosing the coarse and fine ROM samples provides two advantages over previous methods. First, it provides superior spurious performance because the fine ROM samples may be chosen by computer optimization rather than by linear interpolation. Secondly, because this method of storage compression is not derived from trigonometric identities, it may be used to compress the storage of any arbitrary function, not just the sine function.

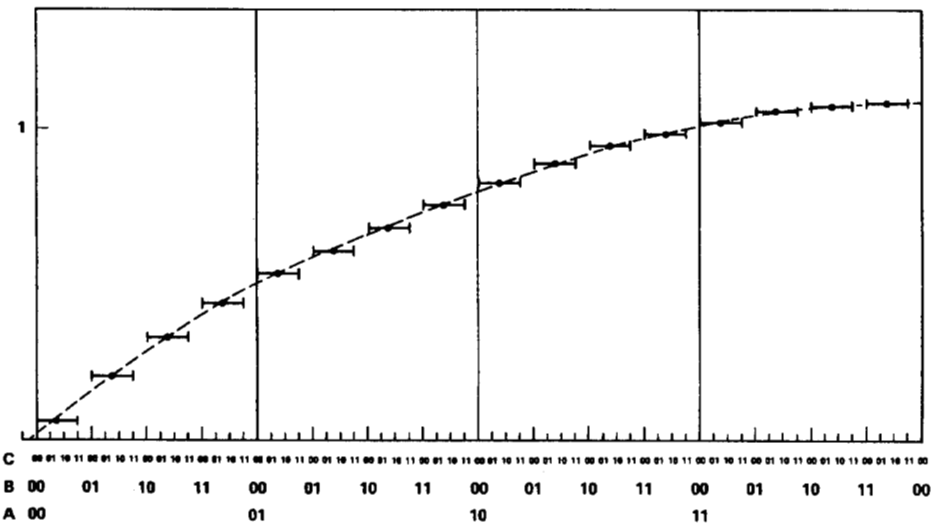

Figure 8. Coarse-Fine Interpolation for (2,2,2) Segmentation of the Phase.

A comparison was performed between the two different methodologies of choosing the ROM samples for the case of a hypothetical design with a phase precision of 12 bits. The segmentation was (4,4,4) for both cases. In the trigonometric derivation, the coarse ROM size was 256 x 11 bits and the fine ROM size was 256 x 4 bits. For the coarse/fine derivation, the coarse ROM size was 256 x 9 bits and the fine ROM size was 256 x 4 bits. The result of the simulations showed a worst case spurious performance of -72.2 dB for the trigonometric generation of the ROM samples and -84.2 dB for the coarse-fine generation of ROM samples. This demonstrates an improvement of 12 dB for the coarse-fine technique while simultaneously requiring less storage.

Sine-Phase Difference Algorithm

Since the coarse-fine algorithm can be used to store any arbitrary function, another technique may be employed to achieve even greater reduction in ROM size. This may be accomplished by storing the the function

$$f(x) = \sin(\pi x/2) - \pi x/2 \qquad (10)$$

instead of the sine function in the look-up table. The sine function is then generated by adding the phase argument from the phase accumulator to the output of the ROM's, as shown in Figure 9. The advantage of this technique is that it reduces the maximum amplitude of the function to be stored in the coarse ROM. Therefore the coarse ROM output word-length may be reduced and the output adder may be simplified. This modification reduces the look-up table storage requirements by at least of 2^{A+B+1} bits. The application of this technique to the hypothetical (4,4,4) segmented 12 bit output DDFS design considered earlier results in a savings of 512 bits in the coarse ROM and an increase in the compression ratio from 11.7:1 to 13.5:1. The penalty for this reduction in ROM storage is the addition of another adder. However, this is an advantageous tradeoff in the high speed VLSI implementation of a DDFS. The ROM propagation delay, which cannot be pipelined, is reduced, increasing the maximum clock frequency of the DDFS. The expense of another adder, which is readily pipelineable, is inconsequential in the full custom VLSI implementation.

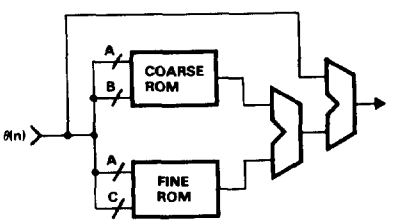

Figure 9. Sine-Phase Difference Algorithm.

Exploitation of Fine ROM Symmetry

The fine ROM size may also be reduced by exploiting the symmetry in the samples. As shown in Figure 10, if the coarse ROM samples are chosen in the middle of the interpolation region, then the fine ROM samples will be approximately symmetric about the $\chi = (2^C - 1)/2$ axis. Thus by using an adder/subtracter instead of and adder to sum the coarse and fine ROM values, the size of the fine ROM may be halved. Some additional complexity must be added to the adder/subtracter control logic if this technique is to be used with the sine-phase difference algorithm, since the slope of the function in equation (10) changes sign at a non symmetric point between 0 and $\pi/2$ on the x-axis. Fortunately, the digital logic required to perform this can be accomplished with less than four logic gates for the 12-bit phase case considered previously.

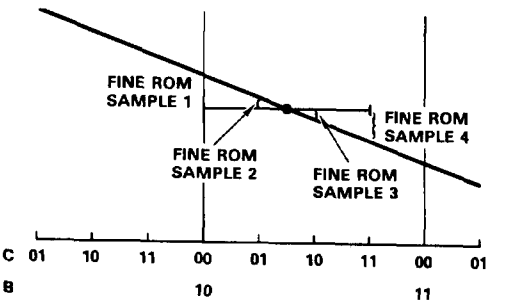

Figure 10. Exploitation of Fine ROM Symmetry.

Since the fine ROM is not generally in the critical speed path, the effective resolution of the fine ROM may be doubled at no extra hardware penalty, rather than halving the ROM. This is an extremely powerful technique because it allows the segmentation of the compression algorithm to be changed, effectively adding an extra bit of phase resolution to the look-up table, which thereby reduces the magnitude of the worst case spur due to phase accumulator truncation.

100 MHz DDFS Architecture Demonstration Vehicle

These architectural techniques were used to design a DDFS to meet a performance goal of better than -90 dB of spurious rejection with the potential of being pipelined to achieve a 100 Mhz clock rate in a standard 1.25 μM CMOS process. A 15-bit phase precision in the look-up table address was chosen, which from Figure 2 is shown to provide -90.3 dB of rejection in the magnitude worst case spur due to phase accumulator truncation. Next, the 15-bit phase must be segmented to address the coarse and fine ROM's. In this design it is assumed that the sine phase difference algorithm will be used, exploiting the symmetry of the fine ROM samples to double the effective fine ROM resolution. It is also assumed that the phase accumulator modification shown in Figure 5 is used, and that the quarter wave symmetry of the sinewave will be exploited, reducing the effective look-up table address requirements to 13 bits of phase precision for $0 < \theta < \pi/2$. To obtain the optimal segmentation of the 13-bit phase address to the coarse-fine ROM's, infinite output word-lengths were assumed in the simulations. Furthermore, it was heuristically determined that optimal segmentations of the phase argument have values of B and C that differ by at most 1, and that segmentations with values of B > C perform better than B < C. Therefore, the optimization space of the word-lengths A, B, and C, were constrained such that B = C or B = C + 1. Subject to these constraints, and that A + B + C = 13, the spurious response of the possible segmentations of the 13-bit phase were simulated. The results show the magnitude of the worst case spur in the spectrum plotted against the word-length variable A, and are pictured in Figure 11. A key concept to note in the graph is that the amount of storage required increases exponentially with increasing A, since the total ROM storage is given by $2^A(2^B + 2^C)$. Therefore the minimum value of A that meets the spurious rejection requirement should be chosen. The value of A = 4 corresponding to -95.33 dB meets the -90 dB requirement, which results in a phase segmentation of A = 4, B = 4, and C = 5. One anomaly in the graph in Figure 11 is the decreasing spurious performance as A is increased past 6 bits. This is a result of the fine ROM symmetry assumptions breaking down for small values of B. A similar set of simulations for a design without the exploitation of fine ROM symmetry show a monotonically increasing spurious performance with increasing values of A, as expected from theory.

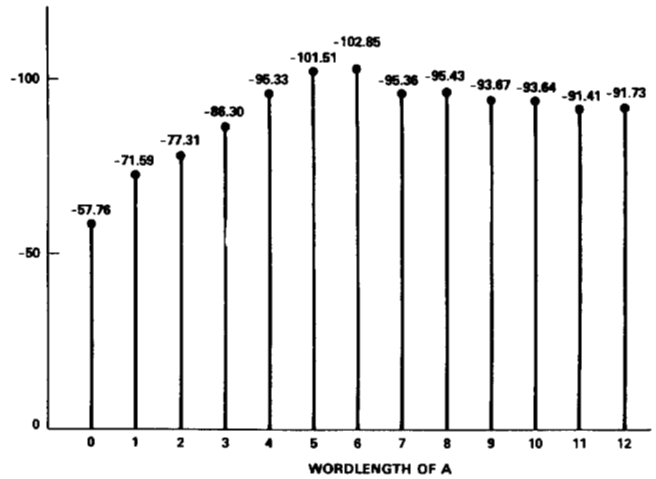

Figure 11. Worst Case Spurious Response for Different Phase Segmentations.

For the (4,4,5) segmentation of the phase, the effect of various output quantizations of the coarse ROM were simulated to determine the minimum word-length required to achieve the -90dB performance goal. The results of these simulations are shown in Figure 12. From these simulations, an output word-length of 9 bits was chosen for the coarse ROM, providing -89.9dB of spurious rejection including the effects of fine ROM quantization. The fine ROM word-length is provided as part of the simulation algorithm, where it is derived from the maximum amplitude of the samples calculated using equation (8). This resulted in a fine ROM word-length of 4 bits for a coarse ROM word-length of 9 bits.

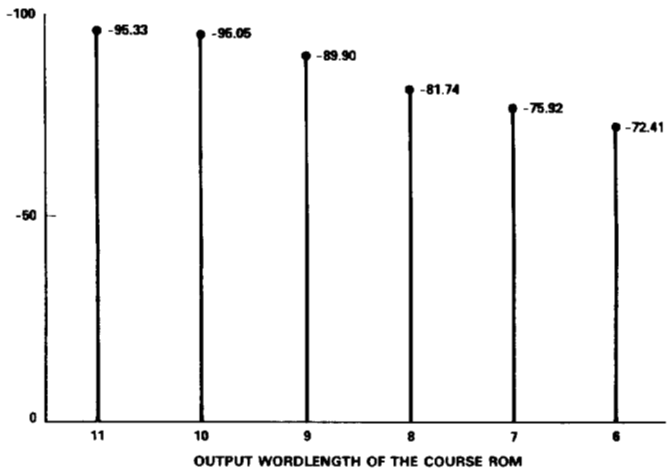

Figure 12. Spurious Response for Different Coarse ROM Output Word-Length Quantizations.

As a final optimization step, the amplitude values of the function to be stored in the look-up table can be scaled to provide improved performance in the presence of amplitude quantization. To accomplish this, the coarse and fine ROM samples were regenerated after multiplying the pre-quantized sinewave by a scaling constant that was varied in very small increments in the neighborhood of unity. The results of these simulations, which are graphed in Figure 13, show that a 1.75 dB improvement in the worst case spurious response can be achieved by using a scaling constant of .999994 instead of 1.0 to generate the coarse and fine ROM samples. This results in a negligible reduction in the amplitude of the synthesizer output, while providing a worst case spurious that is now limited by the spurs due to phase accumulator truncation rather than amplitude quantization.

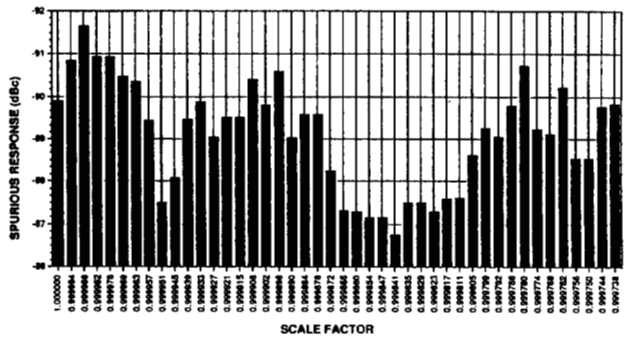

Figure 13. Spurious Improvement From Scaling Sinewave Amplitude.

DDFS Architecture Demonstration Vehicle

Figure 14 shows the detailed logic block diagram of the DDFS design exploiting all architectural innovations discussed in this paper. The design has a simulated worst case spurious response of -90.3 dB using only 3,072 bits of ROM storage, providing a total storage compression ratio of 37:1. The simulated output spectrum of this design is depicted in Figure 15, showing the high degree of digital spurious rejection attained using this architecture. This design provides a 14-bit sinewave output, where the most significant 12 bits represent the ideal 12-bit rounded samples of a sinewave. This design is presently being implemented as a single VLSI chip in TRW's VHSIC 1.25µM CMOS process. Detailed circuit simulations predict the chip will be capable of operating at a clock rate of 100 Mhz, providing a .023 Hz frequency resolution over a 33 MHz bandwidth. The key circuit structures have already been fabricated in TRW's CMOS process as a test chip, shown in Figure 16, and have validated all logic and circuit simulations. The final design is scheduled to enter fabrication in November '88.

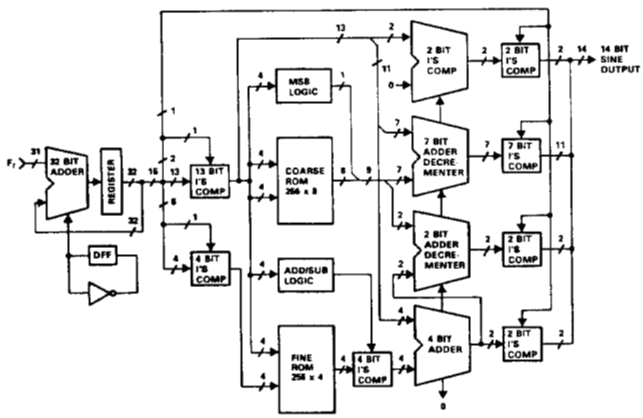

Figure 14. -90.3 dB Spurious DDFS Architecture.

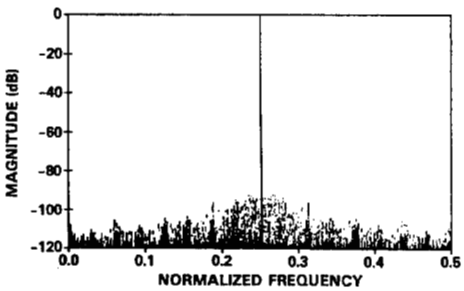

Figure 15. Simulated DDFS Output Spectrum.

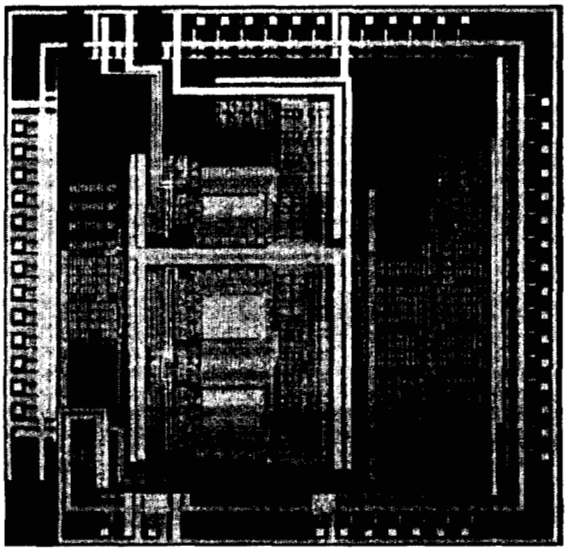

Figure 16. 1.25 μM CMOS DDFS Test Chip.

Summary

Techniques for the design of VLSI architectures for direct digital frequency synthesis have been introduced, which allow for the optimization of the spurious response in the presence of finite word length effects. These optimization techniques exploit certain number theoretic properties of the phase accumulator to make the exhaustive simulation of DDFS performance in the presence of different system nonlinearities computationally feasible. These design techniques have been applied to design a 14-bit output DDFS with a simulated spurious performance of -90.3 dB and a level of pipelining capable of allowing a 100 MHz clock rate in a 1.25 μM CMOS process.

Acknowledgements

The authors wish to thank M. Miscione of TRW Microelectronics Center for providing VLSI fabrication services and partial funding for this research. P. Hadinger and R. Smith of TRW Defense Communications Division also provided partial financial sponsorship and many valuable technical discussions on the systems applications of DDFS's.

References

[1] J. Tierney, C.M. Rader, and B. Gold, "A Digital Frequency Synthesizer," IEEE Trans. Audio Electroacoust., vol. AU-19, pp. 48-57, 1971

[2] H.T. Nicholas III and H. Samueli, "An Analysis of the Output Spectrum of Direct Digital Frequency Synthesizers in the Presence of Phase-Accumulator Truncation," Proc. 41st annual Frequency Control Symposium USERACOM, Ft. Monmouth, N.J., May 1987, pp.495-502

[3] D.A. Sunderland, R.A. Strauch, S.S. Wharfield, H.T. Peterson, and C.R. Cole, "CMOS/SOS Frequency Synthesizer LSI Circuit for Spread Spectrum Communications" IEEE Journal of Solid State Circuits, vol. SC-19, pp.497-505, Aug 1984

[4] R.E. Lundgren, V. S. Reinhardt, and K. W. Martin, "Designs and Architectures for EW/Communication Direct Digital Synthesizers", Research and Development Technical Report SLCET-TR-85-0424-1, U.S. Army Laboratory Command, Fort Monmouth, New Jersey, Aug 1986

[5] Albert L. Bramble, "Direct Digital Frequency Synthesis", Proc. 35th annual Frequency Control Symposium, USERACOM, Ft. Monmouth, N.J., May 1981, pp.406-414

[6] J. Gorski-Popiel, Frequency Synthesis: Techniques and Applications, New York: IEEE press, 1975

[7] V. Manassewitsch, Frequency Synthesizers, Theory and Design, New York: John Wiley, 2nd Ed., 1979

[8] H.T. Nicholas, "The Determination of the Output Spectrum of Direct Digital Frequency Synthesizers in the Presence of Phase Accumulator Truncation", Thesis, Masters of Science in Electrical Engineering, UCLA, 1985.