Methods of Mapping from Phase to Sine Amplitude in Direct Digital Synthesis

Jouko Vankka

***Abstract*—There are many methods for performing functional mapping from phase to sine amplitude (e.g., ROM look-up, coarse/fine segmentation into multiple ROM's, Taylor series, CORDIC algorithm). The spectral purity of the conventional direct digital synthesizer (DDS) is also determined by the resolution of the values stored in the sine table ROM. Therefore, it is desirable to increase the resolution of the ROM. Unfortunately, larger ROM storage means higher power consumption, lower reliability, lower speed, and greatly increased costs. Different memory compression and algorithmic techniques and their effect on distortion and trade-offs are investigated in detail. A computer program has been created to simulate the effects of the memory compression and algorithmic techniques on the output spectrum of the DDS. For each memory compression and algorithmic technique, the worst case spurious response is calculated using the computer program.**

I. Introduction

This paper first gives a description of the conventional direct digital synthesizer. The most elementary technique of compression is to store only $\pi/2$ rad of sine information, and to generate the ROM samples for the full range of 2π by exploiting the quarter-wave symmetry of the sine function. Beyond that, the methods of compressing the quarter wave memory include: a trigonometric identity, the Nicholas method, the use of Taylor series and the CORDIC algorithm. A different approach of phase-to-sine-amplitude mapping is the CORDIC algorithm, which uses an iterative computation method [1]. The penalties paid for different methods are the increased circuit complexity and the distortion due to the sine memory compression. Because the possible number of generated frequencies is large, it is impossible to simulate all of them to find the worst case situation. If the Nicholas modified phase accumulator is used, then only one simulation is needed to determine the worst case signal-to-spur level [2]. In this paper a case of 14-bit phase to 12-bit amplitude mapping is analyzed. A computer program has been created to simulate the effects of the memory compression and algorithmic techniques on the output spectrum of the DDS. For each memory compression and algorithmic technique, the worst case spurious response is calculated using these premises. Some examples of commercial circuits using the above methods are also presented.

Manuscript received April 23, 1996; accepted August 26, 1996.

J. Vankka is with the Helsinki University of Technology, Laboratory of Signal Processing and Computer Technology FIN-02150 Espoo, Finland (e-mail: jvan@mariko.hut.fi).

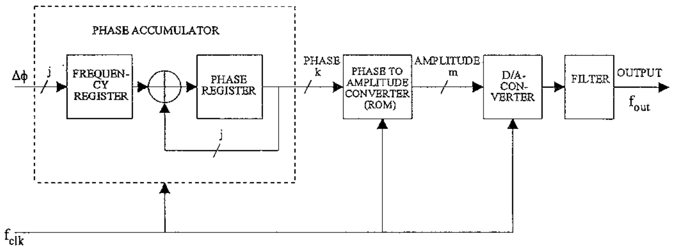

Fig. 1. Simplified block diagram of the direct digital synthesizer.

II. Conventional Direct Digital Synthesizer

The direct digital synthesizer is shown in simplified form in Fig. 1. The DDS has the following basic blocks: phase accumulator, phase-to-amplitude converter (generally a sine ROM), digital-to-analog converter, and filter [3]. The phase accumulator consists of a j-bit frequency register which stores a digital phase increment followed by a j-bit full adder and a phase register. The digital input phase increment is entered in the frequency register. At each clock pulse this value is added to the value previously held in the phase register. The phase increment represents a phase angle step that is added to the value of the previous step at every $1/f_{\text{clk}}$ seconds to produce a linearly increasing digital value. The phase value is generated using the modulo 2^j overflowing property of the j-bit phase accumulator. The rate of the phase accumulator overflows is the output frequency

$$f_{\text{out}} = \frac{\Delta\phi f_{\text{clk}}}{2^j}, \quad \Delta\phi < 2^{j-1}, \tag{1}$$

where $\Delta\phi$ is the phase increment word, f_{clk} is the clock frequency and f_{out} is the output frequency. The constraint in (1) comes from the sampling theorem. The phase increment word in (1) is an integer; therefore, the frequency resolution is found by setting $\Delta\phi = 1$

$$\Delta f = \frac{f_{\text{clk}}}{2^j}. \tag{2}$$

The read only memory (ROM) is a sine lookup table which converts the phase information into the values of a sine wave. In the ideal case of infinite precision of the phase and with no amplitude quantization, the output sequence of the table is given by:

$$\sin\left(2\pi\frac{\phi(n)}{2^j}\right), \tag{3}$$

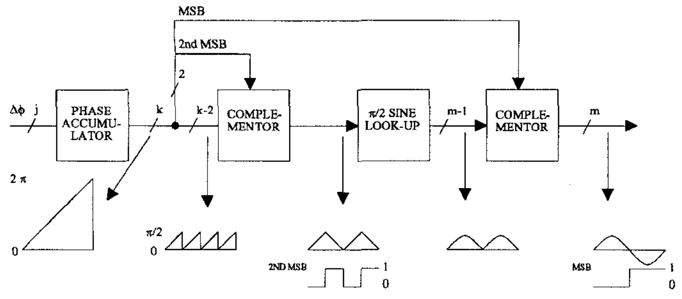

Fig. 2. Logic to exploit quarter wave symmetry.

where $\phi(n)$ is (the 2^j-bit) phase register value (at the nth clock period). The numerical period of the phase accumulator output sequence is defined as the minimum value of P for which $\phi(n) = \phi(n+P)$ for all n. The numerical period of the phase accumulator output sequence (in clock cycles) is:

$$P = \frac{2^j}{\mathrm{GCD}(\Delta\phi, 2^j)}, \tag{4}$$

where $\mathrm{GCD}(\Delta\phi, 2^j)$ represents the greatest common divisor of $\Delta\phi$ and 2^j. The numerical period of the sequence of the samples fetched from the sine ROM will have the same value as the numerical period of the sequence generated by the phase accumulator [2], [4]. Therefore, the spectrum of the output waveform of the DDS prior to the digital-to-analog conversion is characterized by a discrete spectrum consisting of P points. The ROM output is presented to the D/A-converter, which develops a quantized analog sine wave. The filter removes the high frequency sampling components and provides a pure sine wave output.

The size of the ROM is $(2^k \times m)$, where k is the wordlength of the truncated phase address, and m is the wordlength of the ROM. The number of words in the ROM will determine the phase quantization error and the number of bits in each word will determine the amplitude quantization error. It is desirable to increase the resolution of the ROM, but larger ROM storage means higher power consumption, lower reliability, lower speed, and greatly increased costs.

III. Exploitation of Sine Function Symmetry

A well-known technique is to store only $\pi/2$ rad of sine information, and to generate the sine lookup table samples for the full range of 2π by exploiting the quarter-wave symmetry of the sine function. The decrease in lookup table capacity is paid by the additional logic necessary to generate the complements of the accumulator and lookup table output.

The details of this method are shown in Fig. 2. The two most significant phase bits are used to decode the quadrant, while the remaining $k-2$ bits are used to address a one-quadrant sine lookup table. The most significant bit determines the required sign of the result, and the second most significant bit determines whether the amplitude is increasing or decreasing. The accumulator output is used "as is"

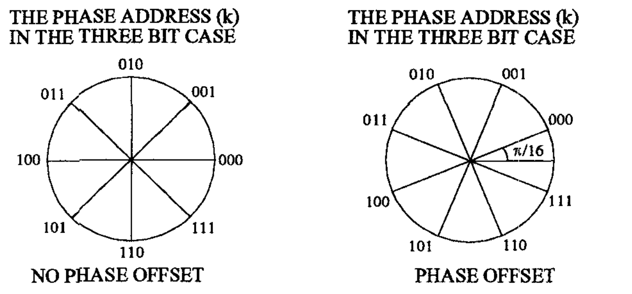
THE PHASE ADDRESS (k) IN THE THREE BIT CASE
010
011
001
100
000
101
111
110
NO PHASE OFFSET
THE PHASE ADDRESS (k) IN THE THREE BIT CASE
010
001
011
000
π/16
100
111
101
110
PHASE OFFSET

Fig. 3. 1/2 LSB phase offset is introduced in all phase addresses. In this case 1/2 LSB correspond $\pi/16$. The 1/2 LSB phase offset is added to all the sine lookup table samples. In this figure, 1's complementor maps the phase values to the first quadrant without error.

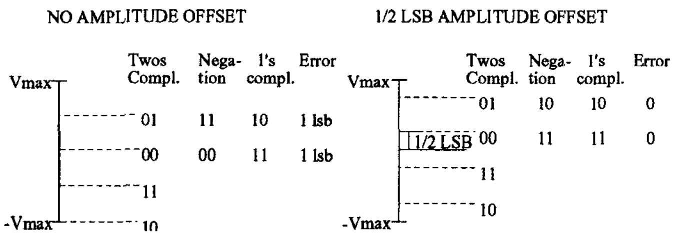

Fig. 4. 1/2 LSB offset is introduced into the amplitude that is to be complemented; then the negation can be carried out with 1's complementor without error in Fig. 2. There must also be a 1/2 LSB offset in the ROM output to the D/A-converter.

for the first and third quadrants. For the second and fourth quadrant, the phase bits must be complemented so that the slope of the saw tooth is inverted. As shown in Fig. 2, the sampled waveform at the output of the lookup table is a full 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.

In most practical DDS digital implementations, numbers are represented in 2's complement format. Therefore, the 2's complementing must be used to take the absolute value of the quarter 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 a number that is to be complemented, then a 1's complementor may be used in place of 2's complementor without introducing an error [5]. This provides savings in hardware since a 1's complementor may be implemented as a set of simple exclusive-or gates. This 1/2 LSB offset is provided by choosing the lookup table samples such that there is a 1/2 LSB offset in both the phase and amplitude of the samples [5], Figs. 3 and 4. If there is no phase offset in Fig. 3, then 0 and $\pi/2$ have the same phase address to the quarter wave memory, and one more address bit is needed to distinguish these two values.

IV. Compression of the Quarter-Wave Sine Function

In this section the quarter wave memory compression will be investigated. The width of the sine lookup table is reduced before taking advantage of the quadrant symme-

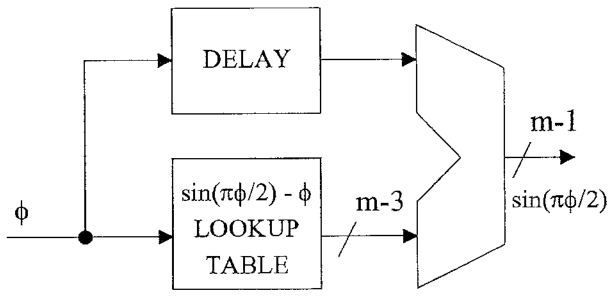

Fig. 5. The sine-phase difference algorithm.

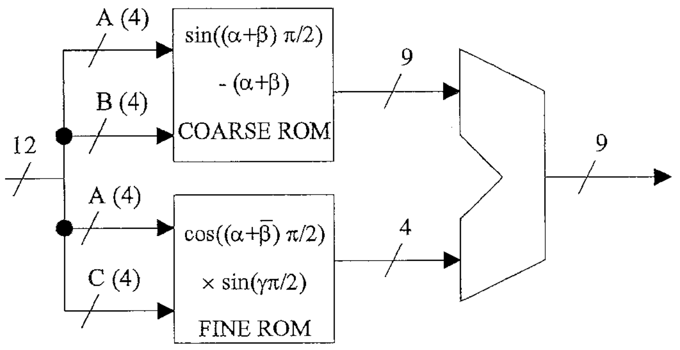
A (4)
B (4)
12
A (4)
C (4)
sin((α+β) π/2) - (α+β) COARSE ROM
cos((α+β̄) π/2) × sin(γπ/2) FINE ROM
9
4
9

Fig. 6. The block diagram of modified Sunderland architecture for quarter wave sine function compression.

try of the sine function. The two most significant phase bits are used to decode the quadrant, while the remaining $k-2$ bits are used to address a one-quadrant lookup table. First, a sine-phase difference algorithm will be presented [5]. This algorithm is used in all the subsequent quarter wave compression techniques except for the CORDIC algorithm. The compression techniques are: a trigonometric approximation, the so-called Nicholas' architecture, the Taylor series method, and the CORDIC algorithm. In each method the total compression ratio, the size of the memory, the worst case spur level, and additional circuits are presented in Table I. The amplitude values of the compressed quarter wave memory could be scaled to provide improved performance in the presence of amplitude quantization [5]. However, the optimization of the value scaling constant provides only negligible improvement in the amplitude quantization spur level, so it is beyond the scope of this paper.

A. Sine-Phase Difference Algorithm

Compression of the storage required for the quarter-wave sine function is obtained by storing the function

$$\boxed{f(\phi) = \sin\left(\frac{\pi\phi}{2}\right) - \phi} \tag{5}$$

instead of $\sin(\pi\phi/2)$ in the lookup table (Fig. 5). Because

$$\max\left[\sin\left(\frac{\pi\phi}{2}\right) - \phi\right] \approx 0.21 \max\left[\sin\left(\frac{\pi\phi}{2}\right)\right], \tag{6}$$

this saves 2 bits of amplitude in the storage of the sine function [5]. The penalty for this storage reduction is the introduction of an extra adder at the output of the look-up table to perform the operation

$$\left[\sin\left(\frac{\pi\phi}{2}\right) - \phi\right] + \phi. \tag{7}$$

However, this is an advantageous trade-off in the high speed VLSI implementation of the DDS. The sine lookup table propagation delay, which cannot be easily pipelined, is reduced, increasing the maximum clock frequency of the DDS. The expense of another adder, which is readily pipelineable, is negligible in the full custom VLSI implementation.

B. Modified Sunderland Architecture

The original Sunderland technique is based on simple trigonometric identities [6]. There are two modifications of the original Sunderland paper. After the paper, a method for performing the two's complement negation function with only an exclusive-or, which does not introduce errors, has been published [5]. This method works by introducing the 1/2-LSB offsets into the phase and amplitude of the sine ROM samples as described in Section III. The sine-phase difference algorithm has been published after Sunderland paper [6], too.

The phase address of the quarter of the sine wave is decomposed to $\phi = \alpha + \beta + \gamma$, with the word-lengths of the variables: $\alpha \rightarrow A$, $\beta \rightarrow B$, and $\gamma \rightarrow C$ (Fig. 6). In this way the 12 phase bits are divided into three 4 bit fractions such that $\alpha < 1$, $\beta < (2^{-4})$, $\gamma < (2^{-8})$. The desired sine function is given by:

$$\begin{aligned}\sin\left(\frac{\pi}{2}(\alpha+\beta+\gamma)\right) &= \sin\left(\frac{\pi}{2}(\alpha+\beta)\right)\cos\left(\frac{\pi}{2}\gamma\right)\\ &\quad + \cos\left(\frac{\pi}{2}(\alpha+\beta)\right)\sin\left(\frac{\pi}{2}\gamma\right).\end{aligned} \tag{8}$$

Given the relative sizes of α, β, and γ, this expression may be approximated by:

$$\begin{aligned}\sin\left(\frac{\pi}{2}(\alpha+\beta+\gamma)\right) &\approx \sin\left(\frac{\pi}{2}(\alpha+\beta)\right)\\ &\quad + \cos\left(\frac{\pi}{2}\alpha\right)\sin\left(\frac{\pi}{2}\gamma\right).\end{aligned} \tag{9}$$

The approximation is made closer by adding (2^{-5}) (the average value of β) to α in the second term. In Fig. 6 the size of the upper memory is reduced by the sine difference algorithm. The access time of the upper memory is the most critical due to its larger size. If γ is subtracted from the lower memory, the result will be negative, and more logic is needed in the adder. The coarse ROM provides low resolution samples, and the fine ROM gives additional resolution by interpolating between the low resolution samples in Fig. 6.

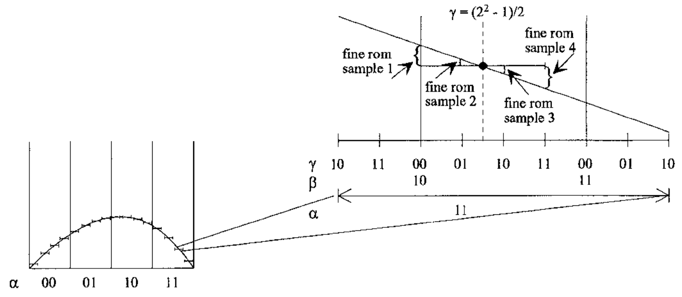

Fig. 7. The fine ROM samples are used to interpolate between the coarse samples, and the symmetry in the fine ROM samples around the $\gamma = (2^C - 1)/2$ axis. Here $C = 2$.

C. Nicholas Architecture

An alternative methodology for choosing the samples to be stored in the ROMs is based on numerical optimization [5]. The phase address of the quarter of the sine wave is defined as, $\phi = \alpha + \beta + \gamma$, with the word-length of the variable α to be A, the word-length of β to be B, and γ to be C. The variables α, β form the coarse ROM address, and the variables α, γ form the fine ROM address. In Fig. 7 the coarse ROM samples are represented by the dot along the solid line, and the fine ROM samples are chosen to be the difference between the value of the "error bars" directly below and above that point on the solid line. In Fig. 7, the function is divided into four regions, corresponding to $\alpha = 00, 01, 10$, and 11. Within each region, only one interpolation value may be used between the error bars and the solid line for all the same γ values. The interpolation value used for each value of γ is chosen to minimize either the mean square or the maximum absolute error of the interpolation within the region [5].

Further storage compression is provided by exploiting the symmetry in the fine ROM correction factors, Fig. 7. If the coarse ROM samples are chosen in the middle of the interpolation region, then the fine ROM samples will be approximately symmetric around the $\gamma = (2^C - 1)/2$ axis. Thus, by using an adder/subtractor instead of an 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/subtractor control logic if this technique is used with the sine-phase difference algorithm, since the slope of the function in (5) changes sign at a nonsymmetry point between 0 and $\pi/2$ on the x-axis. For example, the digital logic required to perform this can be accomplished with less than four logic gates for the 13-bit phase case [5]. Since the fine ROM is generally not in the critical speed path, the effective resolution of the fine ROM may be doubled, rather than halving the fine ROM. 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.

Computer simulations showed that the optimum partitioning of the ROM address word lengths to provide a

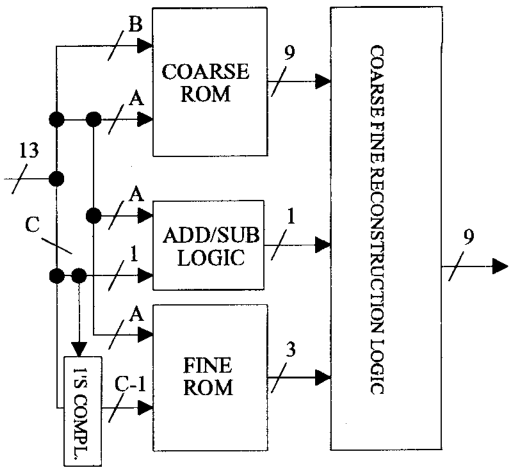

Fig. 8. The sine function generation logic of Nicholas' architecture.

13-b phase resolution was $A = 4$, $B = 4$, and $C = 5$, using the notation in Fig. 8 [5]. The simulations showed that the mean square and the minimum-maximum error criterion give the same maximum spur level in this segmentation. The architecture for this ROM compression technique is shown in Fig. 8. The amplitude values of the coarse and fine ROMs could be scaled to provide improved performance in the presence of amplitude quantization [5]. The optimization of the value scaling constant provides only a negligible improvement in the amplitude quantization spur level, so it is beyond the scope of this paper.

In a modified version of the above architecture the symmetry in the fine ROM samples is not utilized [7], so the extra bit of the phase resolution to the ROM address is not achieved. Therefore, the modified Nicholas architecture uses a 14-to-12-bit instead 15-to-12-bit phase to amplitude mapping in our case. Some hardware is saved, because an adder instead of an adder/subtractor is used to sum the coarse and fine ROM values, and the adder/subtractor control logic is not needed. The difference between the modified Nicholas architecture and the modified Sunderland architecture is that the samples stored in the sine ROM are chosen using the numerical optimization in the modified Nicholas architecture.

The IC realization of the Nicholas architecture is presented in [8], where a CMOS chip has the maximum clock frequency of 150 MHz. Analog Devices has also used this sine memory compression method in their CMOS device, which has the output word length of 12 bits and 100 MHz clock frequency [9]. The IC realization of the modified Nicholas architecture is presented in [7], where the CMOS quadrature digital synthesizer operates at 200 MHz clock frequency. The modified Nicholas architecture has also been used in [10], where a CMOS chip has four parallel ROM tables to achieve four times the throughput of a single DDS. The chip that uses only one ROM table has the clock frequency of 200 MHz [7]. Using the parallel architecture with four ROM tables, the chip attains the speed of 800 MHz [10].

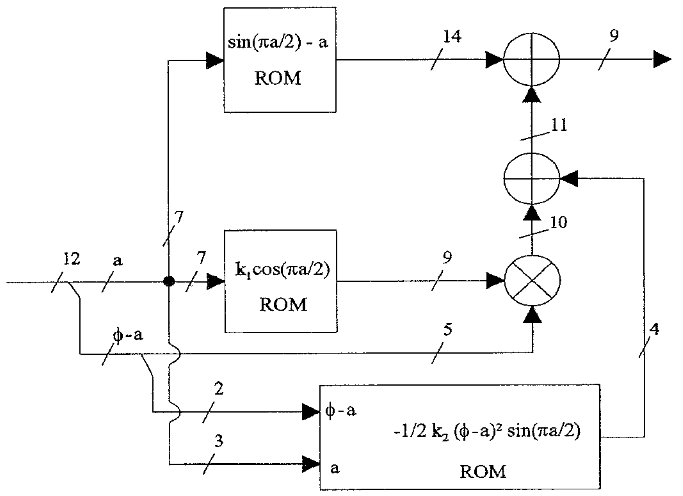

Fig. 9. Taylor series approximation for the quarter sine converter.

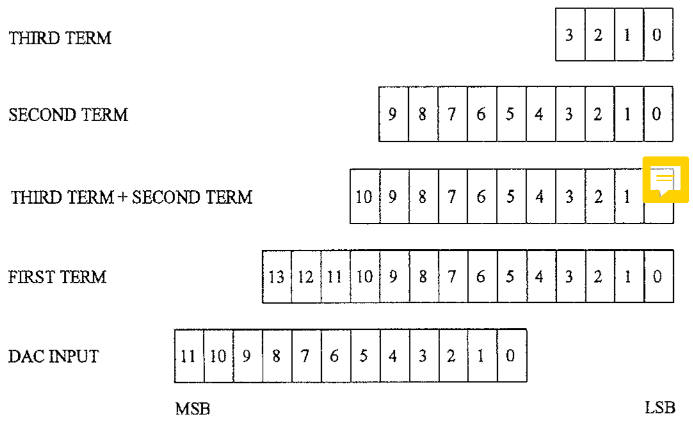

Fig. 10. Relative bit positions of multi-bit data words used in implementing the circuit of Fig. 9.

D. Taylor Series Approximation

The phase address "ϕ" is divided into the upper phase address "a" and the lower phase address "$\phi - a$" [11]. The Taylor series is performed around the upper phase address $(\phi = a)$

$$\sin\left(\frac{\pi}{2}\phi\right) = \sin\left(\frac{\pi}{2}a\right) + k_1(\phi - \alpha)\cos\left(\frac{\pi}{2}a\right) - \frac{k_2(\phi-\alpha)^2 \sin\left(\frac{\pi}{2}a\right)}{2} + \cdots, \tag{10}$$

where k_n represents a constant used to adjust the units of each series term. The adjustment in units is required because the phase values are given in angular units. Therefore, it is necessary to have a conversion factor k_n, which includes a multiple of $\pi/2$ to compensate for the phase units. The Taylor series (10) is approximated by taking three terms in Fig. 9. While additional terms can be employed, their contribution to accuracy is very small as shown in Fig. 10 and, therefore, of little weight in this application. Other inaccuracies present in the operation of current DDS designs override the finer accuracy provided by successive series terms.

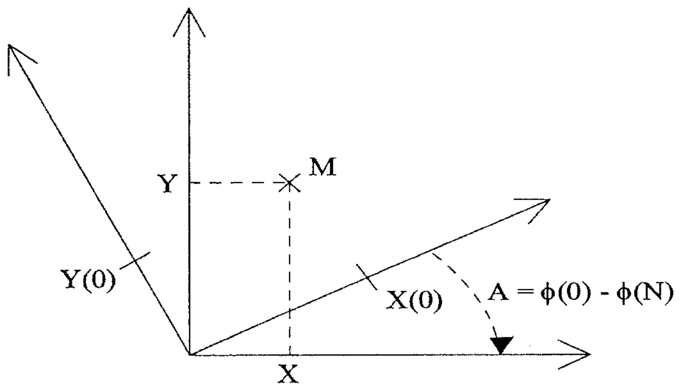

Fig. 11. "Rotation," given $X(0)$, $Y(0)$ and $\phi(0)$ find X and Y.

The seven most significant bits of the input phase are selected as the upper phase address "a", which is transferred simultaneously to a sine ROM and a cosine ROM as address signals as shown in Fig. 9. The output of the sine ROM is the first term of the Taylor series and it is taken to the first adder, where it will be summed with the remaining terms. The size of the sine ROM is reduced by the sine difference algorithm. The output of the cosine ROM is configured to incorporate the predetermined unit conversion value k_1. The cosine ROM output is the first derivative of the sine. The least significant bits $(\phi - a)$ are multiplied by the output of the cosine ROM to produce the second term. The third term is computed in a ROM by combining the second derivative of $\sin((\pi a)/2)$ and the square of the lower phase address "$\phi - a$". This is done by selecting the upper bits of "$\phi - a$" and "a" values as a portion of the address for the ROM. This is possible since the last term only roughly contributes 1/4 LSB to the D/A-converter input, as shown in Fig. 10. As with the cosine ROM, the unit conversion factor is included in the values stored in the ROM. The third term ROM output is combined with the multiplier output in a second adder, and subsequently combined with the first term ROM output in the first adder.

QUALCOMM has used the Taylor series approximations in their device, which has the output word length of 12 bits and 50 MHz clock frequency [12]. This DDS is CMOS device.

E. CORDIC Algorithm

The CORDIC algorithm performs vector coordinate rotations by using simple iterative shifts and add/subtract operations, which are easy to implement in hardware [1]. In Fig. 11, a pair of rectangular axes is rotated clockwise by the angle A by the CORDIC algorithm, then the coordinates of a point M transform from $(X(0), Y(0))$ to (X, Y)

$$\begin{aligned} X &= X(0)\cos(A) - Y(0)\sin(A) \\ Y &= Y(0)\cos(A) + X(0)\sin(A), \end{aligned} \tag{11}$$

where A is the total rotation angle after N CORDIC iterations

$$A = \phi(0) - \phi(N) = \sum_{n=0}^{N-1} u(n)a(n), \tag{12}$$

where $\phi(0)$ is the desired rotation angle, $a(n) = \arctan(2^{-n})$ and $u(n) = \text{sign}(\phi(n))$. The absolute value of the angle approximation error can be bounded by [13]

$$|\phi(N)| = \left|\phi(0) - \sum_{n=0}^{N-1} u(n)a(n)\right| \leq \arctan(2^{-N+1}). \tag{13}$$

The total number of iterations, N, is determined by the accuracy desired. CORDIC is a bit-recursive algorithm, which means that each iteration increases the accuracy of the results by approximately one bit. The truncation due to the finite precision in fixed point arithmetic causes errors, too. Both the scaling operation (see below) and the truncation errors must be taken into account when selecting the optimal number of iterations [13]. The CORDIC algorithm can be easily derived from (11). For angles 0 to $\pi/2$ it can be described as follows

```
Initiation: Given $X(0), Y(0), \phi(0)$
for $n = 0 : N-1$ /* CORDIC equations */
$u(n) = \text{sign}(\phi(n))$ /* sign detection */
$X(n+1) = X(n) - u(n) \times Y(n) \times 2^{-n}$ /* iteration */
$Y(n+1) = Y(n) + u(n) \times X(n) \times 2^{-n}$ /* iteration */
$\phi(n+1) = \phi(n) - u(n) \times \arctan(2^{-n})$ /* angle updating */
End $n$-loop
$X = X(N)/K(N)$ /* scaling operation */
$Y = Y(N)/K(N)$ /* scaling operation */
                                                            (14)
```

The CORDIC processor either adds or subtracts a series of known angle values so that the value of $\phi(n)$ is driven to 0. The known angles (arctan) could be precomputed and stored in the CORDIC processor. The decision of whether to add or subtract the next angle in the series is based on the sign of $\phi(n)$, i.e., if $\phi(n)$ is negative, then the next angle is added to $\phi(n)$ to drive it closer to 0. If $\phi(n)$ is positive, then the next angle is subtracted from $\phi(n)$. The $(X(n), Y(n))$ coordinate pair is put through the series of angular rotations as $\phi(n)$ but with opposite sign (i.e., if 45° is subtracted from $\phi(n)$, then $(X(n), Y(n))$ is rotated + 45°). The rotation is not a pure rotation but a rotation-extension. The scaling operations compensate for this magnitude change. The scaling factor for a particular value of N is constant, so it could be precomputed and stored in the CORDIC processor. The scaling factor is

$$K(N) = \prod_{n=0}^{N-1} \sqrt{1 + u_n^2 2^{-2n}}. \tag{15}$$

If the initial values are chosen to be $X(0) = 1$, $Y(0) = 0$ and $\phi(0)$ is formed using the remaining $k-2$ bits of the phase register value from the DDS. From (11) the result will be $X = \cos(\phi(0) - \phi(N))$, $Y = \sin(\phi(0) - \phi(N))$, where $\phi(N)$ is the angle approximation error. If the initial values are chosen to be $X(0) = 1/K(N)$, $Y(0) = 0$, and $\phi(0)$ contains the remaining $k-2$ bits of the phase register value from the DDS, then there is no need for the scaling operation after the CORDIC iterations. However, in the simulations the signal-to-spur level was worse than the case of the first initial values. Because $1/K(N)$ is approximately $0.607253 (N > 13)$ and not power of two, the worse spur level was due to accumulated truncation errors in fixed arithmetic. The wordlength of the CORDIC iterations should be larger than in the case of the first initial values. The amplitude of the output waveform could be modulated by changing the scaling factor, but it is better to do the scaling after the iterations in fixed arithmetic. The value of Y may then be transferred to an appropriate D/A-converter. The architecture for quarter sine wave generation employing this technique is shown in Fig. 12.

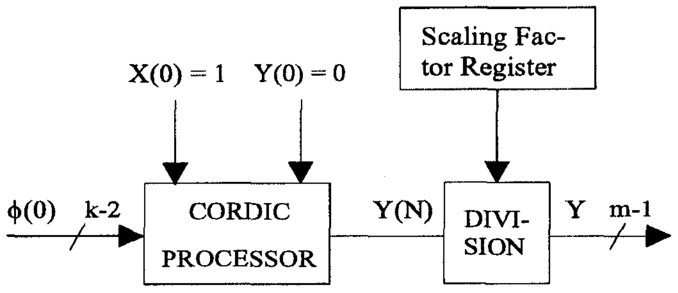

Fig. 12. The CORDIC processor for quarter sine wave converter.

The implementation of the CORDIC cell requires adders, barrel shifters, multiplexers and registers. The most speed-critical component will be the adder/subtractor due to the carry propagation. The throughput of the CORDIC datapath can be improved by introducing redundant number representation in the internal computation and eliminating the carry propagation from each addition/subtraction [14]–[16], but then the required chip area will increase [16]. The sign detection in the CORDIC algorithm would also be more complicated than in the nonredundant case, and the conversion and the reconversion to/from the redundant representation reduces the gain in the computation speed. To increase throughput, one could use pipelined processors with the penalty that the pipelined structure would require a large amount of chip area [17], [18].

The CORDIC algorithm is suitable for the DDS with more than 14 output bits, where the sine look-up table even with compression is still too large for high speed operation. It is also effective in solutions where there is the need of in-phase and quadrature components simultaneously, because the algorithm calculates both. For example GEC-Plessey I/Q splitter has 20 MHz clock frequency with 16-bit phase and amplitude accuracy [19], and Raytheon Semiconductor's DDS has 25 MHz clock frequency with 16-bit phase and amplitude accuracy [20].

TABLE I
Summary of Memory Compression and Algorithmic Techniques.

Method	ROM Needed	Total compression ratio	Additional circuits (not including quarter and sine difference logic)	Worst case spur (below carrier)	Comments
Uncompressed memory	$2^{14} \times 12$ bits	$1:1$	-	-97.23 dBc	reference
Mod. Sunderland architecture	$2^8 \times 9$ bits $2^8 \times 4$ bits	$59:1$	adder	-86.91 dBc	good spur level and simple
Nicholas architecture	$2^8 \times 9$ bits $2^8 \times 3$ bits	$128:1$ Note $(k = 15, m = 12)$	adder/subtractor adder/subtractor control logic	-88.94 dBc	best compression ratio
Taylor series approzimation with three terms	$2^7 \times 14$ bits $2^7 \times 9$ bits $2^5 \times 4$ bits	$64:1$	2 adders multiplier	-97.04 dBc	need multiplier
CORDIC algorithm	-	-	14 pipelined stages, 18-bit inner wordlength	-84.25 dBc	much computation

V. Simulation

A computer program (in Matlab) has been created to simulate the direct digital synthesizer in Fig. 1. The memory compression and algorithmic techniques have been analyzed with no phase truncation (the phase accumulator length = the phase address), and the spectrum is calculated prior to the D/A-conversion. The number of points in the DDS output spectrum depends on $\Delta\phi$ (phase increment word) via the greatest common divisor of $\Delta\phi$ and 2^j $(\text{GCD}(\Delta\phi, 2^j))$ (4). Any phase accumulator output vector can be formed from a permutation of another output vector regardless of the initial phase accumulator contents, when $(\text{GCD}(\Delta\phi, 2^j)) = 1$ for all values of $\Delta\phi$ (see Appendix). A permutation of the samples in the time domain results in an identical permutation of the discrete Fourier transform (DFT) samples in the frequency domain [2]. This means that the spurious spectrum due to all system nonlinearities can be generated from a permutation of another spectrum, when $(\text{GCD}(\Delta\phi, 2^j)) = 1$ for all $\Delta\phi$, because each spectrum will differ only in the position of the spurs and not in the magnitudes [2]. When the least significant bit of the phase accumulator is forced to one (Nicholas modified phase accumulator [2]), it causes all of the phase accumulator output sequences to belong to the number theoretic class $(\text{GCD}(\Delta\phi, 2^j)) = 1$, regardless of the value of $\Delta\phi$. Only one simulation is needed to be performed to determine the value of the worst case spurious response due to the system nonlinearities. In addition, the modified phase accumulator averages errors introduced by the D/A-converter, which is, however, beyond the scope of this text. The number of samples has been chosen (an integer number of cycles in the time record) so that problems of leakage in the fast Fourier transform (FFT) analysis can be avoided and unwindowed data can be used. The FFT was performed over the output period (4). The size of the FFT was 16384 points (except in Nicholas architecture 32768 points).

VI. Conclusion

Table I comprises the summary of memory compression and algorithmic techniques. The best spur level (almost the same level as with the ideal 12-bit rounded samples of a sine wave) is achieved by the Taylor series approximation with three terms. The Nicholas architecture has the best compression ratio in Table I, but it uses a 15-to-12-bit instead 14-to-12-bit phase to amplitude mapping in our case. The original Sunderland architecture produced worst case spur level of about -72.2 dBc [5], but the modified Sunderland architecture gives about 14 dB performance improvement. In the CORDIC algorithm the spur level is high, because the quantization step sizes are not equal (nonlinearity) due to the rotations by $\arctan(2^{-n})$. The number of the iterations (14) in the CORDIC processor for the quarter sine wave converter are more than needed for the 11-bit accuracy because of the high spur level.

Table II shows how much memory and additional circuits are needed in each memory compression and algorithmic technique to meet the spectral requirement for the worst case spur level, which is about -85 dBc due to the sine memory compression. In the DDS, most spurs are normally not generated by digital errors but rather by the analog errors in the D/A-converter. The spur level (-85 dBc) from the sine memory compression is not significant in DDS applications because it will stay below the spur level of a high speed 12-bit D/A-converter [21]. Differing from Table I, two terms are used for Taylor series approximation in Table II. Then, all memory compression and algorithmic techniques in Table II are comparable with almost the same worst case spur. In Table II the modified Nicholas architecture [7] is used and therefore the compression ratio and the worst case spur level are different from that in the Nicholas architecture [5] in Table I. The difference between the modified Nicholas architecture and the modified Sunderland architecture is that the samples stored in the sine ROM are chosen according to the numerical optimization

TABLE II
MEMORY COMPRESSION AND ALGORITHMIC TECHNIQUES WITH WORST CASE SPUR LEVEL DUE TO THE SINE MEMORY COMPRESSION SPECIFIED TO BE ABOUT −85 dBc.

Method	Needed ROM	Total compression ratio	Additional circuits (not including quarter and sine difference logic)	Worst case spur (below carrier)	Comments
Uncompressed memory	$2^{14} \times 12$ bits	1 : 1	-	−97.23 dBc	reference
Mod. Sunderland architecture	$2^8 \times 9$ bits $2^8 \times 4$ bits	59 : 1	adder	−86.91 dBc	simple
Mod. Nicholas architecture	$2^8 \times 9$ bits $2^8 \times 4$ bits	59 : 1	adder	−86.81 dBc	simple
Taylor series approximation with two terms	$2^7 \times 9$ bits[†] $2^7 \times 5$ bits[††]	110 : 1	adder multiplier	−85.88 dBc	need multiplier
CORDIC algorithm	-	-	14 pipelined stages, 18-bit inner wordlength	−84.25 dBc	much computation

[†] The first term ROM size, which is reduced by the sinedifference algorithm.
[††] The cosine ROM Size.

in the modified Nicholas architecture. In the 14-to-12-bit phase-to-amplitude mapping the numerical optimization gives no benefit, because the modified Sunderland architecture and modified Nicholas architecture give the same spur levels.

APPENDIX

A. Phase Accumulator as a Permutation Generator

The phase accumulator can be considered as a permutation generator, where each value of $\Delta\phi$ provides a different permutation of the values from 0 to $2^j - 1$ given by

$$_{\Delta\phi}\phi(n) = (n\Delta\phi) \mod 2^j. \tag{16}$$

Fig. 13 shows that any phase accumulator output vector can be formed from the permutation of another output vector regardless of the initial phase accumulator contents, when $(\mathrm{GCD}(\Delta\phi, 2^j)) = 1$ for all values of $\Delta\phi$. Fig. 14 shows that the time vectors are formed from values, which have the property $(\mathrm{GCD}(\Delta\phi, 2^j)) = 2$. From Fig. 14 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 time output vector for $\Delta\phi$ can be formed from a permutation of the individual elements of the vector for $\Delta\phi = 1$

$$_{\Delta\phi}\phi(n) = {}_1\phi((n\Delta\phi) \mod 2^j), \tag{17}$$

when $\Delta\phi$ and 2^j are relatively prime (Fig. 13). As was shown in (17), each input time vector may be formed from a permutation of another time vector by permuting the indices using $(n\Delta\phi) \mod 2^j$. The converse follows from

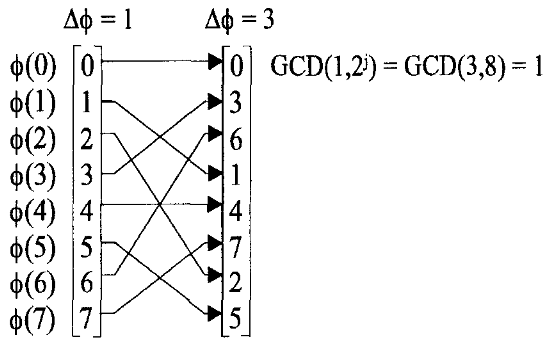
Δϕ = 1
Δϕ = 3
ϕ(0) ϕ(1) ϕ(2) ϕ(3) ϕ(4) ϕ(5) ϕ(6) ϕ(7)
0 1 2 3 4 5 6 7
0 3 6 1 4 7 2 5
GCD(1,2^j) = GCD(3,8) = 1

Fig. 13. Time series vectors for a 3-bit phase accumulator for $\Delta\phi = 1$ and $\Delta\phi = 3$. The column vector for $\Delta\phi = 3$ can be formed from a permutation of the values of the $\Delta\phi = 1$ vector, regardless of the initial phase accumulator contents.

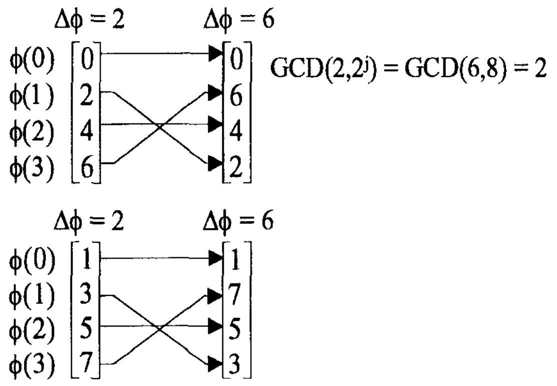

Fig. 14. Time series vectors for a 3-bit accumulator for $\Delta\phi = 2$ and $\Delta\phi = 6$.

the existence of a unique integer $0 \leq J < 2^j$ satisfying the relation

$$\Delta\phi J \mod 2^j = 1. \tag{18}$$

This is a fundamental result of number theory which requires that $\Delta\phi$ and 2^j are relatively prime [22]. In a sense J is the multiplicative inverse of $\Delta\phi$. From the above equation it follows that $\Delta\phi$ and J must be odd because 2^j is even. Therefore, J and 2^j are too relatively prime.

The sine output DDS operates by applying some memoryless nonlinear function $s\{\}$ to the phase accumulator output to produce the sine function. The DFT of the phase-to-amplitude converter output using (17) is:

$$S\{_{\Delta\phi}\phi(m)\} = \sum_{n=0}^{2^j-1} s\{_1\phi((n\Delta\phi) \mod 2^j)\} \times W_{2^j}^{mn} \quad m = 0, 1, \ldots, 2^j - 1, \tag{19}$$

where the period of the phase accumulator is 2^j when $\Delta\phi$ and 2^j are relatively prime (4), and $W_{2^j} = e^{-j2\pi/2^j}$. Equation (18) can be used to show that the permutation samples in the time domain result in the same type permutation in the frequency domain by defining the new index

$$q = (n\Delta\phi) \mod 2^j, \tag{20}$$

and noting that

$$\begin{aligned} qJ \mod 2^j &= J((n\Delta\phi) \mod 2^j) \mod 2^j \\ &= n\Delta\phi J \mod 2^j. \end{aligned} \tag{21}$$

Substituting from (18), (21) becomes

$$n = qJ \mod 2^j. \tag{22}$$

Reindexing (19) using (20) and (22), then

$$\begin{aligned} S\{_{\Delta\phi}\phi(m)\} &= \sum_{q=0}^{2^j-1} s\{_1\phi(q)\} W_{2^j}^{m(qJ \mod 2^j)} \\ &= \sum_{q=0}^{2^j-1} s\{_1\phi(q)\} W_{2^j}^{q(mJ \mod 2^j)} \\ &= S\{_1\phi((mJ) \mod 2^j)\} m = 0, 1, \ldots, 2^j - 1. \end{aligned} \tag{23}$$

The above equation establishes that the permutation of the samples in the time domain results in the same type permutation of the DFT samples in the frequency domain, because J and 2^j are relatively prime. This means that the spurious spectrum due to all system nonlinearities can be generated from a permutation of another spectrum, when $\mathrm{GCD}(\Delta\phi, 2^j) = 1$ for all $\Delta\phi$, because each spectrum will differ only in the position of the spurs and not in the magnitudes.

References

[1] J. E. Volder, "The CORDIC trigonometric computing technique," *IRE Trans. Electron. Comput.*, vol. EC-8, pp. 330–334, Sept. 1959.

[2] H. T. Nicholas and H. Samueli, "An analysis of the output spectrum of direct digital frequency synthesizers in the presence of phase-accumulator truncation," in *Proc. 41st Annu. Freq. Contr. Symp.*, 1987, pp. 495–502.

[3] A. L. Bramble, "Direct digital frequency synthesis," in *Proc. 35th Annu. Freq. Contr. Symp.*, USERACOM (Ft. Monmouth, NJ), May 1981, pp. 406–414.

[4] D. L. Duttweiler and D. G. Messerschmitt "Analysis of digitally generated sinusoids with application to A/D and D/A converter testing," *IEEE Trans. Commun.*, vol. COM-26, pp. 669–675, May 1978.

[5] H. T. Nicholas, H. Samueli, and B. Kim, "The optimization of direct digital frequency synthesizer performance in the presence of finite word length effects," in *Proc. 42nd Annu. Freq. Contr. Symp.*, 1988, pp. 357–363.

[6] 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 J. Solid-State Circuits*, vol. SC-19, pp. 497–505, Aug. 1984.

[7] L. K. Tan, and H. Samueli, "A 200 MHz quadrature digital synthesizer/mixer in 0.8 μm CMOS," *IEEE J. Solid-State Circuits*, vol. 30, pp. 193–200, Mar. 1995.

[8] H. T. Nicholas, and H. Samueli, "A 150-MHz direct digital frequency synthesizer in 1.25-μm CMOS with −90-dBc spurious performance," *IEEE J. Solid-State Circuits*, vol. 26, pp. 1959–1969, Dec. 1991.

[9] Analog Devices AD 9955 Data Sheet, Rev. 0, 1994.

[10] L. K. Tan, E. W. Roth, G. E. Yee, and H. Samueli, "A 800 MHz quadrature digital synthesizer with ECL-compatible output drivers in 0.8 μm CMOS," *IEEE J. Solid-State Circuits*, vol. 30, pp. 1463–1473, Dec. 1995.

[11] L. A. Weaver, and R. J. Kerr, "High resolution phase to sine amplitude conversion," U. S. Patent 4,905,177, Feb. 27, 1990.

[12] Qualcomm Q2334, Technical Data Sheet, June 1991.

[13] Y. H. Hu, "The quantization effects of the CORDIC algorithm," *IEEE Trans. Signal Processing*, vol. 40, pp. 834–844, Apr. 1992.

[14] M. D. Ercegovac, and T. Lang, "Redundant and on-line CORDIC: application to matrix triangularization and SVD," *IEEE Trans. Comput.*, vol. 39, pp. 725–740, June 1990.

[15] N. Takagi, T. Asada, and S. Yajima, "Redundant CORDIC methods with a constant scale factor for sine and cosine computation," *IEEE Trans. Comput.*, vol. 40, pp. 989–995, Sept. 1991.

[16] J. Lee, and T. Lang, "Constant-factor redundant CORDIC for angle calculation and rotation," *IEEE Trans. Comput.*, vol. 41, pp. 1016–1025, Aug. 1992.

[17] G. Gielis, R. van de Plassche, and J. van Valburg, "A 540 MHz 10b polar-to-cartesian converter," *ISSCC Digest Technical Papers*, pp. 160–161, Feb. 1991.

[18] A. Madisetti, A. Kwentus, and A. N. Wilson, Jr., "A sine/cosine direct digital frequency synthesizer using an angle rotation algorithm," *ISSCC Digest Technical Papers*, pp. 262-263, Feb. 1995.

[19] GEC-Plessey Semiconductors, Data Sheet PDSP16350 I/Q Splitter/NCO, Dec. 1993.

[20] Raytheon Semiconductor Data Book, Data Sheet TMC2340, 1994.

[21] Burr-Brown IC Data Book-Data Conversion Products, Data Sheet DAC650, 1994.

[22] J. H. McClellan and C. M. Rader, *Number Theory in Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1979.

Jouko Vankka was born in Helsinki, Finland, on July 12, 1965. He received the Diploma Engineer degree from the Helsinki University of Technology in 1991. Since 1994 he has been with the Laboratory of Signal Processing and Computer Technology at the Helsinki University of Technology. He is currently working as a research scientist toward the doctoral degree. His current research interests include design and implementation methods of DSP algorithms.