A Simple DDS Architecture with Highly Efficient Sine Function Lookup Table

ZhengYu Wang, M. C. Frank Chang, Jessica Chiatai Chou
Electrical Engineering Department
University of California, Los Angeles
Tel: 310-794-9176
wzy@ee.ucla.edu

ABSTRACT

A simple architecture for direct digital frequency synthesis (DDS) is presented. The proposed architecture uses a sampling–only-algorithm (SOA) to achieve a high compression ratio of 558 in realizing the sine function look up table, which is higher than the prior arts.

Categories and Subject Descriptors

B.7.0 [Integrated Circuits]: General

General Terms

Algorithms, Performance, Design.

Keywords

Frequency Control, Direct Digital Synthesis, DSP.

1. INTRODUCTION

Direct digital frequency synthesizer (DDS) has the unique feature of fast and fine frequency switching that is required by modern communication systems. The main design tradeoff in realizing DDS is between the ROM size and output SFDR. ROM size grows exponentially as the phase truncation is reduced. Large phase truncation, however, causes large output spurs. In this paper, we introduce an improved Taylor series algorithm to reduce the ROM size efficiently. The error introduced by one's complement operation is compensated by the carry of output adder. Simulation result shows that the output SFDR is mainly determined by the phase truncation, but not data compression.

2. BASIC IDEA

2.1 Algorithm

Taylor series of sine function is:

$$\sin(x + delta) = \sin(x) + slope \bullet delta + \Delta \quad (1)$$

where $slope = \cos(x) = \sin(\frac{\pi}{2} - x)$ (2)

So, $\sin(x + delta) = \sin(x) + \sin(\frac{\pi}{2} - x) \bullet delta + \Delta$ (3)

Based on the stored sample 'sin(x)' and the slope 'sin(π/2-x)', the sine value of object phase 'x+delta' can be calculated according to (1). We also observe that the slope does not need to be stored, because the slope of one sample 'sin(x)' is just the value of another sample 'sin(π/2-x)' if samples are evenly distributed in phase range of 0~π/2. The phase 'π/2-x' can be easily found by 2's complement of 'x', provided that the number of divisions is a power of 2. Reference [1] mentioned a similar idea but from a different viewpoint. We also observe that if the stored samples (or slopes) are selected at the center of each phase division (Fig. 1(a)), then the approximation (2) is better than the case when samples are selected at the beginning of each phase division, because the approximated 'delta' phase is reduced (Fig. 1(b)).

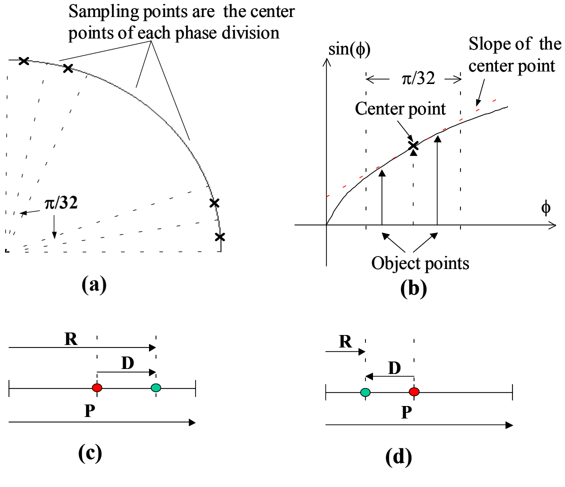

Sampling points are the center points of each phase division
π/32
(a)
sin(ϕ)
Slope of the center point
π/32
Center point
ϕ
Object points
(b)
R
D
P
(c)
R
D
P
(d)

Figure 1. Phase Division

2.2 One's Complement

To further simplify the hardware, we replace all the 2's complements with 1's complements which are used to map the phase of 'sample' to the phase of 'slope'; map the 'object phase' above π/2 into the phase range of 0~π/2 and map the 'residue phase' in phase division to 'delta phase'.

GLSVLSI'04, April 26-28, 2004, Boston, Massachusetts, USA

When mapping sample phase ‘X’ to slope phase ‘π/2-X’, there should be no offset error because we have already selected the samples at the center of each phase division.

When mapping object phase ‘X’ above π/2 into the phase range of 0~π/2, if we assume that ‘X’ is the mid point of each phase sub-division, and the data from accumulator is the index of ‘X’ (but not object phase itself)**,** then there is no offset error as well.

When mapping the residue phase ‘R’ to the delta phase ‘D’ in each phase division ‘P’ (Fig. 1(c) and (d)), the phase offset caused by 1’s complement has to be considered. In each phase division, we use the lower PL bits of truncated data from the accumulator to represent the index of residue phase ‘R’. We also assume that ‘R’ is the mid point of each phase sub-division. The delta phase ‘D’ represents the absolute phase difference between the mid point of phase division and the residue phase. The sine value is approximated as the product of D and the slope at the mid point of phase division. If R is above the mid point of phase division, then D is equal to R minus 0.5P, which can be found by simply setting MSB of R to zero and cause –0.5LSB phase offset. If R is below the mid point of phase division, then D is equal to 0.5P minus R, which can be found by 2’s complement of R and cause +0.5LSB phase offset. If 2’s complement is replaced with 1’s complement in order to reduce hardware, the resulting D will have –0.5LSB phase offset.

To accommodate these offsets, the size of the multiplier has to be increased. Fortunately, we can avoid this by adding or subtracting 1LSB to the amplitude output. Using the existing carry of the output adder, this compensation actually does not require any extra hardware.

3. ARCHITECURE

3.1 General Description

Fig. 2 shows the architecture. The truncated W bits from the accumulator are separated into 3MSB+PH+PL bits. Among these, the three most significant bits are used to generate the full sine wave, the following PH bits are the index to phase divisions in the phase range of 0~π/4 or π/4~π/2, and the least significant PL bits are the index to phase sub-divisions in each phase division mentioned above.

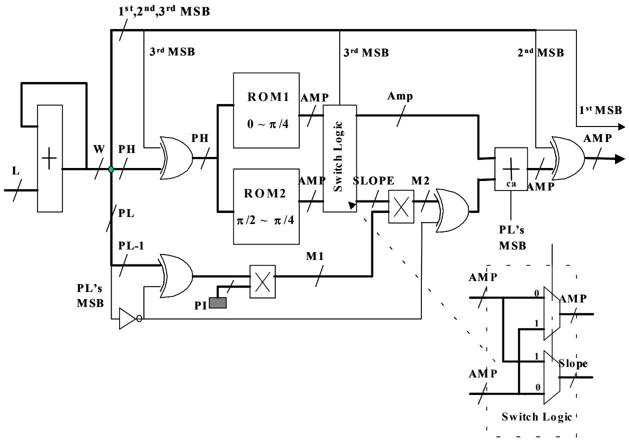


Figure 2. Architecture

The 1st MSB is used as a sign bit of half sine wave. The 2nd MSB is XOR’ed (one’s complement) with the quarter sine wave from the adder output to generate the half sine wave. The 3rd MSB is XOR’ed with PH bits to generate the index of quarter sine wave. When the 3rd MSB is 0, PH bits represent the phases in 0~0.25π range, which are the index to the sampled sine wave data in ROM1, and also the index to the slope data in ROM2. When the 3rd MSB is 1, the reversed PH bits represent the phases in 0.5π~0.25π range, which are the index to the sampled sine wave data in ROM2, and also the index to the slope data in ROM1. The stored data in ROMs are directly obtained from quantized sine values without any optimization. Therefore, the total ROM size is $2 \times (2^{PH} \times AMP)$ bits.

PL bits are the index of ‘residue’ phase X in phase division (Fig. 1(c~d)). The width of the phase division is P ($= \frac{0.25\pi}{2^{PH}}$). The MSB of PL bits are XOR’ed with the following PL-1 bits to generate the index of delta phase ‘D’. If PL’s MSB is 1, then the residue phase ‘R’ is above the mid point of the phase division (Fig. 1(c)). D is equal to R-0.5P, which can be represented by PL-1 bits directly. If the PL’s MSB is 0, then R is below the mid point of the phase division (Fig. 1(d)). D is equal to 0.5P-R, which can be found by reversing the PL-1 bits. Both of the above operation result in -0.5 LSB phase offset error that should be compensated latter. D is then multiplied by constant π to generate the real ‘delta’ phase in radians. The resulting product is multiplied by the slope and added to the sampled data to approximate the sine amplitude.

3.2 Carry Compensation

It is quite simple to compensate for the 0.5LSB phase offset error. When residue phase is above the mid point, the error is -0.5LSB phase times the slope (due to add operation), thus adding 1LSB amplitude to the final output. When it is bellow the mid point, the error is 0.5LSB phase times the slope (due to subtract operation), thus subtracting 1LSB amplitude from final output. The adding and subtracting of 1LSB is through the carry of the adder at the output end of the DDS, which is referred as carry compensation. This carry compensation operation does not consume extra hardware because carry operation is required by subtraction operation originally. Although this compensation is not complete, simulation shows that the resulting error is reduced efficiently.

3.3 Hardware Summary

Hardware used in proposed DDS is summarized in table 1. It shows that both the ROM size and the extra blocks are minimized, which are composed of one small hardwired multiplier, one small multiplier and one adder.

4. DESIGN EXAMPLES

4.1 Implementation

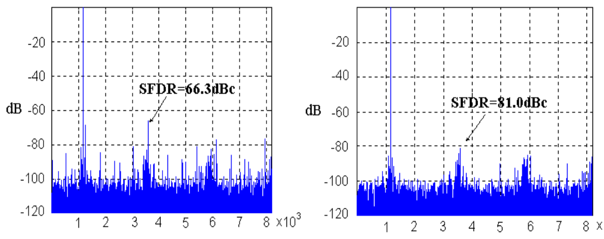
Table 2 shows the width of data paths for sample designs of 11bit and 12bit DDS. The names of data paths are also shown in architecture (Fig. (2)). The ROM size is 352 bits (32 X 11). The stored data are just quantized sine values of corresponding phases without any optimization. The design is first modeled with the C-program, and then implemented with Verilog at gate level, using TSMC 0.18um CMOS technology and IBM 7hp SiGe technology separately.

4.2 Test Plan

When the least significant bit of the phase accumulator is forced to one, all outputs of the phase accumulator belong to the number theoretic class ($GCD(\Delta\phi, 2^j) = 1$)[4], regardless of the values of the input control word. Thus only one simulation is needed to determine the value of the worst-case spurious response. The selected control word is 1199104, which means that the output signal frequency is about 1/14 of the clock frequency. (1199104≈2^{24}/14).

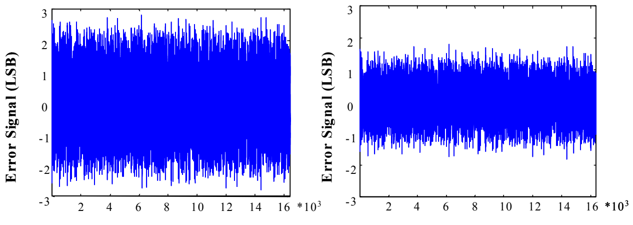
4.3 Simulation Result

The simulated FFT results of the case without and with the carry compensation are shown in Fig. 3(a) and Fig. 3(b), respectively. The carry compensation improved the SFDR to 81dB. Fig. 4(a) and Fig. 4(b) show the error signal of the case without and with carry compensation (the error signal here is equal to the DDS output value minus ideal sine value). It is shown that the carry compensation reduces the error signal by a factor of 2. The simulation also shows that the proposed DDS can work with clock frequency above 1GHz and 2GHz for TSMC and IBM process separately without any circuit optimization (using standard cell in library and with slow-slow corner condition). The design is now under optimization.



(a) Without "Carry Compensation" (b) With "Carry Compensation"

Figure 3. Output Spectrum



(a) (b)

Figure 4. Error Signal

4.4 Comparison

Comparison between the proposed DDS and the state-of-the-art designs is given in Table 3. It is shown that the ROM size in the proposed DDS is less than half that of the prior designs, and the effect on output SFDR is minimal.

5. CONCLUSION

The proposed SOA DDS architecture uses a very small lookup table (352 bits) for sine function implementation. The design technique is also simple because the stored data does not require any optimization. Combined with small additional circuits, this architecture is suitable for high speed and low power DDS designs.

6. ACKNOWLEDGMENTS

The authors wish to thank Jazz and IBM for supplying the process models and supporting the project.

7. REFERENCES

[1] F. Curticapean and J. Niittylahti *A hardware efficient direct digital frequency synthesizer*. The 8th IEEE International Conference on Electronics, Circuits and Systems,Sept.2001, 51~54

[2] L.K. Tan, Edward W.Roth, Gordon E.Yee and Henry Samueli *A 200MHz direct digital synthesizer/mixer in 0.8 um CMOS*. IEEE JSSC, Mar. 1995, 193~200,

[3] J.Vankka, Mikko Waltari, Marko Kosunen and Kari A.I.Halonen *Methods of mapping from phase to sine amplitude in direct digital synthesis*. IEEE Trans. Ultrasonics Ferroelectrics, and Frequency Control., Mar. 1997, 526~531

[4] H.T.Nicolas and Henry Samueli *A 150MHz direct digital frequency synthesizer in 1.25um CMOS with –90dBc spurious performance*. IEEE JSSC, Dec. 1991, 1959~1969

[5] S.Liao and L.G. Chen *A low-power low-voltage direct digital frequency synthesizer*. Proc. International Symp. On VLSI Systems, Technology and Applications, 1997, 265~269

Table 1. Blocks Summary

Block	Size	Function
Accumulator	L bits	Generate phase index
Two ROMs	2^{PH} by AMP	Store sampled sine wave data in phase range of 0π~0.25π(ROM1) and 0.5π~0.25π(ROM2)
Switch Logic	Two multiplexer controlled by 3rd MSB	When 3rd MSB is 0, the data from ROM1 is fed into adder, the data from ROM2 is fed into multiplier. When 3rd MSB is 1, vice-versa.
Hardwired multiplier	PL-1 by PI (can be reduced because of constant input(π))	Transfer phase index into phase
Multiplier	M1 by SLOPE (can be reduced because only M2 output bits are useful)	Multiply delta phase and slope
Adder	AMP by M2	Generate amplitude output

Table 2. Data path width of 11bit and 12bit DDS

Design	L	W	PH	PL	SLOPE	AMP	PI	M1	M2	SFDR (dBc)
11bit	24	13	4	6	5	10	3	5	3	75.2
12bit	24	14	4	7	7	11	4	7	5	81.0

Table 3. Comparison

Method	ROM Size	Compress Ratio	Phase Resolution	Amplitude Resolution	SFDR	Additional Circuit
Uncompressed	196608	1:1	14	12	N/A	N/A
Tan[2]	3072	64:1	14	12	84.3	2 adders
Vankka[3]	1792	110:1	14	12	85.9	2 adders, multiplier
Nicholas[4]	3072	128:1	15	12	90.3	Adder, control logic
Liao[5]	832	236:1	14	12	80	2 X 4 adders
Curticapean[1]	832	236:1	14	12	85	Adder, multiplier
Proposed	352	558:1	14	12	81.0	Adder, hardwired multiplier, multiplier