International Journal of Engineering Research and General Science Volume 3, Issue 5, September-October, 2015
ISSN 2091-2730

A Novel ROM based DDFS Architecture for Portable and Wide band Communication

Parameshwara M. C* and Srinivasaiah H. C+

*Assistant Professor, Department of E & CE, Vemana Institute of Technology, VTU, Bangalore-34, Karnataka, India.

*Email:pmcvit@gmail.com, Phone: +919620902171

+Professor, Department of TCE, Dayananda Sagar College of Engineering, VTU, Bangalore-78, Karnataka, India.

+Email:hcsrinivas@gmail.com

Abstract— In this paper we propose a novel architecture for 'Read Only Memory (ROM)' based 'Direct Digital Frequency Synthesizer (DDFS)' based on simple sine trigonometric approximation formula. In this proposed architecture, to synthesize the sine wave from 0 to π/2 we use the sine samples from 0 to π/4 and the cosine samples from 0 to π/4, a digital multiplier, and a scaling block. Further the quarter wave symmetry of the sine wave is explored to derive the remaining samples that are required to synthesize the sine wave from π/2 to 2π. In the proposed DDFS architecture the sine and cosine samples are stored in two separate ROMs this saves the ROM area about 14.6 % than the traditional ROM based DDFS where the single ROM is used to store the sine samples from 0 to π/2. Further at every input clock cycle one sample from each ROM is read and multiplied to generate the first quadrant sine samples corresponding to an angle [0, π/2) with magnitude being mapped to [0, 0.5) this results in doubling the frequency of the output signal with the magnitude being halved. Further the halved magnitude of the output sine wave can be restored to full scale [0, 1) by using a scaling factor of '2' (or multiplied by 2). To verify and validate our proposed novel concept we have designed and simulated an 11-bit DDFS using MATLAB-Simulink tool. The calculated SFDR for the plotted frequency spectrum of the designed 11-bit DDFS is -66 db.

Keywords— Direct Digital Frequency Synthesizer, DDFS, Frequency synthesizer, sine ROM, cosine ROM, sine -approximation, SFDR.

1. INTRODUCTION

The tremendous growth in modern wireless communications [1] and the rapid advancements in semiconductor IC technologies [2] have made direct digital frequency synthesizers (DDFSs) as an inevitable choice for frequency synthesis. The traditional communication systems are augmented with analog phase locked loop frequency synthesizers (PLLs) because of their high output frequency capabilities and high spectral purity. In spite of these advantages the analog PLLs suffer from various issues like long frequency tuning time, high phase noise, closed loop stability issues and also bulky when implemented [5-20]. On the other hand the DDFSs offer many promising advantages viz. fine frequency steps in sub-hertz range, fast frequency switching times, smooth frequency transitions, ease of fabrication, low cost, and low power. Thus these advantages of DDFSs have made them an indispensable integral part of modern high speed digital communication systems. The simplified block diagram of standard DDFS is shown in Fig.1. This block diagram has been derived from Joseph Tierney et al. [3]. The DDFS shown in Fig.1 has an 'L' bit phase accumulator, a phase to sine amplitude mapper (or converter), a linear digital to analog converter (DAC), and a low pass filter.

Further depending on how the 'M' bit phase accumulator output is mapped onto corresponding sine amplitude of 'K' bits, the traditional DDFS are classified into two types, viz. digital recursion and direct computation based DDFSs. The digital recursion based DDFS such as CORDIC based DDFS suffer from phase noise issues. On the other hand the direct computation based DDFSs are less sensitive to phase noise where the sine samples are directly computed and stored in ROM look-up table. The stored samples are used to synthesize the digital sine wave. The sine samples can be directly computed by using the following equation [3].

$$exp\left[j\left(\frac{2\pi}{N}\right)[nk+C]\right] = cos\left(\left(\frac{2\pi}{N}\right)[nk+C]\right) + j\,sin\left(\left(\frac{2\pi}{N}\right)[nk+C]\right).$$

Where 'k' is the frequency index, 'n' is the time index, 'N' is a design parameter, and 'C' is a constant. The direct computation based

standard DDFS architectures are further classified into three main types viz. ROM, ROM less, and sine computation based architectures.

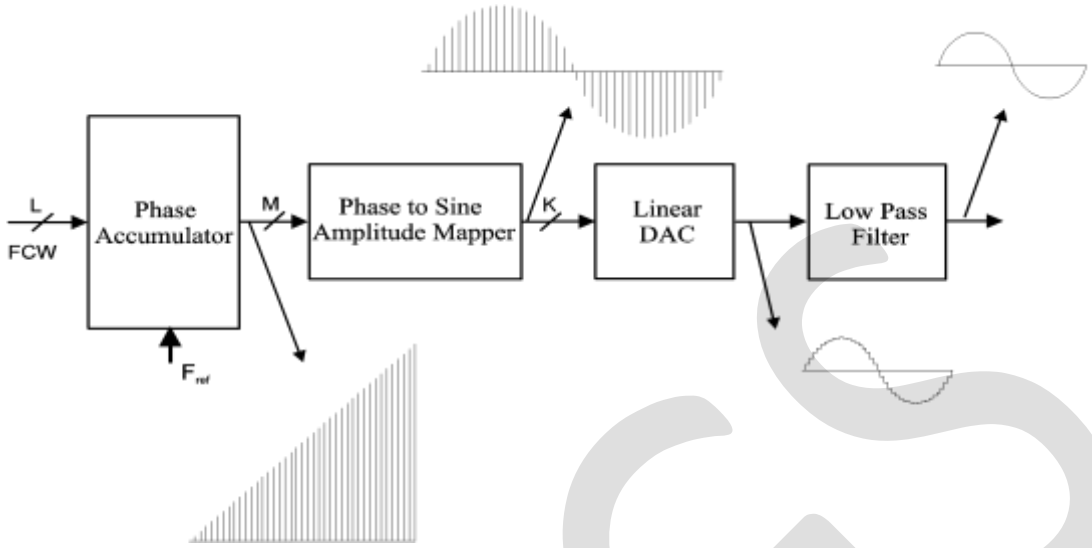


Fig. 1. Simplified block diagram of direct computation based standard DDFS [3].

In the aforementioned classification the ROM less and the sine computation based architectures are more complex and tedious in terms of hardware design and implementation. Whereas the ROM based DDFS architectures are simple in terms of hardware and hence easy to design and implement. The critical block in ROM based architectures is the ROM block that store the sine samples required to synthesize the sine output. The two parameters that affect the performance of ROM based DDFS are the size of ROM look up table and the ROM word access time. The size of the ROM look up table can be decided based on the number of words that are required to store and number of binary bits that are needed to represent each ROM word. The ROM size doubles for every 1-bit increase in input address word. The ROM word access time is the important parameter that decides the maximum frequency of operation of the ROM based DDFS. The ROM word access time is exponentially proportional to the size of the ROM. Thus it is very important to reduce the size of the ROM and hence to minimize the ROM word access time to improve the overall performance of the DDFS. In the till date literature, there are several state-of-the art ROM compression algorithms listed and are available to optimize the size of the ROM such algorithms are [5-18]:

- ROM compression using phase truncation
- Sine-phase difference
- Coarse-Fine ROM using Hutchinson algorithm
- Coarse-Fine ROM using Sunderland algorithm
- Coarse-Fine ROM using Nicholas algorithm
- Double Trigonometric approximation
- Quad line approximation
- Langlois's technique
- Liao's technique
- Parabolic approximation
- Quantization and Error ROM techniques

In this paper we propose a novel architecture for ROM based DDFS using simple trigonometric approximation. The proposed architecture uses two sub ROMs namely sine-ROM (ROM-1) and cosine-ROM (ROM-2), a multiplier, and a scaling block. In this architecture the ROM-1 stores the sine samples only from 0 to $\pi/4$ and ROM-2 stores the cosine samples only from 0 to $\pi/4$, thus saving about 14.6 % overall size of the ROM as compared to conventional ROM based architectures. The ROM word access time is also reduced which ultimately enhances the speed of the DDFS.

Rest of this paper is organized as follows: Section-2 presents the conventional ROM based DDFS architecture and its salient features. Section-3 describes about the concept of the proposed architecture of the DDFS. Section-4 discusses the MATLAB-simulink based implementation of the proposed architecture with 11 bit DDFS as a case study. Finally conclusions are drawn in section -5.

2. CONVENTIONAL ROM BASED DDFS ARCHITECTURE

The block diagram of conventional ROM based DDFS [5] is as shown in the Fig. 2. The block diagram consists of an 'M' bit phase accumulator, a π/2 sine ROM look-up table, an 'M-2' bit phase complementor, and an 'N-1' bit sine amplitude complementor. The DDFS has two inputs viz. an 'L' bit frequency control word (FCW) and a reference clock (F_{ref}). The phase accumulator accumulates an 'L' bit FCW on every cycle of F_{ref} and overflows after $2^L - 1$ clock cycles. Thus the phase accumulator generates $2^L - 1$ sine ROM addresses for each overflow and generates maximum number of addresses when FCW=1. For example if L =32 bit and FCW=1 then the phase accumulator generates 2^{32}-1 ROM word addresses. Further corresponding to $2^L - 1$ address locations we need to store the sine samples in the π/2 sine ROM and considering again the aforementioned example with L=32 bit and FCW =1, we need to store about 2^{32}-1 sine samples which is ~4 billion words. Thus, it is highly impractical to store all the words because the ROM size becomes very huge. Further the larger ROM size leads to higher power dissipation and longer word access time and hence not suitable for portable and wideband applications, where power, speed, and bandwidth (BW) are more critical. Thus one has to reduce the ROM size to fit DDFS in the aforementioned applications. There are many state-of-the art ROM compression techniques that are available to reduce the size of the ROM look up table. The Foremost technique is the phase truncation where the LSBs of the phase accumulator output are truncated to reduce the number of phase addresses required to scan the ROM, this introduces an error called 'phase truncation error'. Thus to reduce the size of the ROM look-up table phase truncation is inevitable.

In the Fig. 2 at the output of the phase accumulator 'M' most significant bits (MSBs) would be retained after the phase truncation of appropriate number of least significant bits (LSBs). Thus the phase accumulator is now required to address 2^M address locations instead of original $2^L - 1$ address locations, this greatly reduces the size of ROM look-up table at the cost of phase truncation error. Further the first two MSBs (M1 and M2) out of M bits are used to explore quarter wave symmetry of the sine wave by controlling the phase and sine amplitude complement blocks. The second MSB M2 is used to control phase complement, M2 = '0' implies phase increment and M2 = '1' implies phase decrement. The first MSB M1 is used to control sine amplitude complement, M1= '0' implies sine amplitude increment or decrement in the first half cycle of the sine wave and M1= '1' implies second half cycle of the sine wave. The M-2 output bits of the phase complementor are used to generate address locations for the π/2 sine ROM, thus the maximum number of address locations that are generated corresponding to FCW=1 is 2^{M-2}-1 and hence sine ROM needs to be stored only 2^{M-2}-1 words instead of $2^L - 1$. The 'N-1' number of bits is used to represent each word in π/2 sine ROM. Thus the total size of the π/2 sine ROM is given by $(2^{M-2}\text{-}1) \times (N - 1)$ bits. Thus the DDFS finally synthesizes an 'N" bit sine wave at its output and the frequency of the synthesized output signal is given by

$$F_{out} = FCW \times F_{res}$$

Where F_{res} is the frequency resolution and is give by

$$F_{res} = \frac{F_{ref}}{2^L}$$

Since the DDFS is basically a sampled system and hence the minimum frequency at which DDFS operates is dictated by Nyquist theorem. Thus the minimum frequency ($F_{ref} = F_{min}$) of the input clock signal that is required to operate DDFS should be at least equal to twice the maximum frequency of the DDFS output signal. However this is just a theoretical limit but in practice for all practical purposes the limitation on input reference clock is given by [4].

$$F_{ref} = F_{min} \geq 2.5 \times max(F_{out})$$

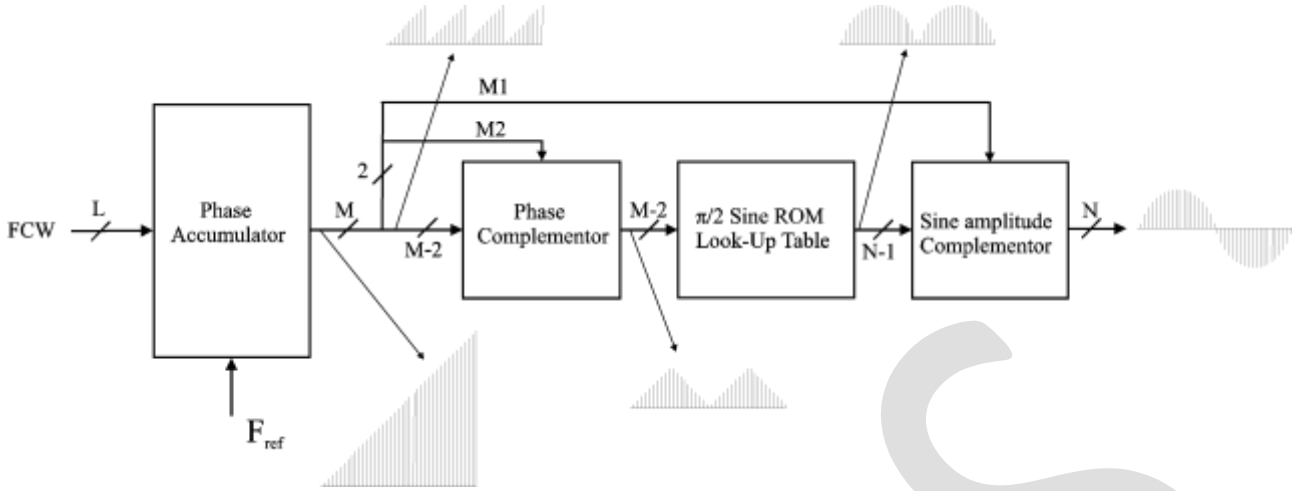


Fig. 2. Simplified block diagram of the conventional ROM based DDFS using sine quarter wave symmetry [5].

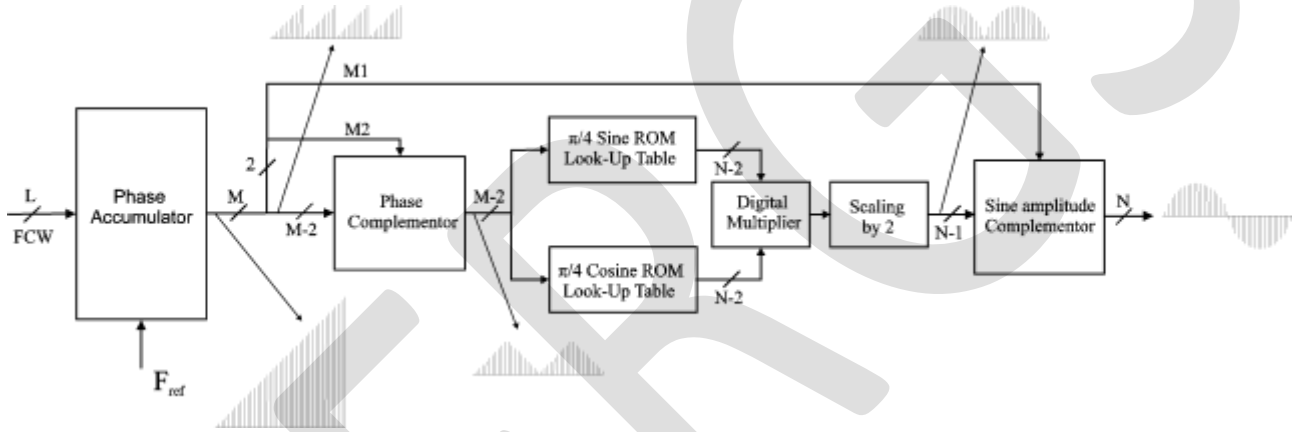


Fig. 3. Block diagram of proposed novel ROM based DDFS using sine quarter wave symmetry.

3. PROPOSED NOVEL ROM BASED DDFS ARCHITECTURE

The block diagram of the proposed novel architecture of ROM based DDFS is as shown in Fig. 3. This block diagram is derived based on sine trigonometric formula. Let us consider the sine angle summation formula, that is

$$sin(A + B) = \sin A \cos B + \cos A \sin B$$

Where 'A' and 'B' are two angles and suppose if $A = B$ then,

$$sin(A + A) = \sin A \cos A + \sin A \cos A$$

$$sin(2A) = 2 \sin A \cos A$$

Thus by using the above concept we can express $sin\left(\frac{\pi}{2}x\right)$ that is used to derive the samples to store in a π/2 sine ROM look-up table as

$$sin\left(\frac{\pi}{2}x\right) = 2\, sin\left(\frac{\pi}{4}x\right) cos\left(\frac{\pi}{4}x\right)$$

where 'x' is the phase value in the range [0, 1). Thus, we can express a π/2 sine ROM look-up table into a π/4 sine sub ROM look-up table and a π/4 cosine sub ROM look-up table.

The block diagram of proposed ROM based DDFS architecture derived based on the above concept is shown in the Fig. 3. The Fig. 3 consists of an 'M' bit phase accumulator, a phase complementor, a sine amplitude complementor, a π/4 sine sub ROM, a π/4 cosine sub ROM, a digital multiplier, and a scaling block. The M-2 bits of the phase accumulator are used to address two sub ROM blocks.

Each sub ROM stores $2^{M-2} \times N$-2 bits. The output of two sub ROM blocks are applied to the digital multiplier and subsequently scaled up by a factor 2 in order to map the amplitude of the samples corresponding to an angle between [0 π/2). At the output of scaling block N-1 bits should be retained and remaining bits would be truncated. Finally with N-1 bits along with the first MSB M1 are used to synthesize the N –bit sine wave. The main advantages of this architecture can be explained below:

- The ROM size is reduced as compared to conventional DDFS architecture, for example if M=10 and N=8, then conventional DDFS ROM requires to store $(2^{M-2}-1) \times (N-1) = (2^8-1) \times 7 = 1785$ bits, where as the proposed DDFS ROM stores $(2^{M-3}-1) \times (N-2) \times 2 = (2^7-1) \times 6 \times 2 = 1524$ bits. Thus we can save about 14.6 % overall ROM area as compared to the conventional method.
- The ROM word access time is reduced due to reduction in number of words and word size. This reduction in ROM word access time is achieved at the cost of an extra multiplier and scaling block.
- The extra multiplier delay is not that much significant compared to ROM word access delay. Suppose if multiplier delay is comparable with the delay of total ROM word access time, then the delay can be reduced by using pipelining or any other speed optimization techniques, whereas ROM operation cannot be pipelined and hence reducing the total ROM word access time at the cost of an extra multiplier does not have much impact on overall performance of the DDFS system.
- The ROM word access speed is two times faster than the conventional ROM based DDFS; this is because at every clock we read two words, one from π/4 sine sub ROM look-up table and one from π/4 cosine sub ROM look-up table. This doubles the BW of the DDFS system.
- There is a wide scope for further ROM size reduction using state-of-the art ROM compression techniques as listed in section-1.
- The proposed architecture can be modified as multiplier free by replacing multiplier using shift and add techniques.
- Considering all those above merits the proposed architecture is a best candidature for the applications such as portable and wide band communication systems where power and bandwidth are critical.

4. DESIGN AND SIMULATION OF AN 11-BIT DDFS BASED ON NOVEL ARCHITECTURE

To verify the feasibility of the proposed ROM based DDFS architecture, we have designed an 11-bit DDFS with L=16-bits, M=12-bits, and N=11-bits using MATLAB-simulink tool version R2012a. The Fig. 4 shows the design and implementation of an 11-bit DDFS. Fig.4 (a) is a 12-bit phase accumulator, where the unsigned 16-bit integer FCW value is stored in a register and is accumulated at every reference clock cycle. Further the output of 16-bit phase accumulator is converted into 16-bit binary using integer to binary conversion block and this output of the phase accumulator is represented by port '1' (output1) and is applied as an input to the Fig. 4 (b) represented by port '1' (input1). Fig. 4 (b) is the phase complementor sub system, the main functionality of this sub system is to control phase increment or phase decrement. This sub system consists of a demux block, a complementor block composing a set of XOR gates, and a mux block. The function of the demux block is to truncate 4-LSBs of 16-bit unsigned binary number that is received at its input (input1) and then output the remaining 12 MSBs. Out of the 12 MSBs the first two MSBs referred as 'M11' and 'M10' respectively are used to control the phase increment or decrement and to generate positive or negative sine half cycles. The remaining 10 bits (12MSBs-2 MSBs) viz. M9, M8, ……., M0 are applied to 10 XOR logic gates and each of these 10 bits should be applied at the second input of each respective XOR gate. Whereas the first input of each 10 XOR gates should be connected to the MSB 'M10'. The M10 is used to control phase increment or decrement, the M10 = '0' indicates phase increment and M10 = '1' indicates phase decrement. The outputs of 10 XOR gates are applied as an input to the mux. The mux block combines 10 XOR gates output into a single bus wide 10 bits. The output of this mux block indicated by port '2' (output2) is applied as an input (input2) to the sub system shown in Fig. 4 (c).

The Fig. 4 (c), is a phase to amplitude converter sub system, this sub system consists of binary to integer conversion block, sine and cosine look-up tables, a multiplier, a scaling block, and an integer to binary converter block. The functionality of this sub system is explained as follows: the 10 bit binary phase signal that is received at port 2 (input2) is first converted into unsigned 10 bit integer and is then used to address the two sub ROM look up tables viz. sine ROM and cosine ROM look up tables. Where, the sine ROM look up table stores the sine samples from 0 to π/4 and cosine ROM stores the samples from 0 to π/4. At each reference cycle one sample from each look-up table is read and applied to a multiplier block. The multiplier block multiplies the samples received from the look-up tables and then outputs the sine samples corresponding an angle [0, π/2). Further the output of the multiplier is scaled up by factor '2' in order to map the amplitude of the sine samples corresponding to an amplitude in the range [0, 1). The sine samples so obtained are in the range [0, 1) are first converted into integer values and subsequently converted from integer to 10 bit binary signal. This binary signal is available at the port '3' (output3).

The Fig. 4 (d) is the sine complementor block and is used to generate the positive or negative half cycle of the sine wave that is to be synthesized. This block consists of demux, a set of XOR logic gates, and a mux block. The working operation of this block is similar to the Fig .4 (b). Here the signal to be complemented is sine amplitude and the signal that is used to control this complementary operation is the very first MSB of the phase accumulator truncated output that is 'M11'. The M11 = '0' implies the positive half cycle

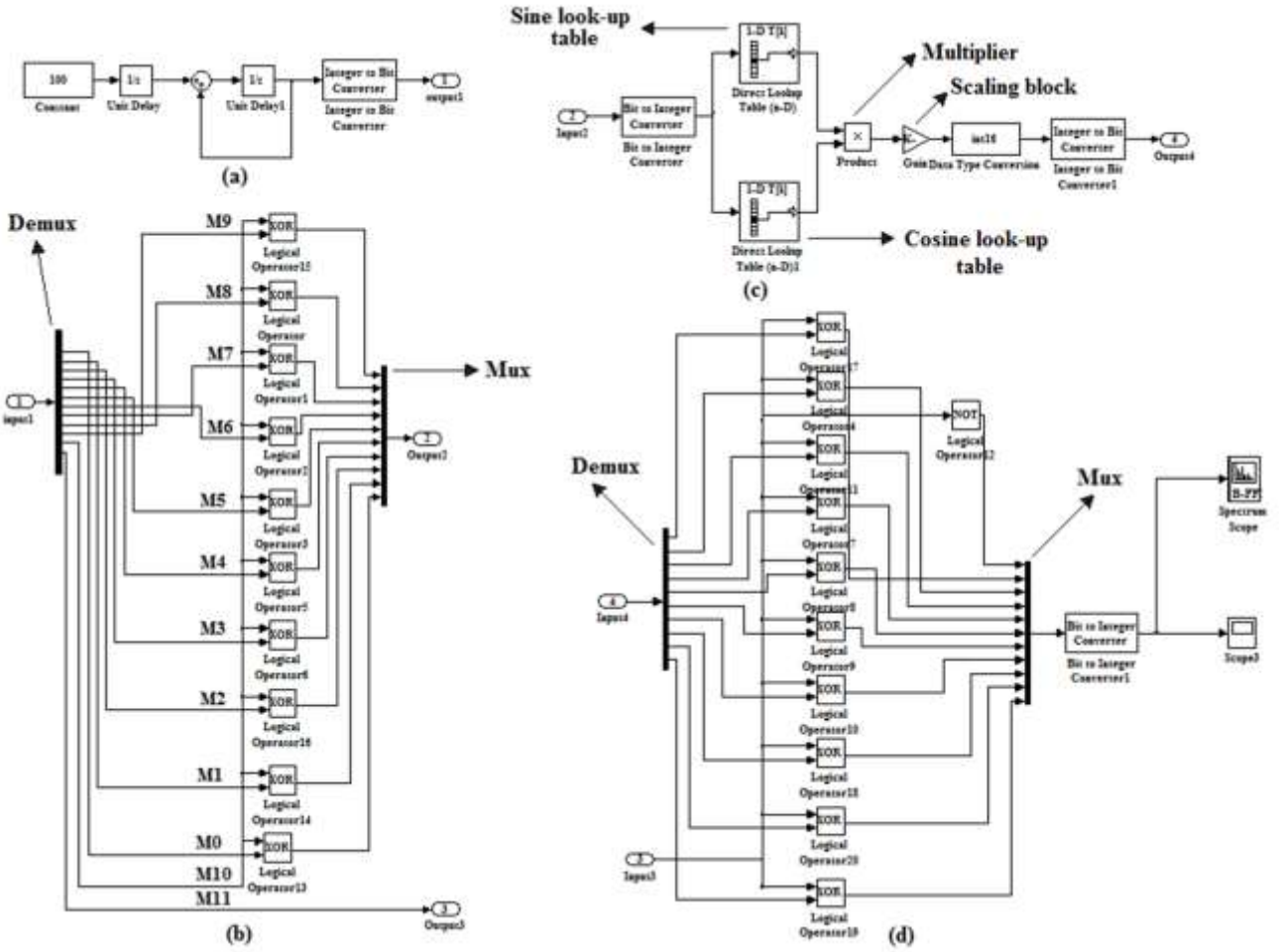
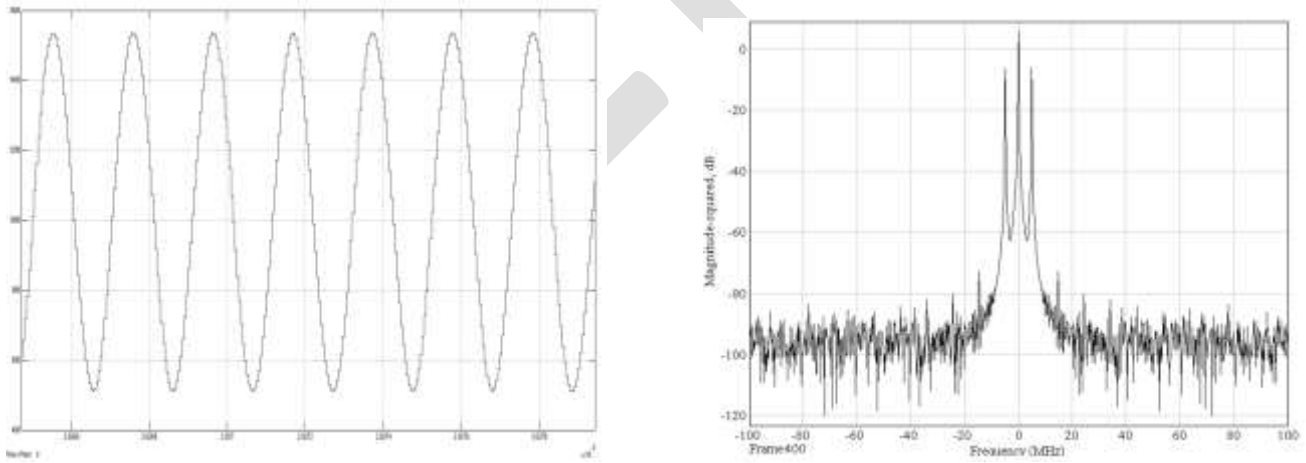


Fig. 4. An 11-bit DDFS designed and implemented in MATLAB-Simulink tool: (a) Phase accumulator (b) Phase complementor (c) Phase to sine converter (d) Sine amplitude complementor.



(a) (b)

Fig. 5. Simulation results of 11-bit DDFS for FCW =100: (a) Synthesized sine wave (b) Frequency spectrum of the synthesized sine wave .

of sine wave and M11 = '1' implies negative half cycle of the sine wave that is to be synthesized.

The design is simulated with FCW =100 and Fref = 200 MHz and the simulated results shows that the sine wave that is synthesized is having an output signal frequency of 4.88 MHz and measured SFDR indicates a spectral purity of -66 db. The synthesized sine wave and its frequency spectrum plot are shown in the Fig. 5 (a) and Fig. 5 (b) respectively. Further the ROM size and the ROM

compression ratio that is achieved using the proposed technique with some of the standard compression technique are compared and the results are tabulated in Table-1. For L=16, M=12, and N=11 The proposed architecture achieves the total ROM size of 9216 bits and compression ratio of 78.22 with respect to uncompressed memory without phase truncation.

ACKNOWLEDGMENT

Authors acknowledge management, Vemana Institute of Technology and Dayananda Sagar Group of Institutions (DSI), Bangalore for all their support in pursuing this research in the Research centre, Department of Telecommunication Engineering, Dayananda Sagar College of Engineering, Bangalore, VTU, Karnataka, India.

5. CONCLUSION

In this paper we have proposed a novel architecture for the ROM based DDFS. The proposed DDFS architecture uses two sub ROMs instead of a single ROM. The two sub ROMS are used to store samples of sine and cosine functions corresponding to an angle $[0, \pi/4)$. The remaining samples corresponding to an angle $[\pi/4, \pi/2)$ can be derived using a digital multiplier and a scaling block. Thus by storing the samples in two separate sub ROMs corresponding to angle $[0, \pi/4)$ the proposed architecture reduces the ROM area about 14.6 % compared to the conventional ROM based architecture. Also the proposed architecture generates two cycles of the output signal for each overflow of the phase accumulator corresponding to two samples that are read at every reference clock cycle. This doubles the BW of the DDFS. Further an 11-bit DDFS based on the proposed architecture was designed and simulated using MATLAB-Simulink tool. Also the DDFS synthesizes an 11-bit sine wave with an output signal frequency of 4.88 MHz with a spectral purity of -66 db corresponding to FCW=100. Considering also these merits the proposed architecture can be considered as good candidature for the portable and wideband applications. Also there is a wide scope for further reduction of two sub-ROM sizes by using the state-of-the art compression techniques.

TABLE -1
COMPARISON OF ROM SIZE AND ROM COMPRESSION RATIOS OF PROPOSED ARCHITECTURE WITH THE STANDARD TECHNIQUES FOR L=16-BITS, M=12-BITS, AND N=11-BITS

ROM compression technique used	Total number of ROM bits required	Total compression ratio achieved	Additional hardware required
Uncompressed memory without phase truncation	$2^{16}\times11=720896$	1:1	-
Compressed memory with phase truncation	$2^{12}\times11=45056$	16:1	-
Compressed memory with phase truncation and sine quarter wave symmetry (CM-PTQS)	$2^{10}\times10=10240$	70.4:1	-
Sine phase difference using CM-PTQS	$2^{10}\times8=8192$	88:1	1-adder
Double trigonometric approximation using CM-PTQS	$2^{10}\times7=7168$	100.57:1	1-adder , 1-complementor
Quad line technique using CM-PTQS	$2^{10}\times6=6144$	117.33:1	2-adders, 1-complementor, 1-mux
Proposed technique using CM-PTQS	$2^{9}\times9\times2=9216$	78.22:1	1-multiplier, 1-scaling block

REFERENCES:

[1] F. Akyildiz, David M. Gutierrez-Estevez, and Elias Chavarria Reyes, "The evolution to 4G cellular systems: LTE-Advanced," Elsevier Journal of Physical Communication, pp. 217-244, 2010.
[2] 2013PIDS_Summary.pdf, available at http://www.itrs.net/ITRS.
[3] J. Tierney, C. Rader, and B. Gold, "A Digital Frequency Synthesizer,'' IEEE Trans. Audio Electro acoustics, vol. AU-19, no. **1,** pp. 48-56, Mar 1971.
[4] Emanuele Lopelli, Johan D. van der Tang, and Arthur H. M. van Roermund, "Minimum Power-Consumption Estimation in ROM-Based DDFS for Frequency-Hopping Ultralow-Power Transmitters," IEEE Transactions on circuits and systems—I: Regular Papers, Vol. 56, No. 1, January 2009.
[5] Byung-Do Yang, Jang-Hong Choi, Seon-Ho Han, Lee-Sup Kim, and Hyun-Kyu Yu, "An 800 MHz Low-Power Direct Digital Frequency Synthesizer With an On-Chip D/A Converter*",* IEEE Journal of solid-state circuits, Vol. 39, no. 5, pp. 761-774, May 2005.
[6] H. T. Nicholas, III, H. Samueli, and B. Kim, "The optimization of direct digital frequency synthesizer performance in the presence of finite word length effects," in Proc. 42nd Annual Frequency Control Symposium USER-ACOM, pp. 357–363, May 1988.
[7] 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," in Proc. 41st Annual. Frequency Control Symposium.USERACOM, pp. 495–502, May 1987.
[8] A. Yamagishi, M. Ishikawa, T. Tsukahara, and S. Date, "A 2-V, 2-GHz low-power direct digital frequency synthesizer chip-set for wireless communication," IEEE J. Solid-State Circuits, vol. 33, pp. 210–217, Feb. 1998.
[9] J. M. P. Langlois and D. Al-Khalili, "ROM size reduction with low processing cost for direct digital frequency synthesis," in Proc. IEEE Pacific Rim Conf. Communications, Computers and Signal Processing, pp. 287–290, Aug. 2001.
[10] B.-D. Yang and L.-S. Kim, "A direct digital frequency synthesizer using a new ROM compression method," in Proc. Eur. Solid-State Circuits Conf., pp. 288–291, 2001.
[11] A. M. Sodagar and G. R. Lahiji, "Mapping from phase to sine-amplitude in direct digital frequency synthesizers using parabolic approximation," IEEE Trans. Circuits Syst. II, vol. 47, pp. 1452–1457, Dec. 2000.
[12] S. Liao and L.-G. Chen, "A low-power low-voltage direct digital frequency synthesizer," in Proc. Int. Symposium. VLSI Technology, Systems, and Applications, pp. 265–269, June 1997.
[13] H. T. Nicholas III and H. Samueli, "A 150-MHz direct digital frequency synthesizer in 1.25µm CMOS with 90-dBc spurious performance," IEEE J. Solid-State Circuits, vol. 26, pp. 1959-1969, Dec. 1991.
[14] Loke Kun Tan, and Henry Samueli, "A 200MHz Quadrature Digital Synthesizer/Mixer in 0.8µm CMOS," IEEE Journal of Solid State Circuits, Vol. 30, No. 3, pp. 193-200, March 1995.
[15] 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 JSSC, pp. 497-505, Aug. 1984.
[16] J. Vankka, "Methods of Mapping from Phase to Sine Amplitude in Direct Digital Synthesis", IEEE Tr. On ultrasonics, ferroelectrics, and frequency control, Mar. 1997, pp. 526-534.
[17] B.H. Hutchison, Jr., Frequency Synthesis and Applications, IEEE Press, 1975.
[18] Bar-Giora Goldberg, "Digital frequency synthesis demystified," LLH Technology Publishing, 1999.
[19] Bennet C. Wong, and Henry Samueli, "A 200MHz All-Digital QAM Modulator and Demodulator in 1.2µm CMOS for Digital Radio Applications," IEEE Journal of Solid State Circuits, vol. 26, no 12, pp. 1970-1979, Dec. 1991.
[20] F. P. Chan, M. P. Quirk, and R. F. Jurgens, "High-Speed Digital Baseband Mixer," TDA Progress Report 42-81, Communication System Research Section, January-March, pp. 63-80, 1985.