http://dx.doi.org/10.5755/j01.eee.19.10.5905 ELEKTRONIKA IR ELEKTROTECHNIKA, ISSN 1392-1215, VOL. 19, NO. 10, 2013

FPGA-Based Implementation of a New Phase-to-Sine Amplitude Conversion Architecture

Q. K. Omran[1,2], M. T. Islam[3], N. Misran[1]
[1]*Department of Electrical, Electronic and Systems Engineering, National University of Malaysia, Bangi 43600, Selangor, Malaysia*
[2]*University of Diyala,Diyala, Iraq*
[3]*Institute of Space Science (ANGKASA), National University of Malaysia,*
qahtanen@eng.ukm.my

[1]***Abstract*—The classical structure of linear interpolation-based phase-to-sine mapper (PSM) consists of at least two ROMs for polynomial coefficient storage. Other architectures may include extra ROM for storing residual errors. However, ROMs dissipate high power and occupy a significant amount of the die area. This study presents a new technique that eliminates the ROM by including the computation of segment initial coefficients in the hardware. Therefore, it becomes possible to trim down noticeable hardware resources. The proposed direct digital frequency synthesizer (DDFS) architecture has been encoded in VHDL and synthesized with Quartus II software. Post simulation results show that the proposed design is capable of achieving the theoretical spurious-free dynamic range (SFDR) upper bound when optimal polynomial coefficients are considered. For 32 piecewise linear segments, the SFDR of the synthesized sinusoid is 84.15 dBc. A ROM compression ratio of 597.3:1 was also achieved. The performance of the DDFS is compared with previously presented DDFS techniques and the results show that the proposed design has advantages of high ROM compression ratio and low hardware complexity.**

***Index Terms*—Direct digital frequency synthesizer, DDS, phase to sine amplitude conversion, piecewise linear approximation.**

I. INTRODUCTION

Direct digital frequency synthesizers (DDFSs) are capable of producing sine output waveforms with ultra-thin frequency increments, fast frequency switching, and high spectral purities. The synthesized signal is primarily digital in the DDFS; thus, the DDFS can be incorporated with different digital modulations because of the ease in handling the frequency, phase, and amplitude in the digital domain. Moreover for a number of applications it is required to switch the DDFS output frequency in some predefined pattern. The most obvious application would be a DDFS-based chirp signals used in radar system, spread spectrum communications, and as the stimulus in bio-impedance measurement. Among other frequency synthesis types, DDFS appears the most suitable technique for such applications. DDFS exhibits flexible tune capability over a wide frequency range with very short time. The more important feature is that the duration of chirp signal and its frequency range can be adjusted independently [1].

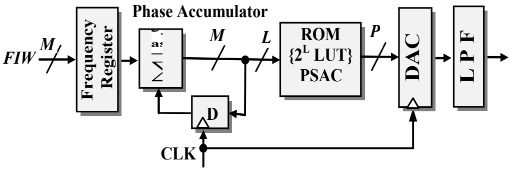

Fig. 1. ROM-based DDFS basic architecture.

The ROM-based DDFS architecture was first introduced by Tierney et al. [2]. As displayed in Fig. 1, the basic structure of the ROM-based DDFS consists of three major blocks; phase accumulator, phase to sine amplitude converter (PSAC), and digital-to-analog converter (DAC). f_{clk} represents the reference clock used by the DDFS and FIW is the frequency instruction word. At each leading edge of the f_{clk}, the phase accumulator adds an M-bit FIW. The accumulated phase value addresses the sine lookup table (LUT) to produce sine waveforms. One period of a synthesized waveform is exactly the overflow of an M-bit phase accumulator. The synthesized frequency output can be expressed as the following

$$F_{out} = FIW \frac{f_{clk}}{2^M}, \tag{1}$$

where $0 \le FIW \le 2^{M-1}$.

For a precise approximation, the ROM-based sine LUT has to be packed with sine amplitude values that correspond to each possible phase value. The phase and amplitude quantization errors are inversely proportional to the depth and width of the ROM, respectively; thus, stretching the ROM for a high spectral purity sinusoidal output is preferable. A large ROM has high power consumption and occupies a large area. These factors negatively influence the performance of the DDFS. Therefore, compressing the size of the ROM without sacrificing spectral purity is essential. Most DDFSs are developed from the architecture shown in Fig. 1. However, during the last four decades, considerable modifications have been introduced and numerous alternative architectures have been proposed to reduce the computational complexity of the PSAC. In general, these

Manuscript received March 12, 2013; accepted June 26, 2013. The research presented in this paper has been financially supported by the University of Diyala, Iraq, Department of Electronic Engineering,and Universiti Kebangsaan Malaysia, Malaysia.

methods can be categorized under three major groups; ROM compression [3]–[5], angle rotation [6]–[8], and piece-wise polynomial interpolation methods [9]–[11].

As stated in [9], [10] the piecewise linear interpolation method is regarded as an efficient technique comparable with other approximation techniques in terms of performances and hardware complexity. A generic PSAC structure based on the linear interpolation technique comprises two ROMs for storing segment initial amplitudes and segment slope coefficients. In this study, we propose a developed version of the standard linear-interpolated PSAC architecture. Our goal is to eliminate the ROM, which stores segment initial coefficients, to minimize system complexity. Once the ROM is eliminated, we expect the target system to exhibit excellent spectral purity and low power consumption with reasonable hardware overhead.

II. Piecewise Linear interpolation Basic Background

In uniform piecewise linear approximation, the first quadrant of the sine function is divided into s segments of equal length. Each segment is approximated with a first-order polynomial. Thus, $p(x)$ can be expressed as the following:

$$p(x) = \begin{cases} c_0 + m_0.(x - x_0)\,, & x_0 \le x \le x_1\,, \\ c_1 + m_1.(x - x_1)\,, & x_1 \le x \le x_2\,, \\ \vdots & \\ c_i + m_i.(x - x_i)\,, & x_i \le x \le x_{i+1}\,, \\ \vdots & \\ c_{s-1} + m_{s-1}.(x - x_{s-1})\,, & x_{s-1} \le x \le x_s\,, \end{cases} \tag{2}$$

where m_i and c_i are the polynomial coefficients of the ith segment, x the input phase scaled to a binary fraction in the interval [0, 1], s is the number of segments that is chosen to be a power of two for further simplification. Fig. 2 depicts the basic structure of the uniform piecewise linear-interpolated PSAC, where two ROMs, one multiplier, and one adder are common blocks. We aim to evaluate the initial coefficients to bypass one of the coefficients ROMs. In the following section, we show that a simple recursive substitution in each polynomial segment enables the segment initial amplitude coefficients to be derived from the slope coefficients; accordingly, ROM elimination becomes doable.

III. The Proposed Modification

As mentioned before the segment initial amplitude coefficients c_i can be obtained by recursive substitution in each segment polynomial. In each subinterval of (2), the sine function is approximated by a first-order polynomial with the following form

$$p(x) = c_i + m_i.(x - x_i)\,, \tag{3}$$

where $x_i \le x \le x_{i+1}, \;\; i = 0..s$.

For a uniform piecewise linear approximation, the segments are equal in length and the segment bounds x_i are equal to (i/s). Starting from the first interval, the segment initial coefficient c_0 and segment lower bound x_0 are equal to zero. Thus, the first segment polynomial is expressed as the following

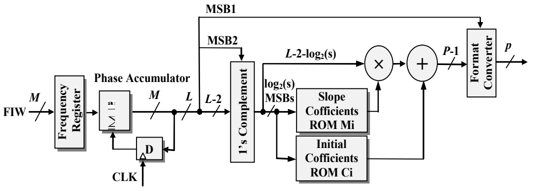

Fig. 2. The linear interpolated DDFS basic structure.

$$p_0(x) = m_0.x\,, \tag{4}$$

where $0 \le x \le \frac{1}{s}$.

By substituting the segment bound 1/(s) into (4) we can find the second segment initial coefficient c_1 as follows

$$c_1 = p_0(x = \tfrac{1}{s}) = m_0.\tfrac{1}{s}\,. \tag{5}$$

Thus, $p_1(x)$, the second segment polynomial

$$p_1 = c_1 + m_1.(x - x_1)\,, \tag{6}$$

where $\frac{1}{s} \le x \le \frac{2}{s}$, can be expressed in terms of slope coefficients by substituting (5) into (6) as follows

$$p_1(x) = m_0.(\tfrac{1}{s}) + m_1.(x - \tfrac{1}{s}), \tag{7}$$

where $\frac{1}{s} \le x \le \frac{2}{s}$.We apply the same procedure for segment number two by substituting the segment boundary (2/s) into (7). Therefore, the third segment initial coefficient c_2 is expressed as the following

$$c_2 = p_1(x = \tfrac{2}{s}) = m_0.(\tfrac{1}{s}) + m_1.(\tfrac{2}{s} - \tfrac{1}{s}) = (m_0 + m_1).(\tfrac{1}{s}). \tag{8}$$

And $p_2(x)$ can readily be expressed as follows

$$p_2(x) = \tfrac{1}{s}(m_0 + m_1) + m_2.(x - \tfrac{2}{s}), \tag{9}$$

where $\frac{2}{s} \le x \le \frac{3}{s}$.

Following the same procedure for the ith segment polynomial

$$p_i(x) = c_i + m_i.(x - x_i)\,, \tag{10}$$

where $\frac{1}{s} \le x \le \frac{i+1}{s}$. We can, in general, deduce the segment initial coefficient c_i as the following

$$c_i = \tfrac{1}{s}(m_0 + m_1 + \cdots + m_{i-1}). \tag{11}$$

By substituting (11) into (10), $p_i(x)$ becomes the following

$$\begin{aligned} p_i(x) &= \tfrac{1}{s}(m_0 + m_1 + \cdots + m_{i-1}) + m_i.(x - x_i) = \\ &= \tfrac{1}{s}\left(\textstyle\sum_{j=0}^{i-1} m_j\right) + m_i.(x - \tfrac{i}{s}), \end{aligned} \tag{12}$$

where $\frac{i}{s} \le x \le \frac{i+1}{s}$. We can then rewrite (2) as follows

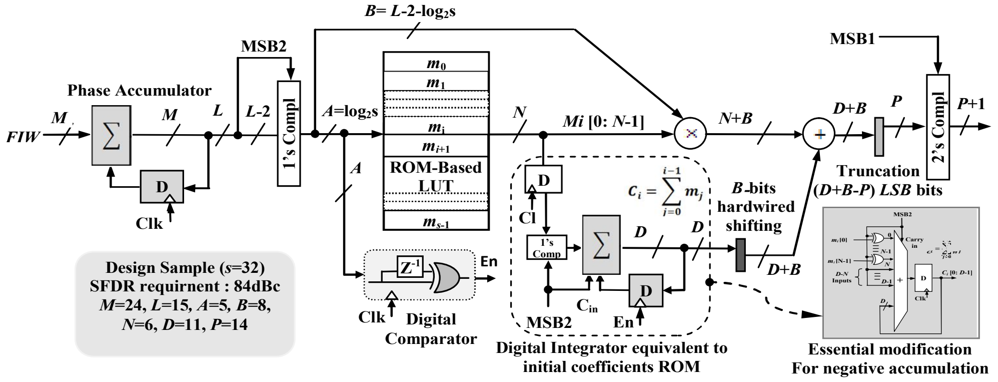

Fig. 3. The proposed DDFS architecture.

$$p(x)=\begin{cases} m_0.x,\ 0\le x<\frac{1}{s}, \\ m_0.(\frac{1}{s})+m_1.(x-\frac{1}{s}),\ \frac{1}{s}\le x<\frac{2}{s}, \\ \vdots \\ \frac{1}{s}\left(\sum_{j=0}^{i-1}m_j\right)+m_i.(x-\frac{i}{s}),\ \frac{i}{s}\le x\le\frac{i+1}{s}, \\ \vdots \\ \frac{1}{s}\left(\sum_{j=0}^{s-2}m_j\right)+m_{s-1}.(x-\frac{s-1}{s}),\ \frac{s-1}{s}\le x\le\frac{s}{s}. \end{cases} \tag{13}$$

At this point, the initial coefficients are successfully replaced by accumulated pervious slope coefficients, thus allowing the ROM to be replaced with a simple accumulator. We show in subsequent sections of the paper that the hardware resources of the counterpart accumulator are significantly less than the replaced ROM hardware resources.

IV. THE PROPOSED DDFS ARCHITECTURE

Based on the theoretical approach presented in the previous section, we introduce a single coefficient ROM architecture displayed in Fig. 3.

The initial coefficients ROM is replaced by a simple digital accumulator, which is depicted in the dashed-line rounded rectangle. The accumulator is simply a digital Integrator in which its output is an integral of the slope coefficient equivalent to the initial coefficient. Furthermore, memory requirements are reduced by a factor of four by exploiting the quadrant symmetry of the sine function. Accordingly the architecture has to perform both positive and negative accumulation. For this purpose the 1's complement block is needed. The accumulator word length is given by

$$D=\left\lceil \log_2\left(\sum_{i=0}^{s-1}dec(m_{qi})\right)\right\rceil. \tag{14}$$

Or simply equal to (N+log$_2$s) bits long, where ⌈.⌉ denotes the ceiling function, dec(m_{qi}) represent the decimal value of the ith quantized slope coefficient, and N the slope coefficient word length.

According to (13), for a given segment i, the accumulator has an instance value of $\sum_{j=0}^{i-1}m_j$ which represents the segment initial coefficient c_i. This value must be kept unchanged during the segment interval. In doing this, the architecture has to initiate one accumulation cycle coincident with each segment's transition. For this purpose, a digital comparator is used to monitor the ROM address bus (the segment selector) for detecting the events of segment's transition. Thus, the comparator output signal En is responsible for initiating the accumulation cycle.

The phase boundary value (π/2) is quantized to L – 2-bits; thus, the segment bound is $B = L - 2 - \text{Log}_2 s$ bits. In this case, the output of the accumulator must be shifted left by B–bits before adding the resulting coefficients to the multiplier output , as a result the adder has a word length of (D+B) bits. Hardwired shifting does not involve any digital gate. Finally, the output of the adder has to be truncated to $P = L - 1$ word length to accommodate the required DAC resolution. The ROM size required for this architecture is $2^A \times$ N bits. Compared with the conventional counterpart architecture, which has an additional initial coefficient ROM of $2^A \times (L - 1)$ bits, the proposed algorithm can save $2^A \times (L - 1)$ memory entries with the penalty of the D-bit additional accumulator.

V. SAMPLE DESIGNS AND PERFORMANCE

Following the proposed architecture shown in Fig. 3, we consider a design sample and show the best possible computational cost in this section. We assume that the first quadrant of the sine function is divided into 32 segments (s = 32). The same design procedure can be used for any different number of segments with similar results. In [9], [10] it is stated that with uniform piecewise linear approximation, the maximum achievable SFDR for a certain number of segments is given by

$$SFDR_{dBc}=20\log(1+16s^2)\approx 24.08+40\log(s). \tag{15}$$

Thus, the targeted SFDR of 84.286 dBc has to be achieved. As a rule of thumb, with L-bits phase resolution, spurs introduced by phase truncation is given by -6.02L dB, thus, the system parameters of this design can be obtained as follows : L = 15 bits, $P = L - 1$ = 14 bits, A = log2 (s) = 5, and $B = L - 2 - A = 8$. The width of the ROM, added accumulator size, multiplier, and adder feed inputs all depend on N, which is the slope coefficient word length.

Hence, to complete the design with minimum hardware overhead, we have to minimize the polynomial coefficient word length N which is the key parameter that determines the performance of the PSM.

By knowing the N, we can easily obtain the accumulator word length D by using (14), the size of the multiplier, and the adder feed inputs. In doing so, the optimal polynomial coefficients have to be obtained first and then quantized on a given number of bits to achieve the targeted SFDR level.

VI. Optimal Polynomial Coefficients

To minimize the approximation error, the minimum-mean-square-error (MMSE) criterion is employed:

$$MMSE = \min_{[m_i=0..s-1]} \int_0^{\pi/2} [\varepsilon_r(x)]^2\, dx, \tag{16}$$

where $\varepsilon_r = \sin(x) - p(x), \quad 0 \le x \le \pi/2$.

With aid of a powerful Maple optimization package, the optimal set of slope coefficients, m_i (i = 0.. s-1) is obtained and presented in Table I.

TABLE I. OPTIMAL SLOPE COEFFICIENTS.

i	*m_i*	*i*	*m_i*	*i*	*m_i*
0	0.99977007	11	0.84493828	22	0.44965506
1	0.99751977	12	0.81766366	23	0.40528070
2	0.99244191	13	0.78842327	24	0.35993011
3	0.98540874	14	0.75728244	25	0.31371196
4	0.97578671	15	0.72431753	26	0.26673977
5	0.96387676	16	0.68960760	27	0.21911842
6	0.94960230	17	0.65323637	28	0.17099434
7	0.93321163	18	0.61529142	29	0.12236189
8	0.91415493	19	0.57586419	30	0.07380440
9	0.89334866	20	0.53504965	31	0.02365137
10	0.87016184	21	0.49294613		

The $p(x)$ has quadrant symmetry; thus, the spectrum of the $p(x)$ is inevitably free of even harmonics and can be expressed as a Fourier sine series: as follows

$$p(x) = \sum_{n=1}^{\infty} b_n Sin(\tfrac{n\pi}{2} x), \tag{17}$$

where $0 \le x \le 1, \quad n = 1,3,5\cdots$, and the b_n, the magnitude of odd harmonics is given by

$$b_n = 2\int_0^1 p(x) Sin(\tfrac{n\pi}{2} x) dx, \tag{18}$$

where $n = 1,3,5$.

Figure 4 shows the spectrum of the $p(x)$, where the largest unwanted frequency component occurs at the harmonic (4s – 1) and has an amplitude of −84.15 dB with respect to the target sinusoid. It is clear from the same figure that the resulting non-quantized optimal coefficient m_i s, can satisfy the theoretical finding of the SFDR upper bound of (15). Figure 5 shows the residual error of the approximated sinusoidal wave. The greatest maximum absolute error is equal to 1.9 × 10^{-4} and can be seen in the last linear segment (i = s).

VII. Quantization of Polynomial Coefficients

To obtain efficient hardware realization, the optimal real-valued coefficients (detailed in Table I) have to be quantized with sufficient finite precision. The rounded quantized coefficient m_{qi} can be obtained by the following

$$m_{qi} = \left\lfloor 2^N m_i + 0.5 \right\rfloor, \tag{19}$$

where [.] denote the floor function, N is the coefficient word length, and 0.5 ensures that the half-way values ($2^N m_i$) are rounded up. Reducing the coefficient word length to minimize the LUT size and simplifying the arithmetic circuitry is highly desirable; however, the resulting poor accuracy due to excessive quantization may further decrease the SFDR level. Thus, the design has to balance circuit complexity against quantize accuracy. To achieve this balance, the SFDR level has to be checked for each quantize accuracy starting from the lowest accuracy.

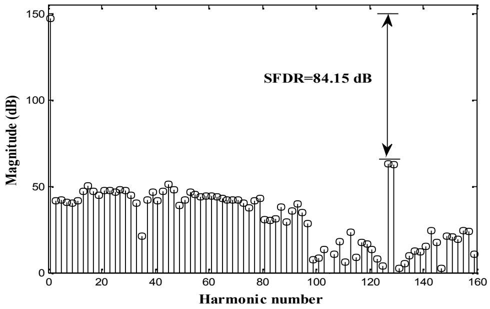

Fig. 4. The spectrum of the $p(x)$.

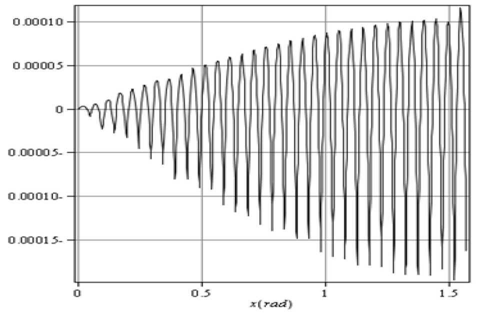

Fig. 5. The residual error of the synthesized curve.

The quantization process is started with N = 4-bits. The resulting spurious level is checked on whether it satisfies the targeted SFDR level. According to Fig. 6, the SFDR with 4-bit coefficient word length is about 73.2 dBc which is far below the theoretical SFDR upper bound. Thus, we have to quantize the coefficients by using a 5-bits and above. From the graph with 6-bit quantization trial, it can meet an SFDR of 82.8 dBc, which is just 1.2 dB below the maximum achievable SFDR. The available results from the 7-bit and 8-bit trials do not exhibit considerable SFDR improvement. Each additional bit will increase the LUT by s bits and extend the accumulator and multiplier size by 1 bit. Consequently, a 6-bit quantization resolution has been

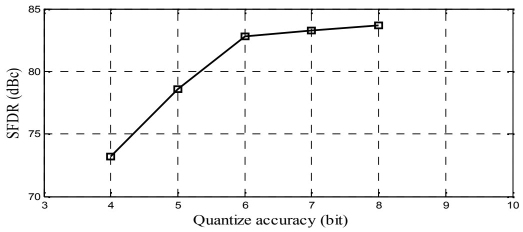

Fig. 6. SFDR level for different quantization accuracy.

considered as a compromise solution with an acceptable spurs level of 82.8 dBc. The quantized slope coefficients are shown in Table II.

TABLE II. QUANTIZED SLOPE COEFFICIENTS.

Slope coefficient m_{Qi} (quantized with 6 bits)											
63	63	62	62	61	61	60	59	57	56	55	53
51	50	48	46	43	41	39	36	34	31	28	25
23	20	17	14	11	8	4	2	1			

Once again, the spectrum of quantized $p_q(x)$ needs to be determined to show the effect of the quantization process. Figure 7 represents the resulting spectrum where the largest unwanted frequency component has an amplitude of −82.8 dBc, which occurs at the harmonic (4s + 1).

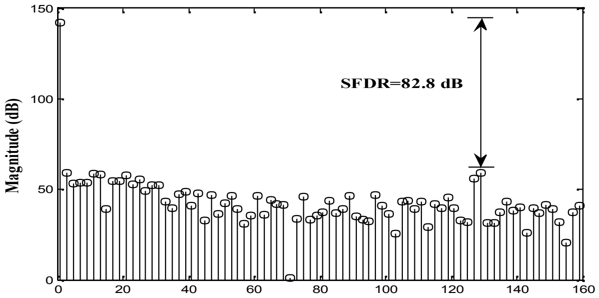

Fig. 7. The spectrum of the quantized p_q (x).

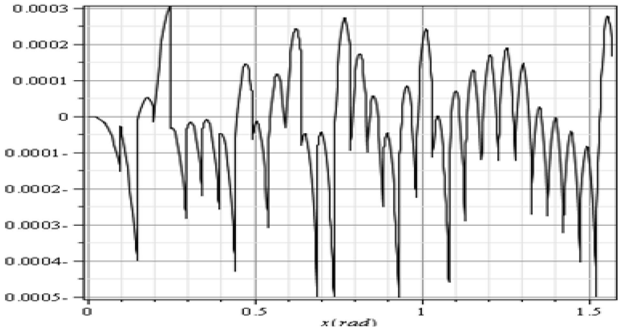

Fig. 8. The approximate error $\varepsilon_{rq}(x)$.

Figure 8 shows the residual error of the approximated sinusoidal wave with a 6-bit coefficient word length. Unlike the non-quantized errors shown in Fig. 5 the quantized errors appear randomly distributed over the segment lines because of the nonlinear truncation and rounding processes.

VIII. Structural Design Improvements

The architecture displayed in Fig. 3 has to be improved to achieve a well-organized hardware. First, for the Multiply-Add circuitry shown in Fig. 9(a), we suggest two scenarios shown in Fig. 9(b) and Fig. 9(c). In the first proposed scheme, the output of the digital integrator after hardwired shifting has to be added to the $m_i.x$ product

$$Add[18:0] = 2^8 c_i[10:0] + m_i x[13:0]. \qquad (20)$$

The size of the first term is 18 bits and its right hand side contain 8 zeroes. Thus, the 8 least significant bits (LSBs) of the $m_i.x$ product are concatenated to the resultant output. The addition can be expressed as follows

$$Add[18:0] = m_i x[7:0] \,\&\, 2^8 \left(c_i[10:0] + m_i x[13:8]\right). \, (21)$$

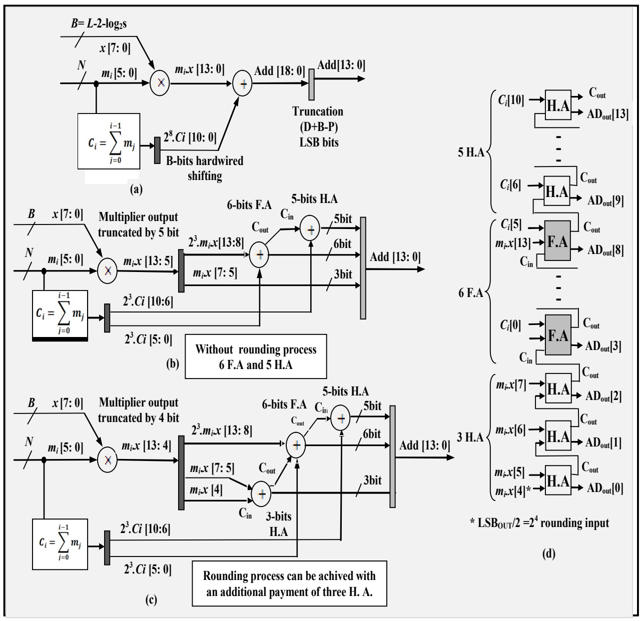

Fig. 9. (a) Multiply-Add data flow; (b) Without rounding; (c) With rounding; (d) The realized circuit.

The notation is used in the VHDL hardware description language. The adder output is $D + B = 18$ bits, which does not match the 15-bit sine output resolution. Thus, the adder output word length has to be truncated by 5 bits. The adder after truncation is defined as follows

$$Add[13:0] = m_i x[7:5] \,\&\, 2^3 \left(c_i[10:0] + m_i x[13:8]\right). \qquad (22)$$

Indeed the first 5 LSBs of the $m_i.x$ products has been truncated (even in final stage). Thus, truncating the the $m_i.x$ product in the early stage is preferable. In doing so the multiplier output become 9 bits, which leads to a noticeable logic gate saving. Following this procedure, the proposed scheme requires a (6 × 8) bit multiplier with 9 bits of output, 6 full adders (F. A), and 5 half adders (H. A). No rounding process has been applied. Defiantly, in the first scenario, the truncation of the 5 LSBs of the $m_i.x$ product will introduce an arithmetic error. For this purpose, the rounding technique must be applied. The rounding process is usually realized by adding a constant value equal to $\mathrm{LSB_{OUT}}/2 = 2^4$ and then truncating the result.Thus, we have to add $m_i.x$ [4] to the final result

$$Add[13:0] = m_i x[4] + m_i x[7:5] + \\ +2^3 \left(c_i[10:0] + m_i x[13:8]\right). \qquad (23)$$

The implementation of such a scheme is displayed in Fig. 9(c). This scheme requires 3 additional H.A payments as a penalty for the rounding process, and the multiplier has to be truncated with only 4 LSBs. The exact realized circuit is shown in Fig. 9(d). The proposed architecture shown in Fig. 3 still requires improvement. The second modification allows the two's complementer to be replaced at the input of the accumulator with a simpler one's complementer. The MSB2 of the input phase feeds the Carry-in of the first adder. Therefore, it becomes possible to save the +1 adder that is essential to perform the two's complement

TABLE III COMPARISON WITH REPORTED WORK.

Architecture	ROM Required (bits)	Compression Ratio	Significant Additional Logic Circuits	SFDR (dBc)	Comments
[9] 2003 Langlois and Al Khalili	Register $2^5\times 14$	256:1 (with respect to $2^{13}\times 14$ ROM)	Three 14-bit adders, three 9-bit 32 to 1 multiplexers, one 14-bit 32 to 1 multiplexer, and 96 inverters	84.2	Highly circuitry needed
[4] 2006 L. S. Chimakurthy, et al.	$2^8\times 12$	37.3:1 (with respect to $2^{13}\times 14$ ROM)	Twice clock frequency, 3 adders, one multiplier, one shift register and frequency divider	89	More circuitry needed with very low compression ratio
[5] 2008 Lai Lin-huil Li Xiao-jinl	328	75:1 (with respect to $2^{11}\times 12$ ROM)	8 adder, 6 shift register , and 4 multiplexers	63.58	Highly circuitry needed with Poor spur level and low compression ratio
Standard uniform Linear Interpolation	$(2^5\times 6)$ + $(2^5\times 14)$	179.2:1 (with respect to $2^{13}\times 14$ ROM)	One adder, one multiplier,	84.2	Low complexity with low compression ratio
The proposed Technique	$2^5\times 6$	597.3:1 (with respect to $2^{13}\times 14$ ROM)	One adder, one multiplier, 11-bit accumulator, and 12 XOR gates	82.8	Low complexity with best compression ratio

Furthermore we have to extend the output of the ROM by (D-N) MSB bits to match the accumulator inputs; otherwise, performing the negative accumulation is impossible.

IX. PERFORMANCE COMPARISON

To validate the proposed algorithm, we code the design sample and traditional piecewise linear interpolation DDFS architectures in the VHDL by using Altera Quartus II 11.0 software with the previously mentioned parameters. The designed projects are implemented after full completion by using Stratix III FPGA (EP3SE50F484C2 device).

An architecture having 32 piecewise linear segments should have a worst-case spur of −84 dBc, which is achieved as well. Figure 10 shows the output spectrum for the output frequency of 0.124 clock frequency with FIW set to 4065.

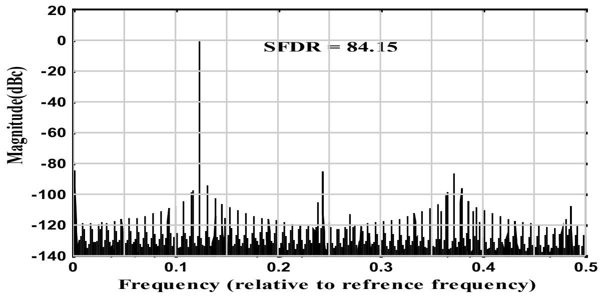

Fig. 10. Calculated output spectrum for $F_{out} = 0.124 f_{clk}$.

The characteristics of the proposed work, along with the standard uniform linear-interpolated DDFSs, are summarized in Table III and are compared with previously published algorithms. Please note that the compression ratio has been calculated with respect to ($2^{L-2}\times P$) ROM size, where L is the phase resolution and $P = L - 1$ is a sine output resolution. As a great advantage of this technique is that, it can replace the ROM size of S × (L − 1) bits required by standard architecture with (N + $\log_2$s) bit accumulator. For example, the ROM size required by the traditional approach for s = 64 is (64 × 15) bits, which can only be replaced by a 12-bit accumulator. The accumulator size is just 1 bit over the architecture of s = 32 while the compression ratio now is 758:1. Compared with the algorithms in [9], [4], and [5], the proposed algorithm exhibits the highest compression ratio with low hardware overhead.

X. CONCLUSIONS

In this paper, we have presented a develop phase-to-sinusoid amplitude conversion architecture based on linear interpolation. The initial coefficient ROM has been replaced by a simple digital Integrator. A generalized single ROM DDFS architecture utilizing this approach was presented, and a particular design with optimal polynomial coefficients of 32 linear segments is discussed. The Multiply-Add circuitry has been minimized, resulting in lower hardware implementation cost. The conventional and develop version designs have been implemented on Altera's Stratix III FPGA (EP3SE50F484C2). It is shown that the proposed approach exhibits an excellent ROM compression ratio with reasonable hardware resources in comparison with previously presented DDFS designs.

REFERENCES

[1] T. Paavle, M. Min, "Discrete-Level Broadband Excitation Signals: Binary/Ternary Chirps", *Elektronika ir Elektrotechnika (Electronics and Electrical Engineering),* no. 6, p. 23–26, 2012.

[2] J. Tierney, C Rader, B. Gold, "A digital frequency synthesizer", *IEEE Trans. Audio and Electro acoustics*, vol. 19, no. 1, pp. 48–57, Mar. 1971. [Online]. Available: http://dx.doi.org/10.1109/TAU.1971.1162151

[3] A. G. M. Strollo, D. De Caro, N. Petra, "A 630 MHz, 76 mW direct digital frequency synthesizer using enhanced ROM compression technique", *IEEE J. Solid-state Circuits*, vol. 42, pp. 350–360, Feb. 2007. [Online]. Available: http://dx.doi.org/10.1109/JSSC.2006.889382

[4] L. S. Chimakurthy, M. Ghosh, F. F. Dai, R. C. Jaeger, "A novel DDS using nonlinear ROM addressing with improved compression ratio and quantization noise", *IEEE Trans. Ultrason., Ferroelectr.,Freq. Control*, vol. 53, no. 2, pp. 274–283, Fab. 2006.

[5] Lai Lin-hui, Li Xiao-jin, Lai Zong-sheng, "A low complexity direct digital frequency synthesizer", in *IEEE 9th Int. Solid-State and Integrated-Circuit Technology Conf.* 2008, pp.1653–1656.

[6] C. Y. Kang, E. E. Swartzlander Jr., "Digit-pipelined direct digital frequency synthesis based on differential CORDIC", *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 53, pp. 1035–1044, May 2006.

[7] J. M. P. Langlois, D. Al-Khalili, "Phase to sinusoid amplitude conversion techniques for direct digital frequency synthesis," in *Proc. IEE Circuit Devices Syst.*, 2004, vol. 151, no. 6, pp. 519–528.

[8] F. Curticapean, K. I. Palomaki, J. Niittylahti, "Quadrature direct digital frequency synthesizer using angle rotation algorithm," in *Proc. IEEE Int. Symp. Circuits Syst. ISCAS'03*, May 2003, vol. 2, pp.81–84.

[9] J. M. P. Langlois, D. Al-Khalili, "Novel approach to the design of direct digital frequency synthesizers based on linear interpolation", *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 50, no. 9, pp. 567–578, Sep. 2003.

[10] D. De Caro, A. G. M. Strollo, "High-performance direct digital frequency synthesizers using piecewise-polynomial approximation", *IEEE Trans. Circuit Syst. I, Reg. Papers*, vol. 52, pp. 324–336, Feb. 2005.

[11] J.-M. Huang, C.-C. Lee, C.-C. Wang, "A ROM-less direct digital frequency synthesizer based on 16-segment parabolic polynomial interpolation", in *Proc. 15th IEEE Int. Conf. on Electronics, Circuits and Systems,* pp. 1018–1021, Sept. 2008.