FPGA Implementations of Single-Multiplier Digital Sine-Cosine Wave Generators

Jassim M. Abdul-Jabbar and Noor N. Qaqos
Computer Eng. Dept. - University of Mosul, Iraq.

Abstract

This paper presents four different realizations of single-multiplier sine-cosine generators based on second-order digital filter structure. FPGA implementations of these four realizations are carried out on FPGA Spartan-3E Kit. Implementation results are compared from the view points of utilization resources and maximum frequency of operation. Another comparison is made between one of implementations of the derived structures and other two recent CORDIC-based implementations. The comparison results indicate that smaller chip area can be achieved in the case of the proposed structure of the sine-cosine generator. In addition, such structure can operate with higher circuit frequency as compared with the two others.

Keywords: Digital Sine-Cosine Generators, Second Order Structure, CORDIC, FPGA Implementation

البناء بإستخدام FPGA لمولدات الجيب والجيب تمام الرقمية أحادية المضرب

د. جاسم محمد عبد الجبار و نور نجيب قاقوس
قسم هندسة الحاسوب- كلية الهندسة- جامعة الموصل

المستخلص

يقدم هذا البحث أربعة نماذج مختلفة لمولدات الجيب والجيب تمام الرقمية أحادية المضرب بالإعتماد على تراكيب من المرتبة الثانية. لقد تم بناء هذه التراكيب الأربعة بإستخدام FPGA Spartan-3E Kit. كما تم مقارنة نتائج البناء من حيث إستغلال المصادر وأعلى تردد للعمل. وقد قدمت مقارنة إخرى لنتائج البناء لأحد التراكيب المقترحة مع تلك النتائج الخاصة ببنائين قدما مؤخراً إعتمدا البناء بـ CORDIC. أظهرت تلك المقارنة إمكانية إستخدام مساحة شريحة FPGA أصغر للتركيب المقترح لمولدات الجيب والجيب تمام, هذا بالإضافة تفوقه في تردد دارة العمل بالمقارنة مع التركيبين الآخرين.

Received: 28 – 10 - 2010 Accepted: 30 – 3 - 2011

1. Introduction

Analog circuits can be used to build sine-cosine wave generators, but the control of the signal parameters is usually difficult. Another easily-controlled, flexible and cost-effective method is to digitally implement such generators. Digital sine-cosine generators have many important practical applications as oscillators in several fields such as orthogonal communication systems, control systems, signal processing systems (*e. g.*, the computation of the Discrete Fourier Transform (DFT)) and instrumentations.

Critically-stable second order structures can generate digital sine-cosine waves according to the angle value that entered to the realization and on the initial values of its variables. The conventional implementation methods of digital sine wave oscillators were first reported by J. Teirney, *et al.* [1] and improved by Schannerberger and Award [2]. Implementation of recursive digital sine wave oscillator using the TMS32010 digital signal processer was introduced by I. Abu-El-Haija, *et al.* [3]. J. Whittington, *et al.* [4] presented a digital sine-cosine generator implemented by FPGA chip as a frequency generator for TIGER radar application.

Recently, FPGA design and implementation of flexible and scalable digital sine-cosine wave generator is reported by E. Garcia, *et al.* [5], based on pipelined CORDIC algorithm. Implementation of CORDIC algorithm for digital sine-cosine using FPGA Spartan-3E Kit is achieved by V. Kumer [6]. Design, test and FPGA implementation for a reconfigurable calculation unit using some methods like a CORDIC algorithm, Look-Up-Table (LUT) and coordinate transformation to compute sine and cosine are also introduced in [7]. In this paper, Implementations of four realizations of single-multiplier digital sine-cosine generators based on second order structures are considered using FPGA Spartan-3E Kit and compared to achieve the best proposed realization that takes less FPGA locations and can operate with a higher intermediate circuit frequency. In addition, the implementation of such best proposed realization is then compared with other two recent CORDIC-based implementations. The resulting reduced implementation facilitates the use of available technology to generate digital sine-cosine waves.

This paper is organized as follows: Besides this introductory section, Section 2 contains different designs and realizations for single-multiplier digital sine-cosine generator. Section 3 describes the FPGA implementations of such generators. Results and comparisons are given in Section 4 to show the superiority of such single-multiplier digital sine-cosine generator realizations in FPGA implementations with the use of 32 bits word-length over two recent works. Finally, Section 5 concludes this paper.

2. Design and Realization of Single-Multiplier Digital Sine-Cosine Generators

The design and realization of digital sine-cosine oscillators are considered in this section. In particular, the design of digital sine-cosine generators that produce the two sinusoidal sequences that are exactly 90 degrees out of phase with each other is first discussed as follows:

Let $s_1[n]$ and $s_2[n]$ denote the two outputs of a digital sine-cosine generator and given by [8]:

$$s_1[n] = \alpha \, sin(n\theta) \quad \ldots (1)$$

and

$$s_2[n] = \beta\, cos(n\theta) \qquad \ldots (2)$$

From (1) and (2), it can be written that

$$s_1[n+1] = \alpha\, sin((n+1)\theta)$$

or $$s_1[n+1] = \alpha\, sin(n\theta)\, cos\theta + \alpha\, cos(n\theta)\, sin\theta \qquad \ldots (3)$$

and

$$s_2[n+1] = \beta\, cos((n+1)\theta)$$

or $$s_2[n+1] = \beta\, cos(n\theta)\, cos(\theta) - \beta\, sin(n\theta)\, sin(\theta) \qquad \ldots (4)$$

Making use of (1) and (2), (3) and (4) can be written in matrix form as

$$\begin{bmatrix} s_1[n+1] \\ s_2[n+1] \end{bmatrix} = \begin{bmatrix} cos\theta & \frac{\alpha}{\beta} sin\theta \\ -\frac{\beta}{\alpha} sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} s_1[n] \\ s_2[n] \end{bmatrix} \qquad \ldots (5)$$

So, to obtain $s_1[n]$ and $s_2[n]$ from $s_1[n+1]$ and $s_2[n+1]$, respectively, two delay units are used in the corresponding structure. That's why digital sine-cosine signals can be generated using a second-order recursive digital filter with poles on the unit circle in the complex z-plane [9]. Thus it is required to compare Eqn. (5) with the equivalent expression of a general second-order structure with no delay free loops in order to arrive at a realization of the single-multiplier sine-cosine generator. The general second-order structure is characterized by the following equation [8]:

$$\begin{bmatrix} s_1[n+1] \\ s_2[n+1] \end{bmatrix} = \begin{bmatrix} 0 & A \\ 0 & 0 \end{bmatrix} \begin{bmatrix} s_1[n+1] \\ s_2[n+1] \end{bmatrix} + \begin{bmatrix} C & D \\ E & F \end{bmatrix} \begin{bmatrix} s_1[n] \\ s_2[n] \end{bmatrix} \qquad \ldots (6)$$

Expression (6) can be implemented using five multipliers as shown in Fig. 1.

By making some abbreviations in (6), it can be rewritten as

$$\begin{bmatrix} s_1[n+1] \\ s_2[n+1] \end{bmatrix} = \begin{bmatrix} AE + C & AF + D \\ E & F \end{bmatrix} \begin{bmatrix} s_1[n] \\ s_2[n] \end{bmatrix} \qquad \ldots (7)$$

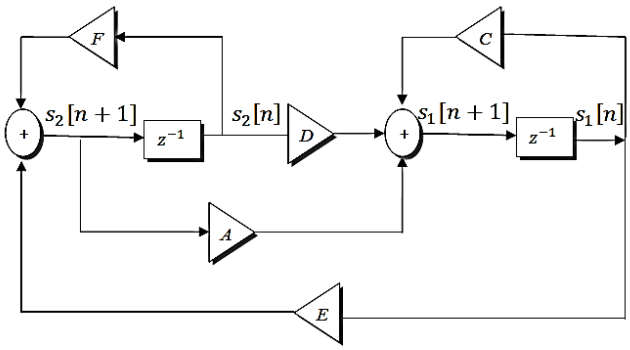

Fig. 1 General second-order structure with no delay free loops.

Comparing Eqns. (6) and (7), the values of *A*, *C*, *D*, *E* and *F* coefficients can be found. Expressing the multipliers *A* and *D* as functions of multiplier *C*, then Eqn. (6) can be written as

$$\begin{bmatrix} s_1[n+1] \\ s_2[n+1] \end{bmatrix} = \begin{bmatrix} 0 & \frac{\alpha(C-cos\theta)}{\beta sin\theta} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} s_1[n+1] \\ s_2[n+1] \end{bmatrix} + \begin{bmatrix} C & \frac{\alpha(1-C.cos\theta)}{\beta sin\theta} \\ -\frac{\beta}{\alpha} sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} s_1[n] \\ s_2[n] \end{bmatrix} \qquad \ldots (8)$$

It can be seen that, equation (8) requires five multipliers for implementation. The number of multipliers can be reduced to only one by substituting specific values for the constant *C*. Different single-multiplier realizations are shown in Figs. 2, 3 and 4 by setting

$C = cos\theta$ & $\left(\frac{\beta}{\alpha} sin\theta = 1 - cos\theta\right)$,
$C = cos\theta$ & $\left(-\frac{\beta sin\theta}{\alpha} = 1 + cos\theta\right)$ and
$C = 0$ & $(\alpha = \beta sin\theta)$, respectively [10].

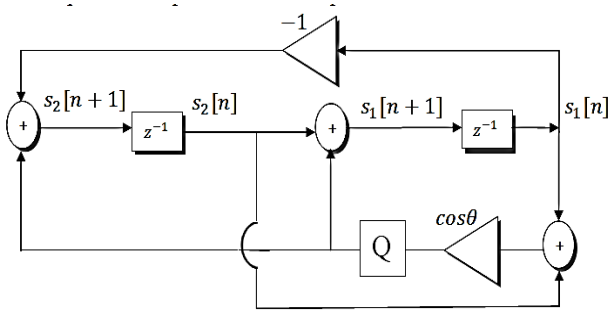

Fig. 2 A single-multiplier sine-cosine generator by setting ($C = cos\theta$) & $\left(\frac{\beta}{\alpha} sin\theta = 1 - cos\theta\right)$.

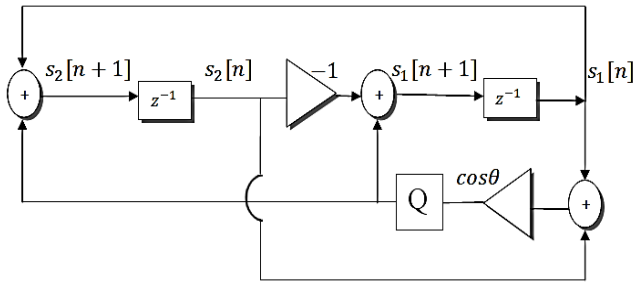

Fig. 3 A single-multiplier sine-cosine generator by setting ($C = cos\theta$) & $(\frac{\beta}{\alpha} = -\frac{1+cos\theta}{sin\theta})$.

In all these figures, Q represents the quantization process after multiplication.

Another method to generate digital sine-cosine waves is by using a second-order critically-stable recursive digital filter with poles on the unit circle in the complex z-plane. The difference equation of the digital system representing the direct form digital sine wave oscillator contains one multiplication operation and one subtraction and can be derived as follows [9], [11]:

Let the generator output be $y(n) = sin\theta n$

Then $y(n-1) = sin\theta(n-1)$ and $y(n-2) = sin\theta(n-2)$

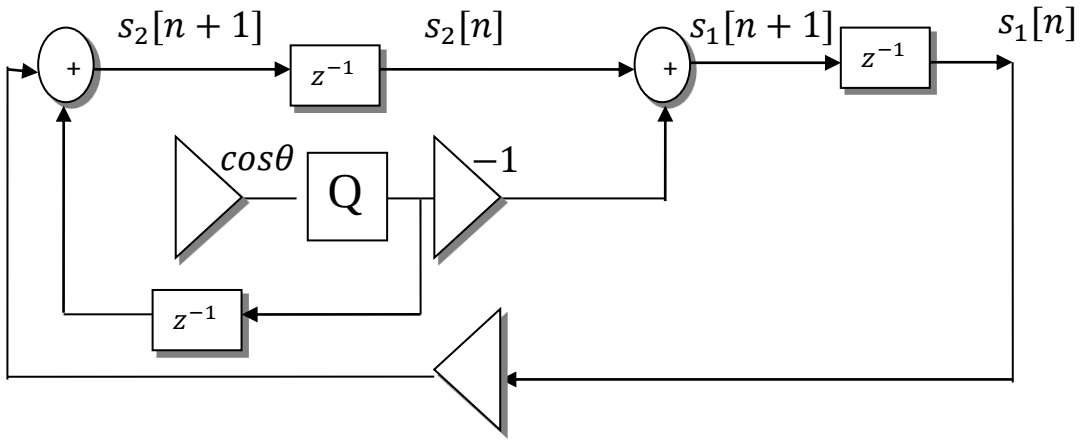

Fig. 4 A single-multiplier sine-cosine generator by setting ($C = 0$) & ($\alpha = \beta sin\theta$).

From the homogenous equations, the difference equation of second order can be written as follows

$$y''(t) + Ay'(t) + y(t) = 0 \quad \ldots (9)$$

Then

$$y(n-2) - A.y(n-1) + y(n) = 0$$

$$y(n) = A.y(n-1) - y(n-2) \quad \ldots(10)$$

To obtain a digital sine-cosine generator with different frequencies, the value of A must be selected to be equal to $A = 2cos\theta$

$$y(n) = 2cos\theta y(n-1) - y(n2) \qquad for \quad n0 \quad \ldots(11)$$

The corresponding block diagram representation of (11) is shown in Fig. 5.

3. FPGA implementations of Single-Multiplier Digital Sine-Cosine Generators

The implementation of any architecture design in programming to a hardware device (such as FPGA) needs the knowledge of the components of the architecture design to make the mapping process of these components to the corresponding hardware components by using a hardware description language such as VHDL. Also an input-output management and different resource allocations must be taken in regard for any hardware implementation. The hardware implementation of the single-multiplier digital sine-cosine generator in Figs. 2, 3, 4 and 5 are shown in the diagrams of Figs. 6, 7, 8 and 9, respectively.

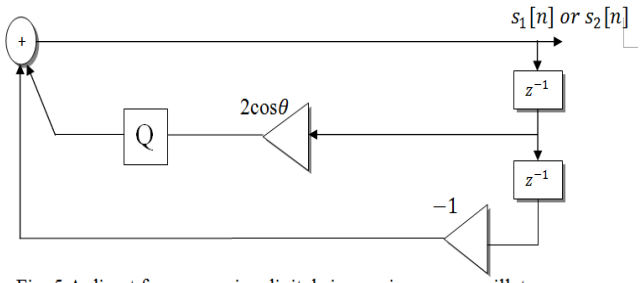

Fig. 5 A direct form recursive digital sine-cosine wave oscillator.

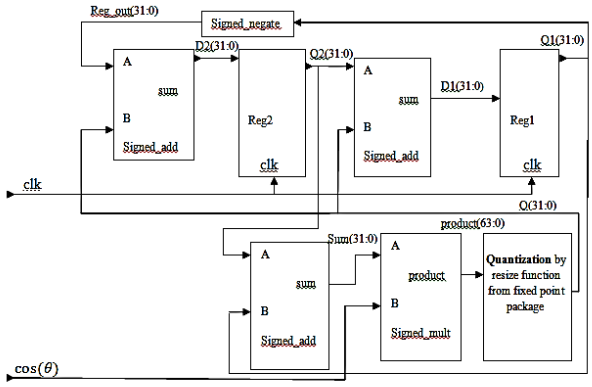

Fig. 6 Hardware implementation of the structure in Fig. 2.

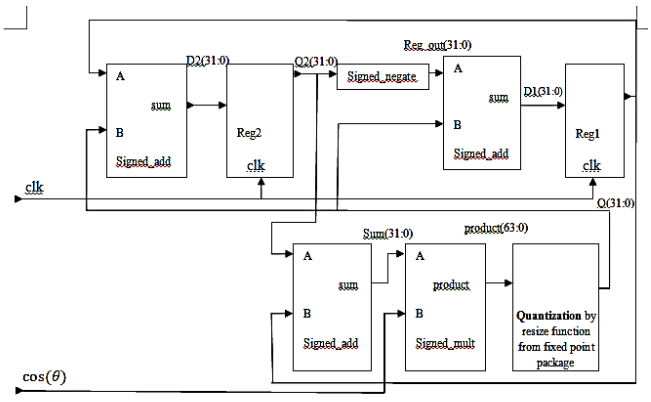

Fig. 7 Hardware implementation of the structure in Fig. 3.

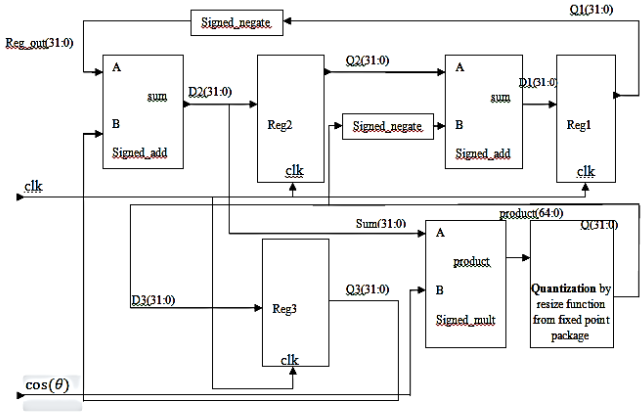

Fig. 8 Hardware implementation of the structure in Fig. 4.

There is more than one approach to describe any architecture design in programming. Each approach for the same architecture will have its own implementation that leads to different resource allocations in the synthesis level. The VHDL language requires very careful steps in order to have a satisfactory efficient implementation of the design to achieve an optimal FPGA design. These steps are as follows:

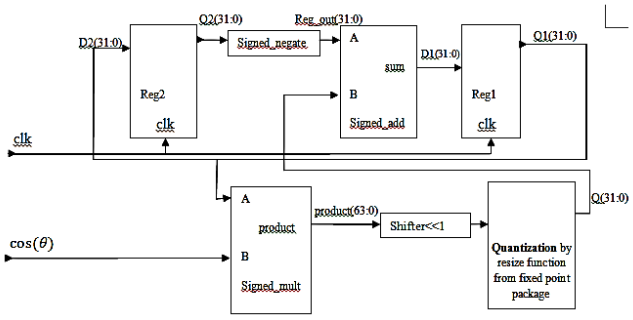

Fig. 9 Hardware implementation of the structure in Fig. 5.

1) Handling delay elements

Single-multiplier digital sine-cosine generators are of second order structures. This means that there are two delay elements in their hardware implementations. These delay elements are used to synchronize the operations in the DSP structures and the timing should be handled carefully so that each signal could carry the right value at exactly the right time. In FPGA, the delay can be accomplished by simply performing an equating operation for the specified signal inside the edge condition of the system clock. This action forces the synthesizer to exploit a D-flip flop that causes the value of the signal to be stored for one clock.

2) Adders and multipliers

A single-multiplier digital sine-cosine generator contains usually three adders and one multiplier. The implementation of multiplier is done using an embedded multiplier 18×18 bits in FPGA [12]. Since the dedicated multiplication process of the generators is of the size 32×32 bits, then four multipliers will be required to implement the multiplication process inside this FPGA.

3) Input-output management

It should be noted that the input to all realizations in Figs. 2 to 5 is *cos(θ)*. The values of *cos(θ)* are stored in an array inside the FPGA as the inputs $cos(\theta_0)$-$cos(\theta_{15})$ to the 16 × 1 Mux. By using the existed four switches (Sw0-Sw3) in SPARTAN-3E kit, the selection of desired angle can be treated as an input angle to the designed system. The output samples of sine and cosine signals are displayed using Liquid Crystal Display (LCD) screen in SPARTAN-3E kit. The LCD screen contains two lines; the first line displays the sine samples (SSam) while the second displays the cosine samples (CSam). The hardware implementation of input-output unit is shown in Fig. 10. Number of bits to represent the inputs *cos(θ)* is 32 bits, number of bits to represent the integer part is 2 bits, while the rest 30 bits are used for fractional part, because the range of values for *cos(θ)* is between -1 and 1.

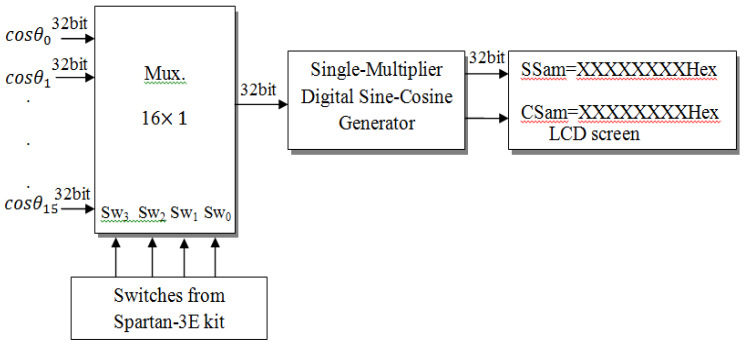

Fig. 10 A hardware implementation for input/output unit for the realizations in Figs. 2, 3, 4 and 5.

4) LCD screen to display results

Spartan-3E FPGA Starter Kit board prominently features a 2-line by 16-character LCD. The FPGA controls the LCD via the 4-bit data interface shown in Fig. 11. Although the LCD supports an 8-bit data interface, the Starter Kit board uses a 4-bits data interface to remain compatible with other Xilinx development boards and to minimize total pin count [13]. The display on LCD screen needs to make some initialization through the turn on between the FPGA chip and the control unit of LCD screen (LCD controller ST7006U) from Xilinx Company production. The logical circuit of LCD screen is implemented using FPGA. The initialization program of LCD screen is included with VHDL programs using LCD screen as a display unit. Table 1 shows the utilization resources exploited and maximum operating frequency when LCD is implemented on FPGA [7].

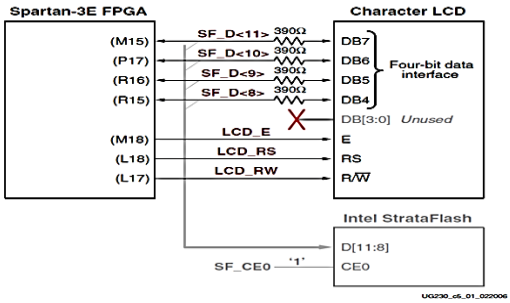

Fig. 11 Lines of data and control between FPGA chip and LCD screen in Spartan-3E kit.

Table 1 Utilization resources and maximum operating frequency for LCD screen.

Resources or Frequency	*Exploited*	*Total*	*Utilization ratio*
Number of Slices	334	4656	7%
Number of Slice Flip Flops	91	9312	0%
Number of 4 input LUTs	639	9312	6%
Number of bounded IOs	137	232	59%
Number of MULT18X18SIOs:	0	20	0%
Number of GCLKs	1	24	4%
Maximum Frequency	83.833MHz		

4. Results and Comparisons

Figures 12, 14, 16 and 18 show the timing diagram of the implementations in Figs. 2, 3, 4 and 5, respectively using ISE9.2i simulator when the input is the $cos30°$ (equal to 0.866025404073298) which represents the value of (376CF5D1) in hexadecimal. Two bits are used for integer part and the rest 30 bits are for fractional part. Figs. 13, 15, 17 and 19 show, respectively the output sine and cosine sequences of the structures of Figs. 2, 3, 4 and 5 when the input is $cos30°$. The utilization resources and the maximum frequency used for implementing the structure in Figs. 2, 3, 4, and 5 are shown in Table 2

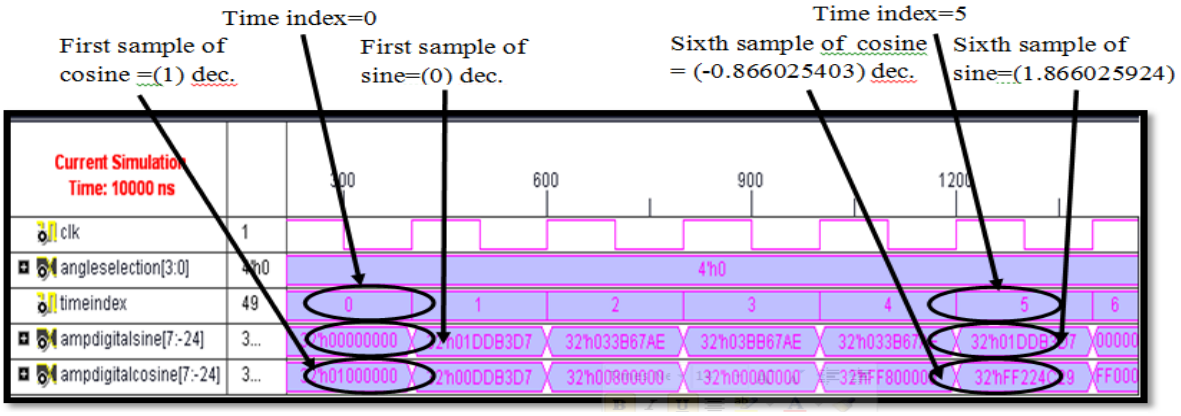

Fig. 12 The timing diagram of the structure in Fig. 2 implementation for $cos30° =$ 376CF5D1 Hex

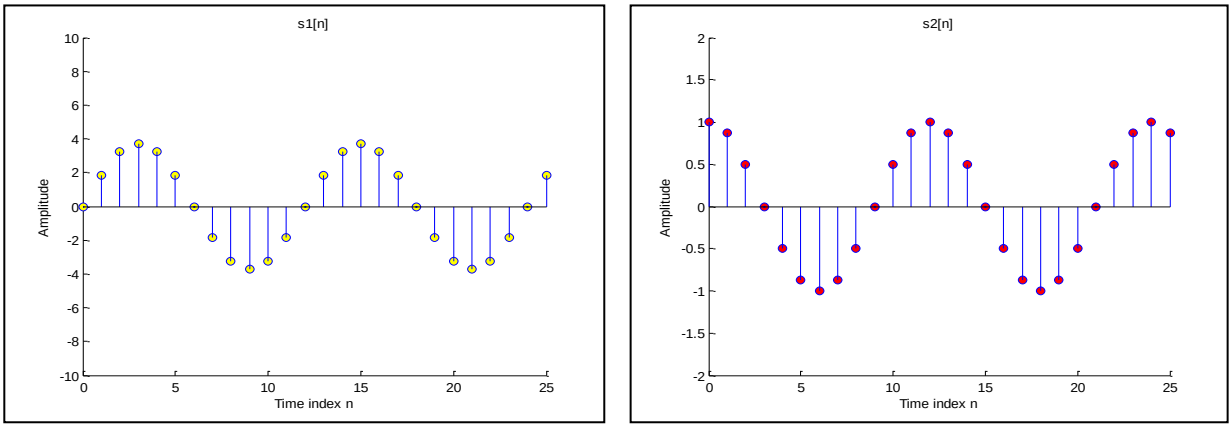

Fig. 13 Samples of sine $s_1[n]$ and cosine $s_2[n]$outputs from Fig. 2 for θ=30°

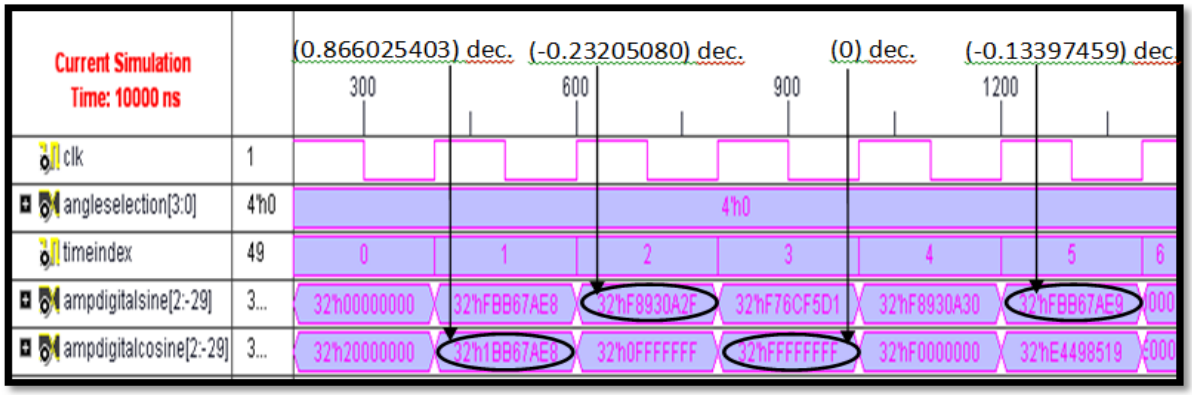

Fig. 14 The timing diagram of the structure in Fig. 3 implementation for $cos30° = 376CF5D1$ Hex.

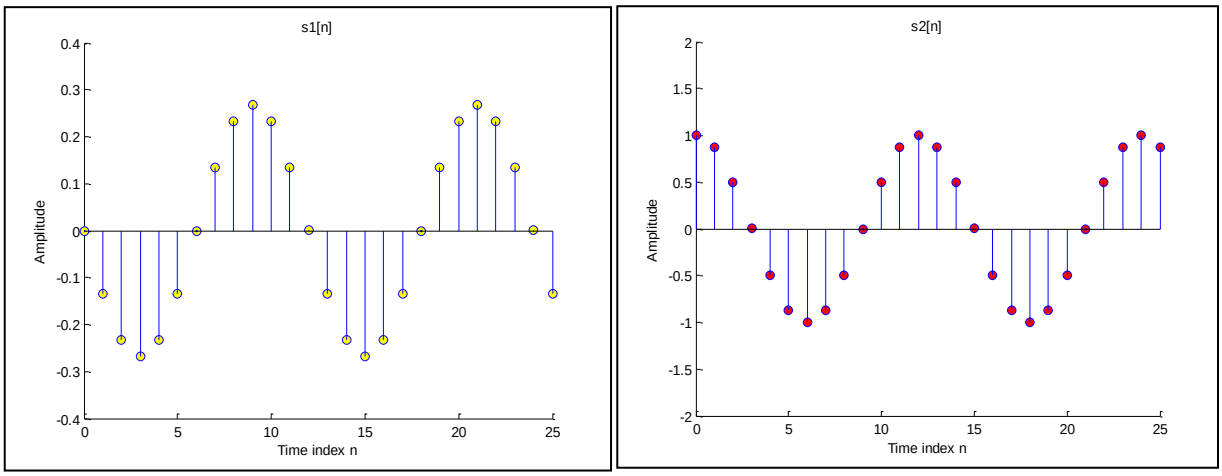

Fig. 15 Samples of sine s_1 [n] and cosine s_2 [n]outputs from Fig. 3 for θ=30°

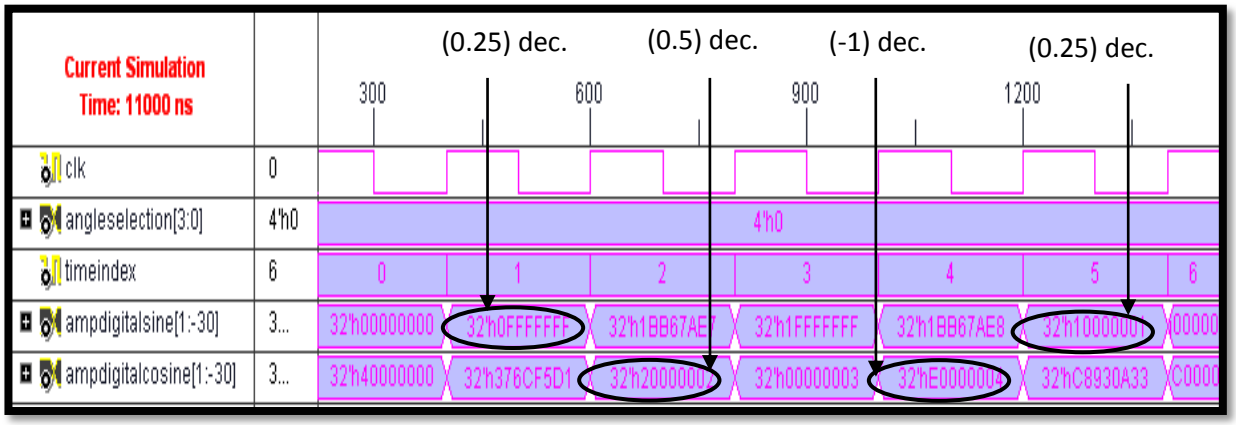

Fig. 16 The timing diagram of the structure in Fig. 4 implementation for $cos30° = 376CF5D1$ Hex.

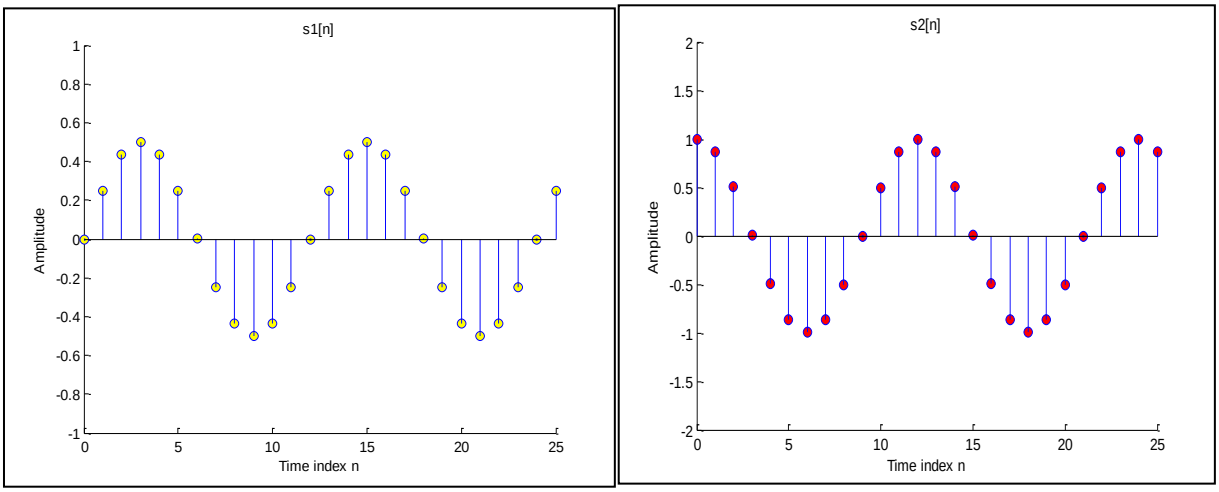

Fig. 17 Samples of sine $s_1[n]$ and cosine $s_2[n]$outputs from Fig. 4 for θ=30°

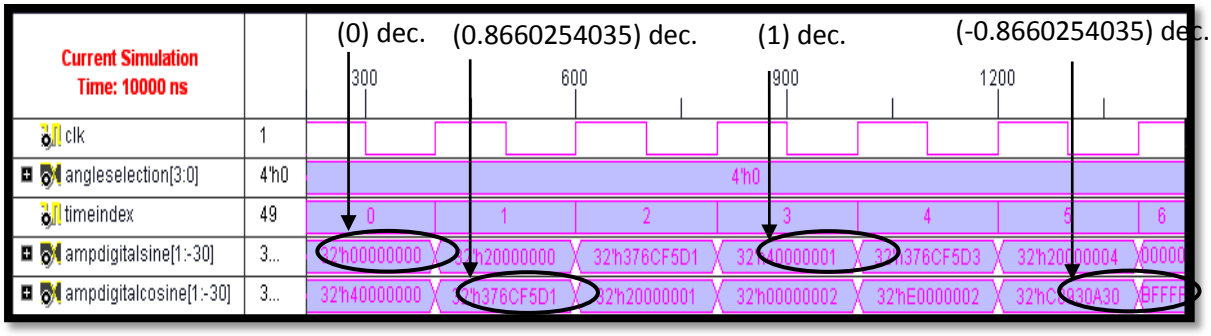

Fig. 18 The timing diagram of the structure in Fig. 5 implementation for $cos30° = 376CF5D1$ Hex.

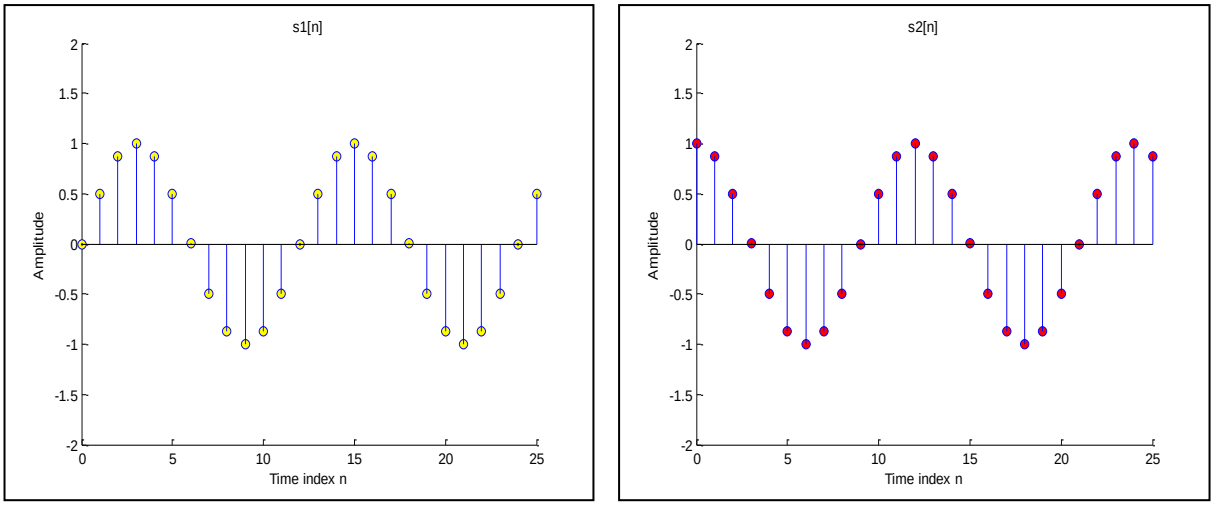

Fig. 19 Samples of sine $s_1[n]$ and cosine $s_2[n]$outputs from Fig. 5 for θ=30°

Table 2 Utilization resources and maximum frequency of the structure in Figs. 2, 3, 4 and 5.

Resources or Frequency		*Total*	*Exploited Allocations*			
			for Fig. 2	*for Fig. 3*	*for Fig. 4*	*for Fig. 5*
Slices	*Number*	4656	690	698	763	1020
	Percentage	100%	14%	14%	16%	21%
Slice Flip Flops	*Number*	9312	250	250	250	385
	Percentage	100%	2%	2%	2%	4%
4 input LUTs	*Number*	9312	1296	1326	1452	1952
	Percentage	100%	13%	14%	25%	20%
Bounded IOs	*Number*	232	109	109	109	109
	Percentage	100%	46%	46%	46%	46%
MULT18X18SIOs	*Number*	20	4	4	8	8
	Percentage	100%	20%	20%	40%	40%
GCLKs	*Number*	24	2	2	2	2
	Percentage	100%	8%	8%	8%	8%
Maximum Frequency		50MHz	32.153MHz	32.523MHz	19.926MHz	32.427MHz

From Figs. 12, 14, 16 and 18, it can be seen that:

1. One sample is generated at each clock period.
2. The angleselection [3:0] is used as an input from the switches in Spartan-3E board to select the desired angle to generate the digital sine-cosine wave.
3. Ampdigitalsine, Ampdigitalcosine are the amplitudes for digital sine and cosine samples, respectively for the given index time.
4. The vector of bits for Ampdigitalsine, Ampdigitalcosine is [7:-24] for Fig. 12 which means that there are 8 bits for representing the integer part and 24 bits for representing the fractional part in fixed point notation. While in Fig. 14, it can be seen that the vector of bits for Ampdigitalsine, Ampdigitalcosine is [2:-29]. This means that there are 3 bits for representing the integer part and 29 bits for representing the fractional part in fixed point notation. And in Figs. 16 and 18, it can be seen that the vector of bits for Ampdigitalsine, Ampdigitalcosine is [1:-30]. This means that there are 2 bits for representing the integer part and 30 bits for representing the fractional part in fixed point notation.

The digital sine-cosine generator structures are found through the implementation results to be very appropriate structures from area allocation point of view. The digital sine-cosine generators are implemented using FPGA Spartan-3E device setting the optimization goal of

the synthesizer to be the area, and a comparison is made with other implementations of two recent works given in Refs. [5] and [7]. The comparison indicates that smaller area can be achieved when using the designed sine-cosine generator (even with Fig. 5 which has the maximum number of slices among other proposed realizations). In addition, Fig. 5 can operate with an intermediate circuit frequency as compared with the corresponding operating frequencies in the recent works. Table 3 shows the comparison results for the digital sine-cosine generator systems.

Table 3 Comparisons with recent implementations of the digital sine-cosine generator systems.

Implementation Technique	*Architecture*	*Slices*	*No. of Multipliers*	*Frequency (MHz)*	*Exploited Device*
The proposed	Fig. 5, Parallel Architecture	(1020) - (334 for LCD) = 686	8	32.427	*Xilinx spartan3E* XC3S500E
Ref. [5]	CORDIC Pipeline	1075	0	124	*Xilinx spartan3* XC3S200E
Ref. [7]	CORDIC Parallel	2053	0	4.752	*Xilinx spartan3E* XC3S500E
	CORDIC Pipeline	2187	0	135.719	*Xilinx spartan3E* XC3S500E

5. Conclusions

FPGA implementations of four realizations of single-multiplier digital sine-cosine wave generators have been presented. 32-bit word-length for the multiplier coefficient and all outputs of arithmetic operations have been utilized. Such implementations have been compared with other two recent CORDIC-based works. The comparisons have indicated that smaller areas and good operating frequencies can be achieved by the FPGA implementations of the proposed realizations.

The comparison results indicate that smaller FPGA location can be achieved even in the case of one of the worst proposed structures (Fig. 5) of the sine-cosine generator. In addition, such structure can take less FPGA locations and can operate with an intermediate circuit frequency as compared with two of the mentioned recent CORDIC-based implementations. It is believed that the proposed reduced implementation facilitates the use of available technology to generate digital sine-cosine waves.

In the proposed sine-cosine wave generators, it should be noted that due the recursive mode of wave samples generation, the required time for generating a wave sample is proportional to the angle of that sample. Such required time also depends on the initial value of the angle θ. In spite of such characteristics, the proposed structure can operate in an on-line application of digital signal modulation systems because the modulation process can be held sequentially with the generation of samples. In addition, the samples of one period of such sine-cosine waves can be transformed from serial to parallel forms for further applications in FFT or DFT structures.

Furthermore, the proposed generators have required only one multiplier in their realizations. This means that a sigle quantization process is to be used after multiplication. Also, besides their simple FPGA implementations, it is believed that the proposed generators possess the advantage of uniform frequency spacing. These properties make them suitable for modern communication and electronic applications, such as FFT and IFFT stages in frequency division multiple access (FDMA) systems for small power, small chip resources,

and efficient physical implementations of potential multimedia wireless communication systems for fourth generation (4G) technologies.

References

[1] J. Teirney, C. Radar, and B. Gold ‘‘ A Digital Frequency Synthesizer,” IEEE Trans. Audio Electroacoust., V3. AU-19, pp. 48-57, Mar. 1971.

[2] M. Schanerberger and S. Award, “The Implementation of A Digital Sine Wave Oscillator Using the TMS320C25 : Distortion Reduction and Applications”, IEEE transactions on instrumentation and measurement, Vol. 39, No. 6, pp. 870-873, December 1990.

[3] I. Abu-El-Haija, M. Al-Ibrahim and M. Al-Jarrah, “Recursive Digital Sine Wave Oscillators Using the TMS32010 DSP”, IEEE, IMTC-94, May 10-12, Hamamatsu, 1994.

[4] J. Whittington, J. Delvin, and T. Salim, “Evaluation of Digital Generation and Phasing Techniques for Transmitter Signals of the TIGER N.Z. Radar”, WARS'02 Workshop on Applications of Radio Science, Leura, NSW, 20-22 February 2002.

[5] E. Garcia, R. Cumplido and M. Arias, “ Pipeline CORDIC Design on FPGA for A Digital Sine-Cosine Waves Generator”, 3rd International Conference on Electrical and Electronics Engineering, Department of Computer Science, Puebla, Mexico, pp. 104-107, September 6-8, 2006.

[6] V. kumer, “FPGA Implementation of DFT Using CORDIC Algorithm”, M. Sc. Thesis, Electronics and Communication Engineering Department, Thapar University, Patiala-147004 (INDIA), June 2008.

[7] E. Ali, “Design and Testing A Reconfigurable Calculation Unit Using FPGA”, M. Sc. Thesis, Computer Engineering Department, Mosul University, 2008.

[8] S. Mitra, “Digital Signal processing”, McGraw-Hill, New York, *NY*, 10020, *ISBN* 0-471-48422-9, 2001.

[9] M. Al-Ibrahim, “A Multifrequency Range Digital Sinusoidal Oscillator with High Resolution and Uniform Frequency Spacing”, IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing, Vol. 48, No. 9, September 2001.

[10] J. M. Abdul-Jabbar and N. N. Qaqos, “Performance Analysis of Single- Multiplier Digital Sine-Cosine Generators”, Submitted for publication in Alrafidain Engineering Journal, College of Engineering, University of Mosul, Mosul, Iraq.

[11] A. Abu-El-Haija and M. Al-Ibrahim, “Digital Oscillator Having Low Sensitivity and Roundoff Errors”, IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-22, No. 1, January 1986.

[12] “Using Embedded Multipliers in Spartan-3 FPGAs”, © 2003 Xilinx, Inc., XAPP467 (v1.1) May 13, 2003. http\\www.xilinx.com.

[13] “Spartan-3E FPGA Starter Kit Board User Guide”, © 2006-2008 Xilinx, Inc., UG230 (v1.1) June 20, 2008. http\\www.xilinx.com.

The work was carried out at the college of Engineering. University of Mosul