CHAPTER 4
DESIGN OF DIRECT DIGITAL FREQUENCY SYNTHESIZER

4.1 PREAMBLE

The ever increasing low cost but high performance telecom and consumer electronics systems require Data Converters such as ADC & DAC with customized capabilities in respect of resolution and speed. Static parameters such as DNL error, INL error, Offset error and Gain error characterizes the low speed behaviour of Data Converters. On the other hand, dynamic parameters such as Signal to Noise Ratio (SNR), Spurious Free Dynamic Range (SFDR), Total Hormonic Distortion (THD), Signal to Noise and Distortion Ratio (SINAD), and Intermodulation Distortion (IMD) characterize the high speed behaviour of ADC. The dynamic characterization of the Data Converter is critical in telecommunication and embedded control applications. It is well known that Automatic test equipment and Built-in self test methods are the popular methods used in industry for dynamic characterisation of Data Converters. In traditional automated test equipment (ATE) the dynamic characterization of data converters require spectrally pure and coherent sinusoidal signal (IEEE 1241-2000), (Jingbo et al 2010). Hence the sinusoidal signal source used has to be tuned for coherency to avoid spectral leakage during the spectral estimation of the device under test. Because of such fine tuning requirement, the process becomes very cumbersome and time consuming.
Joseph et al (1996) proposed that the spectral leakage during mixed signal circuit testing can be reduced by using windowing techniques. A predefined Hamming and Hanning windowing techniques are chosen to eliminate the spectral leakage in spectral estimation (Alberto et al 2006). However, irrespective of windowing, there will always be a compromise in accuracy, in respect of spectral leakage minimization, frequency and amplitude precision, side lobe amplitude reduction and main lobe broadening (Daniel et al 2007). In this Chapter a digitally controlled sinusoidal signal generator is proposed to facilitate cost effective and accurate dynamic characterization of Data Converters.

4.2 BUILT IN DIRECT DIGITAL FREQUENCY SYNTHESIZER

Avanindra Madisetti et al (1999) proposed a DDFS with an angle-rotation algorithm which employs pipelining approach for the generation of sinusoidal signal. The modular architecture provides arbitrary precision by way of cascading more angle-rotation stages in the data path.

Bellaouar et al (2000) proposed a low-power direct digital frequency synthesizer (DDFS) architecture. It uses a lookup table for sine and cosine functions with additional hardware. The computation of the generated sine and cosine functions is based on the linear interpolation between the sample points. A DDFS with 60 dBc spectral purity for a 9-bit output is reported.

A Built-in Direct Digital Frequency Synthesizer (DDFS) has been suggested in IEEE 1241-2000 for the generation of controllable and highly coherent sinusoidal signal with minimum hardware overhead for testing the ADC.
Kroupa et al (2001) proposed a technique that combines direct digital frequency synthesizer and PLL (phase locked loops) with frequency dividers in the feedback path. To arrive at even closer channel spacing, the author introduced the so called Fractional-N frequency synthesizers. A sigma delta modulator is used as a means of spurious signal reduction.

Chua-Chin Wang et al (2002) proposed a direct digital frequency synthesizer employing a trigonometric quadruple angle method. The method uses a 3 square multiplier and the spectral purity is shown to be 130 dBc for a 13-bit output resolution.

Florean et al (2003) proposed a method using a look up table based on CORDIC algorithm which uses an iterative angle computation with rotation angles between $-\pi/2$ and $\pi/2$ for the generation of sinusoidal signal. This method produces 16 bit sine and cosine waveforms with spectral purity of 114 dBc and having tuning latency of 2 cycles.

Caro (2004) has proposed a direct digital frequency synthesizer based on optimized polynomial expansion of sine and cosine functions. Polynomial computation is done by canonical-signed-digit (CSD) hyper folding technique. This method provides a spectral purity of 80 dBc with an output resolution of 12-bit.

Usman Hai et al (2005) proposed a compressed look-up table based DDFS architecture using sine wave symmetry from 0 to $\pi/2$ to address the power issue in wireless communication applications. He has reported that the method
possesses a spectral purity of 33 dBc with 16 words look up table having word size of 9-bit.

Kesoulis et al (2007) proposed a DDFS based on a look-up table, which performs a functional mapping from phase to amplitude of sinusoidal signal. He uses a sine phase difference algorithm with a modified amplitude reduction technique. He observed a spectral purity of 95.11 dBc using 14336-bit memory of look-up table with 18-bit word size.

Lai Lin-hui et al (2008) proposed a low complex direct digital frequency synthesizer (DDFS) based on a look-up table. The look-up table for sine and cosine functions is split into a coarse precision ROM and a fine precision ROM. The results show that the total size of ROM in this proposed DDFS architecture is 328 bit and the spectral purity is 63.58 dBc.

Xiaojin Li et al (2009) presented a direct digital frequency synthesizer (DDFS) using phase to sinusoidal amplitude conversion blocks based on the two segment fourth-order parabolic approximation. Squarer and constant multipliers have been proposed for the computation in the DDFS. The method shows that the resolution is up to 14-bit and the spectral purity of about 90 dBc.

Wan Shuqin et al (2009) presented a Direct digital frequency synthesizer (DDFS) based on modified CORDIC algorithm with reduced hardware and reduced iteration clocks. The first rotation is implemented by using a CORDIC realized in pipeline and carry-save arithmetic. The directions of the CORDIC rotations are computed in parallel by using a look-up table, for the first
rotation. The method shows a spectral purity of 64.22 dBc with 14-bit word length.

An equi-section division method utilizing the symmetry property and amplitude approximation of a sinusoidal waveform to design a direct digital frequency synthesizer (DDFS) is proposed. The sinusoidal phase of a one quarter period is divided into equi-sections by Shiann-Shiun Jeng (2010). The error between each line segment value and the sinusoidal amplitude value is stored in a look-up table to reconstruct the actual sinusoidal waveform. The upper/lower bound of the maximum error value stored in another table is derived to determine the minimum required memory and word length with respect to the bit number of the equi-sections. The method shows the spectral purity of 52.168-dBc with 14- bits word length.

Yao-Hua Chen et al (2010) proposed a 8\textsuperscript{th} order even polynomial approximation of a cosine function with square circuits for realizing DDFS with reduced chip size and minimal spurious free dynamic range (SFDR). The method has a spurious free dynamic range (SFDR) of 95 dBc for 13-bit resolution.

4.3 TRADATIONAL DDFS ARCHITECTURE

DDFS is a fine resolution sinusoidal waveform generator which can generate signals with stable performance, in terms of amplitude, phase and frequency. DDFS implementation mainly relies upon integer arithmetic, allowing implementation on any hardware platform. The traditional DDFS architecture is shown in Figure 4.1. It consists of a phase accumulator, a phase to amplitude converter and a digital-to-analog converter (DAC). The phase accumulator is a digital circuitry consisting of an adder with a feedback and an input as frequency
control word (FCW). The phase accumulator receives an input called FCW which is linearly incremented in accordance with clock. The output of phase accumulator is fed into the phase to amplitude converter which is a sine/cosine generator that converts digital phase values into its equivalent digital sinusoidal amplitude. Most of the methods use the ROM based look-up table for sine/cosine generation. The linearly increasing output of the phase accumulator provides the address to the ROM to access the sine/cosine amplitude values stored in the look-up table (Vankka and Halonen 2001). The digital amplitude values from the phase to amplitude converter are then fed into the DAC to obtain the required sinusoidal output. The spectral purity of the sinusoidal output of the DDFS is dependant on the word size of the ROM and number of words in the look-up table.

![Figure 4.1 Traditional DDFS Architecture](image)

Reducing hardware complexity by optimizing the number of ROMs used has been one of the major challenges in DDFS design (Vankka 1997). The impact of memory size on the power consumption and silicon area is very critical for implementing integrated circuits and system on a chip. Many efficient compression techniques that can reduce the size of look-up table have been proposed (Vankka 1997) (Vankka and Halonen 2001).

Techniques which can reduce the ROM size while retaining the spectral purity and frequency resolution include exploitation of trigonometric identities and approximation of the sine function using Taylor series, parabolic
approximation (Amir M. Sodagar et al 2003) and polynomial approximation wherein the samples of sine amplitude are computed from the digital phase contents.

Sunderland (1984) architecture splits the ROM into two smaller memories. The Nicholas architecture improves the Sunderland architecture to obtain higher ROM-compression ratio (32:1). Kesoulis et al (2007) uses sine phase difference algorithm with modified amplitude reduction technique to achieve higher compression levels, thereby reducing the memory size. The quadrant compression technique is able to compress the look-up table and reduce the ROM by 75%. Lai Lin-hui et al (2008) applied sine-cosine symmetry and reported 87.5% of ROM size reduction by storing sine and cosine values from 0 to π/4.

It has been observed from the existing methods that the traditional ROM based DDFS have several drawbacks including higher ROM size, large chip area, high computation complexity, iteration latency and high power consumption. This necessitates the development of DDFS based on polynomial approximation methods with reduced multiplication count. To achieve this, one often subdivides the expansion region into several subintervals which in turn carries out its own polynomial expansion, with less multiplication. However, the reduction in multiplications is done at the cost of extra table to store the polynomial coefficients, as well as the function values of the expansion points. This chapter proposes a novel polynomial expansion method to develop a DDFS with reduced hardware size, less computation complexity and high spectral purity.
4.4 PROPOSED DDFS ALGORITHM

The proposed DDFS design is based on the polynomial approximation method with a judicious choice of polynomial to minimize the hardware requirement. Though the polynomial algorithm used in the design is based on Taylor series polynomial, the design is given as follows.

4.4.1 Taylor Series Approximation

In the Taylor series expansion, the Sine and Cosine functions can be implemented at a point \( \gamma \) given by,

\[
\sin(\theta) = \sin(\gamma) + (\theta - \gamma) \cdot \cos(\gamma) + \ldots .
\]

\[
\cos(\theta) = \cos(\gamma) - (\theta - \gamma) \cdot \sin(\gamma) + \ldots .
\]

Earlier algorithms for synthesizing DDFS are based on Equations (4.1) and (4.2) which utilize the sample magnitude of \( \sin(\theta) \) stored in a ROM and its slope (Jen-Chuan Chih et al 2001).

Neglecting higher order terms in Equations (4.1) and (4.2),

\[
\sin(\theta) \sim \sin(\gamma)+ (\theta - \gamma) \cdot \cos (\gamma)
\]

\[
\cos(\theta) \sim \cos(\gamma)- (\theta - \gamma) \cdot \sin (\gamma)
\]

Where, \( \gamma \) is a constant. Another form of Taylor series expansion for sine and cosine function at a point \( \gamma \) is given by,

\[
\sin(\gamma) = \gamma - \left( \frac{\gamma^3}{3!} \right) + \left( \frac{\gamma^5}{5!} \right) - \left( \frac{\gamma^7}{7!} \right) + \ldots .
\]

\[
\cos(\gamma) = 1 - \left( \frac{\gamma^2}{2!} \right) + \left( \frac{\gamma^4}{4!} \right) - \left( \frac{\gamma^6}{6!} \right) + \ldots .
\]
For $N$-bits precision, the terms after \((1-(\gamma^2/2!))\) in \(\cos(\gamma)\) expansion and \(\gamma\) in \(\sin(\gamma)\) expansion can be ignored because the magnitude of the higher order terms are less than \(2^{-(N+1)}\).

The Equations (4.5) and (4.6) are further simplified by neglecting higher order terms as given by

\[
\sin(\gamma) \approx \gamma \\
\cos(\gamma) \approx 1 - (\gamma^2 / 2!) \quad (4.7)
\]

By applying the Equations (4.7) and (4.8) in Equation (4.3)

\[
\sin(\theta) \approx \gamma + (\theta - \gamma) \cdot (1 - (\gamma^2 / 2!)) \\
= \theta - ((\gamma^2 / 2) \cdot \theta) + \gamma^3 / 2
\]

Neglecting higher order terms further,

\[
\sin(\theta) \approx (1 - (\gamma^2 / 2)) \cdot \theta \\
(4.9)
\]

The \(\gamma\) value for sine wave, denoted as \(\gamma_s\), is obtained using Equation (4.9) as,

\[
\gamma_s = \sqrt{2 \cdot (1 - \sin(\theta) / \theta))} \\
(4.10)
\]

Where, \(\theta\) ranges from 0\(^{\circ}\) to 45\(^{\circ}\).

Similarly, by applying Equation (4.7) and (4.8) in Equation (4.4),

\[
\cos(\theta) \approx (1 + (\gamma \cdot \gamma / 2)) - (\theta \cdot \gamma) \\
(4.11)
\]

The \(\gamma\) value is cosine function, denoted as \(\gamma_c\), is obtained using Equation (4.11) as,

\[
\gamma_c^2 - 2(\theta \cdot \gamma_c) - 2(\cos(\theta) - 1) = 0 \\
(4.12)
\]

Where, \(\theta\) ranges from 0\(^{\circ}\) to 45\(^{\circ}\). The minimum of the two roots from Equation 4.12 is used for the computation.
Since the approximation is for the ranges from 0 to \(\pi/4\), the remaining quadrant’s values are obtained by the principle of sine/cosine symmetry (Shu-Chung Yi et al 2006). The sine/cosine waveform is symmetrical between the range \([\pi, 2\pi]\) and \([0, \pi]\). Also, the sine waveform from \(\pi/2\) to \(\pi/4\) is the same as the cosine from zero to \(\pi/4\), and the cosine waveform from \(\pi/2\) to \(\pi/4\) is the same as the sine from zero to \(\pi/4\). Thus, it is sufficient to compute the sine and cosine values from 0 to \(\pi/4\) using Equations 4.9 and Equations 4.11. The entire sine wave is generated from the translations given in Table 4.1. The first quadrant is regrouped as 1A & 1B and similarly other quadrants are regrouped as 2A & 2B, 3A & 3B and 4A & 4B. When the phase value is 0 to \(\pi/4\) (0º to 45º), the sine value of the phase is computed from the sine generator straight away. When the phase value is 45º to 90º, the sine value of the phase is computed by subtracting the phase value from 45º and then by passing to a cosine generator. Similarly for other quadrants the sine values are computed.

<table>
<thead>
<tr>
<th>counter value (bits)</th>
<th>Quadrant index</th>
<th>Phase value ((\theta)) Degrees</th>
<th>Sine computation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0</td>
<td>1A</td>
<td>0(\leq\theta\leq)45</td>
<td>(\sin(\theta))</td>
</tr>
<tr>
<td>0 0 1</td>
<td>1B</td>
<td>45(&lt;\theta\leq90)</td>
<td>(\cos(45-\theta))</td>
</tr>
<tr>
<td>0 1 0</td>
<td>2A</td>
<td>90(&lt;\theta\leq135)</td>
<td>(\cos(\theta))</td>
</tr>
<tr>
<td>0 1 1</td>
<td>2B</td>
<td>135(&lt;\theta\leq180)</td>
<td>(\sin(45-\theta))</td>
</tr>
<tr>
<td>1 0 0</td>
<td>3A</td>
<td>180(&lt;\theta\leq225)</td>
<td>-(\sin(\theta))</td>
</tr>
<tr>
<td>1 0 1</td>
<td>3B</td>
<td>225(&lt;\theta\leq270)</td>
<td>-(\cos(45-\theta))</td>
</tr>
<tr>
<td>1 1 0</td>
<td>4A</td>
<td>270(&lt;\theta\leq315)</td>
<td>-(\cos(\theta))</td>
</tr>
<tr>
<td>1 1 1</td>
<td>4B</td>
<td>315(&lt;\theta\leq360)</td>
<td>-(\sin(45-\theta))</td>
</tr>
</tbody>
</table>
4.4.2 Proposed Architecture

Figure 4.2 shows the block diagram of the proposed DDFS architecture to complement the above discussed algorithm. It consists of phase generator, quadrant selector, sine/cosine generator, digital multiplexers and a DAC in the output stage. The phase generator (PG) of the proposed DDFS receives a frequency control word (FCW) as an input from the system to which the proposed DDFS is to be integrated. The phase generator output is then fed to a quadrant selector which selects the phase corresponding to the quadrant requirement. The output of the quadrant selector is fed to a sine/cosine generator. The multiplexer in the output stage selects the output either from sine generator or cosine generator to complete a full cycle of sinusoid as given in Table 4.1. The full cycle is generated by exploiting the symmetry of sine wave with respect to zero crossings over one full period (Shu-Chung Yi et al 2006).

4.4.2.1 Phase generator

The phase generator consists of a phase accumulator, a maximum range register (MRR), and a digital comparator as shown in Figure 4.3. The phase accumulator is designed using an adder and a latch. The adder has two inputs and one of the inputs is a frequency control word (FCW) and input is fed back from the latch. The FCW is added with the same value on each clock signal and the value is accumulated in the latch. For digital realization, the phase is quantized as

\[ \theta = n2^N \]  

Where \( N \) is the size of the frequency control word and \( n \) is the value to be stored in maximum range selector register that takes on integer numbers in the range \( 0 \leq n \leq 2^N \). The value of \( N \) and \( n \) determine the resolution of the phase generator. The phase accumulator output \( \theta \) is quantized in the range 0 to \( \pi/4 \) periodically (i.e. 0° to 45°). The maximum range value (n) for the maximum
phase angle of the phase generator (45°) is $\pi/4 \ast 2^N$. The integer equivalent of the maximum range is stored in a memory called maximum range register. On each clock signal given to the latch, the phase accumulator generates consecutive phases. The comparator compares the output of the latch with MRR, when they are equal the comparator rises active high signal to reset the accumulated value in the latch. The phase accumulated value (PAV) from the output of the latch, the maximum range register output, and the comparator’s output (detect) are connected to quadrant selector.

Figure 4.2 Block diagram of the proposed DDFS
4.4.2.2 Quadrant selector

A quadrant selector consists of a subtractor, a multiplexer and a 3-bit counter as shown in Figure 4.4. It receives the output of the comparator (detect), the value from the MSR and output of the phase accumulator (PAV).

The phase generator is designed for generating phase value from 0 to $\pi/4$, and to generate a full cycle of a sinusoidal the phase generator needs to generate this phase range for 8 times. The comparator in the previous stage detects the maximum range and generates the detect signal 8 times. The detect
signal is used as a clock signal to the 3-bit counter so as to count the number of roll-overs and to identify the quadrant indices. The output of the 3-bit counter enables the multiplexer to choose the appropriate phase for sine computation. The MRR value and the phase values of each quadrant indices (1B, 2B, 3B & 4B) are fed to the subtractor in sequence. The phase values of other indices are routed directly to MUX. The phase value of each of the indices and the LSB output of the 3-bit Counter are multiplexed to produce the phase for Sine/Cosine function.

4.4.2.3 Sine/Cosine Generators

The Figure 4.5 shows the block diagram of the Sine/Cosine generators, which generate sine and cosine value for the input phase between 0 to $\pi/4$. The arithmetic circuit is synthesized as per the Taylor expansion (Equation 4.9) and (Equation 4.11) for sine and cosine generation respectively. This module receives the phase value and quadrant indices from the quadrant selector. The phase value is in the range from 0 to $\pi/4$, and this value is given to two section of arithmetic circuit for generating equivalent sine and cosine value. The two arithmetic circuit computes its equivalent sine and cosine value from the given phase value and $\gamma$ value. The sine and cosine values are computed using $\gamma_s$ and $\gamma_c$ respectively as given by in the Equation 4.9 and Equation 4.11. The values of $\gamma_s$ and $\gamma_c$ are computed in a separate hardware block given in Figure 4.6.

A full cycle of sinusoidal signal is generated from two arithmetic sections with the assistance of quadrant index bits namely LSB and MSB-1. These two bits decide the output from any one of these two arithmetic sections. The selection of arithmetic is given in the Table 4.2. It is clear from the table that an EX-OR of LSB and MSB-1 selects a cosine generator, otherwise a sine generator is selected.
Table 4.2 Arithmetic section selection for the quadrant index bits

<table>
<thead>
<tr>
<th>Quadrant Index Bit</th>
<th>Quadrant Index</th>
<th>Phase value (θ) In degree</th>
<th>Arithmetic Section selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSB-1   LSB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0       0</td>
<td>1A</td>
<td>0 ≤ θ ≤ 45</td>
<td>sin section ((1 - (γ^2 / 2)))</td>
</tr>
<tr>
<td>0       1</td>
<td>1B</td>
<td>45 ≤ θ ≤ 90</td>
<td>cosine section ((1 + (γ^2 / 2)))</td>
</tr>
<tr>
<td>1       0</td>
<td>2A</td>
<td>90 ≤ θ ≤ 135</td>
<td>cosine section</td>
</tr>
<tr>
<td>1       1</td>
<td>2B</td>
<td>135 ≤ θ ≤ 180</td>
<td>sin section</td>
</tr>
<tr>
<td>0       0</td>
<td>3A</td>
<td>180 ≤ θ ≤ 225</td>
<td>-(sin section)</td>
</tr>
<tr>
<td>0       1</td>
<td>3B</td>
<td>225 ≤ θ ≤ 270</td>
<td>-(cosine section)</td>
</tr>
<tr>
<td>1       0</td>
<td>4A</td>
<td>270 ≤ θ ≤ 315</td>
<td>-(cosine section)</td>
</tr>
<tr>
<td>1       1</td>
<td>4B</td>
<td>315 ≤ θ ≤ 360</td>
<td>-(sine section)</td>
</tr>
</tbody>
</table>

Figure 4.5 Sine/cosine generators for 0 to π/4
The quadrants 3A, 3B, 4A and 4B require negation in the output of the DDFS, and this is done by MSB of the quadrant index bits. Once a full cycle of sinusoidal wave is obtained the entire module starts for a new cycle.

4.4.2.4 γ Accumulator

Figure 4.6 shows the block diagram of the γ accumulator employed for generating γ values. It consists of an adder, latch, comparator, subtractor and a multiplexer. Also the module comprises of two registers, one register to store the \( \gamma_{\text{max}} \) value (γ value corresponding to \((\pi/ 4 \times 2^N)\)th clock cycle) and another register to store the γ step value (linear incremental step size of the γ) which is a product of frequency control word (FCW) and \( \gamma_{\text{max}}/(\pi/ 4 \times 2^N) \).

For each clock input, the adder with the latch generates an accumulated γ value with the step size of \((\gamma_{\text{max}}/(\pi/ 4 \times 2^N))\times \text{FCW}. \) This accumulated γ value is fed to a comparator and a subtractor.

The comparator compares accumulated γ value with the \( \gamma_{\text{max}} \) value for every instance of clock. When the accumulated γ value reaches to \( \gamma_{\text{max}} \), the latch is cleared to zero and γ accumulation is started from initial value.

The subtractor subtracts accumulated γ value with the \( \gamma_{\text{max}} \) value for every instance of clock. The subtracted value of the γ accumulator is fed to multiplexer and the accumulated γ value is also given to the multiplexer. One of the inputs of the multiplexer is selected by the LSB of the counter in quadrant selector and given to sine/cosine generator.
Thus, $\gamma$ accumulator is employed to generate $\gamma_s$ values between 0 to $\pi/4$. Similar hardware architecture is used to generate $\gamma_c$ values. The primary difference between the architectures employed to generate $\gamma_s$ and $\gamma_c$ values lies in the $\gamma_{\text{max}}$ register content. The appropriate values to be stored in the $\gamma_{\text{max}}$ registers are determined from (Equation 4.10) and (Equation 4.12).

![Block diagram of $\gamma$ accumulator](image)

**Figure 4.6 Block diagram of $\gamma$ accumulator**

### 4.5 RESULTS AND DISCUSSIONS

The Phase generator, quadrant selector, sine/cosine generator, and $\gamma$ accumulator are coded in Verilog and implemented in Altera FPGA DE1 kit. A digital to analog circuit is assembled in a vero board. The DAC board and FPGA kit are interfaced through GPIO. The value of $N$ is assumed to 11-bit, the value of $n$ is computed as in Equation 4.3 using the symmetry of sine and cosine over $0\leq0\leq\pi/4$ varies from 0 to 1608. The resolution of the proposed DDFS is $4.8843e^{-004}$. A Matlab program is written to calculate the values of $\gamma_s$ and $\gamma_c$ theoretically using Equations 4.10 – 4.13. $\gamma_s$ values range from 0.0003 to 0.4465 and $\gamma_c$ values range from 0.0005 to 0.6091. For the above values, the output waveform of the designed DDFS is captured by a computer using Agilent
34401A through RS232 interface. The captured data are fed to the Matlab program for extracting the graph. The spectrum of the captured data is shown in Figure 4.15.

Figure 4.7 and Figure 4.8 show the comparison plot between ideal and theoretical sine and cosine values for theoretical $\gamma$ values in the time domain. It can be seen that the ideal values of the sine wave and values obtained from the proposed method coincides very well. Figure 4.9 show the spectrum obtained using the theoretical $\gamma$ values. It can be seen that the SFDR of the proposed DDFS using the theoretical $\gamma$ values is nearly 193 dBC.

Figure 4.7 Comparison plot for sine values between ideal and proposed method using LUT for $\gamma_s$ sets
Figure 4.8 Comparison plot for cosine values between ideal and proposed method using LUT for $\gamma_c$ sets

Figure 4.9 Spectrum of the proposed DDFS using LUT for $\gamma$ values
Figure 4.10 and Figure 4.11 show the comparison plot between ideal and proposed sine and cosine values for linear increment of $\gamma$ values. As it is seen from the Figure 4.10, the ideal and the proposed method coincide very well proving the accuracy of the proposed algorithm. Whereas in the Figure 4.11 it can be observed that there exists a slight deviation between the ideal and the proposed method. This is due to the predominant contribution of $\gamma$ values.

![Comparison Plot for sine values between ideal and proposed method using $\gamma$ accumulator](image)

**Figure 4.10** Comparison plot for sine values between ideal and proposed method using $\gamma$ accumulator
Figure 4.11 Comparison plot for cosine values between ideal and proposed method using $\gamma$ accumulator

Figure 4.12 shows the spectrum for DDFS using linear increment of $\gamma$ values. As it is seen from the graph, the SFDR performance of the DDFS using linear increment of $\gamma_a$ and $\gamma_c$ over the range $0 - \pi/4$ is about 135.623 dBc. Thus, using the linear increment $\gamma$ values, SFDR is reduced to 135.95 dBc as compared to actual $\gamma$ values set of 193dBc. A simple combinational circuitry can automatically generate the $\gamma$ values if one knows the range of $\gamma$ values that is the maximum and minimum values of $\gamma$, and the linear equation. Thus, using the linear increment $\gamma$ values, the ROM size is drastically reduced.
Figure 4.12 Spectrum of the proposed DDFS using $\gamma$ accumulator in simulation

Figure 4.13 Deviation plot for $\gamma_s$ values using LUT $\gamma$ values and $\gamma$ accumulator
Figure 4.14 Deviation plot for $\gamma_c$ values using LUT $\gamma$ values and $\gamma$ accumulator

Figure 4.13 shows the comparison between theoretical and the linear increment $\gamma_s$ values of $\gamma_s$. It is inferred from the graph that the linear increment $\gamma_s$ values are in line with the theoretical $\gamma_s$ values. But, the variation between theoretical $\gamma_c$ and linear increment $\gamma_c$ is slightly larger due to the strong dependency of cosine function with $\gamma$. This can be observed from Figure 4.14. As it can be seen from Figure 4.12, this deviation does not affect the applicability of the proposed algorithm.

The SFDR from the measured data for the sine wave is 130.3 dBc, which is comparable with theoretical calculation of 135.3 dBc.
Table 4.2 shows a comprehensive comparative study of various performance parameters of ADC test bed namely, bit resolution, SFDR, SNR, ROM table size and hardware overhead for various DDFS methods reported in the literature with that of the proposed method. It can be noted that the measured SFDR is 130.3 dBc in the proposed method with only 11 bit resolution, as compared to the SFDR of 130.3 dBc with 15-16 bit resolution used by most methods. Moreover in this method since linear increment of $\gamma$ is used, the requirement of ROM has been reduced to minimal, thus reducing the hardware requirement.
**Table 4.3 Performance Summary of DDFS**

<table>
<thead>
<tr>
<th>Method</th>
<th>Bit resolution</th>
<th>SFDR (dBc)</th>
<th>SNR (dB)</th>
<th>ROM table size (bits)</th>
<th>Hardware Overhead / Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Madisetti et al (1999)</td>
<td>16</td>
<td>100</td>
<td>92.2</td>
<td>–</td>
<td>Multiplier-Less Feed-Forward Data Path</td>
</tr>
<tr>
<td>Bellaouar et al (2000)</td>
<td>11</td>
<td>60</td>
<td>60.48</td>
<td>416</td>
<td>Linear Interpolation between Sample Points</td>
</tr>
<tr>
<td>Jen-Chuan et al (2001)</td>
<td>16</td>
<td>100</td>
<td>92</td>
<td>512</td>
<td>3 adders and 4 multipliers</td>
</tr>
<tr>
<td>Langlois et al (2003)</td>
<td>14</td>
<td>96.2</td>
<td>78.92</td>
<td>960</td>
<td>64 Linear Segments</td>
</tr>
<tr>
<td>Sodagar et al (2003)</td>
<td>12</td>
<td>66.8</td>
<td>69</td>
<td>728</td>
<td>Parabolic Initial Guess</td>
</tr>
<tr>
<td>Usman et al (2005)</td>
<td>12</td>
<td>45</td>
<td>70</td>
<td>64</td>
<td>Sine Wave Symmetry from 0 to $\pi/2$</td>
</tr>
<tr>
<td>Shu-Chung et al (2006)</td>
<td>17</td>
<td>80</td>
<td>98</td>
<td>384</td>
<td>Trigonometric Double Angle formula, 3 adders and 3 multipliers</td>
</tr>
<tr>
<td>Ru Xin et al (2007)</td>
<td>18</td>
<td>99.6</td>
<td>99.3</td>
<td>–</td>
<td>Area Optimized DDFS</td>
</tr>
<tr>
<td>Kesoulis et al (2007)</td>
<td>12</td>
<td>63.12</td>
<td>65</td>
<td>448</td>
<td>Modified Amplitude Compression Technique</td>
</tr>
<tr>
<td>Lai Lin-hui et al (2008)</td>
<td>10</td>
<td>63.58</td>
<td>58.2</td>
<td>328</td>
<td>2 adders, 2 shift registers</td>
</tr>
<tr>
<td>Tze-Yun et al (2009)</td>
<td>16</td>
<td>84.4</td>
<td>90.2</td>
<td>64</td>
<td>Multiplier Less, Pipelined Data Path</td>
</tr>
<tr>
<td>Xiaojin et al (2009)</td>
<td>14</td>
<td>90</td>
<td>75</td>
<td>–</td>
<td>Two Segment Fourth Order Parabolic Approximation Technique</td>
</tr>
<tr>
<td>Wan Shuqin et al (2009)</td>
<td>16</td>
<td>84.4</td>
<td>91.8</td>
<td>69</td>
<td>Twenty eight adders, 52 shifters, 14 latches, 20 multiplexers</td>
</tr>
<tr>
<td>Proposed</td>
<td>11</td>
<td>130.3</td>
<td>64</td>
<td>–</td>
<td>Using gamma accumulator</td>
</tr>
</tbody>
</table>
4.6 CONCLUSION

A novel DDFS architecture has been proposed to generate a sinusoidal signal. The proposed method is implemented based on Taylor Series polynomial approximations and improves the spectral purity remarkably. The replacement of look-up table with a digital combinational and sequential circuits results in the reduction of hardware complexity and also aids in faster computation. Results shows that the spectral purity of the proposed DDFS is as high as 135.6 dBc for 11-bit resolution compared to 90 dBc for 14-bit resolution attainable with hybrid CORDIC algorithm.