

## The Optimization of Direct Digital Frequency Synthesizer Performance by New Approximation Technique

<sup>1</sup>Govind S. Patel and <sup>2</sup>Sanjay Sharma

<sup>1</sup>ECED, RIET, Faridabad, India

<sup>2</sup>ECED, Thapar University, Patiala, India

**Abstract:** In this study, an optimized Direct Digital Frequency Synthesizer (DDFS) utilizing Piecewise Linear Approximation is introduced. The proposed technique allows successive read access to memory cells per one clock cycle using time sharing. The output values will be temporarily stored and read at a later time. The output of this system is a reconstructed signal that is a good approximation of the desired waveform. As a result, the DDFS only needs to store fewer coefficients and the hardware complexity is significantly reduced. The proposed DDFS has been analyzed using MATLAB. The SFDR of synthesized achieved is 84.2 dBc. To prove the better performance of proposed DDS architecture it is compared favorably with several existing DDS architectures. In future it can also be used to improve the performance of Hybrid DDS-PLL Synthesizers.

**Keywords:** DDFS, frequency synthesis, ROM and piecewise linear approximation

### INTRODUCTION

The study first gives a description of the conventional DDFS. It is able to generate single-phase or quadrate sinusoids with excellent frequency resolution, good spectral purity and very fast frequency switching and phase continuity on switching. Owing to their unique characteristics, DDFS play an important role both in modern communication systems (including spread spectrum and frequency-hopping systems) and in measurement instrumentation. Simple modification to the phase-generation circuitry produces synthesized chirps useful in radar and electronics warfare systems (Dan, 2008).

The most elementary technique of compression is to store only  $\pi/2$  Rad of phase information and to generate the ROM samples for the full range of  $2\pi$  by exploiting the quarter-wave symmetry of the sine function. Beyond that, the methods include: a Trigonometric Identity, the Nicholas method, the use of Taylor Series, Hutchison Algorithm and CORIDIC algorithm. For each memory compression and algorithm technique, the worst case spurious is calculated using these premises.

The digital frequency synthesis approach employs a stable source frequency i.e., reference clock to define times at which digital sinusoidal sample values are produced. These samples are converted from digital to analog format and smoothed by reconstruction filter to produce analog frequency signals. A DDFS typically consists of a Phase Accumulator (PA) and a sine Lookup Table (LUT). The input to the phase

accumulator is a frequency control word, which determines the periodicity of the phase accumulator. The PA is updated to the frequency control word or tuning word, at each clock, the output of the PA is fed to the LUT. The output of the LUT is then converted to an analog signal using a digital to analog converter. The size of the LUT depends on the length of the n-bit PA. If n is large then the LUT becomes too large, which is not desirable. This slows down the speed of the DDS and results in higher power consumption. To reduce the size of the LUT, a technique of Phase Truncation (PT) is employed. Since in this technique part of the phase generated by the PA is truncated that gives rise to spurs in output spectrum. To minimize these spurs, dither is added to the system that reduces the spurs in output spectrum. Since, the DDFS is a digital system clock jitter also introduces noise in the output spectrum. Jitter is an abrupt and unwanted variation of one or more signal characteristics, such as the interval between successive pulses, the amplitude of successive cycles, or the frequency or phase of successive cycles (Webb, 1970). With advances in design and process technology, today's DDFS devices are very compact and draw little power.

The study investigates and proposes a new DDFS architecture which is based on Piecewise Linear Approximation.

### CONVENTIONAL DDS ARCHITECTURE

DDS consist of main three components; Phase Accumulator, ROM and DAC. Phase Accumulator,

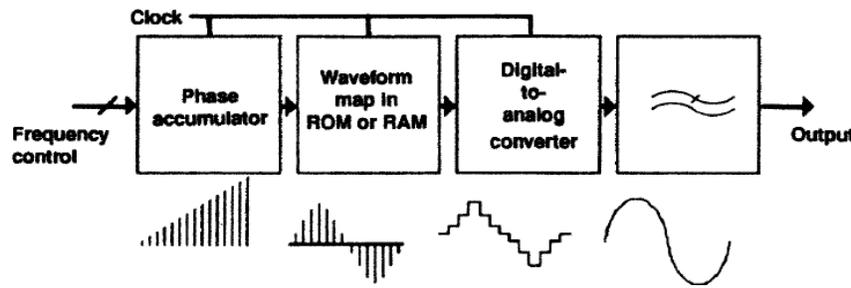


Fig. 1: Basic block diagram of DDFS

which is controlled by Clock  $f_c$ , is accumulated by Step Length  $K$ .  $M$  bit binary code, which is the output of Phase Accumulator, is seen as the address of ROM wave and addressing ROM. Amplitude Code, which is the output of ROM, via D/A converter turns into Step Wave, then be smoothed by Low Pass Filter and complex signal waves, which is decided by Amplitude Code stored in ROM, can be gained finally. Thus, any waves can be generated by DDS. The process of wave generating is: ROM stores wave data, via every Count Value of Address Counter, corresponding to a Location Address of Wave Storage, circularly reads data of every location in turn, then send it to D/A converter, which convert it into corresponding Analog Quantity of output voltage and ultimately be filtered by Low Pass Filter generating smooth waves (Tierney *et al.*, 1971).

The basic DDS structure is shown in Fig. 1,  $M$  means Frequency Control Word;  $f_c$  means Clock Frequency,  $W$  means the Word Length of Phase Accumulator,  $D$  means ROM Data Bit and the Word Length of D/A converter.

**Frequency tuning equation:** A sine wave is generally expressed as  $x(t) = \sin(\omega t)$  which is non-linear and not easy to generate. The angular rate is given by  $\omega = 2\pi f$  where,  $\omega$  is the angular frequency. For an  $n$ -bit accumulator the output signal will have the frequency specified:

$$f_{out} = \frac{\Delta p * f_c}{2^n} \tag{1}$$

The phase rotation for that period can be determined by:

$$\Delta p = \omega * \Delta t \tag{2}$$

**Phase accumulator:** Continuous-time sinusoidal signals have a repetitive angular phase range of 0 to 360 degrees. The counter's carry function allows the phase accumulator to act as a phase wheel in the DDFS implementation. The PA is a modulo- $M$  counter that increments its stored number each times it receives a clock pulse. The number of discrete phase points contained in the  $p$  wheel is determined by the resolution

of the PA ( $n$ -bits), which determines the tuning resolution of the DDFS. The basic tuning equation for DDFS architecture:

$$f_{out} = \frac{M * f_c}{2^n} \tag{3}$$

where,

- $f_{out}$  = Output frequency of the DDS
- $M$  = Frequency control word
- $f_c$  = System clock
- $n$  = Length of the phase accumulator (in bits)

Any change to the value of  $M$  results in immediate and phase-continuous changes in the output frequency.

**Phase-to-amplitude converter (ROM/LUT):** In this study, the DDFS's ROM is a sine Look up Table; it converts digital phase input from the accumulator to output amplitude. The accumulator output represents the phase of the wave as well as an address to a word, which is the corresponding amplitude of the phase in the LUT. This phase amplitude from the ROM LUT drives the DAC to provide an analog output (Lin-hui, *et al.*, 2008). It is also called a digital Phase-to-Amplitude Converter (PAC). In an ideal case with no phase and amplitude quantization, the output sequence of the look up table is given by:

$$\sin\left(\frac{2\pi * P(i)}{2^n}\right) \tag{4}$$

**Digital-to-analog converter and filter:** The phase accumulator computes a phase (angle) address for the look-up table, which outputs the digital value of amplitude-corresponding to the sine of that phase angle to the DAC. The DAC, in turn, converts that number to a corresponding value of analog voltage or current. The DAC and rest of the system run at the same reference clock for synchronization.

### QUADRANT COMPRESSION TECHNIQUE

The size of the ROM can be reduced by more than 75% by taking advantage of the fact that only one

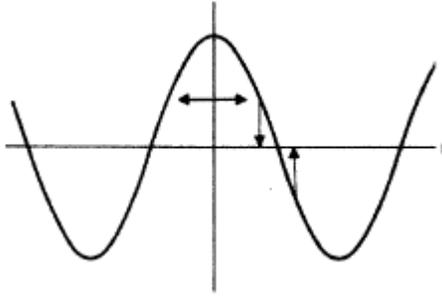


Fig 2: Sine quadrant symmetry

Table 1: Quadrants table

Phase	MSB	MSB-1	Sine
$0 < A < 90$	1	0	SinA
$90 < A < 180$	1	1	Sin(90-A)
$180 < A < 270$	0	0	-SinA
$270 < A < 360$	0	1	-Sin(90-A)

quadrant of the sine (cosine) needs to be stored, since the sine has the symmetric property (Bar-Giora, 1999), as shown in Fig. 2.

Thus, for  $0 \leq \alpha \leq 90$ ,  $\sin(90 - \alpha) = \sin((90 + \alpha)$ ,  $\sin(270 - \alpha) = \sin((270 + \alpha)$ ,  $\sin(\alpha) = \sin(-\alpha)$  and  $\sin \alpha = -\sin((180 + \alpha)$

The presentation of  $\sin \alpha$  only across the first quadrant  $0 \leq \alpha \leq 90^\circ$  is sufficient to reconstruct all quadrants from the first.

Thus the 2 MSBs of W (the input word) are needed only to control the quadrant and the values of the quadrant need to be manipulated as shown in Table 1. Thus, given  $\sin \alpha$  over only the first quadrant, the operations necessary to flip to the quadrants are as follows:

- Quadrant 1:**  $\frac{1}{2} + \frac{1}{2} \sin i$  i is running index from 0 to  $2^{W-2} - 1$
- Quadrant 2:**  $\frac{1}{2} + \frac{1}{2} \sin(i \text{ complemented})$  every bit inverted
- Quadrant 3:**  $\frac{1}{2} + \frac{1}{2} \sin(i \text{ complemented})$
- Quadrant 4:**  $\frac{1}{2} + \frac{1}{2} \sin i$

For all quadrants, i run from 0 to  $2^{W-2} - 1$ .

In the first quadrant, the sine starts at 0.5 (1000...binary) and curves up to 1 (1111.). Then the address to the ROM goes from the maximum state to zero state, but since all the bits are inverted, it will now curve back down the same way it curved up (1111. to 1000...). In third quadrant, it starts again at 0.5, now 01111., but because of the inversion, the value of the sine is maximized and the total sum declines to 0 (000.) when it climbs back in the fourth quadrant (from 000.. to 0111).

Therefore, to achieve the above equation, the MSB needs to be inverted, to the output and the MSB and MSB-1 need to invert the sine function according to the quadrant the operate in. a typical block diagram is shown in Fig. 3.

The reduction of the ROM size is achieved because of the sine quadrant symmetry and because the inside ROM now maps W-2 input bits to D-1 output bits only, compared to W to D bits originally. The saving of ROM size is therefore:

$$\frac{2^{W-2}(D-1)}{2^W D} = \frac{0.25(D-1)}{D} \approx 75\% \text{ saving} \quad (5)$$

**Beyond that, the other methods include:** a Trigonometric Identity, the Nicholas method, the use of Taylor Series, Hutchison Algorithm and CORIDIC algorithm. For each memory compression and algorithm technique, the worst case spurious is calculated using these premises (Chen and Chau, 2010).

### PROPOSED DDS ARCHITECTURE

The concept of this technique is the same with that used in above quadrant compression technique (Caro and Strollo, 2005). Figure 4 shows the block diagram of the proposed DDFS architecture and its Mimulink architecture in Fig. 5. The MSB2 is used to select the

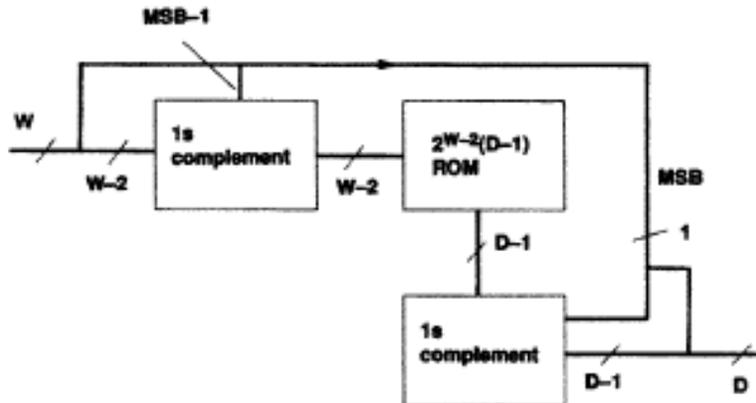


Fig. 3: Sine quadrant compression

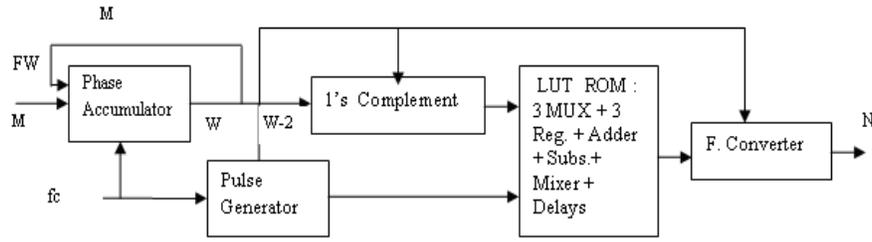


Fig. 4: Proposed DDS

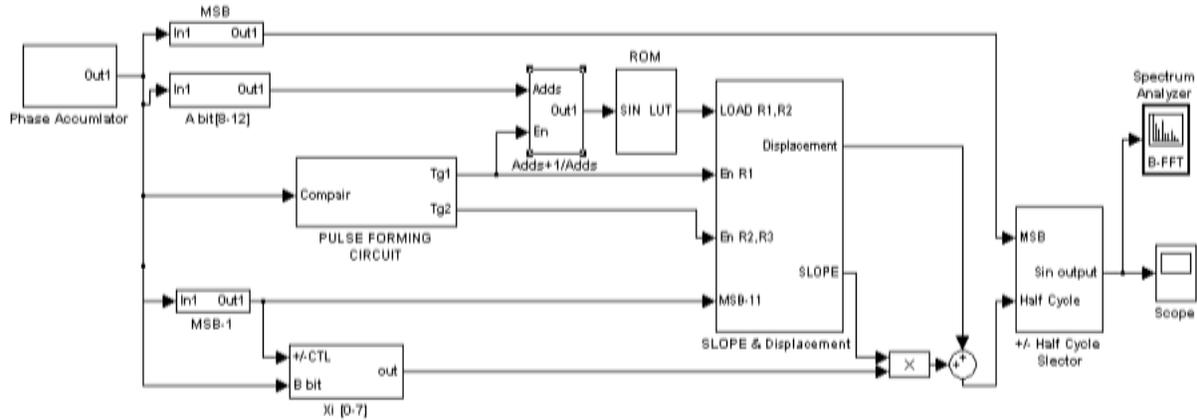


Fig. 5: Proposed model of DDS

quadrants of the sine wave, while the *MSB1* is used to control the format converter. The remaining *W-2* bits are fed into Complement or whose output is split into two parts, the *MSB* part, with *A* bits long, represents the *S* segments and the *LSB* part with *B* bits long, represents an angle  $x$  in the interval  $[0, \pi / (2S)]$ . A multiplexer and its coefficients are the equivalent of a ROM which provide the segment initial amplitudes  $Q_i$ , represented with *D* bits. The proposed architecture also incorporates pulse forming circuit which controls the fetching and loading process of successive  $Q_i$  coefficients. This circuit along with the three storage registers and one Subtract or is essential to perform the task of the slope derivation during the segment interval.

Besides the sine symmetry property, the linear approximation method has been used to approximate the first quadrant of sine function by *S* straight lines; each line is defined by two coefficients,  $P_i$  and  $Q_i$ . The coefficient  $M_i$ , which represents the slope of  $i^{th}$  element. The first quadrant of sine function approximation segment can be calculated from the sine function as follows.

$$P_i = \{ \sin [i \Delta x] - \sin [(i-1) \Delta x] \} / \Delta x \quad 1 \leq i \leq S \quad (6)$$

where,  $\Delta x$  = the length of segment. Above Eq. (6) can be realized easily by subtracting the  $\sin [i \Delta x]$  at successive phase angles and then dividing the result by  $\Delta x$ .

As  $\Delta x$  unsigned constant coefficient, the division can simply be realized by binary operation. The coefficients  $Q_i$ , is equal to  $[\sin (i-1) \Delta x]$  points, As examples for segment number1, ( $Q_1 = 0$ ), yields  $K_1 = P_1 x$  and for segment number 2, ( $Q_2 = \sin \Delta x$ ), in general,  $Q_i = \sin [(i-1) \Delta x]$  for the  $i^{th}$  segment and it can be realized by delaying the pervious  $\sin (i \Delta x)$  by one clock period, hence the realization of the whole  $K_i(x)$  function is accomplished. It is clear that it must get two consecutive sine points at the same time to conduct the process of subtraction and extraction of the slope later. These two sine points can be got only when the corresponding phase angles point simultaneously to their addresses in the sine LUT and that is an inconsequent assumption. As mentioned earlier, the accessing of the memory is valid only once at a specific clock cycle. In this study, we introduce architecture of pulse forming circuit which is performing the task of time sharing and propose the procedure enumerated below to get around this problem.

The value of phase register at any clock period represents the phase of the sine function. As not all of the samples of the sinusoid are stored, only the first *A* bits of the *W-2* phase accumulator output are used to select segment initial amplitudes  $Q_i$ , i.e., it represents the MUX address inputs. The remaining *B* LSB's bits ( $B = W-2-A$ ) represent an angle  $x$  in the interval  $\Delta x$  and are used to calculate the value of the interpolated sine

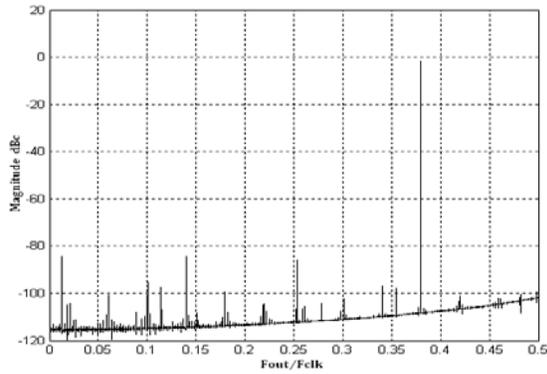


Fig. 6: Output spectrum at 24%

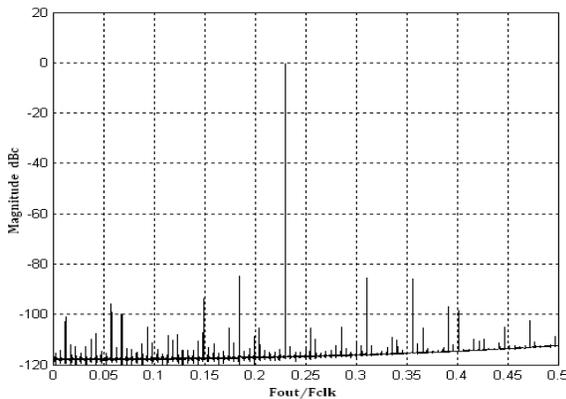


Fig. 7: Output spectrum at 40%

point. It has three simple blocks: digital comparator, pulse narrowing circuit and tapped delay.

- At each clock cycle, the digital comparator examines the MUX Address inputs for detecting the changes in data select inputs i.e., transitions between the segments
- The detected signal will be applied to the Pulse Narrowing Circuit to produce a  $\Delta t$  pulse width signal i.e., trigger1 (tg1), which is usually a fraction of  $1/fclk$
- This signal, tg1 gives an order to advance the Data Select Inputs of MUX by 1, hence the output of the MUX during this time slot is  $\text{Sin} [(i+2)\Delta x]$
- At the same time, the tg1 is used to load this value in register1 (R1)
- After  $\Delta t$ , the data select inputs get back to the previous address value, so the output of MUX will be  $\text{Sin} [(i+1)\Delta x]$
- Trigger2 (tg2) enables register 2 (R2) to load this value
- The content of R2 will be subtracted from the content of R1 and the result will be loaded in register3 (R3), after a specific time which is precisely sufficient to give a chance for signals to be propagated through all gates and settle. Hence, the slope is simply derived and kept unchanged during the segment interval

The main feature of this proposed technique is that there is no need to derive the slope at each clock cycle. By this study, the slope can be derived at the first sample of each segment interval and remains unchanged during entire interval. The ROM Address Bus toggles between two values during this interval. These needless Read Cycles can consume unjustified excessive power. In contrast, the proposed study

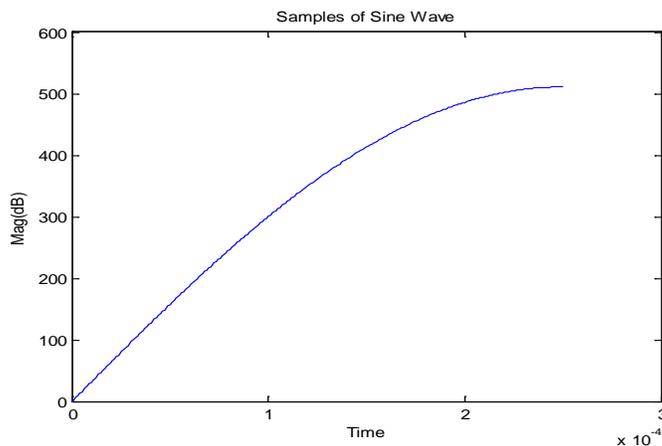


Fig. 8: Generated 256 samples of sine wave

Table 2: Comparison with other architectures

Architecture	Compression ratio	ROM	Additional circuitry	Remarks
Lin-hui <i>et al.</i> (2008)	$2^{11} \times 12$	328	Adder, ROM, shift register, multiplier	Two ROM and slightly more circuitry needed
Chen and Chau (2010)	-	Without ROM	Multiplier, adder, shift registers	Number of multiplier needed
Proposed DDS Architecture	$2^{13} \times 14$ ROM	Register( $2^5 \times 14$ )	Multiplier, adders, register, shift register, delay, pulse generator	Best compression ratio

conducts one Read Cycle per segment and get back to the idle mode during the rest of the interval (Jouko, 1997; Sharma and G.S., 2012; Curticapean *et al.*, 2001).

ROM compression improves its efficiency or its compression ratio with the increase of W and D, the input and output bit widths. Compression ratios of close to 256:1 are achieved and we expect that improvement will be made as more research is applied to this rather interesting aspect of DDS technology (Freeman, 1989; Proakis, 2001).

The spurious level for the DDS is shown in Fig. 6 and 7 for output frequency of 24 and 40% of clock freq. when the input is set up to be 7537 and 12454 respectively. And also Fig. 8 shows the 256 generated samples of the Sine wave.

Table 2 presents a comparison between the proposed architecture and different implementations reported in the recent literature. The comparison shows significant improvement in all features.

## CONCLUSION

A variety of ROM compression algorithms have been presented along with the standard equations necessary to generate the ideal sine digital waveform for any input and output bit size. We have investigated the problem concerning the optimal coefficient choice and the efficient implementation of DDFS circuits with piecewise polynomial approximation. In this study, a novel ROM elimination technique was presented for application in low complex high spectral purity DDFS. Unlike many reported architectures that used complex circuits to compute the sine samples, Only 32 points from a standard sine LUT with fewer registers are required. System complexity is greatly reduced by using an efficient phase to amplitude conversion architecture. It was shown that a Compression ratio of approximately 256:1 and SFDR 84.2 dB were attained. We expect that improvement will be made as more research is applied to this rather interesting aspect of DDS technology. The technique was compared with the existing ones in terms of storage reduction computation and spectral purity. The comparison shows significant improvement in all features. The results show that the

proposed method unifies the existing polynomial approximation methods for sinusoidal DDFS.

## REFERENCES

- Bar-Giora, G., 1999. Digital Frequency Synthesis Demystified. 6th Edn., LLH Technology Publishing, Eagle Rock, VA.
- Caro, D.D. and A.G.M. Strollo, 2005. High performance DDS synthesizers using piecewise polynomial approximation. IEEE T. Circuits Syst. I, 52(2): 324-337.
- Chen, Y.H. and Y.A. Chau, 2010. A direct digital frequency synthesizer based on a new form of polynomial approximations. IEEE T. Consum. Electr., 56: 436-440.
- Curticapean, F., K.I. Palomaki and J. Nittylahti, 2001. Direct digital frequency synthesizer with high memory compression ratio. Electron. Lett. 37(21): 1275-1277.
- Dan, M., 2008. Modulating Direct Digital Synthesizer in a FPGA. VP of Engineering Accelent System Inc., Quick Logic, pp: 143-156.
- Freeman, R.A., 1989. Digital Sine Conversion Circuit for Use in Direct Digital Synthesizer. U.S. Patent No. 4 809 205.
- Jouko, V., 1997. Methods of mapping from phase to sine amplitude in DDS. IEEE T. Ultrason. Ferr., 44(2): 526-534.
- Lin-hui, L., L. Xiao-jin and L. Zong-sheng, 2008. A low complexity direct digital frequency synthesizer. IEEE 9th International Conference on Solid-State and Integrated-Circuit Technology, pp: 1653-1656.
- Proakis, J.G., 2001. Digital Communications. 4th Edn., McGraw- Hill, Boston.
- Sharma, S. and G.S., 2012. A Novel ROM compressed technique for freq. Synthesizer, National Conference of ITMG, India.
- Tierney, J., C.M. Radar and B. Gold, 1971. A digital frequency synthesizer. IEEE T. Acoust. Speech., 19(1): 48-57.
- Webb, J.A., 1970. Digital Signal Generator Synthesizer. US Patent No. 3654450.