CHAPTER 1
Introduction

1.1Motivation

Orthogonal frequency division multiplexing (OFDM) is an attractive technique [1], which is suitable for frequency selective channel such as multipath wireless channel. In Europe, OFDM has been standardized for digital audio broadcasting (DAB) and digital terrestrial video broadcasting (DTVB).Meanwhile, the applications of the OFDM transmission technique for various broadband communication systems and services have been reported in the literature . For example, OFDM has been chosen for the extension of the IEEE 802.11a standard for the 5-GHz frequency band. In this chapter,

Fig 1-1 shows the channelization scheme for IEEE802.11a standard [2], which shall be used with the FCC U-NII frequency allocation. The lower and middle U-NII sub-bands accommodate eight channels in a total bandwidth of 200 MHz. The upper U-NII band accommodates four channels in a 100 MHz bandwidth. The centers of the outermost channels are at a distance of 30 MHz from the band's edges for the lower and middle U-NII bands, and 20 MHz for the upper U-NII band. The OFDM PHY l operates in the 5 GHz band, as allocated by a regulatory body in its operational region. The center frequency is indicated in Figure 1. In a multiple cell network topology, overlapping and/or adjacent cells using different channels can operate simultaneously.

The system uses 52 subcarriers that are modulated using binary or quadrature phase shift keying (BPSK/QPSK), 16-quadrature amplitude modulation (QAM), or 64-QAM

Specifications of Bluetooth and IEEE 802.11b [3] are shown in TABLE 1 and TABLE 2, IEEE 802.11b WLAN builds on the data rate capabilities to provide 5.5Mbits/s and 11Mbits/s payload data rates in addition to the 1Mbps and 2Mbps rates. To provide the higher rates, 8-chip complementary code keying (CCK) is employed as the modulation scheme. An optional mode is provided that allows data throughput at the higher rates (2, 5.5 and 11Mbits/s) to be significantly increased by using a shorter PLCP preamble. Another optional mode replacing the CCK modulation with packet binary

convolutional coding (PBCC) is provided. IEEE 802.11b WLAN is defined for both DSSS and FHSS technique in the 2.4GHz ISM band.

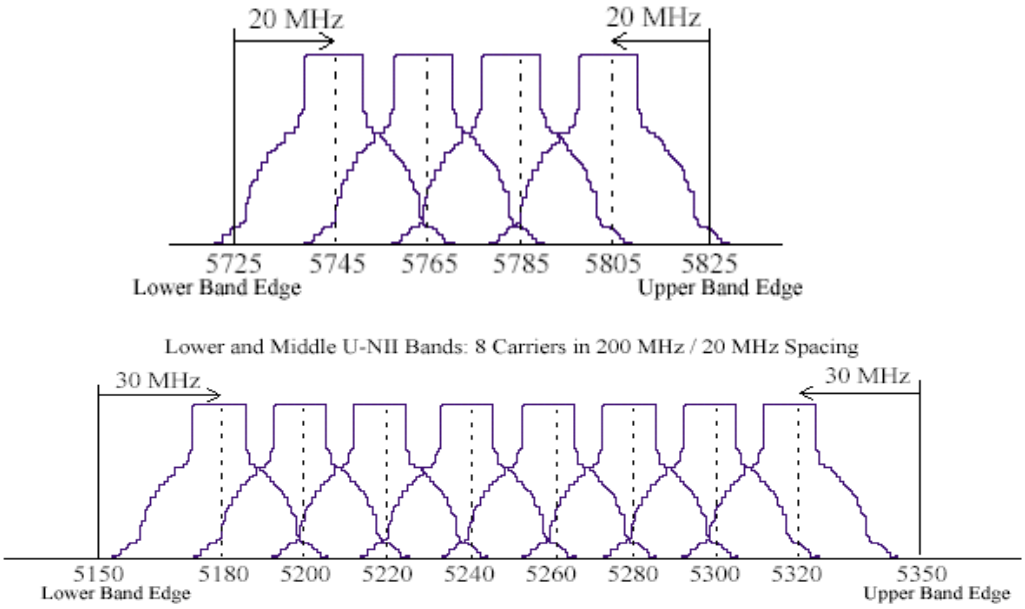


Fig 1-1 OFDM PHY frequency channel plan for the United States

For a 1Mbit/sec FHSS technique of Bluetooth, 2 level Gaussian FSK modulation scheme with a nominal bandwidth bit-period (BT) = 0.5 is used. The symbol {1} shall be encoded with a peak deviation of ($+f_d$), giving a peak transmit frequency of (F_c+f_d), which is greater than the carrier center frequency and vice versa. The deviation factor for the modulation is nominally 0.32. For a 2Mbit/sec FHSS technique, 4 level Gaussian FSK modulation scheme with a nominal bandwidth bit-period (BT) = 0.5 is employed. The deviation factor for the modulation is nominally 0.144 and it will be no less than 0.135. For a DSSS technique, DBPSK modulation scheme is used for 1Mbit/sec transmission and DQPSK is used for 2Mbit/sec enhanced access rate.

In Bluetooth system, that is, a frequency hopping (FH) transceiver architecture, Gaussian Frequency Shift Keying (GFSK) is chosen to allow noncoherent detection capability and constant-envelope signaling. In IEEE 802.11b WLAN system, GFSK is also adopted for FH transceiver architecture. However, DBPSK/DQPSK are selected for direct sequence (DS) transceiver architecture because of its power efficiency and spectrum efficiency. On the other hand, CCK/PBCC are chosen for DS high data rate transceiver architecture because of the Block Coded Modulation (BCM) and the Trellis

Coded Modulation (TCM) concept. Both of the modulation schemes use coded modulation concept to get better performance than conventional modulation schemes. OFDM subcarries are modulated by using BPSK, QPSK, 16-QAM, 64-QAM modulation in IEEE 802.11a

Data Rate	1Mbps
Operating Frequency Range	2402-2480MHz (79 channels)
Number of Operating Channels	79 for US & Europe (2402-2480) 23 for Spain (2449-2471), France (2454-2476)
Operating Channel Center Frequency	Start from 2402GHz with 1MHz step size
Modulation	2-GFSK, BT=0.5, fd=0.28~0.35
Input Dynamic Range	-70~-20dBm
Sensitivity	-70dBm for 0.1% BER

Table 1-1 Bluetooth Specification

Data Rate	1, 2, 5.5 and 11Mbps
Operating Frequency Range	2402-2483.5MHz (13 channels)
Number of Operating Channels	11 for FCC (2400-2472) 13 for ETSI (2400-2483.5)
Operating Channel Center Frequency	Start from 2412GHz with 5MHz step size
Modulation	DBPSK, DQPSK, CCK and PBCC
Input Dynamic Range	-80~-10dBm(1, 2Mbps) -76~-10dBm(5.5, 11Mbps)
Sensitivity	-76dBm for 0.08 FER(max)

Table 1-2 IEEE 802.11b Specification

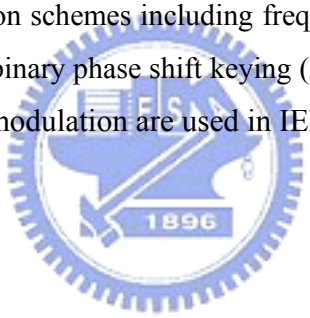
Based on Specifications of Bluetooth, IEEE 802.11b and IEEE 802.11a, we have much interesting on how to obtain a fast switching speed and high-resolution sine waves for modulation scheme. Many methods deal with emerging modern digital techniques used to generate and modulate sine waves. These waveforms are used in almost all radio applications, communications, radar, digital communications, electronic imaging, and so on.

There are many different ways to generate the sine waves. The most traditional and most popular way uses the phase-locked loop (PLL) technology. The PLL is a feedback mechanism locking its output frequency to a reference clock . Because of the mature of the PLL technology, the PLL synthesizers gained popularity for their simplicity and economics. But there are some drawbacks to PLL. In PLL architecture, if we want to get

a higher resolution output, it will much increase the circuit complexity. And due to the feedback structure, the lock-time is variable, and it usually needs many reference clocks to generate the needed output.

In order to get a more precise and fast-switching speed sine waves, the direct digital frequency synthesizer (DDFS) architecture was chosen . In the last few years, direct digital synthesis technology has captured the attention of synthesizer designers and enjoys an unusual popularity. Various designs and architectures are used to implement DDFS, and the applications vary according to the requirements.

In this thesis, we try to find good algorithm to built hardware efficient DDFS. The tuning range can reach from 100KHz to 12.5 MHz, frequency resolution is 95.4hz. Multi-mode modulator is implemented by using proposed DDFS, The modulator consists of four kinds of digital modulation schemes including frequency shift keying (FSK), Double frequency shift keying (DFSK), binary phase shift keying (BPSK) and quadrature phase shift keying (QPSK). Some of these modulation are used in IEEE802.11a standard.



1.2 Thesis Organization

In chapter 2, Principles of Direct Digital Frequency Synthesizer will be discussed, including concepts of Direct Digital Frequency Synthesizer, DDFS parameter and review of existing phase to amplitude algorithm, Spur items were caused by finite output length, phase truncation are also depicted.

In chapter 3, DDFS Architecture and Algorithmic Approximation are described and verified by verilog code ,SFDR of DDFS is calculated by Matlab with Verilog Outputs.

In chapter4 Application of DDFS for multi-mode modulation is illustrated. Function of multi-mode modulator includes frequency shift keying (FSK), Double frequency shift keying (DFSK), binary phase shift keying (BPSK) and quadrature phase shift

keying (QPSK). .

In chapter 5 Verilog simulation and verification for multi-mode modulator will be proceed, Verilog code are synthesized by using Synplify Pro. Altera device EPF10K100ARC240-1 and simple test system are used to verify the function of proposed digital modulator.

In chapter 6 Conclusion

CHAPTER 2
Principles of Direct Digital Frequency Synthesizer

2.1 The Concepts of Direct Digital Frequency Synthesizer

In 1971, Tierney [4] proposed an architecture for direct digital frequency synthesizer (DDFS) with an L-bit frequency control word (FCW) and D-bit output word length, it employs an L-bit accumulator to generate the advancing phase of a sine wave, then uses a ROM table to look-up the D-bit pre-stored sine samples. That is, one period of a sine wave is divided into 2^W segments and a sample is chosen within each segment to be stored in the ROM. According to the current phase accumulator output ,which served as the ROM table address, a sample is looked-up and outputted from the table each clock cycle.

There are a great variety of DDFS implementations since Tierney . However, Most of DDFS designs are based on Tierney, The main block function for DDFS is shown in Fig 2-1. It consists of four elements: phase accumulator, a sine/cosine mapping function (SCMF), digit to analog converter(DAC) and low-pass filter(LPF). Remembering that the presentation of a fixed-amplitude, fixed-frequency, and fixed-phase sine wave is given by $A \sin(\omega t + \varphi)$

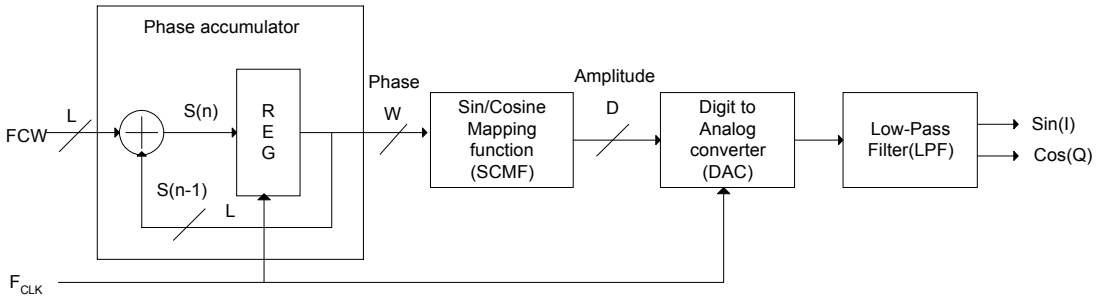


Fig 2-1 Block diagram of the direct digital frequency synthesizer

We can trace the signal buildup as follows: The signal phase is a linear function, The

slope of the phase dθ/dt is the angular frequency ω. To generate the amplitude of the output waveform, it is necessary to transform the phase θ(t) to sin[θ(t)], and this is usually done by using sine/cosine mapping function (SCMF). Since the transformation is nonlinear and most of SCMF use ROM (or RAM) look-up table or computational method for nonlinear transformation.

The output of the SCMF is thus the digital representation of the sine wave signal amplitude(digital samples) and the DAC converts it to an analog sine wave. The LPF removes all the aliasing frequency and causes the signal to appear smooth.. The accumulator is a device that performs the function S(n) = S(n-1) + FCW . Such a device is a digital integrator and produces a linear output ramp whose slope (rate of change) is given by FCW, the input frequency control word. This device is used to generate the phase ωt depends on the clock at which the accumulator runs. The accumulator is operating as an indexer whose output (representing the phase) controls the input of sine/cosine mapping function (SCMF).

Suppose that the accumulator size is L bits, say L = 32 bit binary device. It is therefore able to accumulate from 0 to $2^{32} - 1$. Obviously, above this number, the accumulator overflows and will start from zero again. The rate of the accumulation depends only on the clock rate Fclk = 1/T and FCW. And FCW can be as low as 0, in this case, the accumulator will not increment (equivalent to generating a dc signal)- or any arbitrary number FCW < $2^L - 1$. If we equate 0 with zero phase and $2^{32} - 1$ with 2π, then we have a device that generates phase from 0 to 2π periodically (since the device operates modulo 2^{32}).

For a demonstration of the operation of the accumulator, let us assume that we clock the device, an L = 32 bit accumulator, at Fclk = 2^{32} / 10. Then if FCW = 1, it will take exactly 10 s (2^L clock ticks) to generate 0 to 2π. However, if FCW = 2^{30}, then it will take 10 / 2^{30} s (four clock ticks). Obviously FCW controls the rate of change of the accumulator, and the rate of change of the phase is the frequency ω. In the above example, for FCW = 1, Frequency of DDFS is 0.1 Hz, while for FCW = 2^{30}, the frequency of DDFS is Fclk / 4 Hz. Mathematically, since ω = dθ/ dt, we can rewrite

$$F_{out} = 2\pi \frac{d\theta}{dt} = \frac{F_{clk} FCW}{2^L} \qquad FCW < 2^L - 1 \qquad (2\text{-}1)$$

where

F_{out} : output frequency of DDFS

F_{clk} : input clock frequency

FCW : frequency control word

L : phase accumulator bit length

The constraint in (2-1) come from the sampling theorem .The phase control word in FCW is an integer; therefore, the frequency resolution $F_{res} = \frac{F_{clk}}{2^L}$ is found by setting FCW =1.

The numerical period of the phase accumulator output sequence is defined as the minimum value of P for which θ(N)= θ(N+P) for all N. The numerical period [5] of phase accumulator output sequence (in clock cycle) is:

$$P = \frac{2^L}{GOD(FCW, 2^L)} \qquad (2\text{-}2)$$

where GOD(FCW,2^L) represents the greatest common divisor of (FCW,2^L). An example of frequency control words is FCW_1 = 0001, FCW_2 = 0101 and L = 4,we see that the sequence generated from FCW_1 is q = {q0, q1, q2, q3, q4, q5, q6, q7, q8, q9, q10, q11, q12, q13, q14, q15} and sequence generated from FCW_2 is p = {p0, p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11, p12, p13, p14, p15}, The terms of p sequence are obtained by picking terms from the q sequence when reaching its end. Sequence of p can be rewritten in terms of q sequence p = {q0, q5, q10, q15, q4, q9, q14, q3, q8, q13, q2, q7, q12, q1, q6, q11} In fact, this arrangement can be written as p(n) = q((nxFCW_2)mod2^L), where the sequence index n begins at 0,where FCW_1 is the value after normalization ,and the (.)mod2^L denotes a modulo operation. The time output vectors for FCW can be formed from a permutation of the individual elements of the vectors for FCW = 1

$$_{FCW}\theta(n) = {}_1\theta((nFCW) \bmod 2^L) \qquad (2\text{-}3)$$

Where FCW and 2^L are relatively prime. As shown in (2-3), each input time vector may be formed from a permutation of another time vectors by permuting the indices using

(nFCW) mod 2^L .

The sine output DDFS operates by applying some memory-less nonlinear function s{} to the phase accumulator output to produce the sine function. The DFT of SCMF output using (2-3) is:

$$S\{_{FCW}\ \theta(m)\} = \sum_{n=0}^{2^L-1} s\{_1\theta((nFCW) \bmod 2^L)\} \times W_{2^L}^{mn}$$

$$m=0,1,\ldots\ldots,2^L\text{-}1 \qquad (2\text{-}4)$$

Where the period of phase accumulator is 2^L when FCW and 2^L are relatively prime, and $W_{2^L} = e^{-j2\pi/2^L}$. It can be used to show that the permutation samples in the time domain result in the same type permutation in the frequency domain [5] .

$$S\{_{FCW}\ \theta(m)\} = \sum_{q=0}^{2^L-1} s\{_1\theta(q)\} \times W_{2^L}^{m(qJ \bmod 2^L)}$$

$$= \sum_{q=0}^{2^L-1} s\{_1\theta(q)\} \times W_{2^L}^{q(mJ \bmod 2^L)}$$

$$= S\{_1\ \theta((mJ) \bmod 2^L)\} m = 0,1,\ldots\ldots\ldots,2^L - 1 \qquad (2\text{-}5)$$

The above equation establishes that the permutation of the samples in the time domain results in the same type permutation of the DFT samples in the frequency domain, because J and 2^L are relatively prime. This means that the spectrum due to all system nonlinearities can be generated from a permutation of another spectrum, when GOD(FCW,2^L) = 1 for all FCW, because each spectrum will differ only in the position of the spectrum and not in the magnitudes.

2.2 Sine/Cosine mapping function (SCMF)

Once the phase is generated by the phase accumulator , it is converted into digital representation of output waveform using the sine/cosine-computation block as y(θ)=sin(2π θ/2^W).The synthesized digital signal can be either directly used in digital systems or converted to analog form and then filtered in analog applications.

The SCMF converter is normally the most complex, power hungry and slow sub-unit, this block has therefore been the subject of much research and much innovative architecture has been proposed.

There are a wide variety of methods for SCMF, from full memory to full-hardware approaches. The methods can be placed into three categories.

(1) Based on ROM look-up tables, containing amplitude samples of complete, half, or quarter-period of a sinusoid, and addressed by digital phase, which is generated by the phase accumulator. In the case of half- or quarter-period ROM look-up tables, the other half-or quarter periods are reconstructed from the stored data.

(2) Computational methods [6] [7] [8], trying to compute the sine amplitude samples from the digital phase, which is generated by phase accumulator. Sine/Cos computation by using Taylor series and CORDIC algorithm are examples of this category

(3) Initial guess /correction methods [9] [10] [11], in which an initial guess for the sine/cos function is generated by digital hardware and then is corrected by a small ROM look-up table, containing the difference between the initial guess and the accurate value for the sine/cos amplitude. If the initial guess is properly provided, in each memory location just small correcting data will be stored instead of the whole sine/cos amplitude. It is obvious that the closer the approximation is to the ideal son/cos function, the more memory-wordlength shortening there will be. Compared with the approaches based on ROM lookup tables, these methods require smaller memories, and hence will be faster and consume less power. Also, because of their lower hardware complexity in comparison to computational methods, Hence designer has widely preferred this method to first two categories.

2.2.1 Quadrant Compression

The most elementary method of sine storage compression or reduction of computational hardware implementation is to exploit the symmetry of sin function about 0 and $\pi/2$,as shown in Fig 2-2

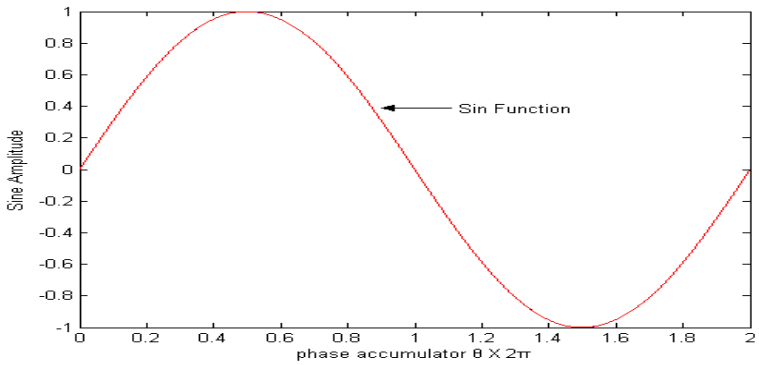


Fig 2-2 Sine quadrant symmetry.

Thus, for $0 \leqq \theta \leqq 90$, $\sin(90-\theta) = \sin(90+\theta)$, $\sin(270-\theta) = \sin(270+\theta)$, $\sin\theta = -\sin(-\theta)$, and $\sin\theta = -\sin(180+\theta)$. The presentation of sin a only across the first quadrant $0 < \theta < 90^\circ$ is sufficient to reconstruct all quadrants from the first quadrant, for example ,the sine function between $\pi/2$ and π may be synthesized from the samples between 0 and $\pi/2$ by taking the phase modulo $\pi/2$ and then taking the absolute value of the phase. Thus the 2 MSBs of W (phase accumulator output bit width) are needed to control the quadrants, and the values of the first quadrant need to be manipulated as shown in Table 2-1.

PHASE	ACCUMULATOR OUT(W) MSB	ACCUMULATOR OUT(W) MSB-1	Sine
$0<\theta<\pi/2$	0	0	$\sin\theta$
$\pi/2<\theta<\pi$	0	1	$\sin(\pi/2-\theta)$
$\pi<\theta<3\pi/2$	1	0	$-\sin\theta$
$3/2\pi < \theta < 2\pi$	1	1	$-\sin(\pi/2-\theta)$

Table 2-1 Quadrant Table.

This is easily implemented in hardware by truncating the phase accumulator output MSB bit and the use the second MSB to full wave rectifies the magnitude of the phase. As shown in Fig 2-3, the sampled waveform at the output of the SCMF is a full wave rectify version of the desired sine wave. The final output sine wave is the generated by multiplying the full wave rectified version by –1 when the phase is between π and 2π .This is accomplished simply by multiplying by the negative of the phase accumulator MSB. Thus, given sin over only the first quadrant, the operations necessary to flip to the other quadrants are shown in Fig 2-3

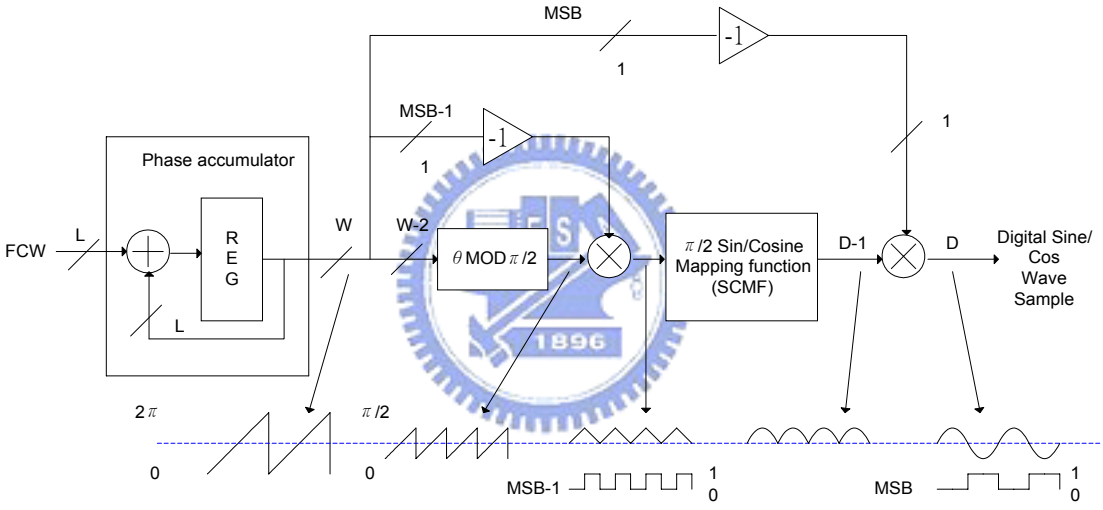


Fig 2-3 Logic exploit quarter wave symmetry

2.2.2 π /2 Sine/Cosine mapping function (SCMF)

From a conceptual point of view, the simplest approach consists of using a ROM-based Look-UP Table(LUT). Many ROM size reduction techniques have been described in the paper [9] [10] [11]. First the quarter wave symmetry of sine function can be exploited to reduce by 4 the number of angles for which a sine amplitude. Truncating the phase accumulator output is another common method, although it induces spurious harmonics. Various angular decomposition methods [12][13] have been proposed to

further reduce memory requirements with LUT-based methods, More recent approaches have attempted to further reduced the ROM size by increasing the amount of calculations performed to obtain a sine approximate the first quadrant of the sine function. These have been based on trigonometric approximation, on Taylor series expansion, on simplified 4^{th} degree polynomial, on Chebyshev polynomial or linear segments of unequal lengths.

2.2.2.1 ROM-based Look-UP Table(LUT) For $\pi/2$ SCMF

The simplest approach for $\pi/2$ SCMF consists of using a ROM-based Look-UP Table(LUT). Its main disadvantage is that the ROM size grows exponentially with the width of the phase accumulator. However, many ROM size reduction techniques have been described in the published paper [9][10][11]. Various angular decomposition methods have been proposed to further reduce memory requirements with LUT-based methods. The typical method for angular decomposition is Sunderland Architecture [12].

The modified Sunderland Architecture is based on simple trigonometry identities , the phase address of the quarter of sine wave is decomposed to $\theta = \alpha + \beta + \gamma$ with wordlength of variables α ->A, β ->B, γ ->C . In this way the 12 phase bits are divided into three 4 bit fractions such that α <1, $\beta < 2^{-4}$, $\gamma < 2^{-8}$. The desired sine function is given by:

$$\sin(\frac{\pi}{2}(\alpha+\beta+\gamma)) = \sin(\frac{\pi}{2}(\alpha+\beta)) \times \cos(\frac{\pi}{2}\gamma) + \cos(\frac{\pi}{2}(\alpha+\beta)) \times \sin(\frac{\pi}{2}\gamma) \tag{2-6}$$

Given the relative sizes of α , β and γ ,this expression may be approximated by :

$$\sin(\frac{\pi}{2}(\alpha+\beta+\gamma)) \approx \sin(\frac{\pi}{2}(\alpha+\beta)) + \cos(\frac{\pi}{2}\alpha) \times \sin(\frac{\pi}{2}\gamma) \tag{2-7}$$

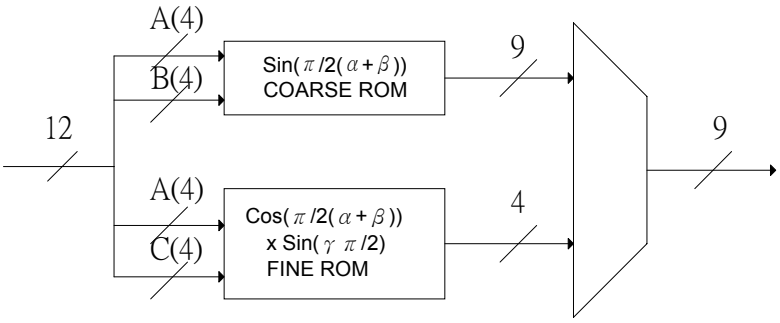


Fig 2-4 Block diagram of modified Sunderland architecture quarter sin function

Fig 2-4 the coarse ROM provides low-resolution samples, and the fine ROM gives additional resolution by interpolating between the low-resolution samples

An alternative methodology for choosing the samples to be stored in the ROM is based on the numerical optimization [13], The phase address of quarter of sine wave is defined as $\alpha+\beta+\gamma$ with word-length of the variable α to be A , the word-length of β to be B, and γ to be C . The variables α , β form the coarse ROM address and the variable α , γ form the fine ROM, the fine ROM samples are chosen to be the difference, the function is divided into four regions, corresponding to α = 00,01,10 and 11. Within each region, only one interpolation value may be used for all the same γ values. The interpolation value used for each value of γ is chosen to minimize either the mean square or the maximum absolute error of the interpolation within the region [13].

Exploiting the symmetry in the fine ROM correction factors provides further storage compression. If the coarse ROM samples are chosen in the middle of the interpolation region , the fine ROM samples will be approximately symmetry around the γ =$(2^C-1)/2$ axis. Thus , by using an adder/subtractor instead of an adder to sum the coarse and fine ROM values, the size of fine ROM may be halved. Some additional complexity must be added to the adder/subtractor control logic if this technology is used with Initial guess/correction method. Since the fine ROM is generally not in the critical speed path , the effective resolution of the fine ROM may be doubled, rather than halving the fine ROM. It allows the segmentation of the compression algorithm to be changed, effectively adding an extra bit of phase resolution to the look-up table, which thereby reduces the

magnitude of the worst-case spur due to phase accumulator truncation.

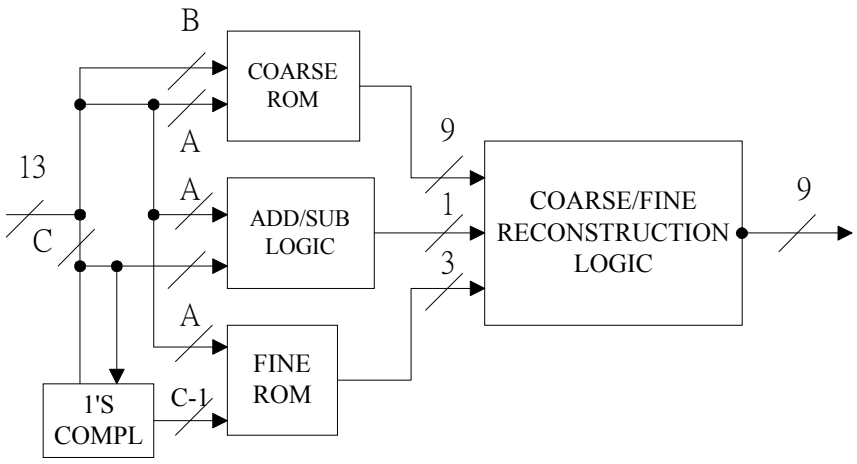


Fig 2-5 The sine function generation of Nicholas's architecture

2.2.2.2 Computational method

Instead of ROM LUT, some computational methods are depicted in some paper [6][7][8], The Taylor and Chebyshev [7] series approximation are also used in DDFS, using either Taylor and Chebyshev approximation in its basic form requires heavy computation and the direct utilization of series approximation in phase-to-amplitude conversion is infeasible. However, the amount of computation can be reduced by using the commonly known summery of sin and cosine signals [14]. In order to achieve a high operation clock frequency, pipeline stages were added to the phase-to-amplitude conversion logic, all the multipliers and squarers were implemented using carry save arithmetic, for further speed up the design, the final addition of the sum and carry vectors was put on a separate pipeline stage. So the computational complexity required by the Taylor and Chebyshev approach is significantly.

Langlois [6] propose to use a hardware-optimized SCMF that approximates the first quadrant of sine function with eight equal-length piecewise line segments, no multiplication was required. Where:

$$S_i(x) = m_i * (x - \frac{i}{8}) + y_i \qquad , i \in [0,7] \tag{2-8}$$

He carefully select the slope of each segment to eliminate the requirement for multiplication by representing each one as a sum of at most two powers of two, also restrict the precision of the slope representation and select equal-length linear segments to reduce control circuitry costs

Mohieldin's [8] proposed architecture is based on the idea of breaking the sine function into linear segments, for a given number of segments, the segments' slopes (k_0,k_1,….) are chosen to minimize the integrated mean square error between the ideal sin(θ) and the approximate piecewise linear P(θ) curves. In order to simplify the implementation of such approximation, the number of segments is chosen to be in power of 2. The point θ i are selected to be equally spaced to further simplify the design. Given the number of piecewise segments, the slope values yielding minimum mean square error (MMSE) are determined where MMSE is expressed as

$$MMSE = \min \int_0^{\theta=\pi/2} [\sin(\theta) - P(\theta)]^2 \, d\theta \qquad (2\text{-}9)$$

For the computation methods, it has been demonstrated that it can achieve ROM-less goal at expense of increasing in hardware complexity, this hardware overhead can be justified for high-speed applications where the large ROM size can represent a bottleneck. For the low-power applications, the reduction in power through ROM size reduction has to be weighed against the increase in overhead power consumption, so as to determined the most suitable compression technique.

2.2.2.3 Initial guess/correction method

In which an initial guess for the sine function is generated by digital hardware and then is corrected by small ROM lookup table, containing the difference of the initial guess and accurate value for sine amplitude, the simplest form of initial guess is to use a straight line as the initial guess for the first quarter-period of sine function, called the sine-phase difference method. In this methods, a straight line represented as y(θ) = 2θ /

π for $0<\theta<\pi/2$,which is proportion to the output of the phase accumulator, is taken as initial guess. Then , a ROM lookup table, which contains the difference between $\sin(\theta)$ and initial guess $2\theta/\pi$,is used to yield an accurate sinusoid. Since $\max[\sin(\theta)- 2\theta/\pi] = 0.21\max[\sin(\theta)]$ for $0<\theta<\pi/2$, this save 2 bits of memory word length.

Another similar work of this category is a double trigonometric approximation, devised by Yamagishi [15] which leads to a memory width reduction of 3 bits, In this approximation, the initial guess for sin amplitude will need less correction, and hence more memory compression will occur. In [16], this paper describes same idea in sine-amplitude approximation called parabolic approximation, a parabola whose maximum and x-axis intersections are the same as that of the sine half period is generated by digital hardware and is considered as the initial guess for the sine function. Such a parabola is expressed as $y(\theta) = (4\theta/\pi)(1-\theta/\pi)$ for $0<\theta<\pi/2$.A comparison between a quarter-period of a sinusoid and approximation error for sine-phase difference method and that of parabolic is $\max[\sin(\theta)- (4\theta/\pi)(1-\theta/\pi)] = 0.056\max[\sin(\theta)]$ for $0<\theta<\pi/2$, parabolic approximation will save 4 bits of memory word length.

Langlois [9] use the straight-line to approximate sine function; the sine function can be approximated with n straight-line segments of the form m_ix+b_i :

$$y(x)=\begin{cases} m_0x+b_0+\varepsilon(x), x_0\le x\le x_1 \\ m_1x+b_1+\varepsilon(x), x_1\le x\le x_2 \\ \cdots\cdots\cdots\cdots \\ m_{n-1}x+b_{n-1}+\varepsilon(x), x_{n-1}\le x\le x_n \end{cases} \qquad (2\text{-}10)$$

Detail calculation can be seen in [9], The concept of this technique is to select n straight lines defined by coefficients m_i and b_i that make points along these straight lines easy to calculate. The coefficients can be selected to bound the value of ε (x) to a maximum such that a desired number of amplitude bits in the ROM are saved. For example, to save 1 bit of storage , the maximum value of ε (x) must less than 0.5.For 2 bits, it must be less than 0.25 and for k bits, less than 2^k .

Soudris [10] choose the new function y(x) to be the (a+b+c)/2, the function y(x) approximates the sin(x) in a good level, so the dynamic range of sin(x) is reduced significantly. Continuing further, it is possible to reduce the dynamic range of derived function sin (x)- (a+b+c)/2 even more by taking advantage of its symmetry, this can be realized by generating the rectilinear segments L1, L2, L3, L4. These segments are implemented by reusing the function (a+b+c)/2 with proper slope faction. so as to approximate the function sin(x)-(a+b+c)/2 in a closer level than the previous case ,By storing, in the coarse ROM table , the samples of the difference between the above function and the value of segment at each point , he can reduce more the memory size according to the previous ascertainment.

2.3 Spur noise of DDFS

The are three sources of noise which are inherent to all DDFS implementations, in addition to the noise generated in the D/A conversion process. The first source of noise is P(n),The distortion due to phase truncation at the input to the sine function computing hardware(SCMF), The second is g(*),which is a nonlinear distortion that is usually present when compressing the storage requirements of the look-up table or approximation method are employed, and the third is A(n),the noise introduced by finite precision of the sine samples generated from SCMF. These noise sources are depicted symbolically in Fig2-6

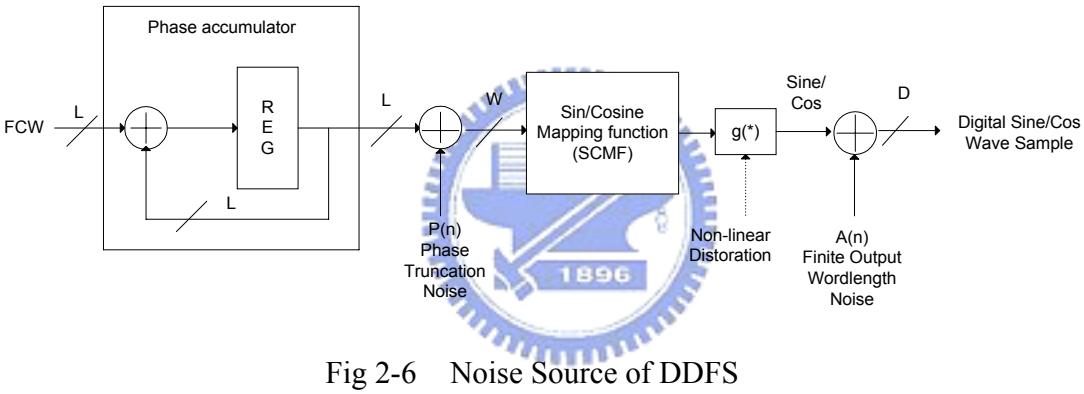


Fig 2-6 Noise Source of DDFS

2.3.1 Output Spectrum of DDFS in the present of Phase truncation

In order to limit the complexity of phase to amplitude such that it can be implemented with reasonable amount of hardware, the phase accumulator output is typically truncated before being fed into SCMF, as shown in Fig2-6. The phase truncation is known to manifest itself as a set of spurious frequencies at the outputs. The first rigorous mathematical treatment for determining the magnitude and spectral location of spurious frequencies (often called “spurs”) was attempted in 1987 by Nicholas and Samueli [18] and resulted in a rather complicated algorithm. In 1993 Kroupa [19] present a simpler algorithm for estimation of spur magnitudes with the introduction of approximation.

The source P(n) is due to the truncation of the phase accumulator bits addressing the sine ROM. Since the amount of memory require to encode the entire width of the phase accumulator would be usually prohibitive, only W of the most significant bits of the accumulator are generally used to calculate the sine-wave samples. The worse case carrier-to-spur ratio due to the phase truncation is [18]

$$\frac{C}{S} = 2^{w} Sinc(\frac{GCD(FCW, 2^{L-W})}{2^{L-W}}), \qquad \text{If} \quad \text{GCD}(\text{FCW},2^{\text{W}})<2^{\text{L-W}} \qquad (2\text{-}11)$$

The phase truncation occurs only, when GCD(FCW,2^{W}) is small than 2^{L-W}, If GCD(FCW,2^{W}) is equal or greater than 2^{L-W}, the phase bits are zeros below 2^{L-W}, no phase error occurs.

Modificated Nicholas Phase Accumulator, This method does not destroy the periodicity of the error sequences, but it spreads the spur power into many spur peaks [18]. If GCD(FCW,2^{L-W}) is equal 2^{L-W-1} in (2-11), the spur power is concentrated in one peak. Then the worst case carrier-to spur ratio is from (2-11)

$$\frac{C}{S} = (6.02W - 3.992) \quad \text{dBC} \qquad (2\text{-}12)$$

Where W is wordlength of phase accumulator output used to address the ROM. If GCD(FCW,2^{L-W}) is equal to 1, the spur power is spread over many peaks. The carrier-to-spur ratio is approximately from

$$\frac{C}{S} = 6.02W \quad \text{dBC} \qquad (2\text{-}13)$$

Compare (2-12) and (2-13) show that the worst case spur can be reduced in magnitude by 3.992 dB by forcing GCD(FCW,2^{L-W}) to be unity, i.e., by forcing the frequency control word to be relatively prime to 2^{L-W}. This causes the phase accumulator output sequence to have a maximal numerical period for all value of FCW, all possible value of phase accumulator output sequence are generate, before any value are repeated.

Arthur Torosyan [17] present simple algorithm for calculation of the output spectrum of DDFS in the presence of phase accumulator truncation in 2001. By using no approximations, a simple formula calculates the magnitudes of the spurious noise frequency (spur) due to phase truncation, their spectral position is also determined by a simple expression. The derivation process itself provides strong insight into spur magnitude and spectral location and it makes evident that the set of spurs due to phase word truncation and the set resulting from finite arithmetic precision are disjoint. Briefly description is as follows.

For Torosyan's discussion of phase word truncation, first assume the SCMF is ideal (with infinite precision output). Let f be an L-bit frequency control word with the rightmost non-zero LSB located in at the W +B position from the MSB. For example, if L = 24, then for W = 15 and B =5 the first non-zero LSB will be at position 20. The word f = 010011011011110101110000 is an example of such a frequency control word, where underlined bits correspond to phase word bit positions that address the SCMF., the set of spurious lines generated by the frequency control word f = 010011011011110101110000 is identical to the set of spurious lines generated by g = 000000000000000000010000(identical in the number and amplitude of spurs). If the frequency control word is g = 000000000000000000010000 and the phase accumulator output is truncated to W = 15 bits before addressing the SCMF, the SCMF input will step once every $2^{(20-15)} = 32$ cycles. Since the SCMF simply outputs the sin and cos values for input phase, the same "redundant" behavior can be observed at the output sin (n) and cos(n) of the DDFS. Notice that SCMF will go through all possible $2^{15} = 32768$ inputs and their corresponding outputs, each repeated 32 times, before repeating the cycle again.

$$S(k) = (1 + e^{-j\frac{2\pi}{2^{20}}k} + ... + e^{-j\frac{2\pi}{2^{20}}31k}) \sum_{n=0}^{2^{15-1}} s(32n) e^{-j\frac{2\pi}{2^{15}}nk}$$

$$= (1 + e^{-j\frac{2\pi}{2^{20}}k} + e^{-j\frac{2\pi}{2^{20}}2k} + ... + e^{-j\frac{2\pi}{2^{20}}31k}) S'(k)$$

For $0 \leq k < 2^{20}$

Where S'(k) is the 2^{15} –point DFT of non-redundant subsequence s'(n) = s(32n). Thus, by summing the finite geometric series, the derivative of the following expression and is depicted in paper of Torosyan [17]:

$$S(k) = V(k)S'(k) \tag{2-14}$$

$$S'_W(k) = \sum_{n=0}^{2^{W-1}} s'(n) e^{-j\frac{2\pi}{2^W}nk} \qquad k = 0,1,\ldots\ldots,2^{W+B}-1$$

$$V_{W,B}(k) = \frac{1-e^{-j\frac{2\pi}{2^W}k}}{1-e^{-j\frac{2\pi}{2^{W+B}}k}} \qquad k = 0,1,\ldots\ldots,2^{W+B}-1$$

$$S_{W,B}(k) = V_{W,B}(k)S'_W(k) \qquad k = 0,1,\ldots\ldots,2^{W+B}-1$$

(2-15)

Where $S'_W(k)$ is periodic in k with period 2^W and one period of $V_{W,B}$ window over 2^B periods of $S'_W(k)$. Since we know the locations of deltas in $S'_W(k)$ not only do we know the locations of the spurious lines created from phase word, but we also know the exact spurious noise line magnitudes since we can evaluate the windowing function V for values of k corresponding to the location of the deltas in $S'_W(k)$. More precisely, the amplitude of the spurious frequency at k, relative to that of the desired sinusoid, is $\left|\frac{V_{W,B}(k)}{V_{W,B}(1)}\right|$. That is in dB, the spur magnitude at k will be:

$$20\log_{10}\left|V_{W,B}(k)\right| - 20\log_{10}\left|V_{W,B}(1)\right| \tag{2-16}$$

2.3.2 Output Spectrum of DDFS in the present of quantization

Finite output word length also leads to the DDFS output spectrum impairment, if it is assumed that the phase truncation does not exist, then the output of the DDFS is given by $s'(n) = \sin(\frac{2\pi n}{2^W}) + A(n)$, where A(n) is the quantization error due to finite output word length D, the output resolution, is an important design parameter, because it has a large effect in the accuracy parameter of the synthesizer. In that respect, it is necessary to define the main accuracy metrics typically used for DDFS. A straightforward method to characterize the accuracy of the synthesized signal is to measure the maximum absolute error (MAE) of actual output valued with respect to the theoretical one.

MAE represents the bound of the error that occurs in the generation of each output sample. Therefore a system performs with MAE approaching the null value exhibits a good performance also with respect to any other accuracy parameters.

Another accuracy metric of a DDFS is the mean square error (MSE). This parameter measures the mean power of the output quantization noise, independently from the spectral quality of the noise itself. Also for MSE we can state a theoretical limit of $LSB^2/12$, obtained realizing the DDFS as an ideal quantizer following a sine generator. If we evaluate the ratio between signal power and MSE, we obtain another accuracy parameter, the signal to noise ratio (SNR), which is typically expressed in dB (SNR_{dB}). Obviously, the maximum SNR of a DDFS is related to the output resolution by the following equations [20]:

$$SNR = \frac{(2^{D-1}-1)^2 LSB^2}{2 * MSE}$$

$$SNR_{dB} = 6.02D + 1.76 \text{ dB} \qquad (2\text{-}17)$$

That, for example, leads to SNRdB = 55.94 dB for a system with an output resolution of D = 9 bits.

2.3.3.Spurious Free Dynamic Range (SDFR)

In Bluetooth, within the ISM band the transmitter shall pass a spectrum mask, that is, in-band spurious emission requirement that is shown in Fig. 2-7(a). The transmitted power shall be measured in a 100 kHz bandwidth using maximum hold. Moreover, the out-of-band spurious emission requirement is shown in Fig. 2-7(b).

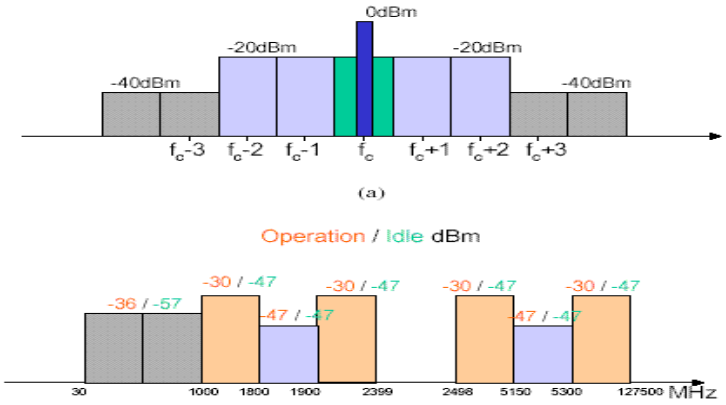


Fig. 2-7 (a) Bluetooth In-Band Spurious Emission Requirement
(b) Bluetooth Out-of-Band Spurious Emission Requirement

In IEEE 802.11b WLAN system, the transmit spectral mask is shown in Fig. 2-8 and the measurements shall be made using a 100kHz resolution bandwidth

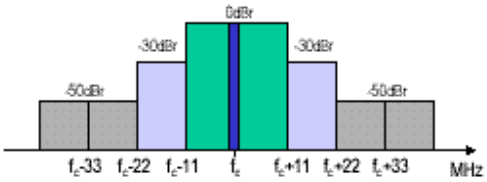


Fig. 2-8 IEEE 802.11b WLAN In-Band Spurious Emission Requirement

In IEEE 802.11a the transmitted spectrum shall have a 0 dBr (dB relative to the maximum spectral density of the signal) bandwidth not exceeding 18 MHz, –20 dBr at 11 MHz frequency offset, –28 dBr at 20 MHz frequency offset and –40 dBr at 30 MHz frequency offset and above. The transmitted spectral density of the transmitted signal shall fall within the spectral mask, as shown in Figure 2-9. The measurements shall be

made using a 100 kHz resolution bandwidth and a 30 kHz video bandwidth.

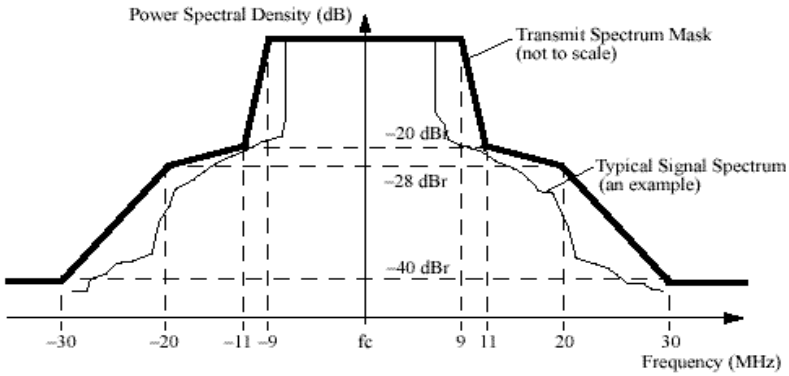


Fig 2-9 Transmit spectrum mask

CHAPTER 3
DDFS Architecture and Verification

In this chapter, the spur items caused by finite output word length, phase truncation and sine/cosine mapping function (SCMF) are discussed. Using the MATLAB to simulate DDFS performance. DDFS architecture and algorithmic approximation is described and verified by verilog code, Matlab is used to calculate SFDR of DDFS with output of ModelSim.

3.1 Spur of Finite Output Word

Accuracy metric of a DDFS is the mean square error (MSE). The signal to noise ratio (SNR), which is typically expressed in dB (SNR_{dB}). Using MATLAB to simulate the finite output word length and the SNR result are shown in Fig 3-1. From the result of simulation, Formula (2-17) can achieve good approximation for SNR of finite output word.

For many applications, the most important accuracy parameter in a DDFS is the spectral purity, often referred as spurious free dynamic range (SFDR), which is defined as the ratio (in dBc) between the fundamental and the largest spurious amplitude in the spectrum of the generated output. Fig 3-2 is the simulation result of MATLAB program, from Fig 3-1 and Fig 3-2, it shows good values SNR generally lead to good values of SFDR too.

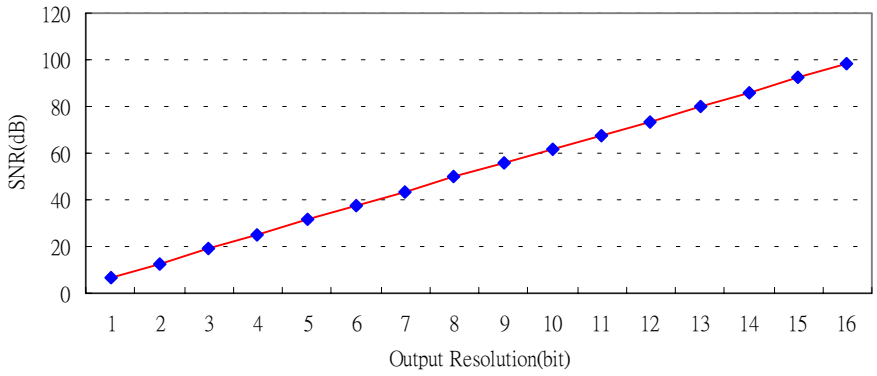


Fig 3-1 SNR(dB) for different output bit length

For the same condition of simulation, it is reasonable for value of SFDR larger than SNR at same bit of quantizer.

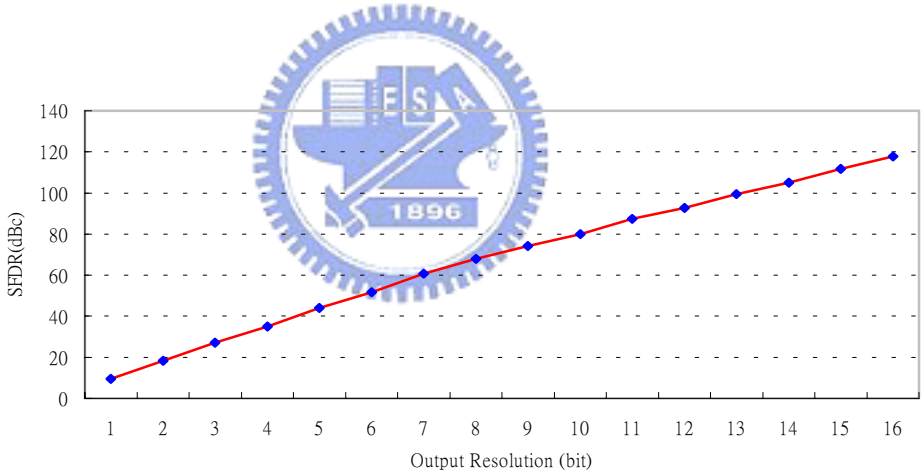


Fig 3-2 SFDR(dBc) for different output bit length

3.2. Spur of phase truncation

Torosyan presents simple algorithm for calculation of the output spectrum of DDFS in the presence of phase accumulator truncation. By using no approximations, a simple formula calculates the magnitudes of the spurious noise frequency (spur) due to phase truncation, their spectral position is also determined by a simple expression. The derivation process itself provides strong insight into spur magnitude and spectral location and it makes evident that the set of spurs due to phase word truncation and the set resulting from finite arithmetic precision are disjoint.

For a DDFS with an L-bit frequency control word and W-bit input SCMF, Matlab program is written to simulate spurious noise of phase truncation based on conclusion of Torosyan (L-bit frequency control word is truncated to be W-bit input SCMF, bits of truncation B = L-W), The following is algorithm for Matlab programming.

(1). Calculate $S'_W(k)$, the 2^W-point of DFT of the SCMF output corresponding to the frequency control word having a single non-zero bit at the position W, and where k denotes the bin number (or frequency index).

(2). For $1 \le B \le (L-W)$ and $V_{W,B}$ given in (2-15), evaluate (2-16) for $k = \{d\,(2^W \pm 1)$; for $1 \le d \le (2^B-1)\}$ to obtain the magnitudes of the "phase truncation" spurs for any frequency control word with its rightmost non-zero bit at position W+B. The entire DDFS output spectrum will be obtained from $S_{W,B}(k) = V_{W,B}(k)S'_W(k)$.

Spurious noise for phase truncation bit based on Torosyan's formula is shown in Fig 3-3. The spur caused by phase truncation is simulated by Matlab program and the result is shown in Fig3-4, two results are consistent, so it is good for algorithm of Torosyan to calculation of the output spectrum of DDFS in the presence of phase accumulator truncation.

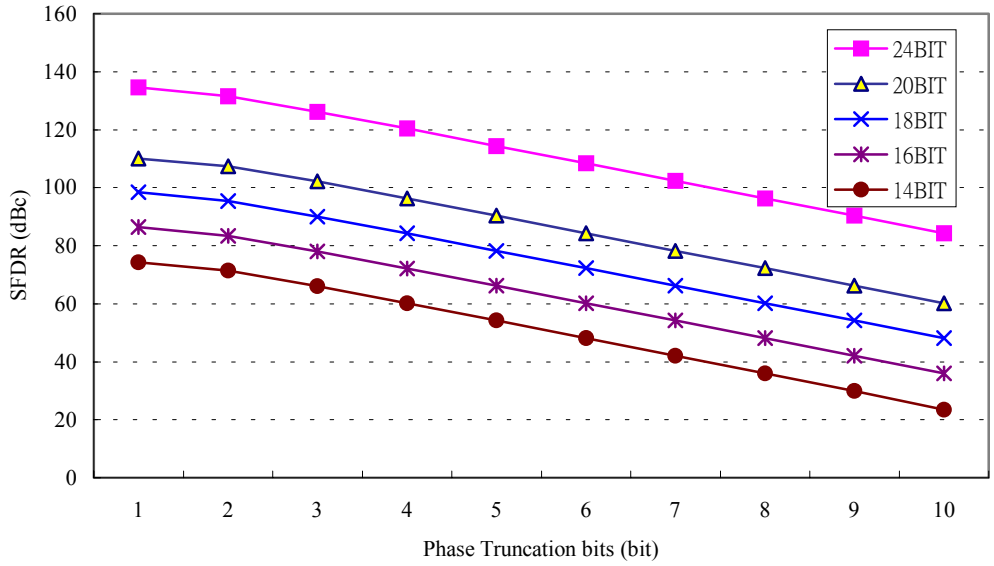


Fig 3-3 SFDR (dBc) for phase truncation bits (based on conclusion of Torosyan)

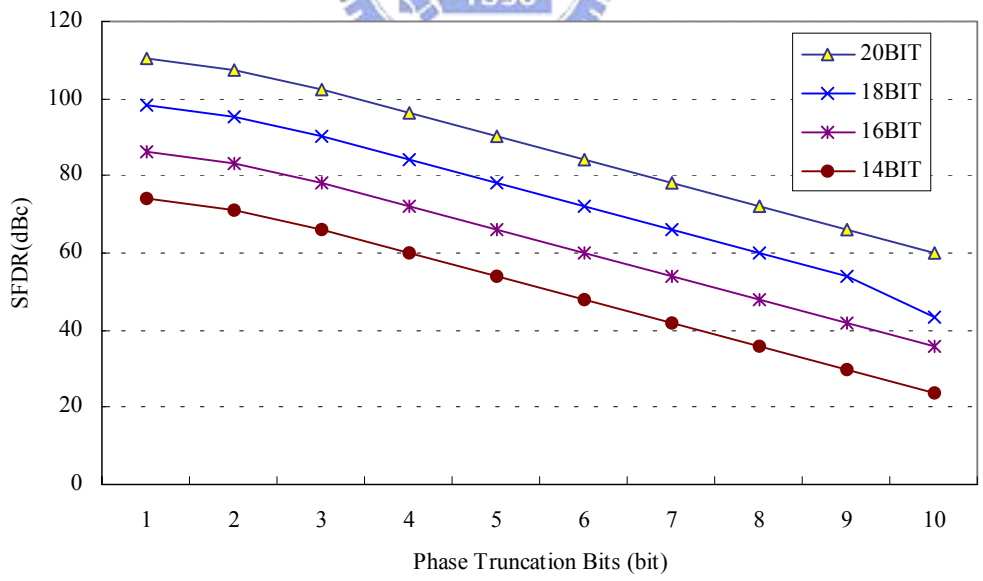


Fig 3-4 SFDR (dBc) for phase truncation bits (based on simulation of MATLAB)

3.3 Sine/Cosine mapping function (SCMF)

Initial guess /correction methods, in which an initial guess for the sine function is generated by digital hardware and then is corrected by a small ROM look-up table, containing the difference between the initial guess and the accurate value for the sine amplitude. These methods require smaller memories, and hence will be faster and consume less power. Because of their lower hardware complexity in comparison to computational methods. Hence the initial guess and error correct ROM are used to approximate the sine function.

Fig 3-5 is the DDFS based on sine-phase difference algorithm, the auxiliary function f(x) (x= A+B+C)

$$f(x)=\begin{cases}(A+B+C)+(A+B+C)/4 & 0\le\theta<\pi/4\\ (A+B+C)+0.25-(A+B+C)/4 & \pi/4\le\theta<\pi/2\end{cases} \quad (3\text{-}1)$$

Using 3rd MSB of phase to choice which segment of curve B (shown in Fig3-6) will be used, for example 3rd MSB = 0 for (A+B+C)/4 and 3rd MSB =1 for 0.25- (A+B+C)/4.

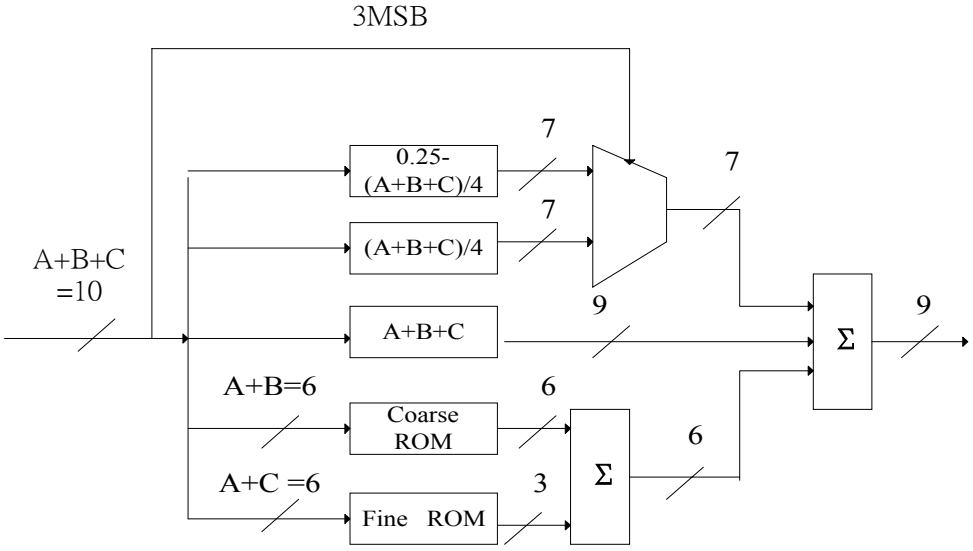


Fig 3-5 The DDFS architecture (sine-phase difference Algorithm)

Auxiliary function f(x) ;that is approximated function is shown in Fig 3-6 , curve A is first approximation function (A+B+C) and curve B is second approximation function by using two segment lines , (A+B+C)/4 for $0 < \theta < \pi/4$, 0.25-(A+B+C)/4 for $\pi/4 < \theta < \pi/2$, curve A+B combines first and second approximation function for quarter quadrant.

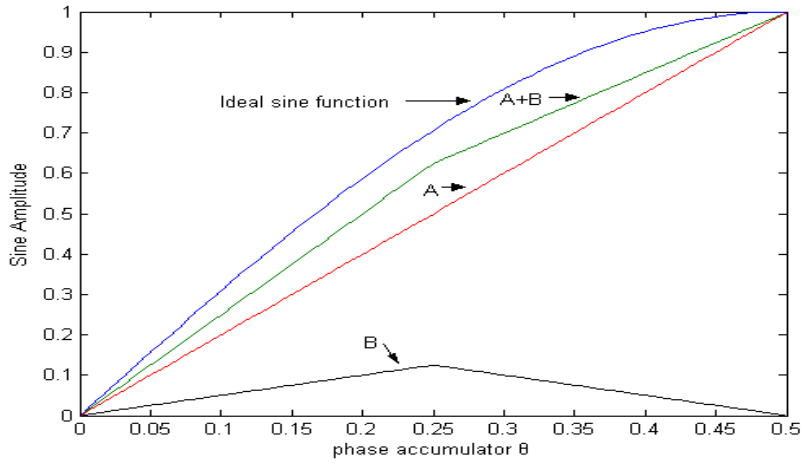


Fig 3-6 Initial guesses techniques using 2-segment approximation

A complete period of the approximation function f(x) and the ideal reference sinusoid are shown in Fig 3-7

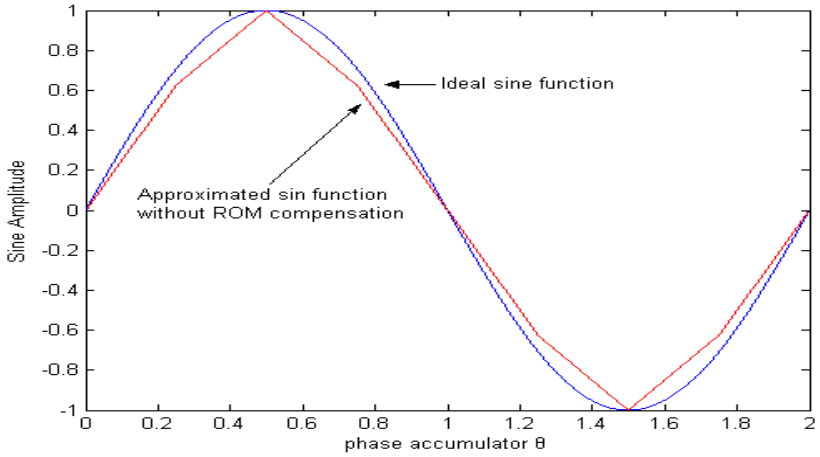


Fig 3-7 Approximated sine function without error correct ROM

Error between sine wave and approximated function f(x) is shown in Fig 3-8.

$$\max[\sin(\theta)-f(x)] = 0.1163 \max[\sin(\theta)] \qquad \text{for } 0 < \theta < \pi/2 \qquad (3\text{-}2)$$

The maximum error is 0.1163 less than 0.125, which saves 3 bits of ROM output, A more detailed study on the errors, depicted in Fig3-8, shows that the approximated function is so chosen that the approximation error is entirely positive.

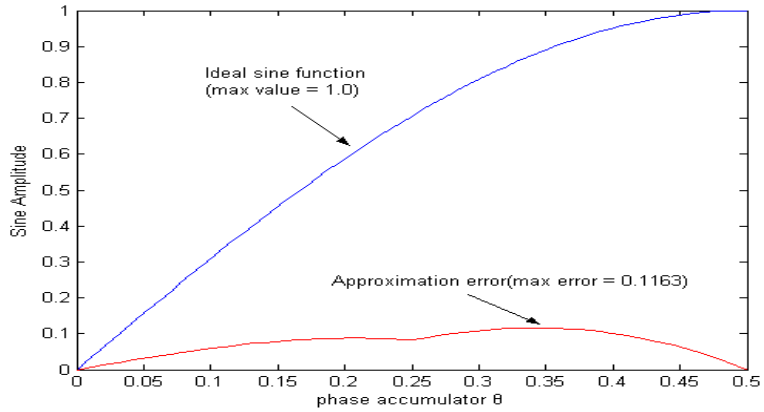


Fig 3-8 Error between sine wave and initial guess function

From a different viewpoint, the likelihood of the approximation to an ideal sinusoid can be judged from its power spectrum. Harmonic levels without error correction are shown in Fig 3-9. Maximum power levels is -27.93 dBc, As can be seen, only odd harmonics have appeared in the spectrum, and their levels are such that they can be neglected in some general applications. However, for precise systems, they should be pushed down by using an error correcting ROM lookup table.

In order to reduce the ROM size, the ROM memory is partitioned into two ROM blocks. The first ROM (coarse ROM) presents the total ROM in less address than the original and the second one can use a form of linear interpolation (Fine ROM). The sum of coarse and fine ROM compensate the error

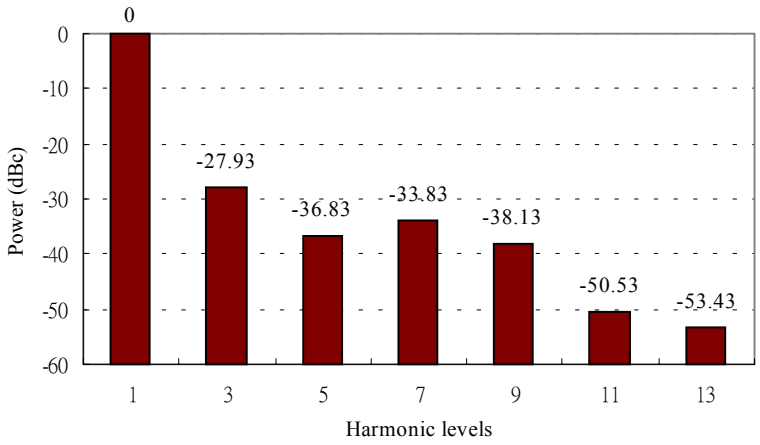


Fig 3-9 Harmonic levels without error correcting ROM

Between the approximation function f (x) and ideal sin function. The coarse ROM output word length may be reduced and this method reduces the lookup table storage requirements at least of 2^{A+B+C} bit totally. The drawback of this reduction of ROM is that another adder/subtract is required

The phase address of quarter sine wave is decomposed to θ = a+b+c with the word length of the variables : a = A, b = B, c = C .The variables A,B form the coarse ROM address, and the variables A,C form the fine ROM address The sine function between [0, π /2] is divided into 2^A different regions . In each region we define 2^B points, which are the samples that stored in coarse ROM by taking minimum error value of coarse interval (2^{A+B} coarse intervals). The fine ROM samples are chosen to be average error value of entire fine interval (2^{A+C} fine intervals), the fine error value is directly below that point. Within each coarse region (same A , different B), only one sets fine correction error value were used for different 2^B coarse interval. Using ROM splitting under sacrificing the performance of SFDR can reduce ROM size.

MATLAB program is used for partitioning ratio study. Furthermore, It was determined by simulating SFDR and ROM size measurements that the segmentations of the address word to the coarse and fine ROM have values of A, B and C. As the value of A increases, the ROM size is increased proportionally (Fig 3-11), Upon to these constraints, the spurious responses of the possible segmentations were simulated (Fig 3-10), the results show the size of the memory plotted against to the word length variable

A, B, C (Fig 3-11).

(A, B, C)	SFDR (dBc)	Max Coarse Error	Max Fine Error	ROM Size (Bits)	Compression Ratio	Remark
(2, 3, 5)	58.71	0.1141	0.0134	576	71.1	2^5x6+2^7x3 = 576
(2, 5, 3)	68.55	0.1141	0.0044	832	49.2	2^7x6+2^5x2 = 832
(2, 4, 4)	66.84	0.1145	0.0078	576	71.1	2^6x6+2^6x3 = 576
(3, 2, 5)	66.11	0.1141	0.019	1216	33.7	2^5x6+2^8x4 = 1216
(3, 3, 4)	66.34	0.1141	0.0087	768	53.3	2^6x6+2^7x3 = 768
(3, 5, 2)	68.33	0.1141	0.0017	1568	26.1	2^8x6+2^5x1 = 1567
(4, 2, 4)	70.52	0.1141	0.0097	1152	27.8	2^6x6+2^8x3 = 1152
(4, 3, 3)	70.98	0.1141	0.0055	1024	40	2^7x6+2^7x2 = 1024
(4, 4, 2)	68.35	0.1151	0.0027	1600	25.6.3	2^8x6+2^6x1 = 1600
(5, 1, 4)	69.03	0.1141	0.0112	1920	21.3	2^6x6+2^9x3 = 1920
(5, 2, 3)	71.31	0.1141	0.0057	1280	32	2^7x6+2^8x2 = 1280
(5, 3, 2)	68.36	0.1151	0.0028	1664	24.6	2^8x6+2^7x1 = 1664
(5, 4, 1)	66.95	0.1156	0.0019	3072	13.3	2^9x6+2^6x0 = 3072

Table 3-1 the simulation result for different phase segmentations

Table 3-1 is the simulation result for different phase segmentations and simulation condition is FCW = 1, Phase to Amplitude bit = 12 bits, no phase truncation and amplitude quantization bits is 10, that is, amplitude output bit is 10 bits. When A = 2, B=4 and C=4, the SFDR is 66.8 dBc, requirement of total ROM size (coarse ROM size 384 bits and fine Rom size 192 bits) is 576 bits. From the simulation result, even we use larger ROM size, SFDR performance is limited by the Rom splitting and quantization, given a certain spectrum specification, the designer has the ability to choose among alternative implementations meeting the design requirement.

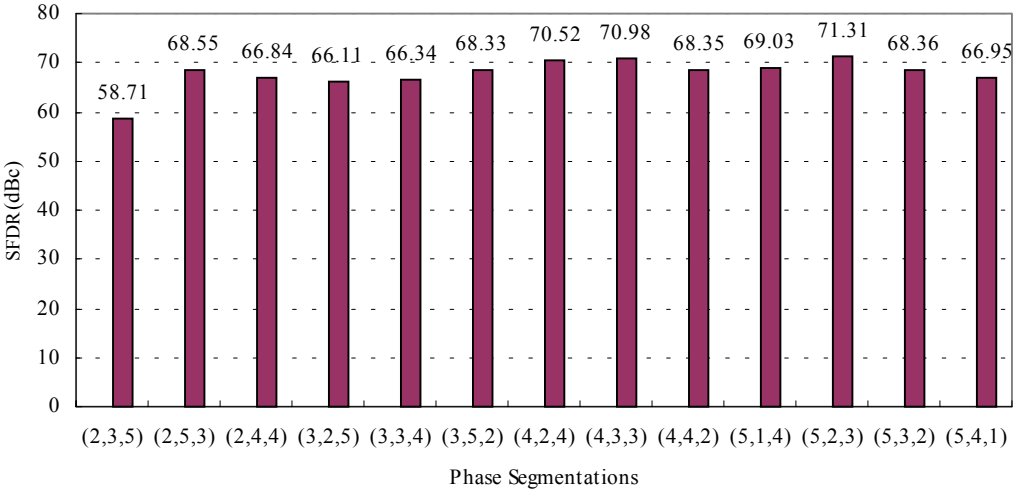


Fig 3-10 Worse Case Spurious Response for partitioning ratio

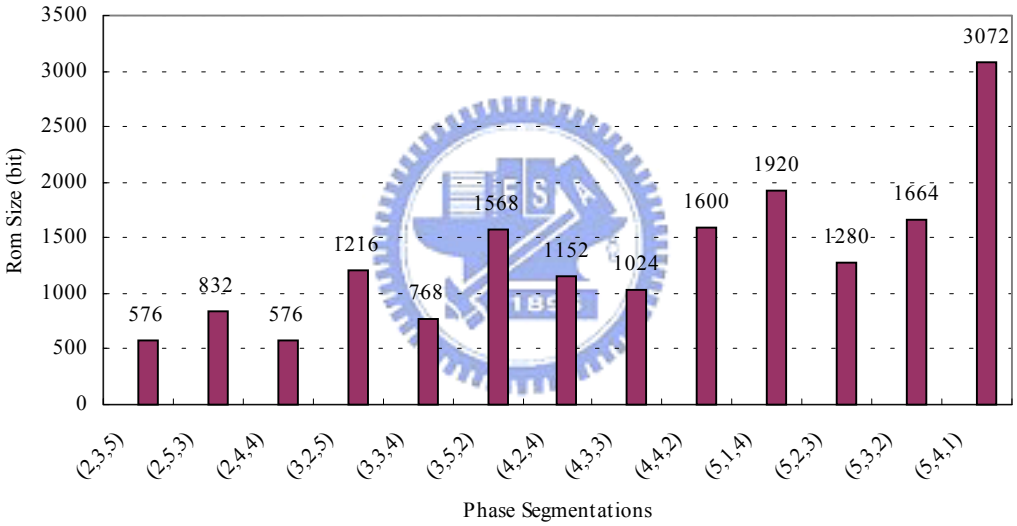


Fig 3-11 ROM Size for partitioning ratio

However, SFDR 66.8 dBc can meet our requirement in the further applications. Proposed DDFS based on sine-phase difference algorithm with ROM segmentation (A=2, B=4 and C=4) is shown in Fig 3-5. The auxiliary function f(x) A+B+C is simple 9-bit phase data and (A+B+C)/4, 0.25- (A+B+C)/4 are 7 bit phase data, The DDFS architecture uses adder blocks for phase data add, further reduction in the size of the ROM is realized by a ROM, segmentation algorithm show in Fig 3-5, The 10-bit phase data of accumulator is divided into three parts. A = 2, B = 4, C = 4 ROM segmentation will be implemented in proposed DDFS. Coarse and fine error correct value are shown in table 3-2 & 3-3. The 2^{12}x10 sine samples are compressed into 2^{6}x6 coarse

samples and 2^6x3 fine samples, compressed ratio is 71:1.

2^A	2^B	Float Value	Int (6 bits)	2^A	2^B	Float Value	Int (6 bits)	2^A	2^B	Float Value	Int (6 bits)	2^A	2^B	Float Value	Int (6 bits)
1	1	0.000615	0	2	1	0.070602	36	3	1	0.081967	42	4	1	0.107484	55
	2	0.005624	3		2	0.073603	38		2	0.087344	45		2	0.104399	53
	3	0.010617	5		3	0.076359	39		3	0.092283	47		3	0.100747	52
	4	0.015578	8		4	0.078857	40		4	0.096774	50		4	0.096524	49
	5	0.020494	10		5	0.081082	42		5	0.100768	52		5	0.091725	47
	6	0.025350	13		6	0.083022	43		6	0.104218	53		6	0.086346	44
	7	0.030130	15		7	0.084663	43		7	0.107190	55		7	0.080383	41
	8	0.034820	18		8	0.085921	44		8	0.109675	56		8	0.073833	38
	9	0.039405	20		9	0.086804	44		9	0.111664	57		9	0.066693	34
	10	0.043872	22		10	0.087346	45		10	0.113150	58		10	0.058959	30
	11	0.048205	25		11	0.087494	45		11	0.114124	58		11	0.050630	26
	12	0.052390	27		12	0.087132	45		12	0.114578	59		12	0.041704	21
	13	0.056413	29		13	0.086388	44		13	0.114271	59		13	0.032178	16
	14	0.060260	31		14	0.085251	44		14	0.113396	57		14	0.022052	11
	15	0.063916	33		15	0.083710	43		15	0.111977	57		15	0.011324	6
	16	0.067368	34		16	0.080667	41		16	0.110008	56		16	0.000006	0

Table 3-2 Coarse error correct values ROM table

A complete period of approximation with ROM compensation is shown in Fig 3-12. From the table 3-3, fine ROM size can be further reduced to 128 bits by using two bits. Of course, it will degrade the performance of SFDR.

2^A	2^C	Float Value	Int (3 bits)	2^A	2^C	Float Value	Int (3 bits)	2^A	2^C	Float Value	Int (3 bits)	2^A	2^C	Float Value	Int (3 bits)
1	1	0.000446	0	2	1	0.000803	0	3	1	0.001551	1	4	1	0.007804	4
	2	0.000000	0		2	0.000134	0		2	0.000464	0		2	0.006166	3
	3	0.001507	1		3	0.001442	1		3	0.001327	1		3	0.006478	3
	4	0.001060	1		4	0.000860	0		4	0.000237	0		4	0.004836	2
	5	0.002567	1		5	0.002214	1		5	0.001159	1		5	0.005144	3
	6	0.002119	1		6	0.001592	1		6	0.000038	0		6	0.003496	2
	7	0.003624	2		7	0.002929	1		7	0.000954	0		7	0.003800	2
	8	0.001223	1		8	0.000326	0		8	0.001860	1		8	0.004101	2
	9	0.002727	1		9	0.001605	1		9	0.002700	1		9	0.004400	2
	10	0.002278	1		10	0.000910	0		10	0.001521	1		10	0.002743	1
	11	0.003781	2		11	0.002144	1		11	0.002306	1		11	0.003037	2
	12	0.003331	2		12	0.001379	1		12	0.001199	1		12	0.001376	1
	13	0.004834	2		13	0.002588	1		13	0.002043	1		13	0.001666	1
	14	0.004382	2		14	0.001920	1		14	0.000932	0		14	0.000000	0
	15	0.005884	3		15	0.003039	2		15	0.001772	1		15	0.000285	0
	16	0.003479	2		16	0.000588	0		16	0.002610	1		16	0.000568	0

Table 3-3 Fine error correct values ROM table

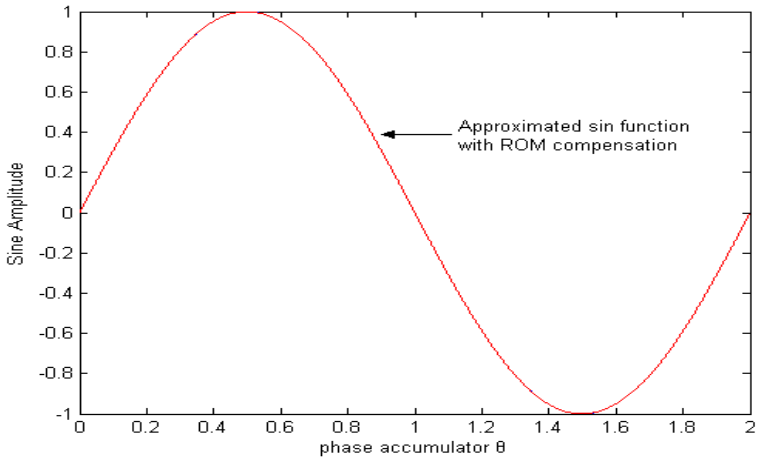


Fig 3-12 Approximated sin function with error correct ROM for one period.

3.4 Architecture of DDFS and Verification

The DDFS architecture of Fig 3-5 is simulated in system level, in order to verify feasibility of the idea. Then, the system was designed by digital circuitry and simulated by verilog code. Phase accumulator is 20 bits width input (FCW) and output of phase accumulator are truncated to 12 bits (12 MSB bits) and feed into SCMF block that convert from phase to amplitude.

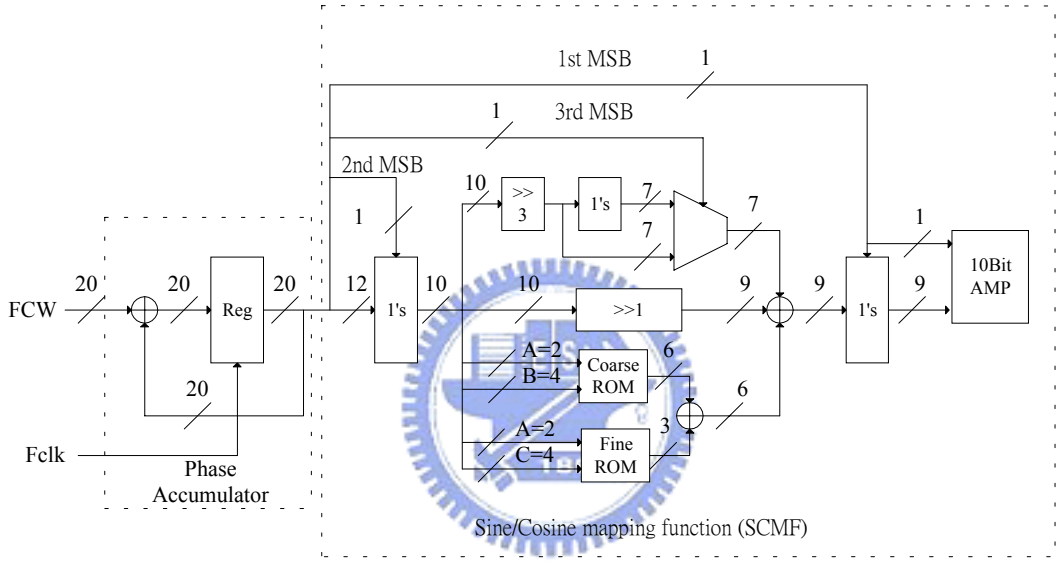


Fig 3-13 Proposed DDFS architecture

The SCMF converter block includes 2 sets of 1's complementary to exploit quarter-wave symmetry. First MSB and second MSB are used to full wave rectify the magnitude of the phase. The final output sine wave is generated by multiplying the full wave rectified version by –1 when the phase is between π and 2π (1st MSB = 1). This is accomplished simply by taking the amplitude bits (9bits) exclusive or with MSB of phase. MSB bit of phase is to be the sign bits, total 10 bits is for sine output amplitude. For the second and fourth quadrant, the phase bits should be complemented so the slope of the saw tooth is inverted, by taking 2^{nd}MSB of phase exclusive or with phase bit to determine whether the amplitude is increasing or decreasing. the full wave output can be recovered by inverting the phase and amplitude appropriately.

In Fig 3-13, the notation {>>N} signifies a right shift by n bits or division by 2^N.

Phase bits shift one bit right and rounds to 9 bits to form A+B+C, phase bits shift three bits right and rounds to 7 bits to be (A+B+C)/4 for phase between 0 and π /4, by taking the inverse bits of (A+B+C)/4 to be 0.25-A+B+C for phase between π /4 and π /2. Third MSB of phase bit determine which segment will be used, that is , 3rdMSB = 0 for phase 0 to π /4, 3rdMSB = 1 for phase π /4 to π /2 . Multiplexer is used for switching (A+B+C)/4 and 0.25-A+B+C under controlled by 3rdMSB phase bit.

Coarse and fine Rom are used as error correcting ROM lookup table. Individual Rom sizes are 384 bits for coarse Rom and 192 bits for fine Rom. Output bit width is 6 bits for coarse and 3 bits for fine Rom. The values of Rom are listed in table 2&3. All the parameters of DDFS were determined by the simulation result of Matlab described in previous section. Output of error corrects Rom (6 bits), the auxiliary function (9 bits) and MSB phase bit were summed to be the output of DDFS (10 bits)

Verilog code has been written to simulate the DDFS architectures. All simulations have been realized with ModelSim. Figure 3-14 show the output bits for frequency control word of h01000 and h04000 (FCW), or an output frequency equal to (h1000/hfffff) x F_{clk} and (h4000/hfffff) x F_{clk}. The simulation is presented in waveform tool of ModelSim. In Fig 3-14, from top to down, they are system clock (clk), system reset (rst), frequency control word (FCW), output of DDFS (10bits), address for coarse ROM (6 bits), output of coarse ROM (6 bits), address for fine ROM (6 bits), output of fine ROM (3 bits), truncated phase address (12 bits), output of coarse ROM (6 bits), output of fine ROM (3 bits), output of multiplex for switching (A+B+C)/4 and 0.25-A+B+C.

The output sinusoid amplitude from simulation of ModelSim is saved to file and use Matlab to do post-processing. The synthesized waveform and the digital phase sweep result from verilog code simulation are shown in Fig 3-15, three waveforms are shown in the figure, the waveforms are full sine amplitude, half sine amplitude and phase accumulation. When phase accumulation output MSB = 1(π to 2π), the full sine amplitude can be achieved from half sine amplitude by proceeding exclusive or with half amplitude(inverse the amplitude). When phase accumulation output 2ndMSB = 1(π /2 to

π or $3\pi/2$ to 2π), by taking 2nd MSB exclusive or with phase bits(inverse the phase), the half amplitude can be recovered from first quadrant. All the property is depicted in the Fig 3-15.

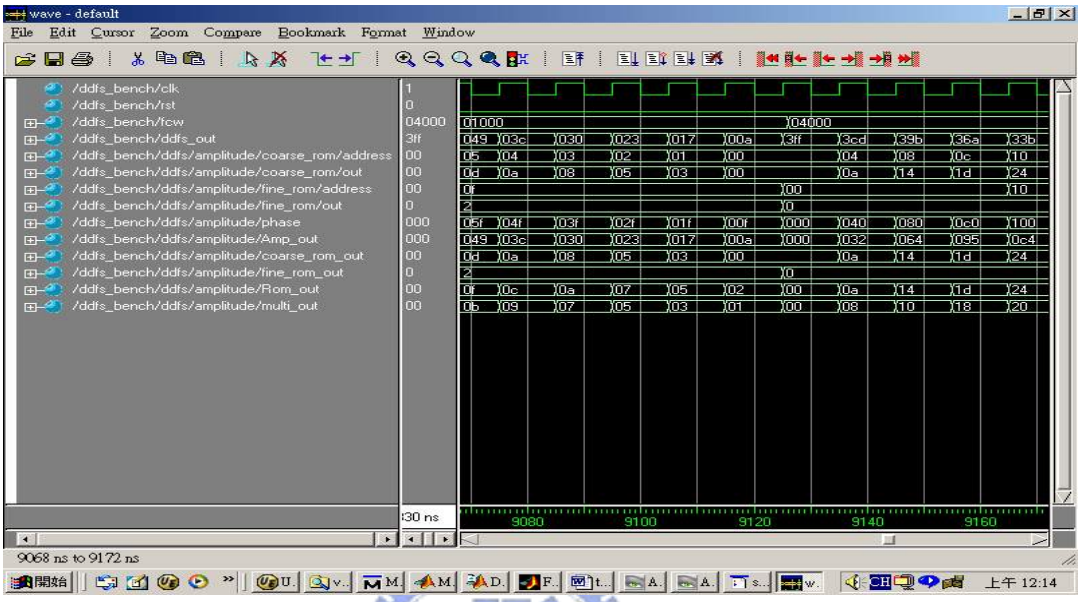


Fig 3-14 Input frequency control word and output of DDFS

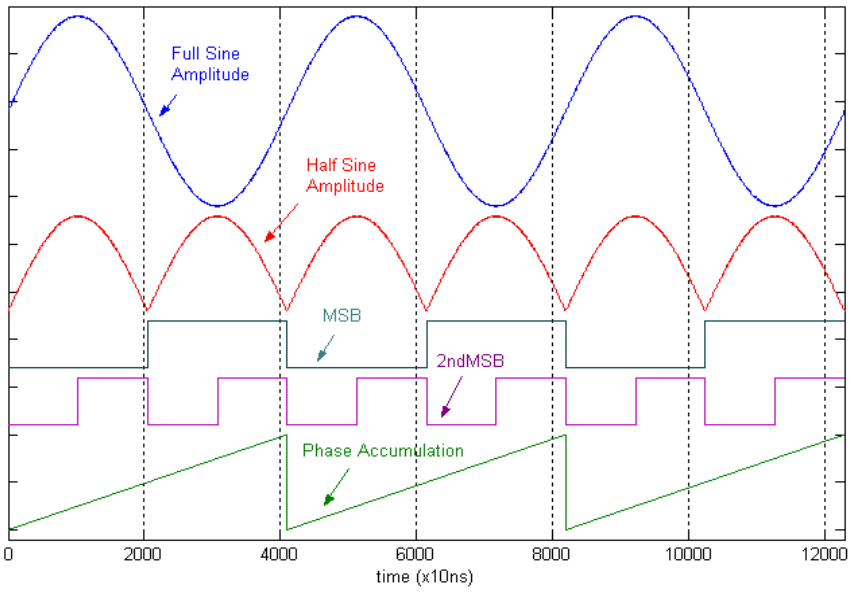
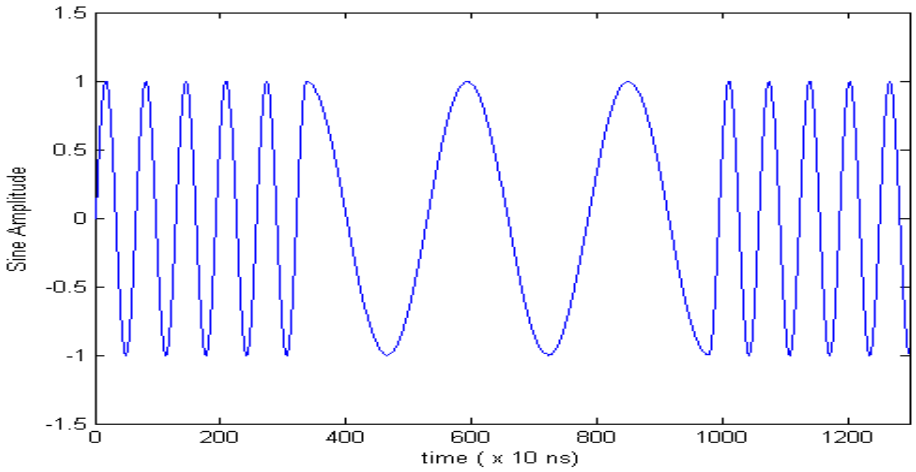


Fig 3-15 Synthesized quasi-sinusoid and digital phase sweep

Frequency–switching behavior is also simulated and the waveform was shown in Fig 3-16. In the beginning, input frequency control word FCW = h04000, after 3360 time unit, change the FCW to h01000, through 5760 time unit, switching back to FCW = h04000. One can find that phase of output waveform is continuous when the frequency is changed.



.

Fig 3-16 Frequency-switching behavior of DDFS

The output spectrum of implementation is calculated in Matlab environment, The FFT was performed over the output period, so the problem of leakage in the Fast Fourier transform (FFT) analysis is avoided. SFDR for various output frequency are simulated and calculated, which is depicted in Fig 3-17.

From the result of verilog simulation, SFDR is affected by output frequency; it is caused by distortion of Sine/Cosine mapping function, output bit quantization and phase truncation; (here did not cover distortion of DAC), individual factor that has an effect on SFDR is described in previous section. It can be seen that SFDR is over 61 dBc for all synthesized output frequency.

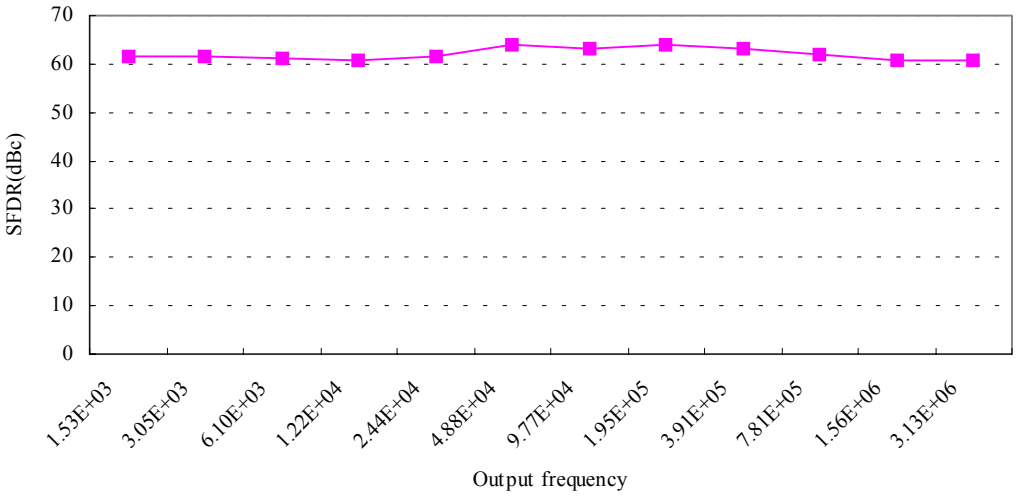


Fig 3-17 SFDR vs. output frequency

3.5 Spec of DDFS

From the result of previous section, the spec of proposed DDFS is listed in table3-4. System clock is 100MHz, tuning Bandwidth is 20MHz, and the dimension of look-up table is 576 bits. The phase resolution is 20 bit with 95.4Hz frequency resolution and is truncated to 12 bits which is input of SCMF, output bits of DDFS is 10 bits with spectral purity better than 61 dBc for all synthesized output frequency.

Parameter	Value	Unit
Frequency Control Word	20	Bits
No. Of Phase Address Bits	12	Bits
No. Of Output Bits	10	Bits
ROM Size	576	Bits
Max. Clock	100	MHz
Frequency Resolution	95.4	Hz
Tuning Bandwidth	12.5	MHz
SFDR	61	dBc

Table 3-4 The spec of DDFS

3.6 Implementation Result and Comparison

The summary of memory compression and algorithmic techniques are listed in Table 3-5. Table 3-5 shows how much memory and how many additional circuits are needed in each memory compression and algorithm technique under the worst case spur level. From the data in table 3-5, and assume that the system output amplitude is quantized with the 9 bits plus a sign bits, then realizes a reduction in ROM size of 25% compare with the Parabolic Approximation, MTA ROM segmentation, of 50% with Modified Nicholas Architecture and of 55% with Modified Sunderland Architecture.

Method	**Need ROM (bits)**	**SFDR (dBc)**	**Additional Circuits**
Uncompressed memory	2^{12} x 10	81.76	-
Quarter sine wave	2^{10} x 9	78.76	-
Modified Sunderland Architecture [12]	2^7 x 7 2^7 x 3	73.59	9-bits adder
Modified Nicholas Architecture [21]	2^7 x 7 2^7 x 2	74.56	9-bits adder
Parabolic Approximation [16]	2^7 x5 2^7 x 1	66.8	Multiplier, 9-bits adder
MTA Quarter period [11]	2^{10} x 4	78.76	7-bitsadd/sub, 9-bits adder*2, Switch
MTA ROM segmentation [11]	2^7 x 4 2^7 x 2	73.84	7-bitsadd/sub, 9-bits adder*2, Switch
DTA with ROM segment	2^6x6 2^6x3	61	6-bitsadder, 9-bits adder

Table 3-5 Comparison of memory size reduction and additional circuit for 12-bits phase to 10 bits amplitude mapping

This implementation is simple and straightforward. The amplitude value

computation utilizes two adders, one is 9-bit adder and the other is 6 bits adder, the parabolic approximation requires full multiplier and a 2's complementor. Since the ROM is generally the hungriest subsystem in a DDFS synthesizer, it is expected that the substantial reduction in ROM size and power consumption more than justifies the modest increase in processing and control costs. Initial investigation show that can reduce the ROM size and maintaining additional circuits cost very low. Of course, additional circuit and Rom size can be increased to improve performance of SFDR.

CHAPTER 4 DIGITAL MODULATION and DDFS

4.1 Binary FSK

Frequency shift keying (FSK) is the most common form of digital modulation in the high-frequency radio spectrum, and has important applications in telephone circuits. Binary FSK (usually referred to simply as FSK) is a modulation scheme typically used to send digital information between digital equipment such as teleprinters and computers. The data are transmitted by shifting the frequency of a continuous carrier in a binary manner to one or the other of two discrete frequencies. One frequency is designated as the "1" frequency and the other as the "0" frequency. The "1" and "0" correspond to binary one and zero, respectively. By convention, "1" corresponds to the higher radio frequency. Fig 4-1 shows the relationship between the data and the transmitted signal. Frequency Shift Keying (FSK) – 1/0 represented by two different frequencies slightly offset from carrier frequency

BFSK: $f_0(t)= A\cos((\omega_c -\Delta\omega)t)$

$f_1(t)= A\cos((\omega_c+\Delta\omega)t)$ (4-1)

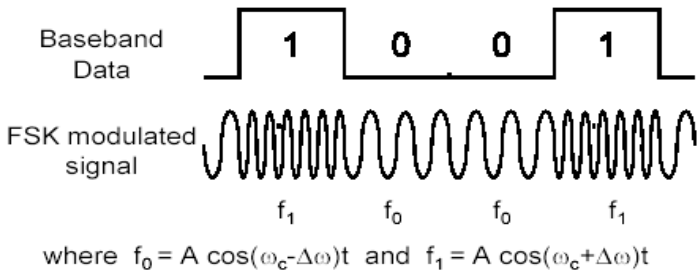


Fig 4-1 Binary FSK

4-2 Double frequency shift keying (DFSK)

DFSK, sometimes called DFS is a scheme to transmit two independent binary data streams by shifting the frequency of a single carrier among four discrete frequencies. Table 4-1 shows a code table to convert two binary bits to one of four output states. For example, if bit X and bit Y is 1 and 0, respectively, the output state is C. If each of the four output frequency states (A through D) is assigned to a corresponding FSK frequency, then it is possible to transmit any two bits as a single element. Fig 4-2 illustrates a typical DFSK signal.

Input Bits		Output Frequency
X	Y	
0	0	A
0	1	B
1	0	C
1	0	D

Table 4-1 one-of-four code tables for two binary bits.

In the example of Fig 4-2, output states A through D are assigned to frequencies f1 through f4, respectively.

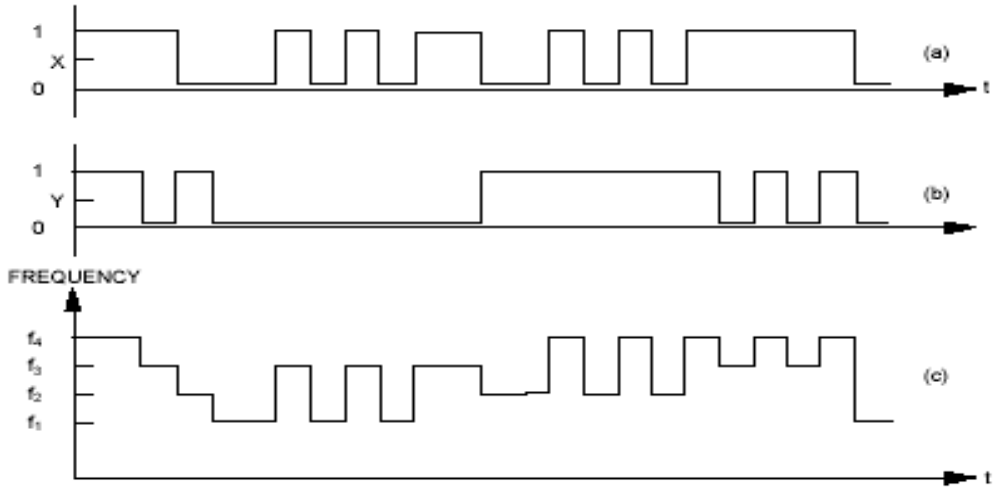


Fig 4-2. DFSK signal.

4-3 PSK

The IEEE 802.11a standard defines OFDM modulation for the high-speed physical layer (PHY) in the 5 Ghz band. The OFDM process provides for a wireless LAN with data payload communication capabilities of 6,9,12,18,24,36,48 and 54 Mbps. transmission and reception of data at the rates of 6,9,12 and 18 Mbps are addressed; The standard specifies 52 subcarriers that are modulated using either binary or quadrature phase-shift keying (BPSK/QPSK) and data rate are 6,9 Mbps for BPSK, 12,18 Mbps for QPSK.

4.3.1 Binary Phase Shift Keying

For binary PSK (BPSK), symbol waveforms have the form:

$$S_0(t) = A\cos(\omega_c + \pi) = -A\cos\omega_c t \quad \text{represents binary "0"}$$
$$S_1(t) = A\cos\omega_c \quad \text{represents binary "1"} \qquad (4\text{-}2)$$

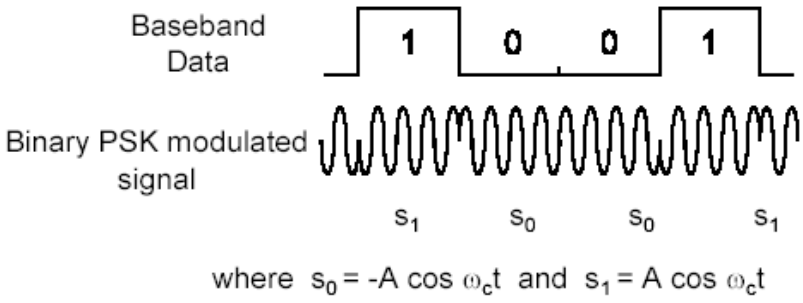


Fig 4-3 BPSK modulation

In 802.11a standard, The OFDM training symbols shall be followed by the SIGNAL field, which contains the RATE and the LENGTH fields of the TXVECTOR. The RATE field conveys information about the type of modulation and the coding rate as used in the rest of the packet. The encoding of the SIGNAL single OFDM symbol shall be performed with BPSK modulation of the subcarriers. The BPSK constellation bit encoding is shown in Fig 4-4 and for BPSK, b0 determines I value, as illustrated in Table 4-2.

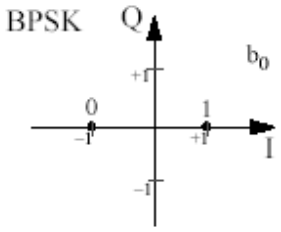


Fig 4-4 BPSK constellation bit encoding

Input bit (b_0)	I-out	Q-out
0	-1	0
1	1	0

Table 4-2 BPSK encoding table

4.3.2 Quadrature Phase Shift Keying

If we define four signals, each with a phase shift differing by 90^0 then we have quadrature phase shift keying (QPSK). The input binary bit stream {dk}, k = 0,1,2... arrives at the modulator input at a rate 1/T bits/sec and is separated into two data streams DI (t) and DQ (t) containing odd and even bits respectively.

DI (t) = d0, d2, d4, ---

DQ (t) = d1, d3, d5, ---

1/T bits/sec and is separated into two data streams DI (t) and DQ (t) containing odd and even bits respectively. A convenient orthogonal realization of a QPSK waveform, s(t) is achieved by amplitude modulating the in-phase and quadrature data streams onto the cosine and sine functions of a carrier wave as follows:

$$s(t) = 1/\sqrt{2}DI(t)\cos(2\pi ft + \pi/4) + 1/\sqrt{2}DQ(t)\sin(2\pi ft + \pi/4) \qquad (4\text{-}3)$$

Using trigonometric identities this can also be written as

$$s(t) = A\cos[2\pi ft + \pi/4 + \theta(t)] \tag{4-4}$$

The pulse stream DI(t) modulates the cosine function with amplitude of ±1. This is equivalent to shifting the phase of the cosine function by 0 or π; consequently this produces a BPSK waveform. Similarly the pulse stream DQ(t) modulates the sine function, yielding a BPSK waveform orthogonal to the cosine function. The summation of these two orthogonal waveforms is the QPSK waveform.

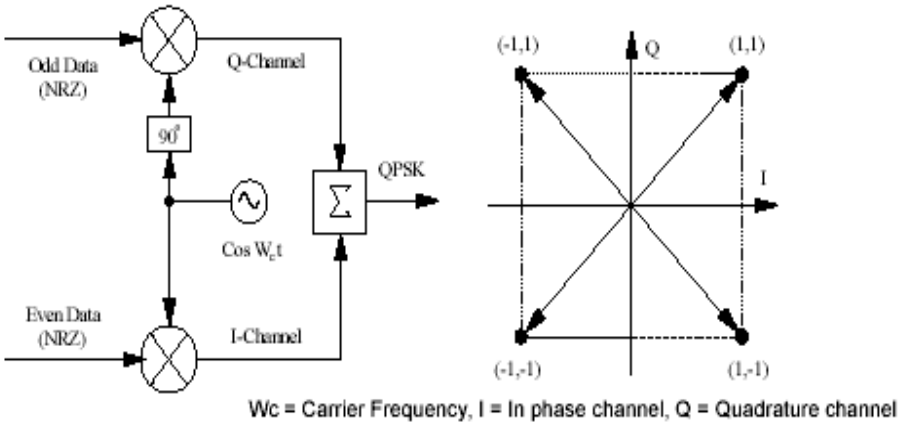


Fig 4-5 QPSK modulations

Each of the four possible phases of carriers represents two bits of data. Thus there are two bits per symbol. Since the symbol rate for QPSK is half the bit rate, twice as much data can be carried in the same amount of channel bandwidth as compared to BPSK. This is possible because the two signals Q and I are orthogonal to each other and can be transmitted without interfering with each other.

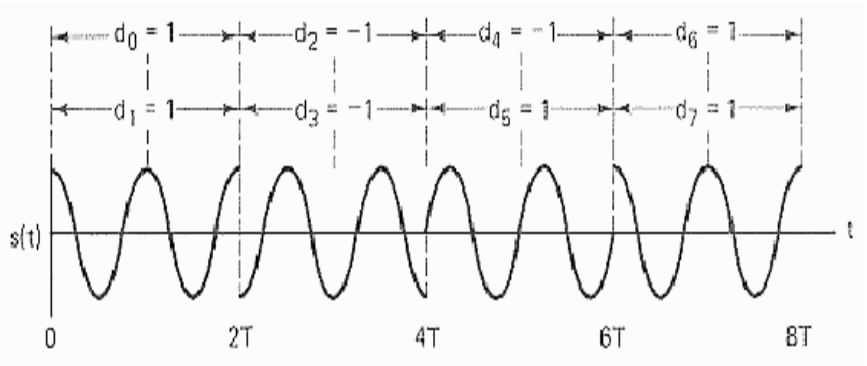


Fig4-6 QPSK

In 802.11a standard, The standard specifies 52 subcarriers that are modulated using quadrature phase-shift keying (QPSK) and data rate are 12,18 Mbps, b0 determines the I value and b1 determines the Q value, as illustrated in Table 4-3.

Input bit (b_0)	I-out	Input bit(b_1)	Q-out
0	-1	0	-1
1	1	1	1

Table 4-3 QPSK encoding table

CHAPTER 5
Digital Modulation Verification

In this chapter, We will use architecture of DDFS depicted in chapter 3 to realize the digital modulation described in chapter 4, the proposed digital modulator will include FSK, DFSK, BPSK and QPSK digital modulation function of base band. Using Synplify Pro to synthesize the verilog code and Altera device EPF10K100ARC240-1 to verify the function of digital modulator.

5.1 Signal Interface

The signal interface of proposed digital modulator is shown in Fig 5-1 and the signal definition are described in Table 5-1

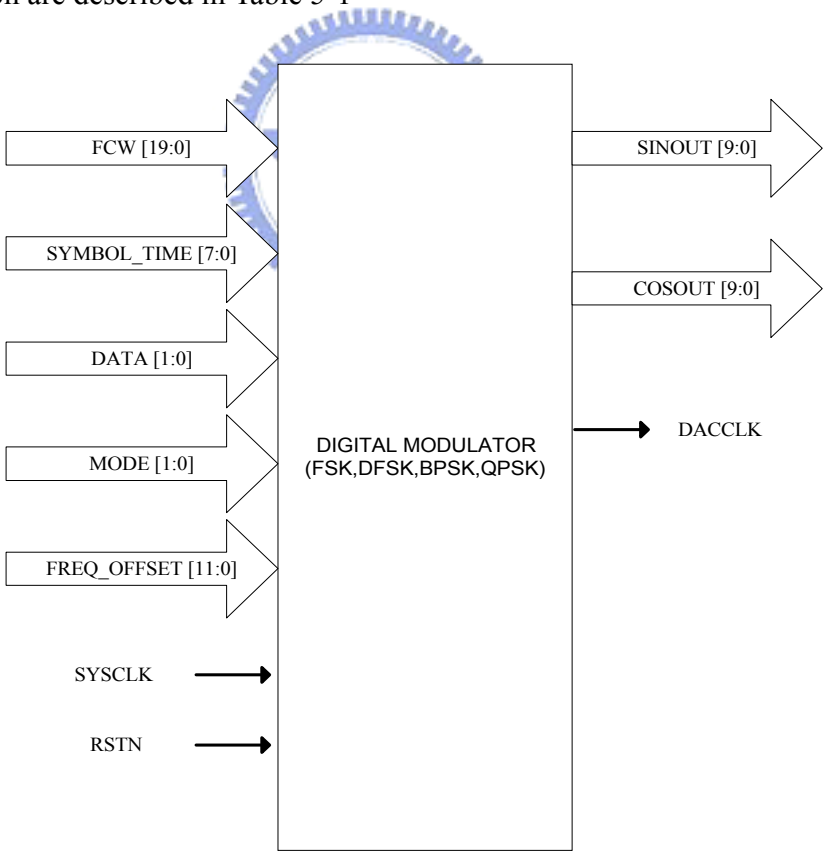


Fig 5-1 Signal interface of digital modulator

Signal name	Type	Description
FCW [19:0]	I	This input is the frequency control word to the DDFS. This word controls the phase accumulator rate, the output frequency of the SINOUT and COSOUT waveform. The output frequency is calculated by the following: F_{out} = FCW [19:0]*(SYSCLK/2^{20})Hz
SYMBOLTIME [7:0]	I	Symbol time, it can be multiple of full sine/cos wave, the range from 0 to 255.
MODE [1:0]	I	These two input bits are used to choice the type of digital modulation, the type of modulation is as follows: 2'b00: FSK 2'b01: DFSK 2'b10: BPSK 2'b11: QPSK
FREQ_OFFSET [11:0]	I	This 12 bit are used in determining the frequency offset for FSK and DFSK FSK: 1'b0: FCW = FCW 1'b1: FCW = FCW+ FREQ_OFFSET DFSK: 2'b00: FCW = FCW 2'b01: FCW = FCW+ FREQ_OFFSET/256 2'b10: FCW = FCW+ FREQ_OFFSET/8 2'b11: FCW = FCW+ FREQ_OFFSET
DATA [1: 0]	I	Data input for digital modulation. Bit 0 is for FSK & BPSK Bit 0-1 is for DFSK & QPSK
SYSCLK	I	This is the system reference clock input to the DDFS. This clock is the sampling clock of output data .the maximum frequency for SYSCLK is 100MHz
RSTN	I	This input is a high asserted global reset. When asserted, the internal phase and frequency control word registers are cleared stopping the output carrier at 0 radians.
SINOUT [9:0]	O	This output is the sine digital amplitude. The output is valid in the rising edge of the DACCLK.
COSOUT [9:0]	O	This output is the cos digital amplitude . The output is valid in the rising edge of the DACCLK.
DACCLK	O	This output is the DAC clock strobe. This clock is the SYSCLK feed to an output pin. The output SINOUT and COSOUT amplitude words will be valid on the rising edge of the DACCLK

Table 5-1 Signal function table of digital modulator

5.2 Simulation Result of Verilog Code

The verilog codes are simulated in ModelSim. The sinusoid amplitude output from simulation of ModelSim is saved to file and use Matlab to do post-processing. The synthesized waveform of sine and cosine for DFSK, FSK, BPSK and QPSK with symbol time are shown in Fig 5-2,5-3,5-4,5-5 respectively.

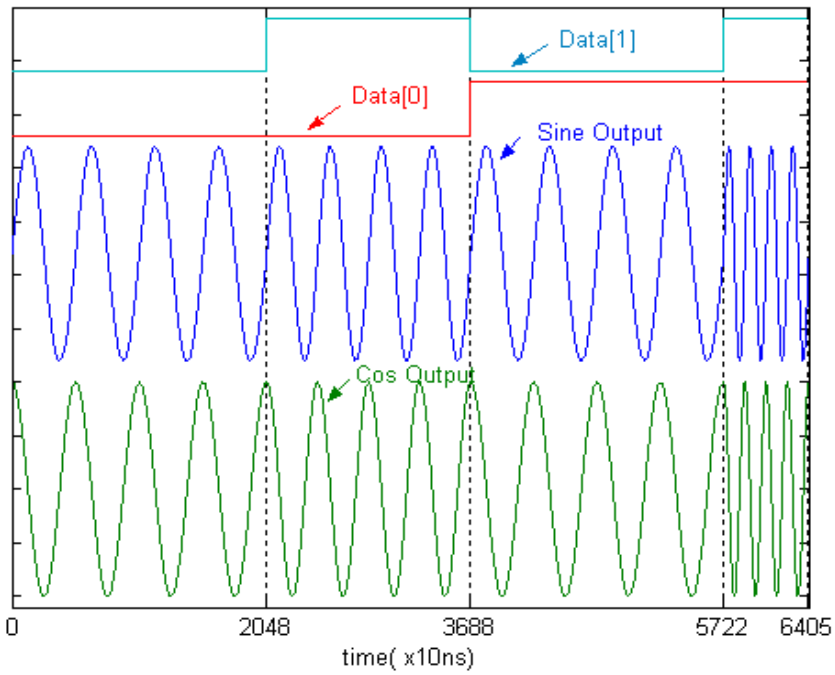


Fig 5-2 DFSK with symboltime = 4

We can easily find that the difference in phase between sine and cosine waveform equal 90° exact in FSK, DFSK and BPSK. In QPSK, phase is defined according to the spec of IEEE802.11a is shown in Fig 5-5.

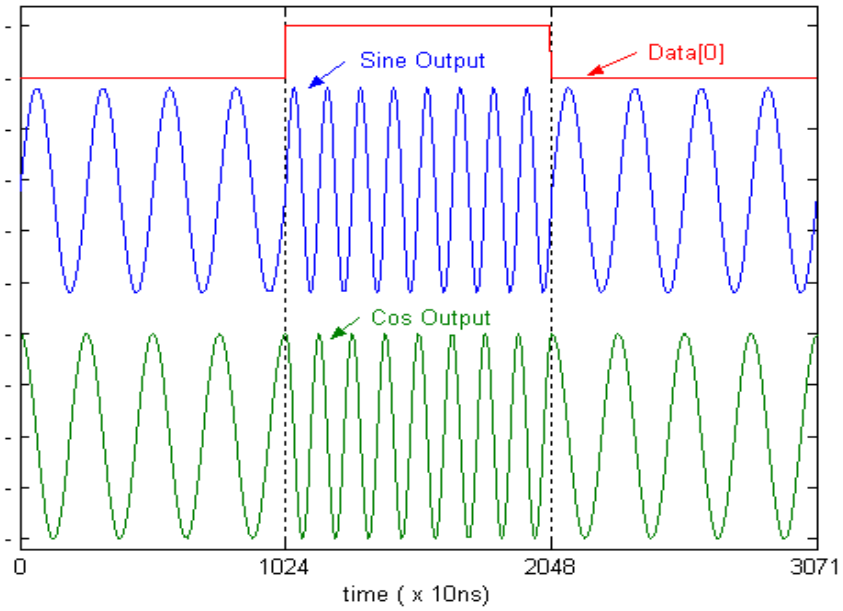


Fig 5-3. FSK with symboltime = 4,8,4

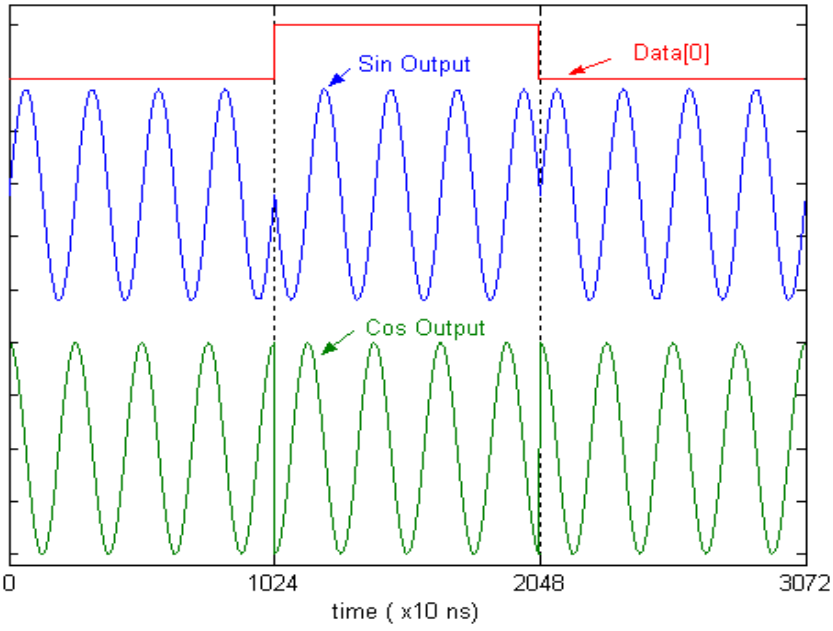


Fig 5-4 BPSK with symboltime = 4

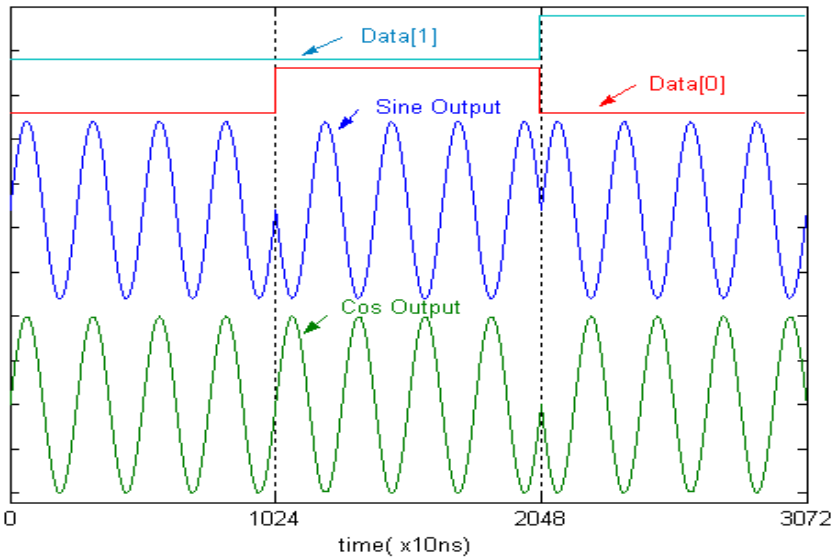


Fig 5-5 QPSK with symboltime = 4

The function of signal symbol time are used to generate number of full sine/cosine wave at setting frequency, it can be seen in Fig 5-6.

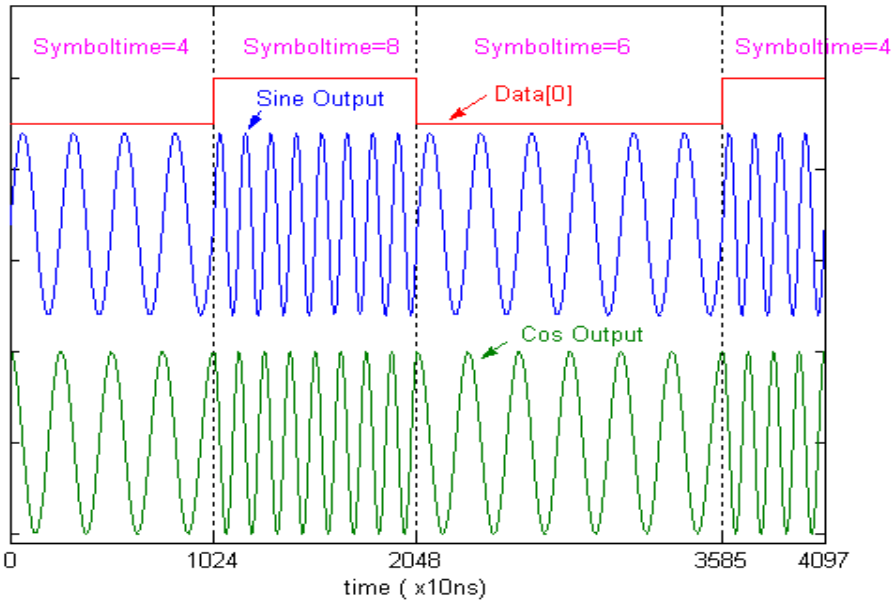


Fig 5-6 FSK with variable symboltime (4,8,6,4)

After hardware description by verilog and function check, we use the Synplify Pro to synthesize the verilog code from the behavior level to gate level, The target device is the Altera Flex10K; the part is EPF10K100ARC240-1, It has 189 I/O pins and 48K

memory bits. We can see each RTL schematic view of verilog module design in Fig 5-7,5-8,5-9,5-10 and 5-11

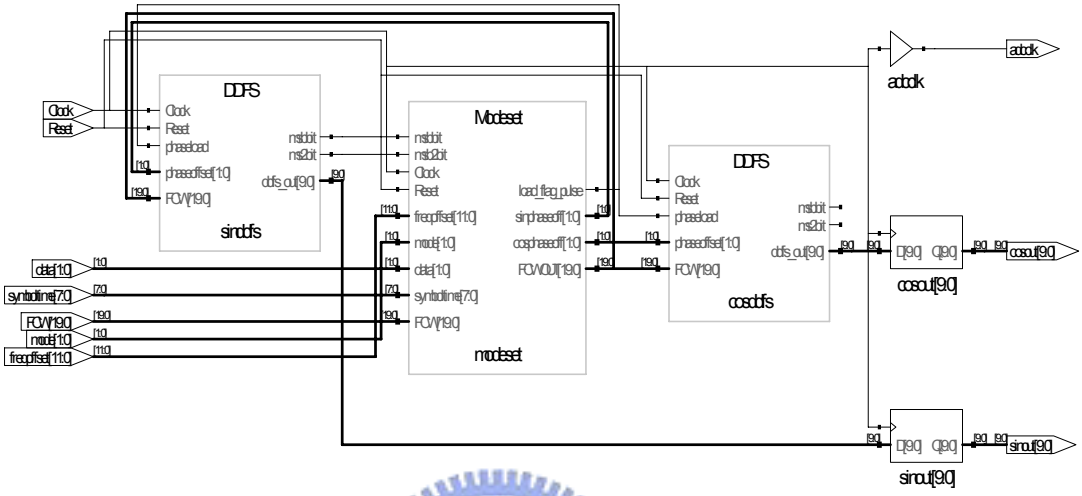


Fig 5-7 Modulator.v RTL schematic view

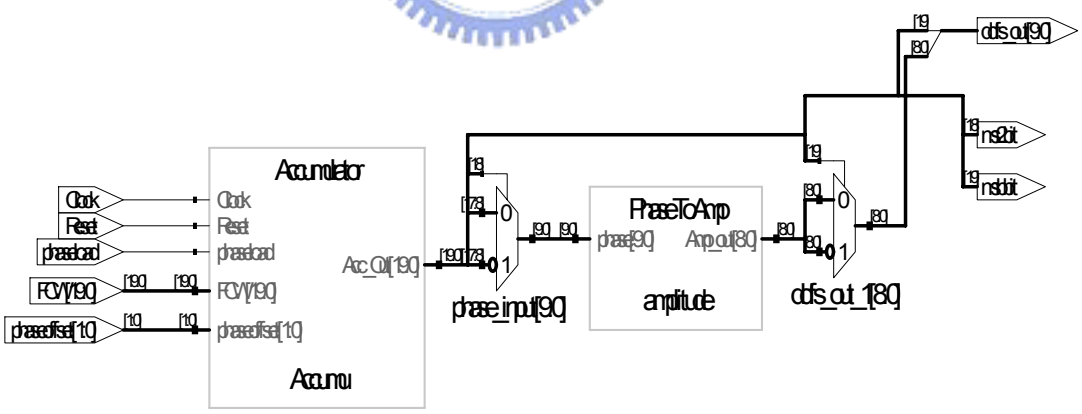


Fig5-8 DDFS.v RTL schematic view

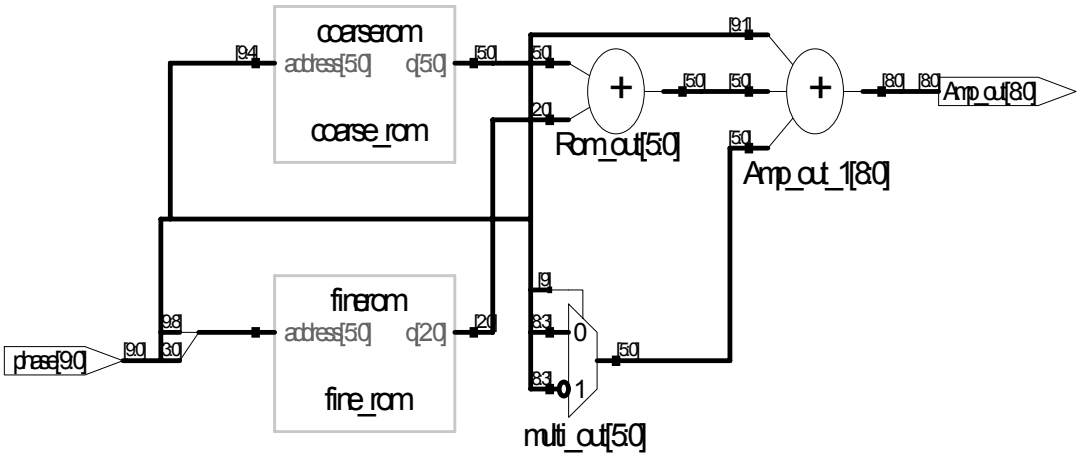


Fig 5-9 Phasetoamp.v RTL schematic view

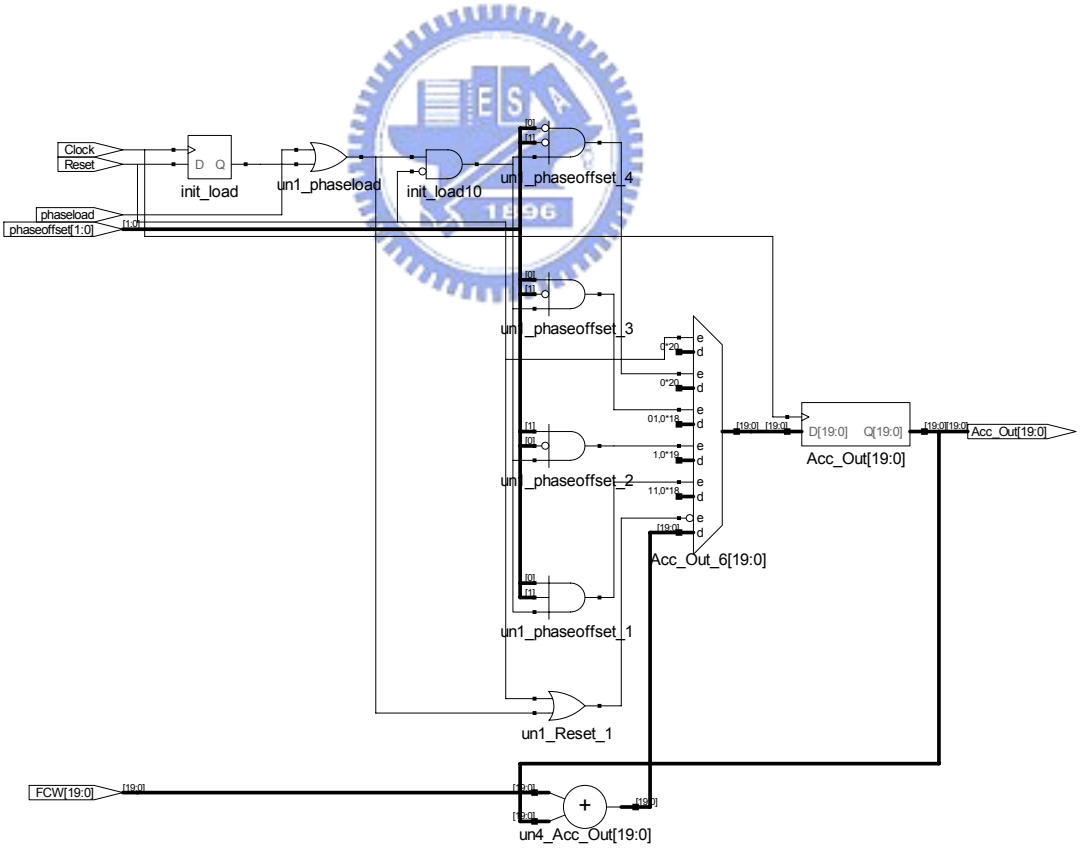


Fig 5-10 Accumulator.v RTL schematic view

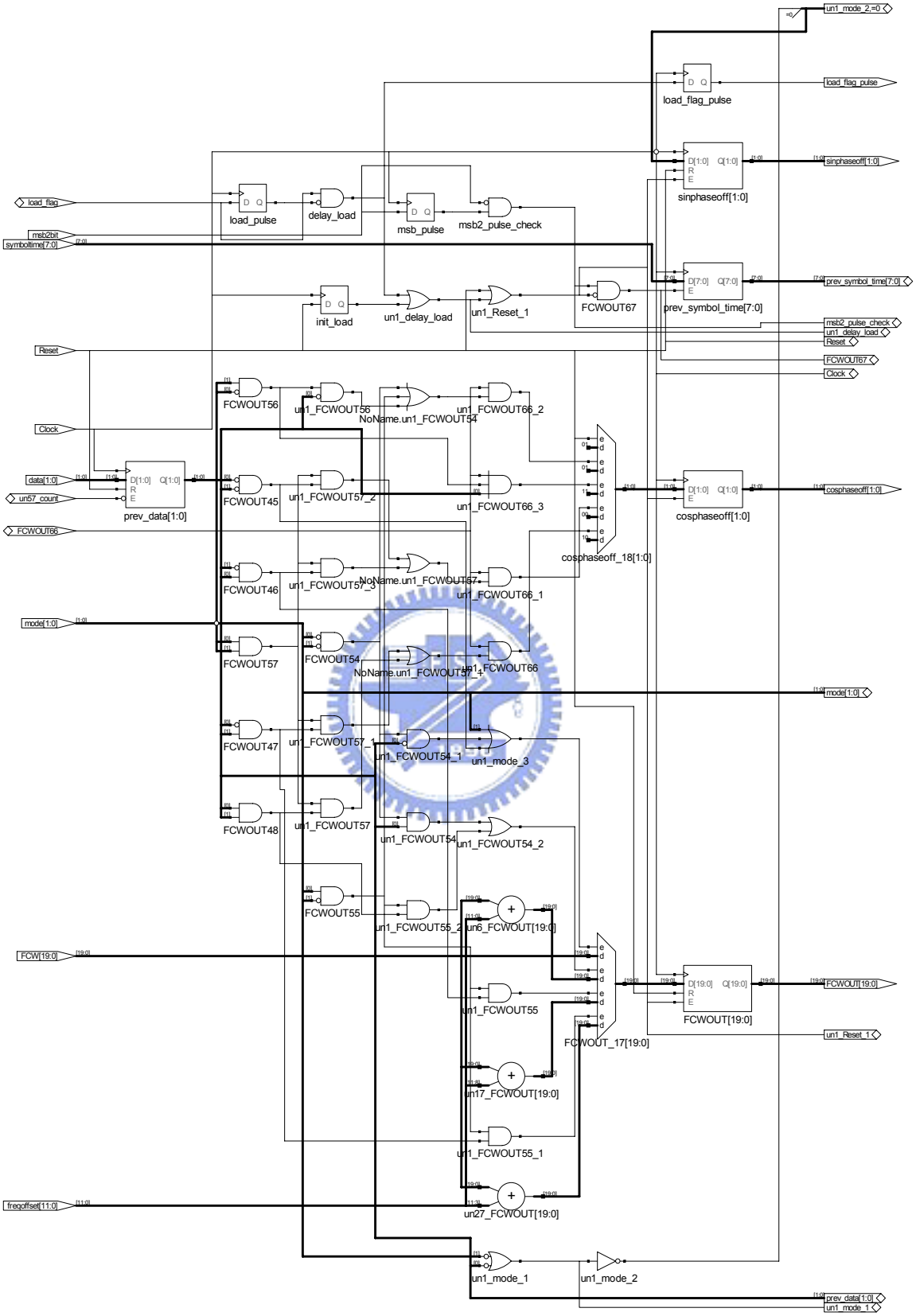


Fig 5-11 Modset.v RTL schematic view

5-3 FPGA Implementation Result

Net list file for verilog code is generated by using the Synplify Pro and transform the net list file from the output of Synplify Pro into SOF file, Altera format file used in Quart environment, then download this SOF file from the computer to the FPGA board and connect the output pin of EPF10K100ARC240-1 to Tektronix TLA 700 logic analyzer.

Family FLEX10KA			
Device EPF10K100ARC240-1			
	Total	Used	%
Logic elements	4,992	238	4%
Memory bits	49,152	1,152	2%

Table 5-2 Resource of device used in digital modulator

From the table 5-2, requirement of resource for device EPF10K100ARC240-1 is 238 logic elements (4% of total), 1152 memory bits (2% of total).

In order to evaluate the digital modulator, a test board is built. Since we only want to verify the algorithm of DDFS, the test system does not include the DAC; the test systems is shown in Fig 5-12 and combine the Altera EPF10K100ARC240-1 with 32Mhz oscillator on a PCB board to design a simple experimental board.

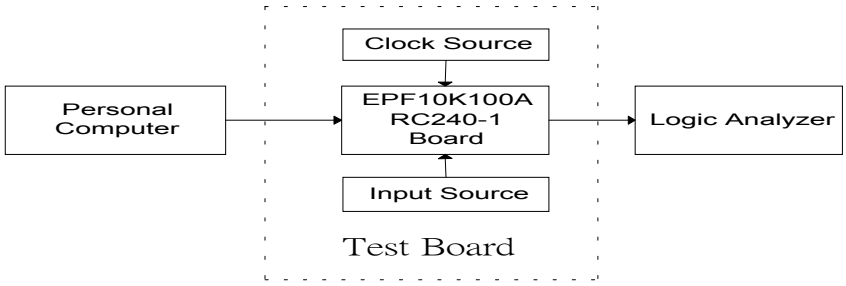


Fig 5-12 Digital Modulator test system

The digital waveforms (captured by Tektronix TLA 700 logic analyzer) of digital modulator are shown in Fig. 5-13,5-14,5-15 and 5-16. There are four types of digital modulation (BPSK, FSK, QPSK, DFSK); the input conditions are depicted as follows:

(1). FCW = h01000, FREQ_OFFSET = hfff, MODE [1:0] = 0, SYMBOLTIME = 4, data input is switched between 0 and 1, the digital waveforms of FSK is shown in Fig 5-14.

(2). FCW = h00400, FREQ_OFFSET = hfff, MODE [1:0] = 1, SYMBOLTIME = 4, data input is switched between 0 and 2, the digital waveforms of DFSK is shown in Fig 5-16.

Fig 5-13 BPSK with symboltime = 4

(3). FCW = h01000, MODE [1:0] = 2, SYMBOLTIME = 4, data input is switched between 0 and 1, the digital waveforms of BPSK is shown in Fig 5-13.

(4). FCW = h01000, MODE [1:0] = 3, SYMBOLTIME = 4, data input is switched between 0 and 2, the digital waveforms of QPSK is shown in Fig 5-15.

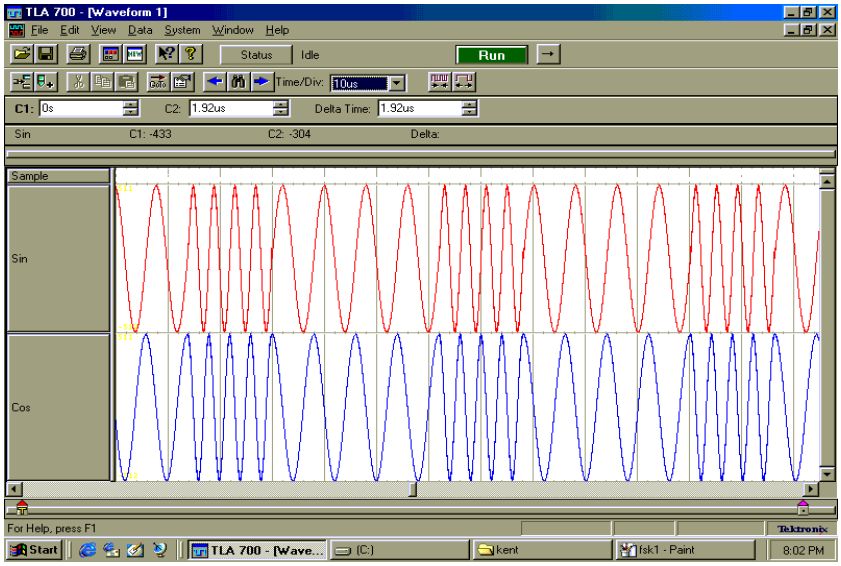


Fig 5-14 FSK with symboltime = 4

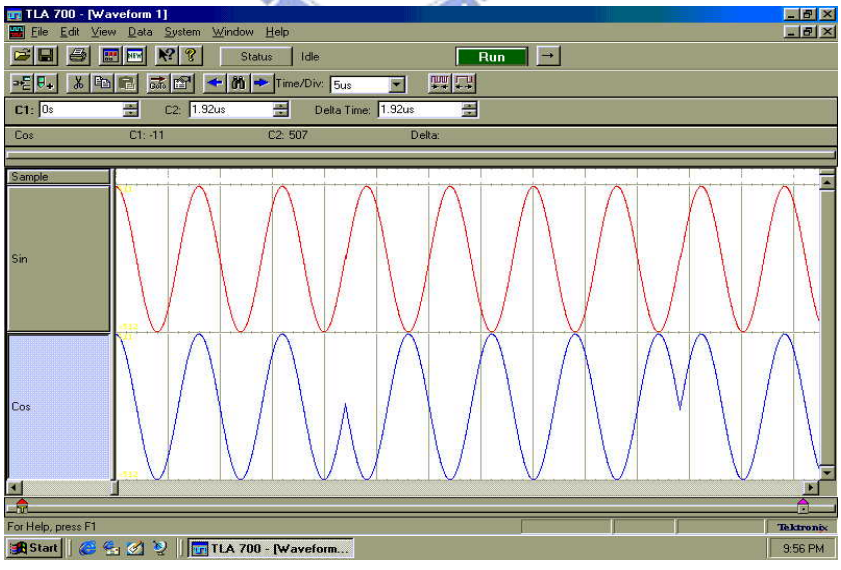


Fig 5-15 QPSK with symboltime = 4

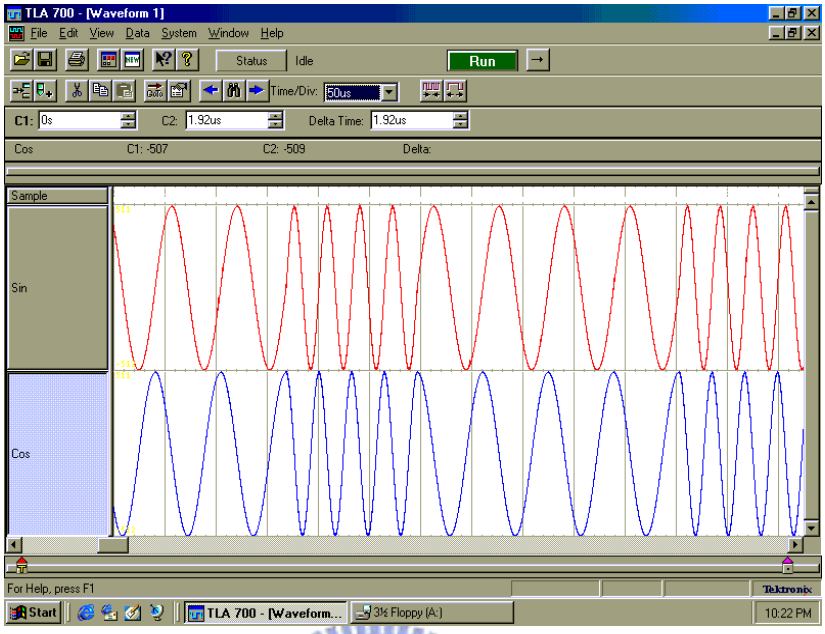


Fig 5-16 DFSK with symboltime = 4

All the results, which are shown in Fig 5-13, 5-14, 5-15, 5-16, are consistent with the results that are simulated in ModelSim. It shows that we can implement the DDFS in the digital modulator to get more precise and fast-switching speed sine waves.

CHAPTER 6
Conclusion

In this thesis, we propose a digital modulator with FSK, DFSK, BPSK and QPSK function by using direct digital frequency synthesizer (DDFS). It provides overview for the DDFS design, the error caused by finite output word length, phase truncation and sine/cosine mapping function (SCMF) are also presented. It depicts the error source of spurious information and method of estimation. Therefore, designers can choice the output word length, number of phase truncation bits and proper SCMF algorithm according to the spec of DDFS.

For algorithm of SCMF in this thesis, this implementation is simple and straightforward. The amplitude value computation utilizes two adders, one is 9-bit adder and the other is 6 bits adder, only adder circuits are required in the additional circuits. No subtractor and multiplier are needed, since the ROM is generally the hungriest subsystem in a DDFS synthesizer, it is expected that the substantial reduction in ROM size and power consumption. Rom table size can be reduced to 576 bits without additional circuit under sacrificing the performance of SFDR. Simulation shows that the worst case of SFDR is 61dBC for various output frequency

It is expected that the substantial reduction in ROM size and power consumption more than justify the modest increase in processing and control costs. Initial investigation show that can reduce the ROM size and maintaining additional circuits cost very low. Of course, Additional circuit and Rom size can be increased in order to improve the SFDR.

We use the proposed DDFS to implement the digital modulator with FSK, DFSK, BPSK and QPSK function; the digital modulator is also with sine/cosine output. Using Synplify Pro to synthesize the verilog code and Altera device EPF10K100ARC240-1 to verify the function of digital modulator; it share the 238 (4%)logic elements and 1152(2%) memory with device.

Finally, we did not use the DAC in the verification of DDFS and digital modulator, but we can combine the design of DDFS and DAC to do more flexible digital modulator (QAM ---- and so on) for the future work.