

**Implementation of Digital Modulation Techniques using Direct Digital
Synthesis**

by

Udaya Shankar N

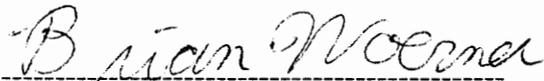
Project submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science

in

Department of Electrical Engineering

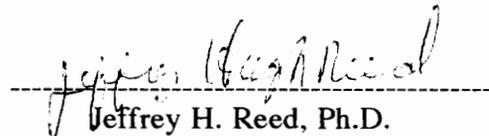
APPROVED:



Brian D. Woerner, Ph.D., Chairman



Timothy Pratt, Ph. D.



Jeffrey H. Reed, Ph.D.

January 1993

Blacksburg, Virginia

C.2

LD
5655
V851
1993
5536
C.2

Implementation of Digital Modulation Techniques using Direct Digital Synthesis

by

Udaya Shankar N

Dr. Brian D. Woerner, Chairman

Department of Electrical Engineering

ABSTRACT

In the past decade, low earth orbit satellite systems have become increasingly popular. There is a growing need for hardware and software to model emerging low earth orbit satellite systems. One possible implementation of the transmitter for a low earth orbit satellite system is with direct digital synthesis equipment. Here we build hardware compatible with the IBM PC and software in ' C ' to simulate a low earth satellite transmitter using a direct digital synthesis system. We present a detailed working analysis of the direct digital synthesis system. The analysis shows the biggest advantage of the direct digital synthesis method is its very fine frequency and phase resolution with the disadvantages being its cost and frequency limitation.

Acknowledgements

I would like to express my sincere appreciation to my major professor, Dr. Brian D. Woerner, for his willing assistance and guidance throughout my project. I would like to thank Dr. Tim Pratt and Dr. Jeff H. Reed for their helpful suggestions and for serving on my Advisory Committee. I am also grateful for their friendship and many stimulating conversations we have had together.

Many thanks to Mr. Sean Kelley for his encouragement and guidance through the course of this project.

Appreciation is also due to the MPRG laboratory for providing facilities on which this project was done.

Finally, I am most grateful to my parents and my brothers for their continued encouragement.

Table of Contents

Chapter 1. Introduction.....	1
1.1 Leosat Systems	1
1.2 Real time Communications	2
1.3 Delayed Communications.....	3
Chapter 2. Spread Spectrum and Frequency Hopping	4
2.1 Model of a Spread Spectrum System.....	6
2.2 Direct Sequence Spread Spectrum	9
2.3 Frequency Hopping Spread Spectrum.....	11
Chapter 3. System Specifications and Hardware Design.....	13
3.1 System Specifications.....	13
3.2 Design Constraints.....	13
3.3 System Design.....	14
3.4 Hardware Design.....	15
3.5 Specific Hardware.....	16
Chapter 4. Results.....	24
4.1 Illustrations	24
4.2 Disadvantage of DDS method.....	26
Chapter 5. Conclusions.....	30
References.....	31
Appendix A. Software	33
Appendix B. Instructions for use.....	39

Chapter 1. Introduction

1.1 Leosat Systems

In the past three decades geostationary systems have dominated the field of satellite communications. A geostationary satellite is usually placed at an altitude of about 36,000km and at this height it makes one complete revolution in 24 hours. Its orbit is placed in the plane of the equator and hence appears to be stationary over a certain place on the earth thus providing communications round the clock. Fixed antennas pointed at the satellites can be used to communicate between distant points on the earth and a world wide coverage can be ensured with only 3 satellites. The disadvantages of GEO systems is that the cost of the launches is expensive and the path loss is large. Also the congestion in the GEO orbit made it imperative to find an alternative for GEO systems.

There are many reasons for the consideration of the LEOSAT systems. The defense need for highly reliable communication systems could not be met by GEO satellites as they were highly vulnerable targets. Also the growing cost of GEO satellites, their increasing mass and the congestion of the geostationary orbit led to the investigation of alternate systems. This eventually resulted in the evolution of LEO satellites.

A LEO satellite is usually positioned at altitudes ranging from 500km-1500km. The satellites can either have inclined or polar orbits. A constellation of many satellites provides continuous round the clock coverage. Also LEO satellites are extremely flexible in their

design and can be launched using recently developed small launch vehicles. Flexibility, global coverage and frequency reuse are the main advantages of LEO systems.

The communication services that can be provided by the LEOSAT systems are telephony, and data transfer. Also the LEOSAT systems can provide local communication (limited in range for a few tens of kilometers), regional communication (limited to a specific region) and global communication. The communications provided by the LEOSAT systems can be classified into two main categories.

1. Real time communications (Big LEOs).
2. Delayed communications (Little LEOs).

1.2 Real Time Communications

Real time services provide nearly instantaneous transfer of information from the source to the destination. Although real time services offer time continuous communications between users, this does not mean that the system offers a permanent link between users: the information is transmitted in the form of packets which are independently routed. Examples of this type are audio/video broadcasting, voice telephony and data message exchange.

Examples of this type are GLOBALSTAR and IRIDIUM. IRIDIUM is a LEOSAT system being developed by the Motorola Satellite Communications. This system is designed to have 6 orbital planes, 11 satellites/plane and 48 beams/satellite. The primary services that will be offered are voice, paging, data, and facsimile services. The system is currently planned to be completed by early 1998. The GLOBALSTAR system consists of a constellation of 48 satellites in circular orbits with 1400 km altitude, with six satellites in each orbital plane, to provide spot coverage. This system provides essentially global mobile personal communications services [16].

1.3 Delayed Communications

Delayed communications store the messages before delivery. A typical message routing is as follows: the end user enters his message into the terminal; the terminal waits for the satellite to be in view and then transmits the message, which is stored on board the satellite. The storage can also be accomplished in relay earth stations distributed all over the world. Later the message is delivered to the end terminal and is stored until the user wishes to retrieve it. Examples of this type are electronic mail, digitized images , computer data, etc. [9]. An example of this type is Orbcomm.

There is a growing need for hardware and software to model emerging LEOSAT systems. One possible implementation of the transmitter for a LEOSAT system is with Direct Digital Synthesis (DDS) equipment. Here we simulate a transmitter using a DDS system. We demonstrate the use of a DDS board for producing digital modulation techniques and we relate theory and practice.

The remainder of this report is organized as follows. An introduction to spread spectrum communication is given next followed by system specifications and hardware design. Finally the results are discussed and conclusions drawn.

Chapter 2. Spread Spectrum and Frequency Hopping

One communication technique which may prove useful for the emerging LEOSAT systems is spread spectrum communication. This technique was being used by the military for quite some time and is now being used for commercial purposes also. In this method essentially the signal is spread over a frequency band which is greater than the minimum bandwidth required to transmit the information. The bandwidth is spread by means of a code word independent of the data. Since coded waveforms are characterized by a bandwidth expansion factor that is greater than unity the signal will occupy a greater bandwidth than it otherwise would. Reception is accomplished by compressing the received signal with a synchronously generated replica of the code word. The code word used for spreading is random in appearance but can be reproduced by deterministic means.

The inherent advantages of spread spectrum systems are:

1. Enhanced Security

A message may be hidden in the background noise by spreading its bandwidth with coding and transmitting the resultant signal at a low average power. The signal is said to be secure because of its low power level and pseudo random code. Since the signal is spread over a wider range of frequencies now, a larger frequency band must be monitored to intercept the signal. Also the power density of the signal to be detected is lowered making the task of detecting the signal even more difficult.

2. Anti-interference

Interference from other users arises in multiple access communications systems in which a number of users share a common bandwidth. At any given time, a subset of these users may transmit information simultaneously over the common channel to corresponding receivers. As the signal is spread over a wide spectrum, the power density per unit bandwidth is lowered thus resulting in less interference to other bands.

3. Anti-jamming

In combating intentional jamming, it is important that the jammer trying to disrupt the communication does not have prior knowledge of the signal characteristics except for the overall channel bandwidth and the type of modulation. The transmitter introduces an element of unpredictability or randomness in each of the transmitted coded waveforms which is known to the intended receiver but not to the jammer. As a consequence, the jammer must synthesize and transmit an interfering signal without the knowledge of the pseudo-random pattern. Also the jammer cannot use signal observation to improve its performance and must rely on jamming techniques which are independent of the signal to be jammed. Note that jamming rejection may be used to reject either intentional or unintentional jamming [3].

Most of the spread spectrum systems use the Code-Division Multiple Access (CDMA) to facilitate multiple access capability. CDMA systems are also referred to as spread spectrum multiple access (SSMA) since the signals are spread over a wide frequency band. Two spread spectrum techniques may be used - direct sequence and frequency hopping. In both the methods each user is allocated a specific code, and the code defines how the signal is spread across the frequency spectrum with time. The code must be known by the receiver so that it can despread the signal. In CDMA systems, the entire

transmission bandwidth is available to all the users at all times. This allows the users to access the channel randomly. A major advantage to spread spectrum as an access technique is its resistance against jamming and other interferences [2].

2.1 Model of a Spread Spectrum System

The block diagram of a basic spread spectrum digital communications system is shown in Figure 1. The signal at the source is sampled and these samples are applied to a quantizer. The quantized samples are applied to an encoder. The encoder responds to each sample by the generation of a unique and identifiable binary pulse pattern. The combination of the quantizer and the encoder is called an analog-to-digital converter. Error correction coding is often included in this stage. This error correction coding is either block encoding or linear convolutional encoding. In block encoding blocks of k information bits are encoded into corresponding blocks of n bits ($n > k$). Thus the signal transmitted over the communications channel is a digitally encoded signal.

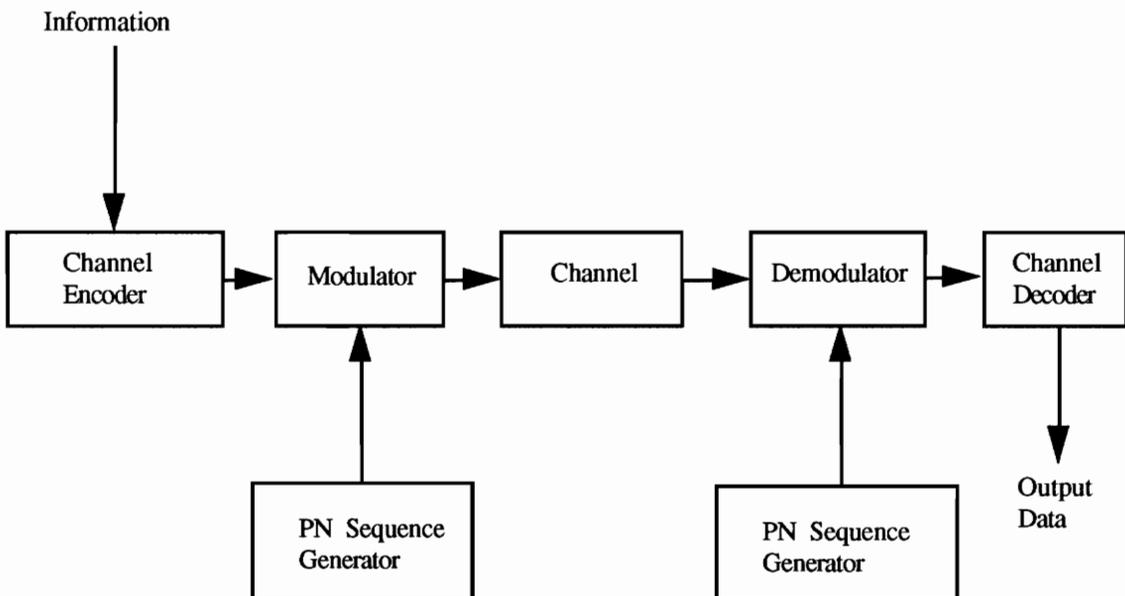


Figure 1. Spread Spectrum System

The binary digits from the encoder are fed into a modulator which maps each bit into an elementary signal waveform. Binary PSK or FSK are commonly used signaling waveforms for transmitting the bits in the coded sequence.

The demodulator can be viewed as a matched filter to the signal waveform. Its output may or may not be quantized. In some systems the decoder is used to make firm decision on whether each coded bit is a 0 or a 1. The sequence of the detected bits is fed into the decoder. The decoder performs the inverse operation of the encoder. The decoder output is the sequence of quantized multilevel pulses. The quantized signal is now reconstituted. In this case since the decoder operates on the hard decisions made by the demodulator, the decoding process is termed as hard decision decoding. In certain other systems the analog output from the demodulator is fed to the decoder. The decoder makes use of the additional information contained in the unquantized samples to recover the information sequence with a higher reliability than that achieved with hard decisions. This is known as soft decision decoding [15].

The two extra blocks are the identical pseudo random pattern generators, one which interfaces with the modulator at the transmitting end and the other which interfaces with the demodulator at the receiving end. The generators generate a pseudo-random binary valued sequence which is impressed on the transmitted signal at the modulator and removed from the received signal at the demodulator. A diagram of a PN sequence generator is shown in Figure 2. It consists of a shift register and a parity generator. Usually type D flip-flops are selected to build the shift register. The input to the first flip-flop is the output of the parity generator. A parity generator generates an output which is at logic 0 when an even number of inputs are at logic 0 and generates an output which is at logic 1 when an odd number of inputs are at logic 1. Generally a parity generator is constructed of an array of exclusive - or gates. A portion of each connection to the parity generator is shown to be dashed in order

to indicate that not all outputs are need to be connected to the parity generator. As a matter of fact, the PN sequence generated depends on the number of flip-flops employed and also on which of these are connected to the parity generator. Furthermore if feedback connections are arranged so that the PN sequence waveform has a maximum period of N where $N = (2 ** r) - 1$ where r is the number of stages of shift registers. This type of PN code generator is called maximum - length sequence generator or m - sequence generator.

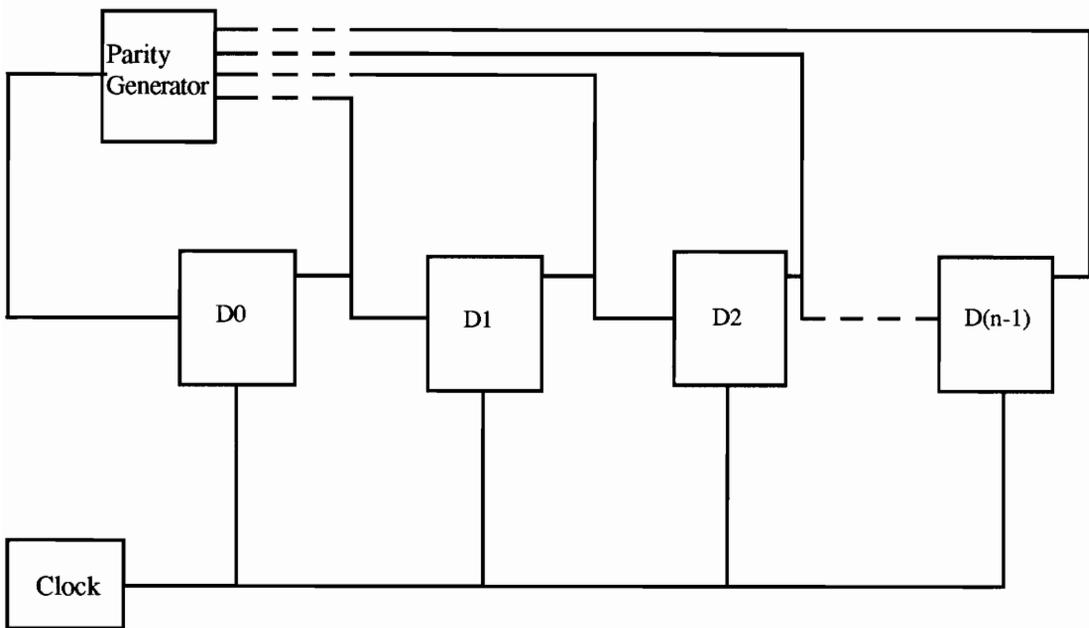


Figure 2. PN Sequence Generator

Properties of Maximum - Length Sequences [4].

Some properties of m - sequences are:

Property 1.

In one period the number of ones is always the one more than the number of zeros.

Property 2.

The modulo 2 sum of any m - sequence when summed chip by chip with a shifted version of the same sequences produces another shifted version of the same sequence.

Property 3.

If a window of width r (where r is the number of stages in the shift register) is slid along the sequence for N shifts, then all possible r -bit words will appear exactly once, except for the all 0 - bit word.

Synchronization of the PN sequence generated at the receiver with the PN sequence contained in the incoming received signal is required in order to demodulate the received signal. Initially, prior to the transmission of information, synchronization may be achieved by transmitting a fixed pseudo-random bit pattern which the receiver will recognize in the presence of interference with a high probability. After time synchronization of the generators is established, the transmission of information may commence.

Interference is introduced in the transmission of the information-bearing signal through the channel. The characteristics of the interference depend to a large extent on its origin and can be categorized as broadband or narrowband relative to the bandwidth of the information bearing signal, and can either be continuous or discontinuous in time.

The two main techniques used to spread the spectrum are direct sequence spread spectrum and frequency hopping.

2.2 Direct Sequence Spread Spectrum

A block diagram of a direct sequence spread spectrum is as shown in Figure 3. In this method the information carrier is modulated with a code sequence which is pseudo randomly generated. The code has a much higher rate than that of the information signal. This results in a much greater bandwidth than for the information alone [15].

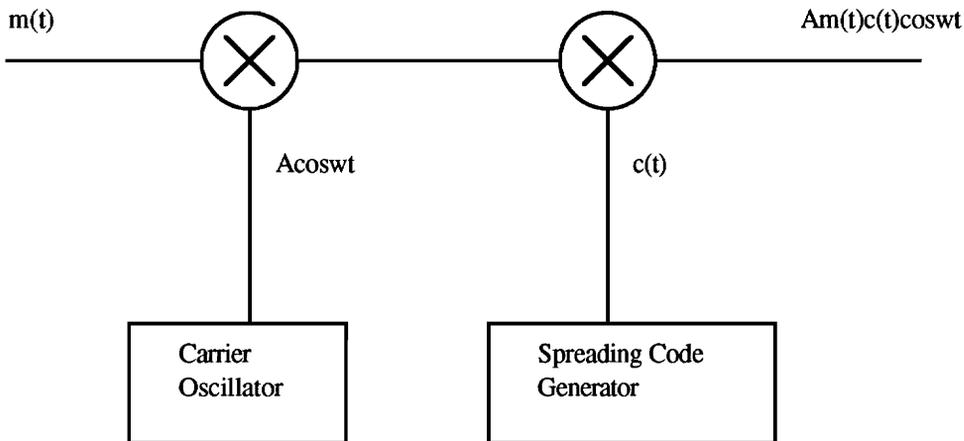


Figure 3. Direct Sequence Spread Spectrum Transmitter

Assume that the information rate is R bits/sec and the available channel bandwidth is W Hz. In order to utilize the entire available channel bandwidth, the phase of the carrier is shifted pseudo - randomly according to the pattern from the PN generator at a rate W times/sec. The reciprocal of W , denoted as T_c , defines the duration of a rectangular pulse which is called a chip and its time duration T_c is called the chip interval.

Corresponding to the transmission time of an information bit, if we define $T_b = 1/R$ the bandwidth expansion factor W/R may be expressed as

$$B_c = W/R = T_b/T_c$$

In practice T_b/T_c is an integer and the ratio of T_b and T_c is the number of chips per information bit.

These waveforms are illustrated in Figure 4.

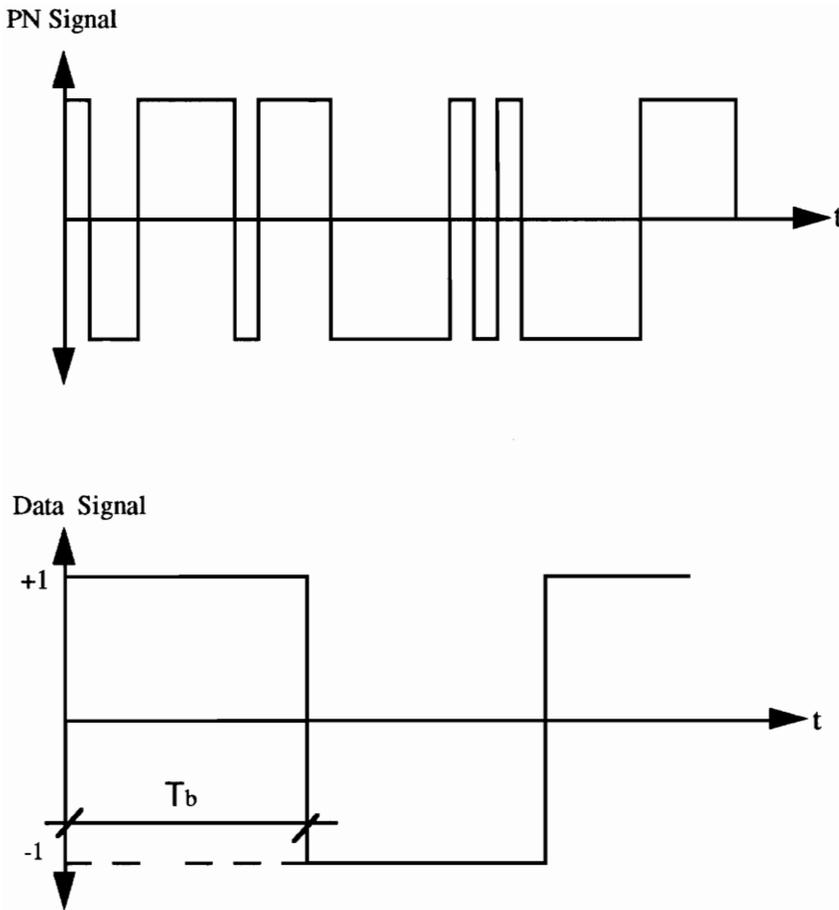


Figure 4. PN and data signals

2.3 Frequency Hopping Spread Spectrum

In this method, the available channel bandwidth is subdivided into a large number of contiguous frequency slots. In any signaling interval, the transmitted signal occupies one or more of the available slots. The selection of the frequency slots in each signaling interval is made pseudo-randomly according to the output from a PN generator transmitter hops to a set of frequencies within the operating band. At the receiver the local oscillator is synchronized to the transmitter sequence to facilitate recovery.

A block diagram of the transmitter and the receiver for a frequency hopped spread spectrum system is as shown in Figure 5. The modulation is usually either binary or M-ary

FSK. If binary FSK is used the modulator selects one of the two frequencies corresponding to the transmission of either a 1 or a 0. The resulting FSK signal is translated in frequency by an amount that is determined by the output sequence from the PN generator which in turn is used to select a frequency that is synthesized by the frequency synthesizer. This frequency is mixed with the output of the modulator and the resultant frequency translated signal is transmitted over the channel. Thus m bits from the PN generator may be used to specify upto $(2^m) - 1$ possible frequency translations.

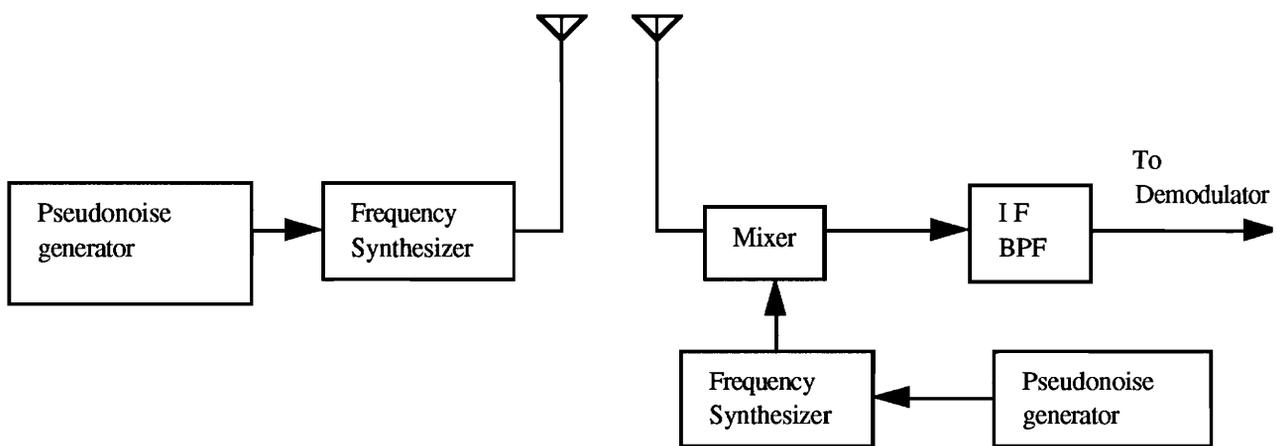


Figure 5. Frequency Hopped Spread Spectrum System

At the receiver, we have an identical PN generator, synchronized with the received signal, which is used to control the output of the frequency synthesizer. Thus the pseudo-random frequency translation introduced at the transmitter is removed at the receiver by mixing the synthesizer output with the received signal. The resultant signal is demodulated by means of an FSK demodulator. a signal for maintaining synchronism of the PN generator with the frequency translated received signal is usually extracted from the received signal [1].

Chapter 3. System Specifications and Hardware Design

Here a low cost LEOSAT system for low rate data communications is examined. The application will be to collect environmental and meteorological data from around 1000 terminals throughout Italy. The satellite will poll the remote terminals as it passes overhead, collects data and download it to the base station at some later time. The time the data needs to be stored on board the satellite may range from a few seconds to one whole day.

The initial use will be to collect environmental data and new applications are being explored.

3.1 System Specifications

The satellite will be placed in orbit approximately 800 km above the earth in a circular orbit. The uplink from the remote terminal to the satellite will operate at frequencies of about 149 MHz. Approximately 2 MHz of the bandwidth will be available for communication in this range. Four simultaneous 2400 bit/sec signals can be accommodated as transmissions along the uplink will occur at a rate of 2400 bits/sec. The downlink from the satellite to the central base station will operate at a frequency of 137 MHz, and at a data rate of 9600 bit/sec [1].

3.2 Design Constraints

1. Interference Rejection

As the system will not have exclusive use of the 149MHz band, other terrestrial

communication systems may result in interference. The system should be capable of suppressing this interference.

2. Low power availability

The remote base station may be located in areas in which a readily available power supply is not found easily. This limits the power consumption of the system.

3. Low cost terminals and off-the-shelf technology.

Since over 1000 terminals will be used the cost of each terminal should not be very high. Also any spread spectrum technique that is implemented should not add a big amount of money to the cost of the terminal. Since the system is scheduled to be deployed within the next couple of years application specific hardware cannot be developed. Any spread spectrum technique which is implemented must be built with tested readily available hardware.

The factors that can be exploited to meet the design constraints are the low data rate and the fact that communications are not real time.

3.3 System Design

Frequency hopping has been selected as opposed to direct sequence spread spectrum for the following reasons. Firstly frequency hopping is affected less in the presence of interference than direct sequence. Secondly it is easier to implement frequency hopping and a number of products are available to implement this method commercially.

A hop rate of 50 hops/sec has been selected. At 2400 kbits/sec, this results in 48 bits/hop and this hop rate can be achieved by the DDS synthesizer chip.

3.4 Hardware Design

The two most appropriate techniques that are available for frequency synthesis are direct digital synthesizers (DDS) and phase locked loop (PLL). DDS is a digital method while PLL is an analog method. The advantages and disadvantages of both the methods are discussed here. DDS provides a very fine frequency and phase resolution. DDS provides the ability to change frequency and phase rapidly while this is not the case with PLL. DDS provides a better frequency resolution than the PLL. The DDS also provides very fine phase resolution. On the other hand the output frequency of the DDS is dependent on the clock frequency and so is limited to around 25 - 30MHz as opposed to that of PLL. PLL can output much higher frequencies than the DDS. PLL can output frequencies in the range of hundreds of MHz. One disadvantage of the DDS is that it consumes more power than PLL. DDS units typically consume 2 - 5 Watts and the PLL consume around 0.5 Watts. The other disadvantage of a DDS is the cost. DDS chips cost around \$400 but require additional circuitry which increases the system cost to around \$1000. On the other hand the cost of a PLL chip is approximately \$5 and with all the required additional circuitry the total cost will be around \$25.

For the present application DDS was chosen rather than PLL due to its switching speed and settling time. When it comes to switching speed, with DDS, the rate at which the frequencies can be changed is limited by the clock speed since the DDS stores samples of the sinusoid. In the case of a PLL, the time response of the feedback loop determines the switching speed. If the frequency step is large it takes a longer time for the loop to adjust to the new frequency. While the typical hop rates for the DDS units range from several khops/sec to several Mhops/sec, the hop rates for PLL are in the range of 10 to 50 hops/sec.

3.5 Specific Hardware

The hardware for the present application consists of the interface board and the DDS board. The DDS board is used to simulate a transmitted spread spectrum signal. This can be considered more as a "test bed" rather than an actual system.

Typically a DDS chip consists of a phase accumulator followed by a sine look-up table as shown in Figure 6. This is followed by a digital-to-analog converter and a lowpass filter. A DDS operates on the principle that a digitized waveform of a given frequency can be generated by accumulating phase changes at a higher frequency. Sampling theory requires that the generated frequency be no more than one half of the clock frequency (Nyquist rate). The phase accumulator consists of a N-bit frequency register and a N-bit phase register. The phase accumulator accepts a N-bit binary word representing the output frequency and outputs a binary word indicating the instantaneous phase to the sine look-up table. The sine look-up table converts the phase accumulator output to a digitized sinusoid by outputting appropriate entries. As no computation is involved, this method provides the ability to change frequency and phase rapidly. In its simplest form this table consists of one complete cycle of a sine wave. The frequency resolution of the DDS is the lowest non-zero frequency it can output.

If F is the frequency of the clock and the number of input bits to the DDS chip is N , the frequency resolution is $F/(2^N)$. Thus by increasing N , the number of input bits, the resolution can be increased. As the output frequency is increased, the number of samples per sinusoid decreases. Since sampling theory states that at least two samples per cycle are required to reconstruct the output waveform, the maximum DDS fundamental output frequency is $F/2$.

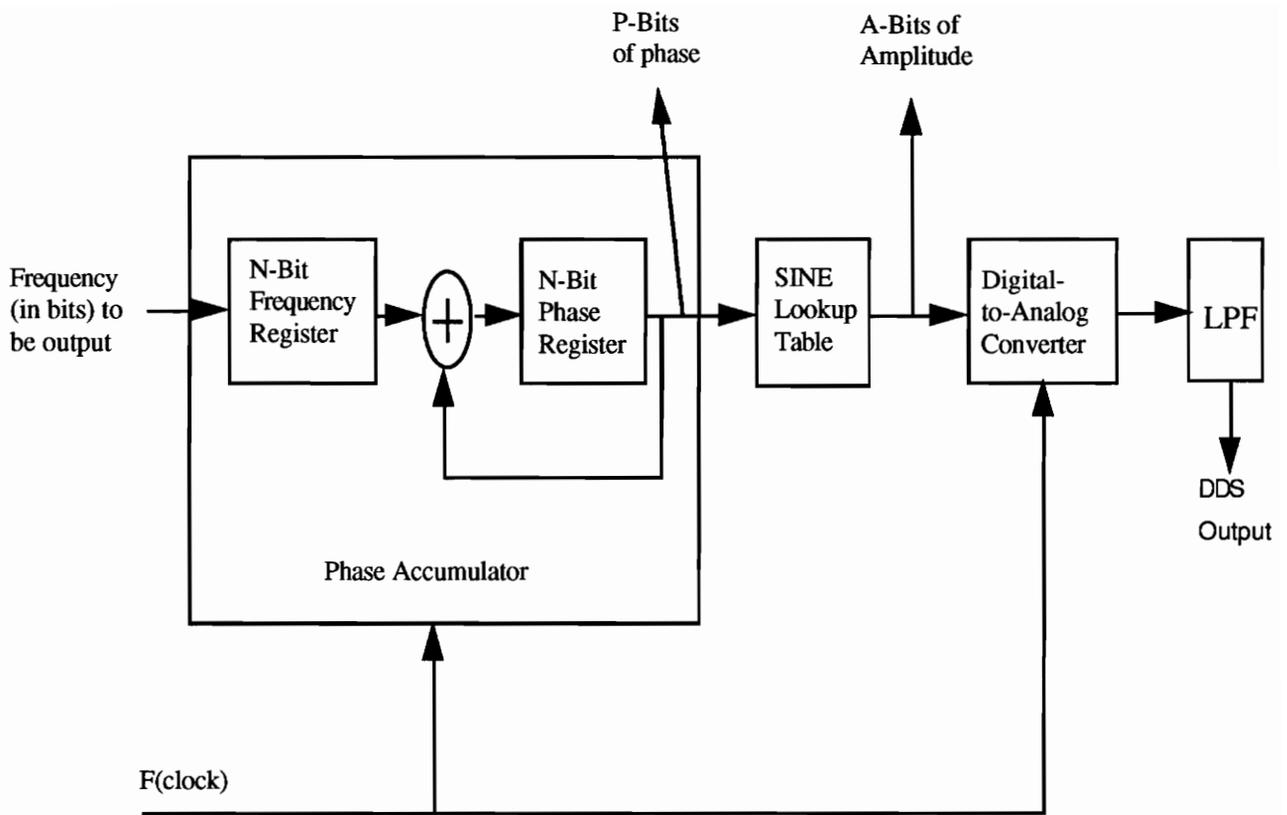


Figure 6. A Generic Direct Digital Synthesis System

The phase accumulation of a generated sine wave whose frequency is equal to $1/8$ th of the clock frequency is illustrated in Figure 7. The circle shows the phase accumulation process of $180/4$ at each clock cycle. The letters on the circle represent the phase value at a given time, and the sine wave shows the corresponding amplitude representation. The phase-to-amplitude conversion occurs in the on-chip sine computation function i.e. in the sine look-up table. Note that the phase increment added during each clock period is $180/4$ degrees, which equals $1/8$ th of a complete cycle.

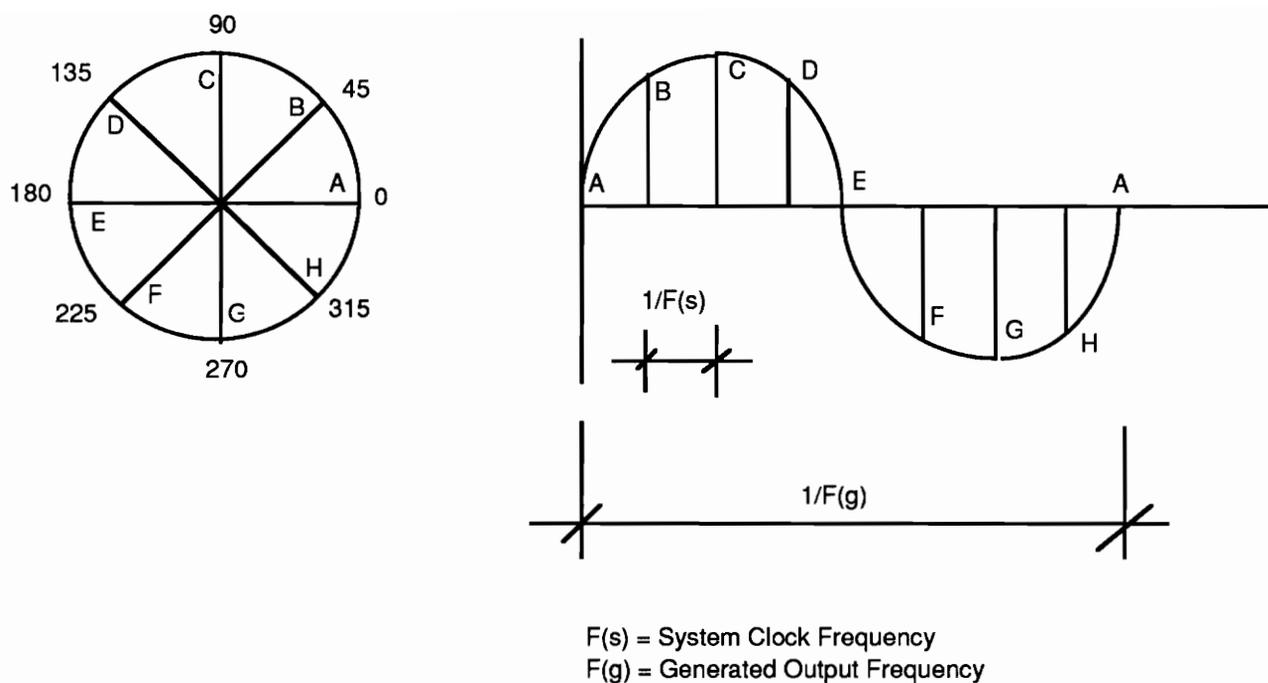


Figure 7. Generation of a sine wave from a sine look-up table

The control of the DDS board is achieved through the modulation port and the tuning port. These are 24 bit ports and the PC bus is 8 bits wide. To achieve a 24 bit output from the 8 bit bus we use the interface board. The interface board is a PC-AT bus interface. A schematic of the interface board used is shown in Figure 8. The interface board contains three 82C55 ICs as shown and additional circuitry to drive the board. The address space provided on the board is in blocks of four in the range of Hex 300-31F and is decoded. The address space used is from Hex 300 - 3B. The three I/O ports PA, PB, PC and the control port of the first 8255 are addressed individually by the addresses Hex 300, 301, 302 and 303. Likewise the ports of the second and the third 8255 are addressed by the addresses from Hex 304 - 3B.

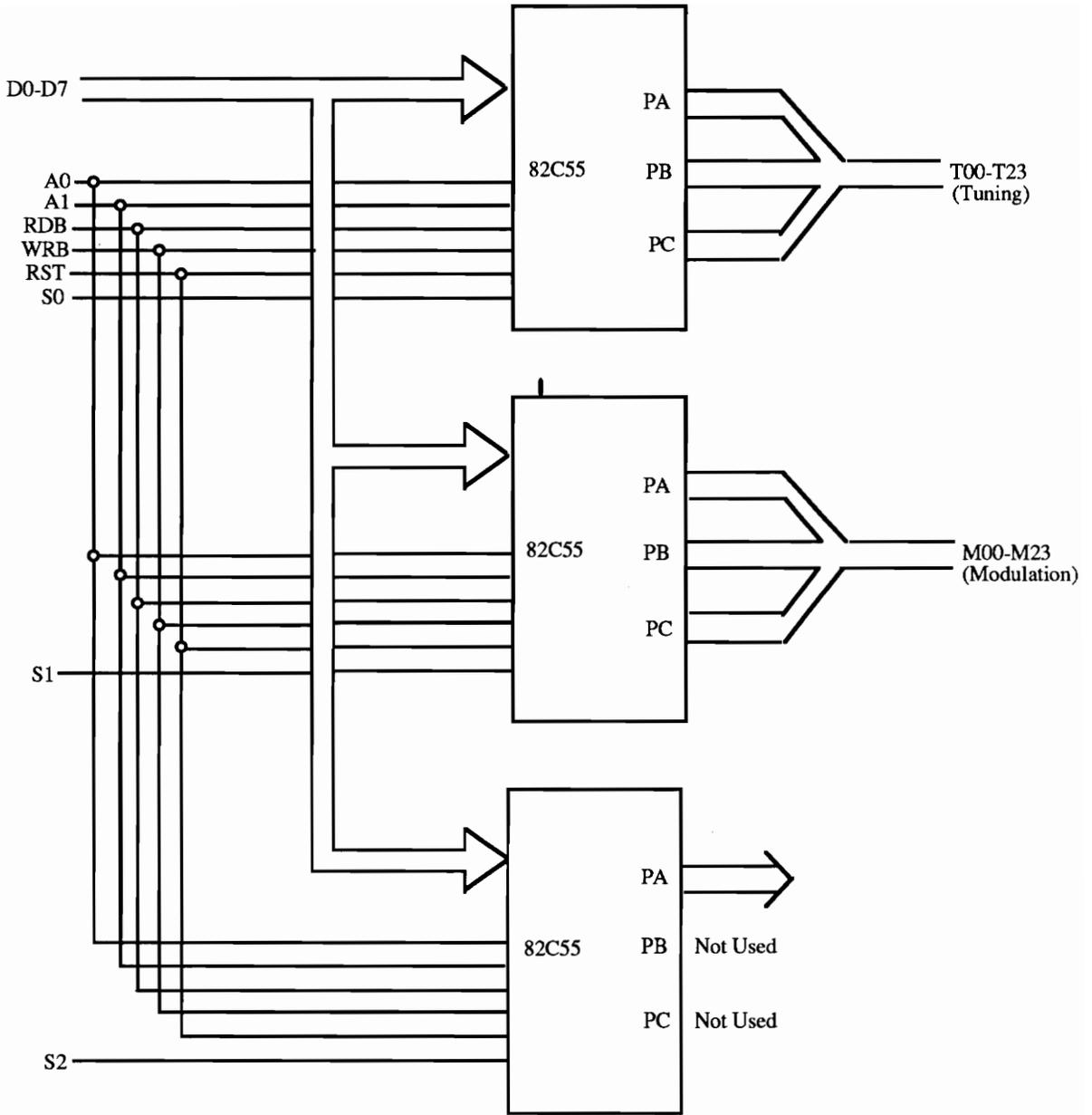


Figure 8. Schematic of the Interface board

All the 8255s are programmed such that the 3 ports PA, PB, and PC are in the output mode. The first 82C55 is used to input the tuning frequency to the tuning port and the second 82C55 is used to input to the modulation port. The third 8255 is used to select the mode in which the DDS board operates - either FM or PM. The ports B and C of this 8255 is not used.

The DDS board used is the DX 2070. The circuit diagram of the DDS board used is shown in Figure 9. The board consists of a digital programmable frequency synthesizer DX2250, two 2274 ROM map chips and one 12-bit-D/A converter TDC 1012. The clock frequency of the board is 20MHz which results in a frequency resolution of $20\text{MHz}/2^{24} = 1.19\text{Hz}$. The highest frequency the frequency synthesizer can output is determined by the Nyquist rate which turns out to be $20/2 = 10\text{MHz}$. Thus the frequency synthesizer can output any frequency in the range of 1.19Hz to 10MHz. The working of the board can be explained as follows.

The frequency synthesizer consists of a 24-bit frequency register and the 24-bit phase register. This is shown in Figure 10. Firstly the frequency synthesizer is set to either FM or PM using the control 8255. In the present board frequency control is achieved through a 24 bit tuning port. To achieve FSK, the frequency to be output is input to the tuning port. In the next clock cycle this 24 bit value is added to the value in the phase

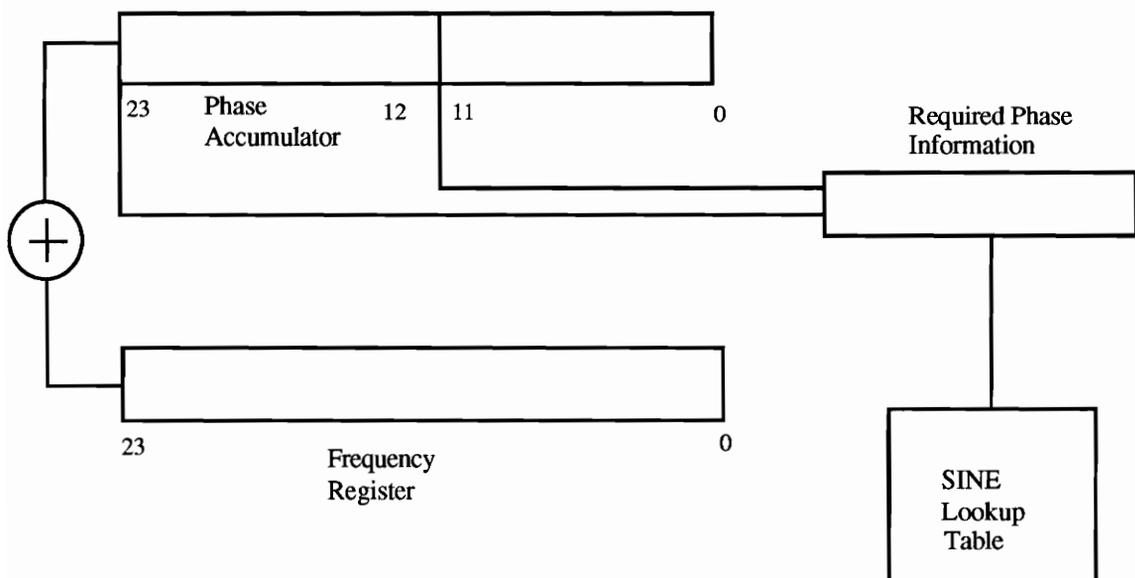


Figure 10. Schematic of the DDS board

accumulator and the result is stored in the phase accumulator. The carry bit beyond the 24th bit is discarded. The phase accumulator will have the required phase bits. The upper 12 bits of the phase accumulator is used to look up the amplitude information in the sine lookup table. Thus only if the upper 12 bits of the phase accumulator change will the output of the sine lookup table change i.e. for two consecutive clock cycles if the value in the upper 12 bits remains unchanged the value output to the sine lookup will be the same.

When a frequency of 1.19Hz is to be output, the 24 bit equivalent of 1.19Hz i.e. "000000000000000000000001" is input to the frequency register through the tuning 8255. This value is added to the value in the phase accumulator (the previous value in the phase accumulator was zero) and the result is stored in the phase accumulator. Thus the phase accumulator will have the entry "000000000000000000000001". During the next clock cycle the value in the frequency register is again added to the phase accumulator and the result stored in the phase accumulator. Thus now the phase accumulator will have the value "000000000000000000000010". But since the upper 12 bits are used in the sine lookup table, the same value will be output by the sine lookup table in both the cases. In this case the upper 12 bits of the phase accumulator change once in 2^{12} clock cycles. Thus the same value is repeatedly output 4096 times. This output wave is the closest approximation to an ideal sine wave that can be produced by the board being used.

When a frequency of 10MHz is to be output, initially a 24 bit representation of "0" will be output first since the phase accumulator will have an initial value of zero. During the next clock cycle the 24 bit representation of 10MHz (which was in the frequency register) is added to the phase accumulator and the result is stored in the accumulator. Now the upper 12 bits which will have the value "100000000000" will be input to the sine lookup and the corresponding value will be output. During the next clock cycle again the 24 bit representation of 10MHz (which was in the frequency register) is added to the phase accumulator and the result is stored in the accumulator. Now the upper 12 bits, which will

have the value "000000000000" (since the carry bit beyond the 24th bit is discarded), is output thus producing a 10MHz sine wave. Thus the upper 12 bits of the phase accumulator change every clock cycle and at any clock cycle either a zero amplitude or the maximum amplitude is output. This output wave will be a very poor approximation of a sine wave (in fact, it will be a square wave) and is the poorest approximation to an ideal sine wave that can be produced by the board. The degree of smoothness of sine waves that can be output will be between those of 1.19Hz and 10MHz.

The sine lookup here is made up by two 12 bit 2274 ROM chips. In these chips are hard coded the values of the amplitude of the sine wave corresponding to the particular phase value. The two chips, have a total of 2^{12} entries. Thus if a frequency of 1.19Hz is to be output all the values in the ROM chips will be made use of repeatedly and the output wave will be closest to a smooth sine wave. This is the smoothest sine wave that can be produced by the board. As the frequency is increased the number of values in the ROM chips will decrease and so the output wave will be an approximation of a sine wave. This approximation puts an upper limit to the frequency the board can output and this is shown in later sections.

Chapter 4. Results

The two main digital modulation techniques - frequency shift keying and phase shift keying were displayed. The disadvantage of the direct digital synthesis method is also illustrated. The software shown in Appendix A was run and the results were recorded on a spectrum analyzer and an oscilloscope and these results are illustrated below.

4.1 Illustrations

The waveform displayed on the spectrum analyzer to illustrate frequency shift keying is shown below in Figure 11.

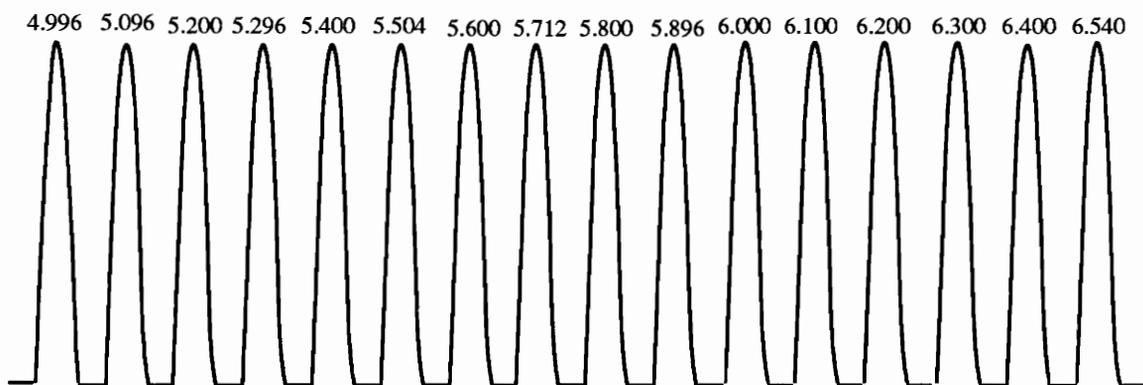


Figure 11. Illustration of FSK

The center frequency used was 5MHz and a delta frequency used was 0.1MHz. The number of bits per symbol used was 4. Thus 16 possible frequencies starting from 5MHz to 6.5MHz need to be transmitted. This can be observed Figure 11.

Another example with the same values with the number of bits per symbol as 2 is shown in Figure 12.

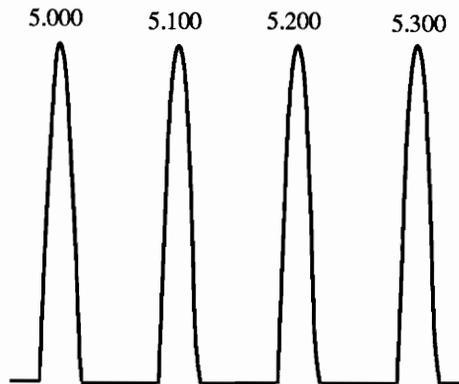


Figure 12. Illustration of FSK

The below waveform in Figure 13, obtained on the oscilloscope, illustrates binary phase shift keying. In order to be able to obtain the waveform on the oscilloscope a low frequency of 500Hz was chosen. Since the only PSK that can be seen on the oscilloscope in our lab is BPSK one bit per symbol was used. The software can simulate other PSK methods like QPSK, MPSK. Since this cannot be captured on the oscilloscope only BPSK has been demonstrated.

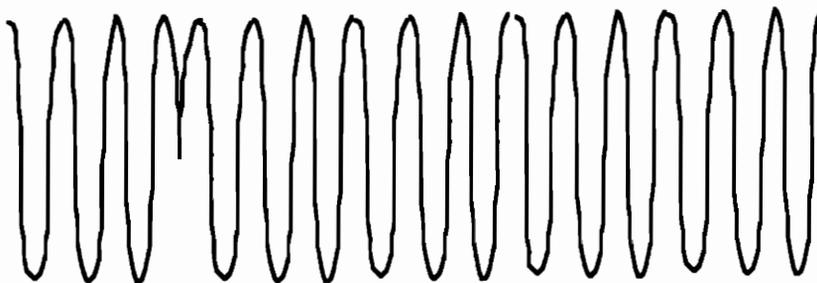


Figure 13. Illustration of BPSK

4.2 Disadvantage of DDS method.

One of the disadvantages of using the DDS technique to simulate the transmitter is that it puts an upper limit on the frequency at which we can operate with it. This is because it approximates a continuous wave with discrete samples. Due to this sampling harmonics are produced and restricting the frequency at which we can operate. This is illustrated with Figure 14.

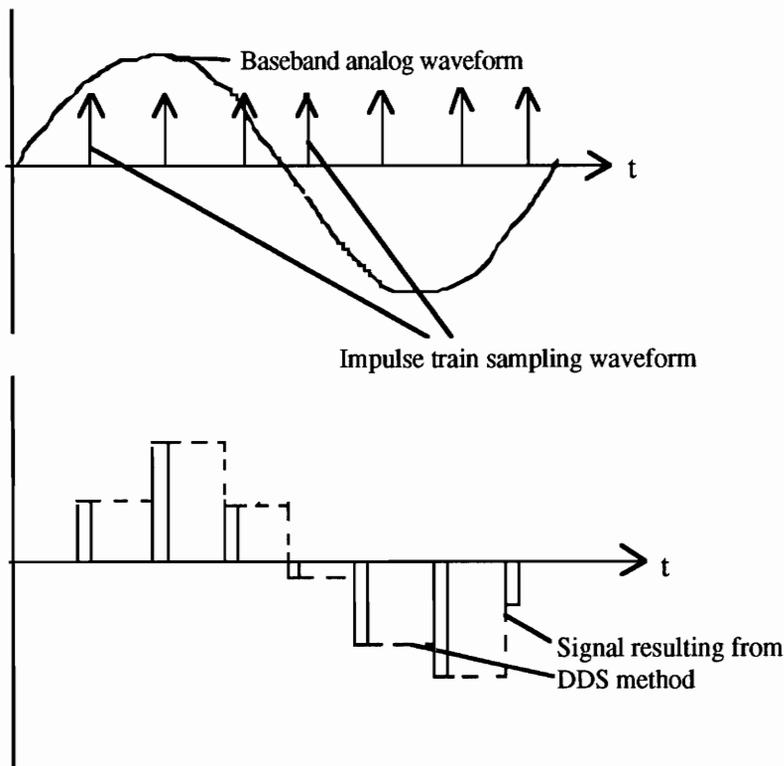


Figure 14. Calculation of harmonics

Consider the sine wave shown. The impulse train sampling waveform is shown along with the sine wave. The thick pulses shown are will result when flat top PAM is used and the dotted lines shown is the waveform resulting from the DDS. The spectrum can be found as follows.

If the baseband, $w(t)$, is an analog waveform bandlimited to B Hz, the instantaneous sampled PAM signal is given by

$$w_s(t) = \sum_{-\infty}^{\infty} w(kT_s)h(t - kT_s)$$

where $h(t)$ denotes the sampling pulse shape and is given by

$$h(t) = 1, t < T_s$$

The spectrum for the above waveform can be obtained by taking the Fourier transform of the above equation. Rewriting in a more convenient form

$$w_s(t) = \sum_{-\infty}^{\infty} w(kT_s)h(t) * \delta(t - kT_s)$$

$$w_s(t) = h(t) \sum_{-\infty}^{\infty} w(kT_s) * \delta(t - kT_s)$$

$$w_s(t) = h(t) * \sum_{-\infty}^{\infty} [w(t)\delta(t - kT_s)]$$

Thus the spectrum can be written as

$$w_s(f) = h(f) \left[w(f) * \sum_k e^{-j2\pi kT_s f} \right]$$

But the sum of the exponential functions is equivalent to the Fourier series expansion where the periodic function is an impulse train. So we have

$$\frac{1}{T_s} \sum_k \delta(f - kf_s) = \frac{1}{T_s} \sum_{-\infty}^{\infty} c_n e^{j(2\pi nT_s)f}$$

Using this

$$w_s(f) = h(f) \left[w(f) * \frac{1}{T_s} \sum_k \delta(f - kf_s) \right]$$

$$w_s(f) = \frac{1}{T_s} h(f) \left[w(f) * \sum_k \delta(f - kf_s) \right]$$

$$w_s(f) = \frac{1}{T_s} h(f) \sum_{-\infty}^{\infty} w(f - kf_s)$$

where

$$h(f) = F[h(t)] = \tau \left(\frac{\sin(\pi \tau f)}{\pi \tau f} \right)$$

The above equation shows that the harmonics should be centered around $(f - kf_s)$, each frequency weighted by the Sinc function $h(f)/T_s$. As we move on either sides of 0Hz the amplitude of the harmonics should decrease. This was verified practically and is illustrated below.

The sampling frequency of the DDS board is 20MHz(f_s) and the frequency of the baseband sine wave used was 2MHz. Thus according to the above analysis the samples should be centered at 20MHz, -20MHz, 40MHz, -40MHz and so on which was observed on the spectrum analyzer as shown below in Figure 15.

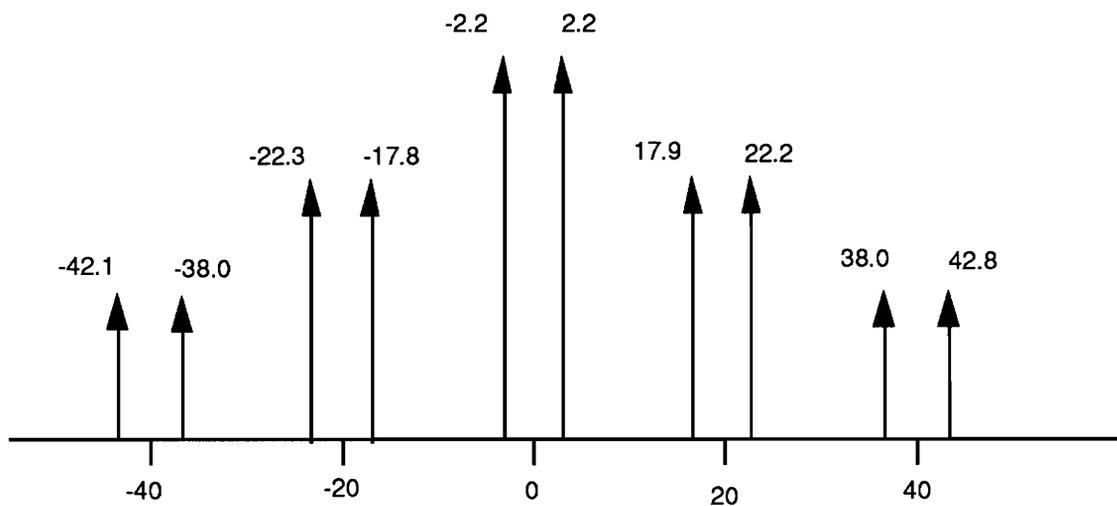


Figure 15. Illustration of harmonics

Another case with the baseband frequency as 5MHz is shown below in Figure 16.

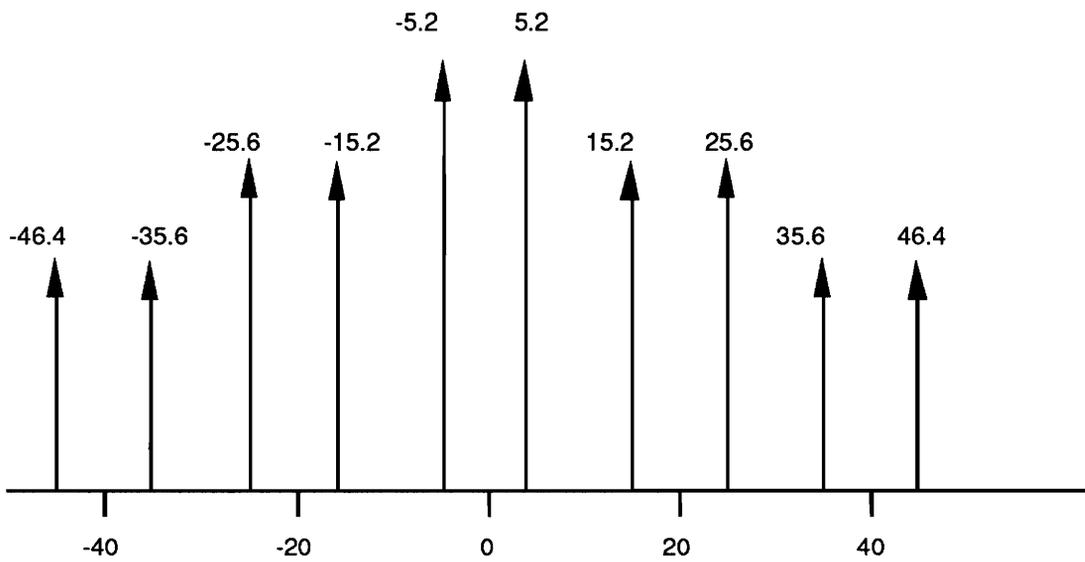


Figure 16. Illustration of harmonics

Chapter 5. Conclusions

In this project the reasons for the emergence of low earth satellite systems were considered. The various aspects of low earth satellite communications were also considered. The concept of direct digital synthesis was introduced.

Hardware was built using the direct digital synthesis board DX 2070 manufactured by Qualcomm Inc. The board consisted of a digital programmable frequency synthesizer DX 2250, two 2274 ROM map chips and one 12 bit D/A converter. The board has a clock frequency of 20 MHz and the maximum frequency the board can generate is 8MHz. In order to make the board compatible with the IBM PC, an interface board was designed.

Software was written in 'C' to program the DDS board. The software contains routines to program the board to any particular frequency or phase and to perform digital modulation techniques.

The reason for the advantages of the direct digital synthesis system, very fine frequency and phase resolution and the ability to change frequency and phase rapidly, is seen. The disadvantages of the direct digital synthesis method are its cost and frequency limitation. The limit of frequency has been demonstrated experimentally to be approximately 40% - 50% of the clock frequency. The frequency limitation with the DX 2070 was found to be 8MHz.

References

- [1] B. D. Woerner, M. J. Feurestein and T. Pratt, "*Spread Spectrum for Low Cost LEOSAT Communications in the VHF Band*", Final Report, 1992.
- [2] T. Pratt and C. W. Bostian, "*Satellite Communications*", John Wiley and Sons: New York, 1986.
- [3] R. E. Zeimer and R. L. Peterson, "*Digital Communications and Spread Spectrum Systems*", New York: Macmillan, 1985.
- [4] L. W. Couch II, "*Digital and Analog Communication Systems*", New York: Macmillan, 1990.
- [5] T. S. Rappaport, "*The Wireless Revolution*", IEEE Communications Magazine, vol. 29, no. 4, pp. 66-79, April 1991.
- [6] R. C. Dixon, "*Spread Spectrum Systems*", New York: Wiley Interscience, 1990.
- [7] D. L. Schilling, R. L. Pickholtz and L. B. Milstein, "*Spread Spectrum goes commercial*", IEEE Spectrum, August 1990.
- [8] R. A. Scholtz, "*The Origins of Spread Spectrum Communications*", IEEE Transactions on Communications, vol. 30, no. 5. May 1982.
- [9] G. Maral, J. D. Ridder, B. G. Evans and M. Richharia, "*Low Earth Orbit Satellite Systems For Communications*", International Journal of Satellite Communications, vol. 9, Feb 1991.
- [10] R. L. Pickholtz, D. L. Schilling and L. B. Milstein, "*Theory of Spread-Spectrum Communications--A Tutorial*", IEEE Transactions on Communications, vol. 30, no. 5. May 1982.
- [11] D. Verhulst, M. Mouly and J. Szpirglas, "*Slow Frequency hopping Multiple Access for Digital Cellular Radiotelephone*", IEEE Journal on Selected Areas in Communications, vol.2,no. 4, July 1984.
- [12] R. A. Scholtz, "*The Spread Spectrum Concept*", IEEE Transactions on Communications, vol. 25, no. 8. Aug. 1977.

[13] L. A. Gerhardt and R. C. Dixon, "*Introduction-Special Issue on Spread Spectrum Communications*", IEEE Transactions on Communications, vol. 25, no. 8, Aug. 1977.

[14] P. Wood, "*Mobile Satellite Services for Travelers*", IEEE Communications Magazine, Nov 1991.

[15] John G. Proakis, "*Digital Communications*", McGraw Hill, Second Edition.

[16] "*Workshop on Personal Satellite Communication Systems*", First International Conference on Universal Personal Communications, September 29, 1992.

Appendix A. Software

In this section the basic routines required to use the DDS board is included.

```
#include <stdio.h>
#include <stdlib.h>
#include <dos.h>
#include <conio.h>
#include <ctype.h>

typedef unsigned int uint;
typedef unsigned long ulong;

/* initialising the 8255 */
void init_tun_8255()
{
    int i;
    int base = 0x300 ;
    outp(base + 3,0x80);
    for(i = 0;i < 3;i++)
        outp(base + i,0x00);
}

void init_mod_8255()
{
    int i;
    int base = 0x304;
    outp(base + 3,0x80);
    for(i = 0;i < 3;i++)
        outp(base + i,0x00);
}

void init_ctrl_8255()
{
    int base = 0x308;
    outp(base + 3,0x80);
}
```

```

    outp(base,0x00);
}

/* initialise the DDS board */
/* have to set the pc mod port to either FM or PM using the third 8255 */
void set_fm()
{
    int base = 0x308;
    outp(base,0x03);
}

void set_pm()
{
    int base = 0x308;
    outp(base,0x07);
}

/* routine to set the DDS board to a particular frequency */
/* assume the 24 bit frequency is in a long integer variable */

void set_freq(ulong data)
{
    int base = 0x300;
    int i;
    uint state;

    for(i = 0;i < 3;i++)
    {
        state = (data & 0XffL);
        data = data >> 8;
        outp(base + i,state);
    }
}

void mfsk(int n,ulong *freq,int ts,int size,uint *data)
{
    int i,j;
    uint mask,sym_count,w;

    mask = ~(~0 << n);
    sym_count = (16/n);

    for( i = 0; i < size; i ++)
    {
        w = data[i];
        for (j = 0; j < sym_count; j++)
        {
            set_freq(freq[w & mask]);
            w = w >> n;
        }
    }
}

```

```

        delay (ts);
    }
}
/* routine to set the board to a certain phase */
/* assume that the 12 bit phase is in a unsigned integer variable */

void set_phase(uint data)
{
    int base = 0x304;
    outp(base + 1, (data & 0xff));
    outp(base + 2, ((data >> 8) & 0xff));
}

void mpsk(int n,int ts, uint size, uint *phase, uint *data)
{
    int i,j;
    long int state;
    uint w,mask, sym_count;
    mask = ~(~0 << n);
    sym_count = (16/n);
    for (i = 0; i < size; i++)
    {
        w = data[i];
        for (j = 0; j < sym_count; j ++)
        {
            state = (w & mask);
            set_phase(phase[state]);
            w = w >> n;

            delay (ts);
        }
    }
}

main()
{
    double freq_start,freq_delta,base_freq;
    ulong start,delta,start_freq,*freq;
    uint *data, size,*phase,start_phase,s_phase;
    int bits,words,ts,N,i,j,n;
    char bit_seq;

    init_ctrl_8255();
    init_tun_8255();
    init_mod_8255();

    printf("FSK or PSK: ");
    scanf("%[FfSsKkPp]", &dig_mod);

```

```

if (toupper(dig_mod) == 'FSK')
{
    set_fm();

    printf("frequency (Hz):");
    scanf("%lf", &freq_start);

    printf("delta (Hz): ");
    scanf("%lf", &freq_delta);

    start = ulong(freq_start * 0.8388608);
    delta = ulong(freq_delta * 0.8388608);

    printf("dwell time (ms): ");
    scanf("%i",&ts);

    printf("bits per symbol (1,2,4): ");
    scanf("%i", &bits);
    N = (1 << bits);

    freq = (ulong *)malloc(sizeof(ulong)*N);
    for(i = 0; i < N; i++)
    {
        freq[i] = (start + (delta * i));
    }

    printf("How many words: ");
    scanf("%i",&words);
    data = (uint *)malloc(sizeof(uint)*words);

    printf("R-random or P-pattern: ");
    fflush(stdin);
    scanf("%[RrPp]",&bit_seq);

    if (toupper(bit_seq) == 'R')
    for(i = 0;i < words;i++)
        {
            data[i] = ulong(rand());
        }
    else if (toupper(bit_seq) == 'P') {
        if (bits == 1)
            for(i = 0; i<words;i++)
                {
                    data[i] = 0x5555;
                }
        else if (bits == 2)
            for(i = 0; i < words; i++)
                {
                    data[i] = 0x1b1b;
                }
    }
}

```

```

        else if (bits == 4) {
            j = 0;
            for(i = 0; i < words/4; i++)
            {
                data[j++] = 0x0123;
                data[j++] = 0x4567;
                data[j++] = 0x89AB;
                data[j++] = 0xCDEF;
            }
            i = words % 4;
            if (i > 0)
                data[j++] = 0x0123;
            if (i > 1)
                data[j++] = 0x4567;
            if (i > 2)
                data[j++] = 0x89AB;
        }
    }
    mfsk(bits,freq,ts,words,data);
}

else if (toupper(dig_mod) == 'PSK'){
    set_pm();

    printf("frequency(Hz): ");
    scanf(" %lf",&base_freq);
    start_freq = ulong(base_freq * 0.8388608);
    set_freq(start_freq);

    printf("starting_phase(degrees): ");
    scanf(" %d",&s_phase);

    start_phase = (s_phase/360) * 4096; 2 ** 12

    printf("bits per symbol(1,2,4): \n");
    scanf(" %d", &bits);
    N = 1 << bits;

    printf("dwell time(ms): ");
    scanf(" %d", &ts);

    phase = (uint *)malloc(sizeof(uint) * N);
    for(i = 0;i < N;i++)
    {
        phase[i] = (start_phase + i*(4096*(16 >> bits)));
    }

    printf("How many words: ");
    scanf(" %d", &words);
    data = (uint *)malloc(sizeof (uint)*words);

```

```

printf("R - random or P - pattern :");
fflush(stdin);
scanf(" %[RrPp]", &bit_seq);

if (toupper(bit_seq) == 'R')
    for(i = 0; i < words; i++)
    {
        data[i] = ulong(rand());
    }
    else if (toupper(bit_seq) == 'P') {
    if (bits == 1)
    for( i = 0; i < words; i++)
    {
        data[i] = 0x5555;
    }
    else if (bits == 2)
    for(i = 0; i < words; i++)
    {
        data[i] = 0x1b1b;
    }
    else if (bits == 4) {
    j = 0;
    for(i = 0; i < words/4; i++)
    {
        data[j++] = 0x0123;
        data[j++] = 0x4567;
        data[j++] = 0x89AB;
        data[j++] = 0xCDEF;
    }
    i = words % 4;
    if (i > 0)
        data[j++] = 0x0123;
    if (i > 1)
        data[j++] = 0x4567;
    if (i > 2)
        data[j++] = 0x89AB;
    }
}
mpsk(bits, ts, size, phase, data);
}
}

```

Appendix B. Instructions for use

The DDS board with the interface board is plugged into the PC. The PC needs to be loaded with a 'C' compiler as the software is written in 'C'. The output of the board is taken from the output port. An SMA connector is inserted on to the output port and the input port of the spectrum analyzer. The two SMAs are connected by a coaxial cable. After the above connections are made the software is run. The spectrum analyzer and the oscilloscope used is manufactured by Tektronics Incorporated.

When the software is executed the user is prompted with the following menu. The bold letters are those that need to be entered by the user.

FSK or PSK: **FSK**

frequency(Hz): **5000000**

delta(Hz): **100000**

dwel time (ms): **2000**

bits per symbol (1, 2 or 4): **2**

how many words: **128**

R - random or P - pattern: **P**

The program starts executing and the waveforms can be obtained on the spectrum analyzer or the oscilloscope.