RESEARCH ARTICLE OPEN ACCESS

FPGA Implementation of A BPSK Modem

Prabhakar. Rapaka*, Vijaya Babu**, G. J. Chitra***.
*Department of Electronics and Communication Engineering, JNT. University, Hyderabad-79
**Department of Electronics and Communication Engineering, JNT. University, Hyderabad-79
*** Department of Electronics and Communication Engineering, JNT. University, Hyderabad-79

ABSTRACT
This paper presents the Implementation of BPSK Modem with a data rate of 1 Mbps (message signal with 4.8 KHz and carrier (NCO) signal with 5MHz) will be modulate with a carrier signal 5MHz by using a Mixer. The BPSK Demodulator accepts digital data input from modulator and performs multiplication on the input data with carrier. It has two main blocks,
The Carrier Recovery will get by using Costas loop and Symbol Timing Recovery using Early Late gate Algorithm. Further it performs Raised cosine filtering (RCF) with a desired Roll-off factor (0.25). Costas loop Consists of Mixer, Loop Filter and NCO (Numerical Controlled Oscillator). The outputs of RCF are apply to AGC (Automatic Gain Control) through that we will get the quantized outputs.
The project will be implemented using VHDL. Simulation will be done to verify the functionality and synthesis will be done to get the NETLIST. Simulation and synthesis will be done using Xilinx ISE 9.2I Tools.

I. INTRODUCTION

In the digital communication, the information is encoded as a set of discrete values (1 and 0). During transmission the information contained in analogue signals will be degraded by noise. Conversely, unless the noise exceeds a certain threshold, the information contained in digital signals will remain intact. Noise resistance represents a key advantage of digital signals over analogue signals. An Analog signal is defined as "The continuous amplitude, phase and frequency with respect variable time".

A Digital signal is defined as "The discrete levels of amplitude, phase and frequency with respect variable time". A Discrete signal is a time series consisting of a sequence of quantities. The digital modulation techniques used to modulate digital information so that it can be transmitted via microwave, satellite or down a cable pair are different to that of analogue transmission. The data transmitted via satellite or microwave is transmitted as an analogue signal. The techniques used to transmit analogue signals are used to transmit digital signals. The problem is to convert the digital signals to a form that can be treated as an analogue signal that is then in the appropriate form to either be transmitted down a twisted cable pair or applied to the RF stage where is modulated to a frequency that can be transmitted via microwave or satellite. Binary-phase shift key modulation (BPSK) modulation technique that can be utilised in a modem.[6] Many modern modems have their origin in deep space telecommunications systems of the 1960s. Digital modulation formats that have high doppler immunity are typically used Waveform complexity tends to be low, typically binary phase shift keying Error correction varies mission to mission, but is typically much stronger than most landline modems.

I.1 Generic BPSK Modem

I.1.1 BPSK Modulator

In this modulation one has as possible results two exit phases for the carrier with a single frequency. An exit phase represents a logical 1 and the other one a logical 0. As the input digital signal changes the state, the phase of the exit carrier moves between two angles that lie 180° outside of phase.Figure1.1 shows a simplified block diagram of a BPSK modulator. The coded signal enters to a multiplexer that commutes the phase of the carrier signal. Depending on the logical condition of the digital input, the carrier is transferred to the output, either in phase or at 180° outside of phase, with the reference carrier oscillator. The output spectrum of a BPSK modulator is only a double lateral band signal with suppressed carrier, where the high and low lateral frequencies are separated from the carrier frequency for a value that is a half of the bit rate. Therefore, the minimal required bandwidth to allow the worst case of the BPSK output signal is similar to the input bit reason.

Figure 1.2 shows the output phase versus the time relationship for a BPSK wave form. The input signal can be $+\cos(\omega t)$ or $-\cos(\omega t)$. The recovery circuit detects and regenerates a carrier signal, as in frequency as in phase with the carrier of the original transmitter. The balanced modulator is a product detector whose output is the product of the two inputs (the BPSK signal and the recovered carrier). Since the only possible outputs are the signals $\cos(\omega t)$ and $-\cos(\omega t)$, the product detector's possible outputs will be:
$\cos2(\omega t) = ½ + ½ \cos(2\omega t)$,
$\cos2(\omega t) = - ½ - ½ \cos(2\omega t)$,

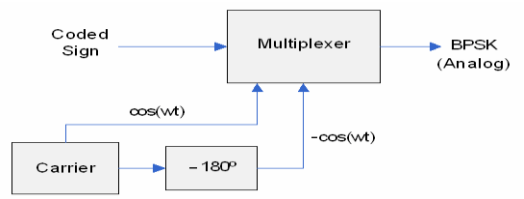


Fig1.1 Generic BPSK Modulator

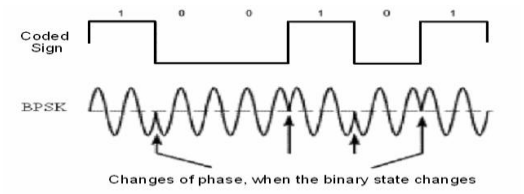
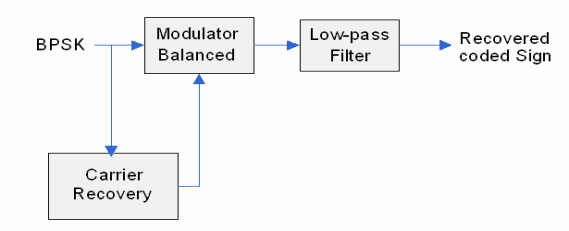


Figure 1.2: BPSK Modulation.

I.1.2 BPSK Demodulator

The demodulation is performed according to the scheme shown below.



FigureI.3: Generic BPSK Demodulator Block Diagram.

To demodulate the signal coming from the channel, a (Multi) block that multiplies the signal for the recovered carrier is used. The pass-low filter FIR separates the continuous signal of + ½ amplitude recovered from the demodulated complex signal and allows to select the zero frequency signal (+1/2 or -1/2). This filter is a pass-low filter, Since at the output of the filter there are signals with ½ amplitude and with ruffled border in each pulse, a comparison block that will provide levels of voltage of ones and zeros, and will avoid the curly of such pulses will be placed.
This paper presents the Implementation of BPSK Modem with a data rate of 1 Mbps (message signal with 4.8 KHz and carrier (NCO) signal with 5MHz) will be modulate with a carrier signal 5MHz by using a Mixer. The BPSK Demodulator accepts digital data input from modulator and performs multiplication on the input data with carrier. It has two main blocks, The Carrier Recovery will get by using Costas loop and Symbol Timing Recovery using Early Late gate Algorithm. Further it performs Raised cosine filtering (RCF) with a desired Roll-off factor (0.25). Costas loop Consists of Mixer, Loop Filter and NCO (Numerical Controlled Oscillator). The outputs of RCF are apply to AGC (Automatic Gain Control) through that we will get the quantized outputs..

II. Implementation of a BPSK Modem

II.1 BPSK modulator module

Data to Binary Phase Shift Keyed (BPSK) signals. BPSK is used extensively in high-speed data transmission., students gain an understanding of the BPSK modulation process, as well as data rate limitations of a BPSK system. Digital communications instructional modules offer superior training in Telecommunications technology. Modules in this system include: (BPSK). The stem is designed to enable students to quickly assemble a functioning communications network. Fully-compatible signal levels and protocols are fed among modules from front panel con connection points. Because the system is composed of prewired, functioning modules, and because connections between modules are made with shielded cables, the routing and trimming of student connections do not affect system performance or measurements. Important test points or test busses are brought to 9-pin computer compatible connectors on the front panels of modules for easy access. Test points and signal outputs are short circuit and overvoltage protected. Power cable clutter is eliminated by means of a unique power connector system. On Enclosure / Supply Regulator module that is part of the instrumentation for Digital Communications.

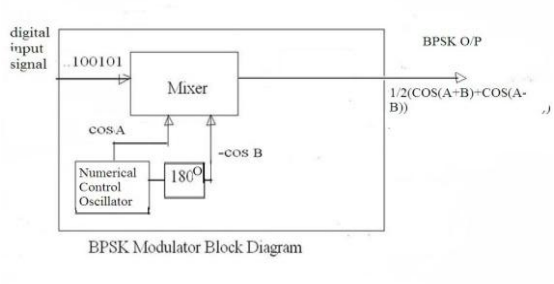


Fig.II.1.BPSK Modulator block diagram.

In BPSK modulation the carrier phase acquires two discrete states (0° and 180°), Which correspond to one bit of the modulation signal. Therefore the symbol period is equal to the

- bit period *Ts* = *Tb*.

The BPSK modulated output is expressed as:

$f(t) = m(t)\cos(2*pi*fct+ 0°)$,

$f(t) = m(t)\cos(2*pi*fct+ 180°)$,

where $m(t)= \pm 1$, fc = carrier frequency.

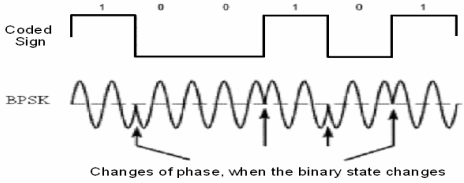


Fig.II.2.BPSK Modulation wave form.

II.2 BPSK demodulator module

BPSK Demodulator demodulates the BPSK signals from the BPSK Modulator and recovers the original data signal. State-of-the-art demodulation techniques with the Costas Loop are utilized. Gain an understanding of BPSK demodulation techniques, as

well as phase ambiguity and bit error rate measurement.

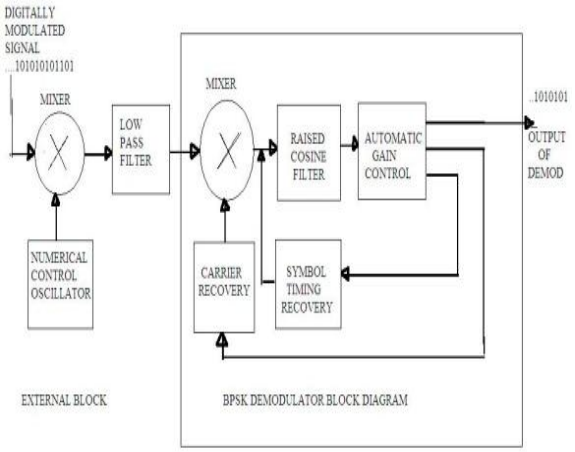
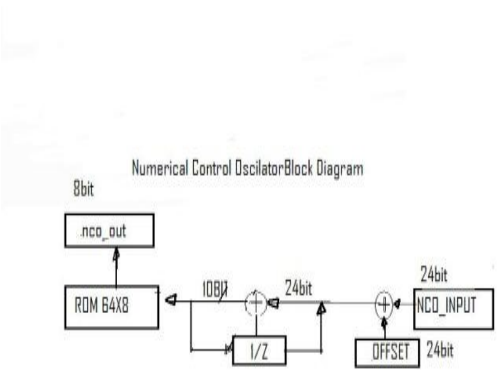


Fig II.3.BPSK Demodulator Block diagram

Costas loop: Costas loop is used for carrier phase recovery. It consists of mixer, LPF and Numerical Controlled Oscillator (NCO). Carrier recovery is required since the receiver typically does not know the exact phase. Implementation of a Costas loop, a local NCO the carrier phase based on the output of a mixer.

NCO (Numerical Controlled Oscillator): In a complete coherent receiver implementation, carrier recovery is required since the receiver typically does not know the exact phase and frequency of the transmitted carrier. In an analog system this recovery is often implemented with a voltage-controlled oscillator (VCO) that allows for precise adjustment of the carrier frequency based on the output of a phase-detecting circuit. In digital application, this adjustment is performed with a numerically-controlled oscillator (NCO).



FigII.4.NCO functional diagram

NCO Implementation

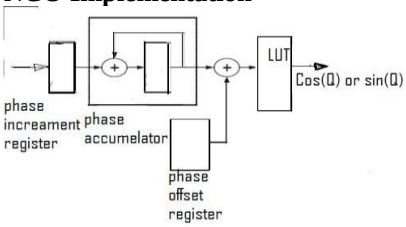


Fig.II.5 NCO implementation diagram

Phase Increment Register: The phase increment register stores the phase value (Δ) that gets added up to the accumulated phase at every clock cycle. The phase increment linearly decides the frequency of the output signal. Hence, this input can be used for frequency shift keying (FSK) modulation. The phase increment is either fixed or read dynamically from an input port, fskin , depending on how the NCO is configured. The output frequency is always a fraction of the clock frequency of the system.

Phase Accumulator: The phase accumulator computes the phase angle value that is used to address the look-up tables used for the output sine signal generation. The phase angle at any cycle is equal to the phase angle at the last cycle plus the phase increment. For cycle i the width of the accumulator is decided by the user parameter, "Phase resolution". For a given accumulator width, phase resolution is highest when the phase increment is equal to 1 and reduces for values greater than 1.

Phase Shift Keying: A constant phase input is added to the accumulated phase before addressing the look-up table. This is useful for implementing phase shift keying (PSK) modulation of the NCO output.

Quantize: The output of phase accumulator (or the optional PSK or dithering module) drives the quantizer The quantizer scales down the accumulator output to reduce the size of the look up table. Assuming the look up table has integer resolution, the quantizer provides a mechanism for fractional phase increments. The Quantizer output width decides the depth of the look-up table and is normally less than the accumulator output width.

Look-up Table: The central part of the NCO is the look-up table which stores the values of the sine wave corresponding to equally spaced phase angles in the $(0,2\pi)$ interval. If the Wave size parameter is equal to "half" or "quarter", sine wave samples corresponding to $(0,\pi)$ or $(0,\pi/2)$ respectively are stored in the look-up table. As cosine can be derived from the sine of a shifted angle, the cosine value, if required, is read from the same look-up table by manipulating the address.

The depth of the look-up table is always a power of 2 and is determined by the user defined parameter Quantizer resolution.The width of the look-up table is, in most cases, equal to the output width.

Sum-of-Angles Memory Reduction: As the sine wave samples are stored in memory in direct digital synthesis NCOs, increasing the phase resolution of the output leads to corresponding increase in the size of the look-up table. The amount of memory required can be greatly reduced by making use of the "sum of angles" trigonometric identity and by using additional multipliers and adders after the memory output. This is achieved by dividing the angle space into coarse sub-divisions and then riting the phase angle as a sum of the nearest coarse angle and an additive corrective angle (fine angle)

Coarse angle: $cj = 2\pi j/C \quad j = 0, 1, 2, ..., C-1$

fine angle: $fk = 2\pi k/cf \quad k = 0, 1, 2, ..., F-1$

The sine and cosine values of θI can be computed using the sine and cosine values of cxand fyusing the following trigonometric identity:

$\text{Sin}(\theta i) = \sin(cx + fy) = \sin(cx)*\cos(fy) + \cos(cx)*\sin(fy)$

$\text{Cos}(\theta i) = \cos(cx + fy) = \cos(cx)*\cos(fy) - \sin(cx)*\sin(fy)$

Improving Quality of Output: A common measure of the output quality of NCO is the Spurious Free Dynamic Range (SFDR). This roughly indicates the degree of power separation between the main lobe and the next strongest side lobe in the power spectral density plot. The SFDR can be improved using either phase dithering or trigonometric correction. Phase dithering diffuses the concentration of phase quantization noise by adding a small random value to the accumulated phase before quantization. Trigonometric correction serves to improve the SFDR in a more deterministic way by adding a correction factor computed from the discarded LSB bits, to the output. The SFDR for the NCO output without dithering or trigonometric correction is approximately equal to 6* Quantizer resolution.

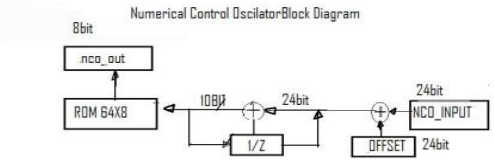


Fig.II.6.NCO oscillator block diagram

COSTAS Loop: The classical Costas loop that is suitable for BPSK/QPSK demodulation is shown in the Figure 1. The system involves two parallel tracking loops operating simultaneously from the same VCO (Voltage-Controlled Oscillator) or NCO (Numerically-Controlled Oscillator). The first loop, called the in-phase loop (or I arm), uses the NCO as in a PLL (Phase Locked Loop), and the second, called the quadrature loop (or Q arm) uses a 90 degree shifted NCO. The I and Q mixer outputs are filtered by single pole Butterworth low pass filters. The I and Q arm filter outputs are multiplied together and the product is scaled and filtered to produce the loop error used to control the NCO. The loop error should settle to a value when the loop is locked. A negative loop error decreases the NCO increment resulting in a lower NCO frequency, and similarly, a positive loop error increases the NCO increment resulting in a higher NCO frequency. The low pass filters in each arm must be wide enough to pass the data modulation without distortion.

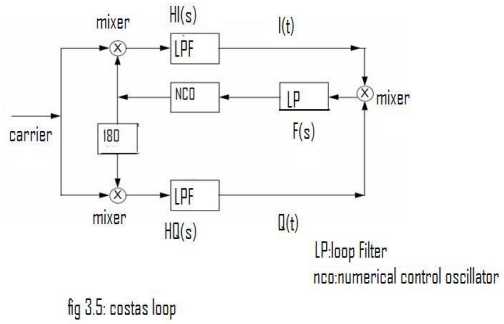


fig 3.5: costas loop

FigII.7.Costas loop

The input to the Costas loop is the waveform written as

$$y(t) = m(t)*\sin(\omega_c t + \psi(t)) + n(t)$$

where m(t) is the BPSK modulation and n(t) is a white bandpass noise. The in-phase mixer generates

$$I(t) = m(t)*\cos\psi_e + n_{mc}(t)$$

while the quadrature mixer generates

$$Q(t) = m(t)*\sin\psi_e + n_{ms}(t)$$

where the mixer noise nmc(t) and nms(t) are low pass demodulated noise processes in the carrier noise n(t). The output of the multiplier is then

$$I(t)Q(t) = m^2(t)\sin(2\psi_e)/2 + n_{sq}(t)$$

where nsq(t) represents all the signal and noise cross-products. The multiplier of the Costas loop can be thought of as allowing the bit polarity of the in-phase loop to correct the phase error orientation of the tracking loop, thereby removing the modulation. When the phase error ψe(t) is small, the Costas loop has the equivalent linear model in Figure 2.

Implementation of Costas

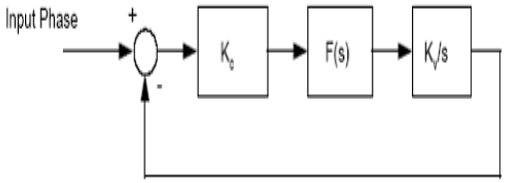


Figure 2: BPSK Costas Equivalent Loop Model

In above figure, Kc is the closed loop gain, which can be expressed as

$$K_c = (m(t)/2)^2 H_l(0) H_a(0) g_c$$

where gc is defined as follows

$$g_c = 4\omega_c / (K_v m^2(t) H_l(0) H_a(0))$$

and where ωc is the cross-over frequency, Kv is the gain of VCO, and F(s) is the transfer function of the loop filter, which is expressed in the following equation

$$F(s) = (sT + 1) / sT$$

where T is the sampling interval. The transfer function of the VCO is Kv/s .Power Detect and Lock Detect An interesting property of Costas loops is that the loop generates signals that can be used for other auxiliary purpose as well. This can be seen by reexamining the in-phase mixed signal and noting the following:

$$I(t)\big|_{\psi e = 0} = Am(t)$$

This shows that when the loop is locked, the in-phase arm produces an output proportional to the input data. Hence the data can be demodulated directly within the Costas loop after phase lock occurs.

$$I^2(t) - Q^2(t) = A^2 m^2(t) \cos(2\psi_e)$$

Squaring, low pass filtering, and subtracting the arm voltages produces an output that indicates phase lock [cos(2ψe)- >1 as ψe->0] and can therefore serve as a lock detector. When ψe = 0, this generates an output proportional to the average signal power, which can also be used for automatic gain control.

$$I^2(t) + Q^2(t) = A^2 m^2(t) + n^2_{ms} + n^2_{mc}$$

Mixer: In telecommunications, a mixer is a nonlinear or time-varying circuit or device that accepts as its input two different frequencies and presents at its output a mixture of signals at several frequencies:

1.the sum of the frequencies of the input signals
2.the difference between the frequencies of the input signals
3.both original input frequencies
4.a balanced mixer passes only a small leakage of the original signal to the output, often implemented as a double balanced mixer which has high isolation of both inputs.
5.un wanted inter modulation products from the inputs.

This nonlinear effect can be created by using a nonlinear electrical component, such as a diode. The time-varying effect can be created using a multiplier circuit such as a Gilbert Cell or passive switches. The manipulations of frequency performed by a mixer can be used to move signals between bands, or to encode and decode them. One other application of a mixer is as a product detector.

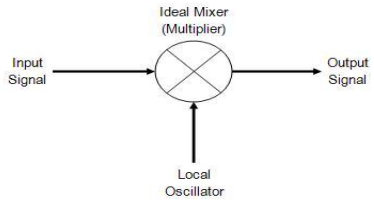


Fig II.8.Ideal Mixer block diagram

Mathematical description: The input signals are, in the simplest case, sinusoidal voltage waves, represent able as

$$v_i(t) = A_i \sin 2\pi f_i t$$

where each *A* is an amplitude, each *f* is a frequency, and *t* represents time. (In reality even such simple waves can have various phases, but that does not enter here.) One common approach for adding and subtracting the frequencies is to multiply the two signals; using the trigonometric identity

$$\sin(A) \cdot \sin(B) \equiv \frac{1}{2}[\cos(A-B) - \cos(A+B)]$$

we have

$$v_1(t)v_2(t) = \frac{A_1 A_2}{2}[\cos 2\pi(f_1 - f_2)t - \cos 2\pi(f_1 + f_2)t]$$

where the sum (f_1 + f_2) and difference (f_1 − f_2) frequencies appear. This is the inverse of the production of acoustic beats.

Multiplication implementation:

There are various ways of multiplying voltages, many of them quite sophisticated. However, as an example, a simple technique involving adiode can be described. The importance of the diode is that it is non-linear (or non-Ohmic), which means its response (current) is not proportional to its input (voltage). The diode therefore does not reproduce the frequencies of its driving voltage in the current through it, which allows the desired frequency manipulation. Certain other non-linear devices could be utilized similarly. The current *I* through an ideal diode as a function of the voltage *V* across it is given by

$$I = I_S \left(e^{\frac{qV_D}{nkT}} - 1 \right)$$

where what is important is that *V* appears in *e*'s exponent. The exponential can beexpanded as

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

and can be approximated for small *x* (that is, small voltages) by the first few terms of that series:

$$e^x - 1 \approx x + \frac{x^2}{2}$$

Suppose that the sum of the two input signals v_1 + v_2 is applied to a diode, and that an output voltage is generated that is proportional to the current through the diode (perhaps by providing the voltage that is present across a resistor in series with the diode). Then, disregarding the constants in the diode equation, the

output voltage will have the form $v_o = (v_1 + v_2) + \frac{1}{2}(v_1 + v_2)^2 + \dots$

The first term on the right is the original two signals, as expected, followed by the square of the sum, which can be rewritten as $(v_1 + v_2)^2 = v_1^2 + 2v_1v_2 + v_2^2$, where the multiplied signal is obvious. The ellipsis represents all the higher powers of the sum which we assume to be negligible for small signals.

As every multiplication produces sum and difference frequencies, from the quardric term of the series we expect to find signals at frequencies $2f_1$ and $2f_2$ from v_1^2 and v_2^2, and $f_1 + f_2$ and $f_1 - f_2$ from the v_1v_2 term. Often $f_1, f_2 \gg |f_1 - f_2|$, so the difference signal has a much lower frequency than the others; extracting this distinct signal is often the principal purpose of using a mixer in such devices as radio receivers. The other terms of the series give rise to a number of other, weaker signals at various frequencies which act as noise for the desired signal; they may be filtered out downstream to an extent, but sensitive applications will require cleaner output and thus a more complicated design.

Raised_cosine Filter (LPF): After frequency translation is implemented the output is passed through a channel Low Pass Filter by means of two root raised cosine filters one for each In-phase and Quadrature-Phase components. Here the filter is basically used to remove higher frequency components. The raised-cosine filter is a particular electronic filter, frequently appearing in telecommunications systems due to its ability to minimize intersymbol interference (ISI). Its name stems from the fact that the non-zero portion of the frequency spectrum of its simplest form (β = 1) is a cosine function, 'raised' up to sit above the *f* (horizontal) axis.

Mathematical description The raised-cosine filter is an implementation of a low-pass Nyquist filter, i.e. one that has the property of vestigial symmetry. This means that its spectrum exhibits odd symmetry about $\frac{1}{2T}$, where *T* is the symbol-period of the communications system.

Its frequency-domain description is a piecewise function, given by:

$$|H(f)| = \begin{cases} 1.0, & |f| \le \frac{1-\beta}{2T} \\ \frac{1}{2}\left[1 + \cos\left(\frac{\pi T}{\beta}\left[|f| - \frac{1-\beta}{2T}\right]\right)\right], & \frac{1-\beta}{2T} < |f| \le \frac{1+\beta}{2T} \\ 0, & \text{otherwise} \end{cases}$$

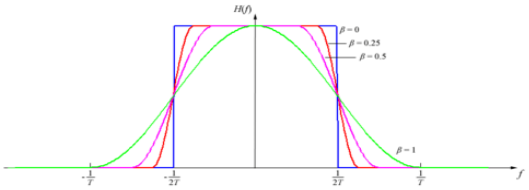
$$0 \le \beta \le 1$$

and characterised by two values; β, the *roll-off factor*, and *T*, the reciprocal of the symbol-rate.

The impulse response of such a filter (assuming linear phase) is given by:

$$h(t) = \mathbf{sinc}\left(\frac{t}{T}\right)\frac{\cos\left(\frac{\pi\beta t}{T}\right)}{1 - \frac{4\beta^2 t^2}{T^2}},$$

in terms of the normalized sinc function.



FigII.Raised Cosine Filter wave form

Amplitude response of raised-cosine filter with various roll-off factors

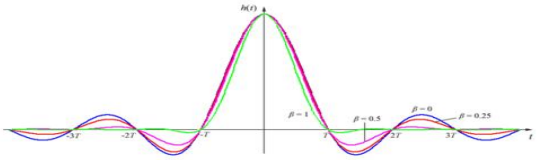


Fig.II.9. Sinc function

Impulse response of raised-cosine filter with various roll-off factors

Automatic Gain Control (AGC) circuit: In the early years of radio circuits, fading (defined as slow variations in the amplitude of the received signals) required continuing adjustments in the receiver's gain in order to maintain a relative constant output signal. Such situation led to the design of circuits, which primary ideal function was to maintain a constant signal level at the output, regardless of the signal's variations at the input of the system. Originally, those circuits were described as automatic volume control circuits, a few years later they were generalized under the name of Automatic Gain Control (AGC) .With the huge development of communication systems during the second half of the XX century, the need for selectivity and good control of the output signal's level became a fundamental issue in the design of any communication system. Nowadays, AGC circuits can be found in any device or system where wide amplitude variations in the output signal could lead to a lost of information or to an unacceptable performance of the system. The main objective of this AGC is to provide the hypothetical reader with a deep insight of the theory and design of AGC circuits ranging from audio to RF applications.

Theory of the Automatic Gain Control system:

To describe an AGC system in terms of control system theory, from pseudo linear approximations to multivariable systems. Each model has its advantages and disadvantages, first order models are easy to analyze and understand but sometimes the final results show a high degree of inaccuracy when they are compared with practical results. From a practical point of view, the most general description of an AGC system is presented in figure 1. The input signal is amplified by a variable gain amplifier (VGA), whose

gain is controlled by an external signal VC. The output from the VGA can be further amplified by a second stage to generate and adequate level of VO. Some the output signal's parameters, such as amplitude, carrier frequency, index of modulation or frequency, are sensed by the detector; any undesired component is filtered out and the remaining signal is compared with a reference signal. The result of the comparison is used to generate the control voltage (VC) and adjust the gain of the VGA.

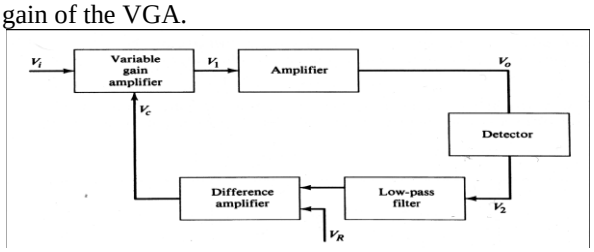


Fig.II.10. AGC block diagram

Since an AGC is essentially a negative feedback system, the system can be described in terms of its transfer function. The idealized transfer function for an AGC system is illustrated in figure 2. For low input signals the AGC is disabled and the output is a linear function of the input, when the output reaches a threshold value (V1) the AGC becomes operative and maintains a constant output level until it reaches a second threshold value (V2). At this point, the AGC becomes inoperative again; this is usually done in order to prevent stability problems at high levels of gain.

Timing Tracking Loop: The next step for the receiver is to sample the message signals at the symbol rate and decide which symbols were sent. Although the symbol rate is typically known to the receiver, the receiver does not know when to sample the signal for the best noise performance. The objective of the symbol-timing recovery loop is to find the best time to sample the received signal.

Timing recovery consists of:

- Timing error measurement
- Timing correction

It can be explained as following:

If a square wave is considered as shown in Figure 1 as a potential recovered in-phase (or quadrature) signal (i.e., the data sent is [+1,−1, +1,−1, . . .]) then sampling at any point other than the symbol transitions will result in the correct data. However, in the presence of noise, the received waveform may look like that shown in Figure 2. In this case, sampling at any point other than the symbol transitions does not guarantee a correct data decision.

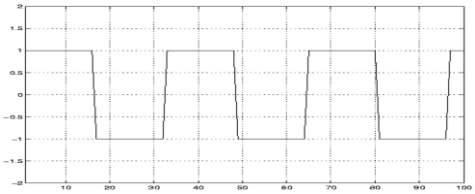
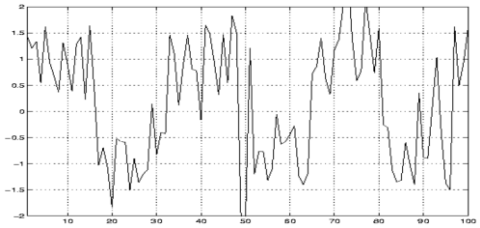


Fig.II.11:Square wave form.



FigII.12: Noisy representation of Fig.II.11

One simple method for recovering symbol timing is performed using a delay-locked loop (DLL).

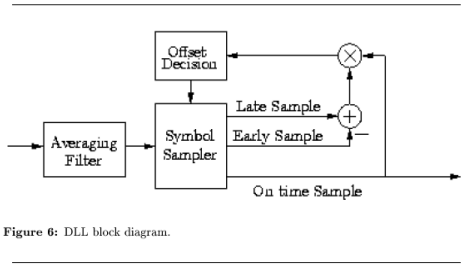


Figure 6: DLL block diagram.

Fig. II.13. Time tracking loop

Consider the sawtooth waveform shown in Fig.II.14, the output of the matched filter with a square wave as input. The goal of the DLL is to sample this waveform at the peaks in order to obtain the best performance in the presence of noise. If it is not sampling at the peaks, we say it is sampling too early or too late.

The DLL will find peaks without assistance from the user. When it begins running, it arbitrarily selects a sample, called the on-time sample, from the matched filter output. The sample from the time-index one greater than that of the on-time sample is the late sample, and the sample from the time-index one less than that of the on-time sample is the early sample. Figure shows an example of the on-time, late, and early samples.

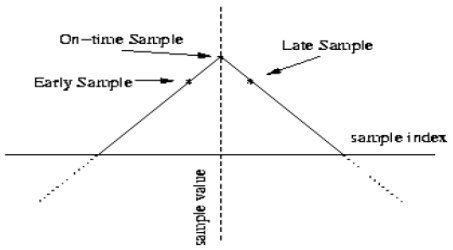


Fig.II.14. Matched filter output.

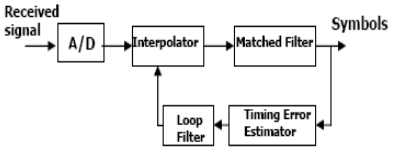
Note in this case that the on-time sample happens to be at a peak in the waveform. The on-time sample is the output of the DLL and will be used to decide the data bit sent. To achieve the best performance in the presence of noise, the DLL must adjust the timing of on-time samples to coincide with peaks in the waveform. It does this by changing the number of time-indices between on-time samples (resampling). There are three cases:

1. If the on-time sample is already at the peak, and the receiver knows that peaks are spaced by Tsymb (time between symbols) samples. If it then takes the next on-time sample Tsymb samples after this on-time sample, it will be at another peak.
2. If the on-time sample is too early. Taking an on-time sample Tsymb samples later will be too early for the next peak. To move closer to the next peak, the next on-time sample is taken Tsymb + 1 samples after the current on-time sample.
3. If the on-time sample is too late. Taking an on-time sample Tsymb samples later will be too late for the next peak. To move closer to the next peak, the next on-time sample is taken Tsymb − 1 samples after the current on-time sample. The offset decision block uses the on-time, early, and late samples to determine whether sampling is at a peak, too early, or too late. It then sets the time at which the next on-time sample is taken. The input to the offset decision block is on − time (late − early), called the decision statistic. The offset decision block could adjust the time at which the next on-time sample is taken based only on the decision statistic. However, in the presence of noise, the decision statistic becomes a less reliable indicator. For that reason, the DLL adds many successive decision statistics and corrects timing only if the sum exceeds a threshold; otherwise, the next on-time sample is taken Tsymb samples after the current on-time sample. The assumption is that errors in the decision statistic caused by noise, some positive and some negative, will tend to cancel each other out in the sum, and the sum will not exceed the threshold because of noise alone. On the other hand, if the on-time sample is consistently too early or too late, the magnitude of the added decision statistics will continue to grow and exceed the threshold. When that happens, the offset decision block will correct the timing and reset the sum to zero. Hence this module locks the sampling frequency and phase to a valid value.

Timing recover:
Timing correction can be performed in several ways:1.Adjusting timing phase of a VCO.2.Using an interpolation filter.3.Adjusting receive filter. Two quantities must be determined by the receiver to achieve symbol synchronization. They are sampling frequency and sampling phase.

Sampling frequency: Locking the sampling frequency requires estimating the symbol period so that samples can be taken at the correct rate. Although this quantity should be known (e.g., the system's symbol rate is specified to be 20 MHz), oscillator drift will introduce deviations from the stated symbol rate.

Sampling phase: Locking the sampling phase involves determining the correct time within a symbol period to take a sample. Real-world symbol pulse shapes have a peak in the center of the symbol period. Sampling the symbol at this peak results in the best signal-to-noise-ratio and will ideally eliminate interference from other symbols. This type of interference is known as intersymbol interference.



FigII.15: Digital filter timing recovery loop

The purpose of the timing recovery loop is to alter, as necessary, the sampling frequency and sampling Phase to sample the matched filter at the peaks. If the timing recovery loop is operating properly, it will provide the downstream processing blocks with symbols that were sampled at the highest SNR points available.

Timing error estimator*:* This block uses an algorithm to generate timing error. The available algorithms are early late sampling, Mueller and Muller algorithm, and Gardner algorithm.

- **Early late gate algorithm**:

Generates its error by using samples that are early and late compared to the ideal sampling point. The generation of the error requires at least three samples per symbol. Thus, it is impractical for systems with high data rates.

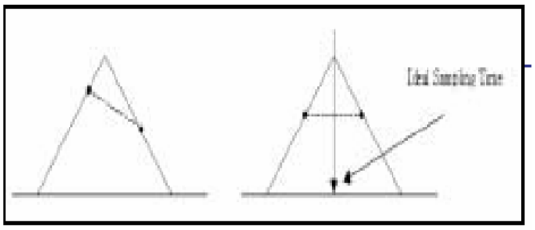


Fig.II.16.Dgital filter output gate pulses

Fig. method of generating error for early late gate algorithm. The left plot shows where sampling is occurring too late. When sampling occurs at the right time, the early and late samples will be at the same amplitude**.**

- **Mueller and Muller algorithm:**

The Mueller and Muller algorithm only requires one sample per symbol. The error term is computed using the following equation:

$En = (\square yn \,.\, yn\square - (\square - 1) yn \,.\, yn\square - 1)$

Where *y*n is the sample from the current symbol and yn-1 is the sample from the previous symbol. The slicer (decision device) outputs for the current, and previous symbol are represented by *^yn* and *^yn-1*, respectively. Examples of the value for the Mueller and Muller error for the cases of different timing offsets are shown in below Figures.

One drawback of this algorithm is that it is sensitive to carrier offsets, and thus carrier recovery must be performed prior to the Mueller and Muller timing recovery.

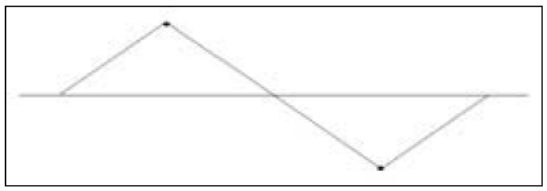


Fig: Correct timing: $e_n = (-1 \cdot 1) - (-1 \cdot 1) = 0$.

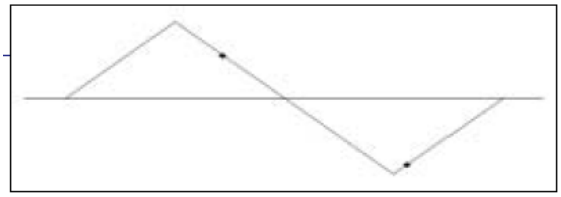


Fig: Timing is fast: $e_n = (-0.8 \cdot 1) - (-1 \cdot 0.5) = -0.3$.

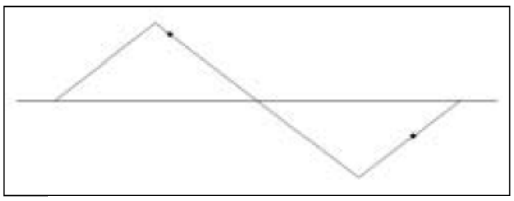


Fig: Timing is slow: $e_n = (-0.5 \cdot 1) - (-1 \cdot 0.8) = 0.3$.

- Other methods of timing recovery:

The ideal case: the transmitter and receiver running off of the same clock. This situation is typically impossible in a wireless communications system. In wired systems, such as computer networks, a timing recovery loop is not needed because synchronization is explicit.

Alternative: the clock frequency transmitted along with the data. The receiver can recover this clock signal with a narrowband band pass filter tuned to that frequency. This method is generally inefficient because the transmission of the clock signal consumes both bandwidth and transmitter power that could have otherwise be

III. Simulation Results

Fig.III.1Modulator output

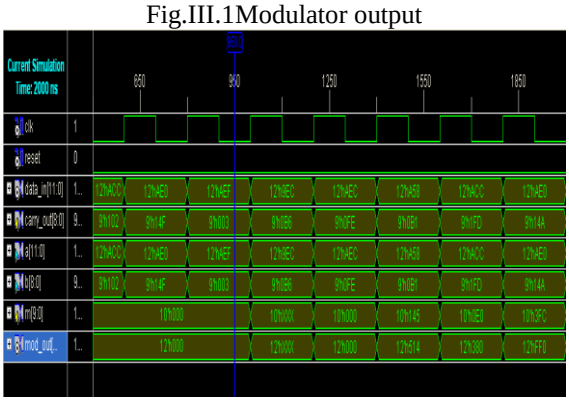


Fig.III.2:Carrier Supressor output

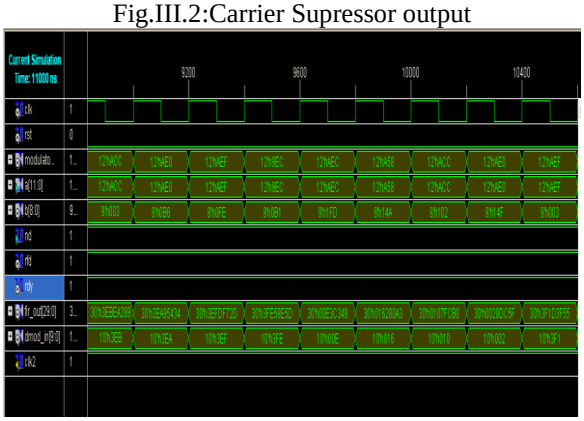


Fig.III.3 Demodulator output

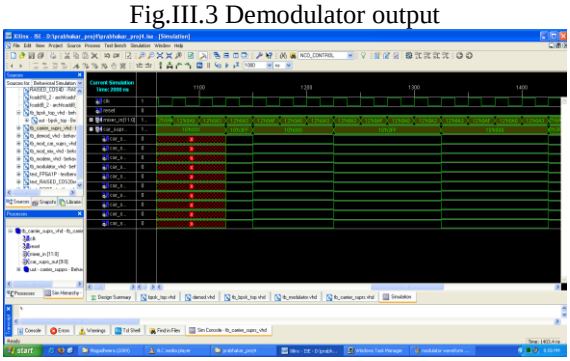


Fig.III.4:Modem out put

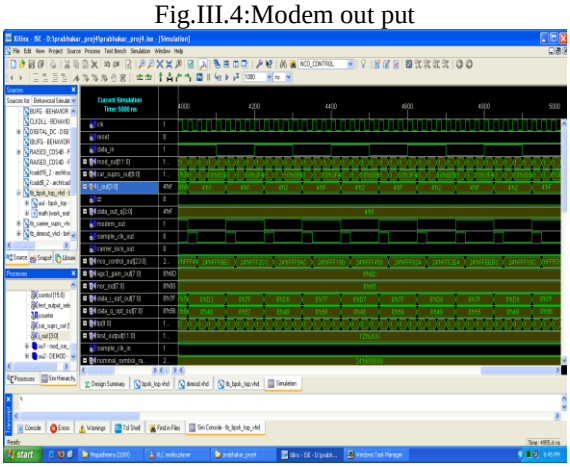
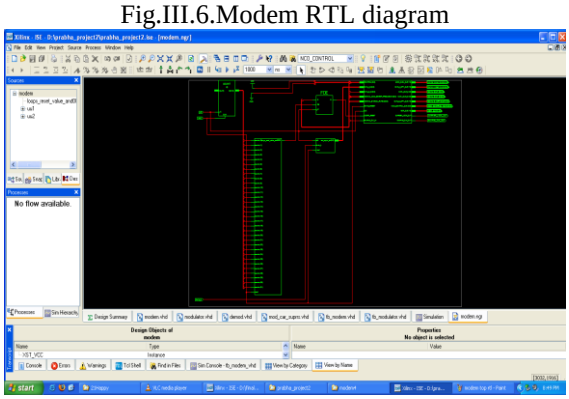


Fig.III.6.Modem RTL diagram



IV. CONCLUSION

BPSK Modem with a data rate of 40 Mbps , message signal with 3.1 KHz and carrier signal with 5MHz is implemented for wireless, Satellite and Telemetry applications. Mat lab soft ware used for generating the filter coefficients for Digital Filter design in Carrier Suppressor block which is used to suppress the modulated carrier in that I used Digital FIR LPF which has a feature of programmability and makes the scheme more repeatable and flexible ie., which can be used for fine and multiple data rate.

The Carrier Recovery is done by using Costas loop and Symbol Timing Recovery using Early Late gate Algorithm . The out put results and data rate of the modem are compared with the specifications , 5Mbps of data rate ,3.1KHz of message.

This paper is implemented using VHDL language, Simulation and Synthesis is done by using Xilinx ISE 9.2I Tool. The results are highly satisfactory.

V. FUTURE SCOPE

BPSK modem can be applicable up to the maximum range of 10 Mbps. It is inherently adopted for QPSK modem which is used up to 20 Mbps of data rate the same design approach can be utilized for high speed speech communication applications i.e. for very high data rates in future. That is high speed wireless LAN Standards IEEE 802.11 has eight data rates (6,9,12,18,24,36,48 and 54Mbps).

Here 6 and 8Mbps modulators are used for OFDM (Orthogonal Frequency Division Multiplexing)Modulation where each sub carrier is BPSK Modulated signal. For 12 and 18 Mbps of QPSK modulators are used for modulated signal. In future could develop an area effective (less complexity of the circuit]) BPSK Design. BPSK Modem supplements the European Telecommunication Standard (ETS) ETS 300 421 which describes the transmission of MPEG-2/DVB transport stream via satellite. where the use of BPSK modulation is required, rather than that specified in ETS 300 421

REFERENCES

[1] *Wireless Digital Communications Modulation & Spread Spectrum Applications*-By DR.KAMIO FEHER (PHI).

[2] *Advanced Electronic Communications Systems*, by WAYNE TOMASI(PHI ,5th ed.

[3] Haykin, Simon (2001*). Communication Systems* (4th ed.). John Wiley & Sons.

[4] Shanmugam, K. Sam. *Digital and Analog Communication Systems.* New York, NY: John Wiley & Sons, 1979

[5] (a)http//www.comblock.com/download/com 1001.pdf (b) www.google.com/satellite communication .pdf

[6] (a) Xilinx ISE 9.2I/IP core gen/communication/dds/document(b) Xilinx ISE 9.2I/IP core gen/DSP/FIR filter/document (c) www.xilinx.com /ISE 9.2i Quick Start Tutorial, Xilinx, Inc.,

[7] (a) *VHDL Premer*, third Edition ,Pearson Education By J.Bhasker. (b) VHDL Programming by Example By Douglas L. Perry

[8] MATLAB-R2007a/FDA(Filter Design Analysis) Tool

Prabhakar.R received M.Tech degree in Digital electronics and communication system and B.Tech in electronics and communication engineering from JITS, Jawaharlal Nehru Technological University, Hyderabad ,AP,India.as.MISTE since 2010. And published few international journals recently.Hyderabad ,AP,India.

Vijaya Babu.E received M.Tech degree in Embedded systems and VLSI Design from the Jawaharlal Nehru Technological University, Hyderabad and B.Tech in electronics and communication engineering from the Osmania University, Hyderabad .A.P ,India

G.J.Chitra received M.E. degree in Systems and Signal Processing from Osmania Campus and B.Tech in electronics and communication engineering from GNITS ,Jawaharlal Nehru Technological University.Hyderanbad.A.P.India.