

BPSK Demodulation for RF Applications

*Yu-Ting Toh
Ali Niknejad, Ed.
Borivoje Nikolic, Ed.*

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2017-91

<http://www2.eecs.berkeley.edu/Pubs/TechRpts/2017/EECS-2017-91.html>

May 12, 2017



Copyright © 2017, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

Acknowledgement

I would like to thank my advisor Professor Ali Niknejad for allowing me to work with him and his invaluable guidance. I would also like to thank my collaborators Dr. Lorenzo Iotti, Calvin Handoko, and Dr. Bo Zhao who worked with me on this project and provided crucial contributions. I could not have done this without them.

BPSK Demodulation for RF Applications

by Yu-Ting Toh

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences,
University of California at Berkeley, in partial satisfaction of the requirements for the
degree of **Master of Science, Plan II.**

Approval for the Report:

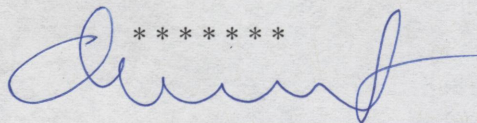
Committee:



Professor Ali Niknejad
Research Advisor

5/12/17

(Date)



Professor Borivoje Nikolic
Second Reader

5/12/2017

(Date)

Abstract

BPSK Demodulation for RF Applications

Yu-Ting Toh

May 12, 2017

In this report, binary phase shift keying (BPSK) demodulation is discussed for an RF application. Various methods are presented but the Costas loop is shown to be the most successful at obtaining the baseband data signal when the carrier IF frequency has a possible range of values and is expected to be a square wave, as opposed to a sinusoidal wave. The loop filter response and filter parameters need to be carefully chosen to maintain loop stability while removing noise interference. Additionally, Manchester encoding and Miller decoding is implemented for the specific application of a probe which detects counterfeit integrated chips by communicating via RF with an on-chip dielet which holds sensitive information.

BPSK Demodulation for RF Applications

Yu-Ting Toh

May 12, 2017

1 Project Overview

1.1 Motivation

Counterfeit integrated chips are pervasive in the government defense industry. Old components may be repackaged and sold as new. Components not up to par with industry standards may be marketed as more reliable than they actually are. They are then introduced and sold into the defense supply chain for profit. Defense systems are required to be very reliable and robust. If counterfeit IC's fail to function as promised, these mission critical systems could be compromised leading to weaknesses in security and defense. This issue has very tangible and often times life-threatening consequences. In May of 2012, the Senate Armed Services Committee found that suspected counterfeit parts exceeded 1 million. Counterfeiters have found ways to circumvent current counterfeit-detection systems. Hence, the Defense Advanced Research Projects Agency (DARPA) started the Supply Chain Hardware Integrity for Electronics Defense (SHIELD) program to “eliminate counterfeit integrated circuits from the electronics supply chain by making counterfeiting too complex and time-consuming to be cost effective.”

1.2 DARPA Specifications

According to DARPA, the specifications are outlined as follows: 100 μm by 100 μm RFID dielets are to be integrated onto chip packaging to provide authenticity information as well as temperature and light sensor information. An external probe verifies authenticity of the IC by providing power to the dielet, setting up a secure communication link, and verifying the source of the IC. The overarching goal is to make counterfeiting IC's not economically sustainable for counterfeiters.

1.3 Dielet Design

In previous years, efforts had been made in the CMOS dielet radio design and power transfer experiments by Dr. Bo Zhao and Nai-Chung Kuo under Professor Ali Niknejad at the Berkeley Wireless Research Center. They investigated wireless power transfer over different frequency ranges and carried out backscattering experiments with different techniques to achieve reduced carrier interference.

1.4 Probe Design

A project was commissioned to develop a probe to interface with dielets developed within DARPA's SHIELD program. A Statement of Work for the Near Field Power and Communications RFID Probe details the interface, controls, timing, and physical requirements. The probe will use near-field inductive coupling to power and communicate with the dielet. It will receive the dielet's encrypted serial number as well as information from its sensors. A USB 2.0 compatible interface with standard micro-B USB connector is used to enable peripheral communications. The probe's primary input power is also provided from the USB port. Additionally, the probe will have 5 indicator LED's controlled by a field-programmable gate array (FPGA) to allow users to identify when host link has been established, dielet link has been established, 5V power is present, alternate power source is active, and FPGA configurable status events have occurred. The last component of the probe interface is a pushbutton read by an FPGA to initiate a dielet authentication sequence.

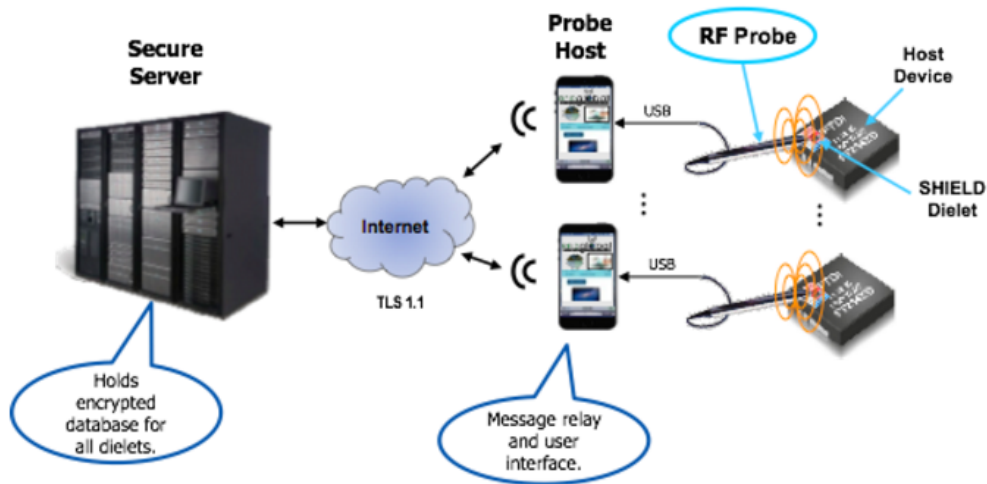


Figure 1: SHIELD System Overview

2 Probe Design Architecture

The probe design consists of five main components: TX chain, RX chain, field-programmable gate array (FPGA), frequency synthesizer, and power conversion. Figure 2 illustrates the functional diagram for the probe.

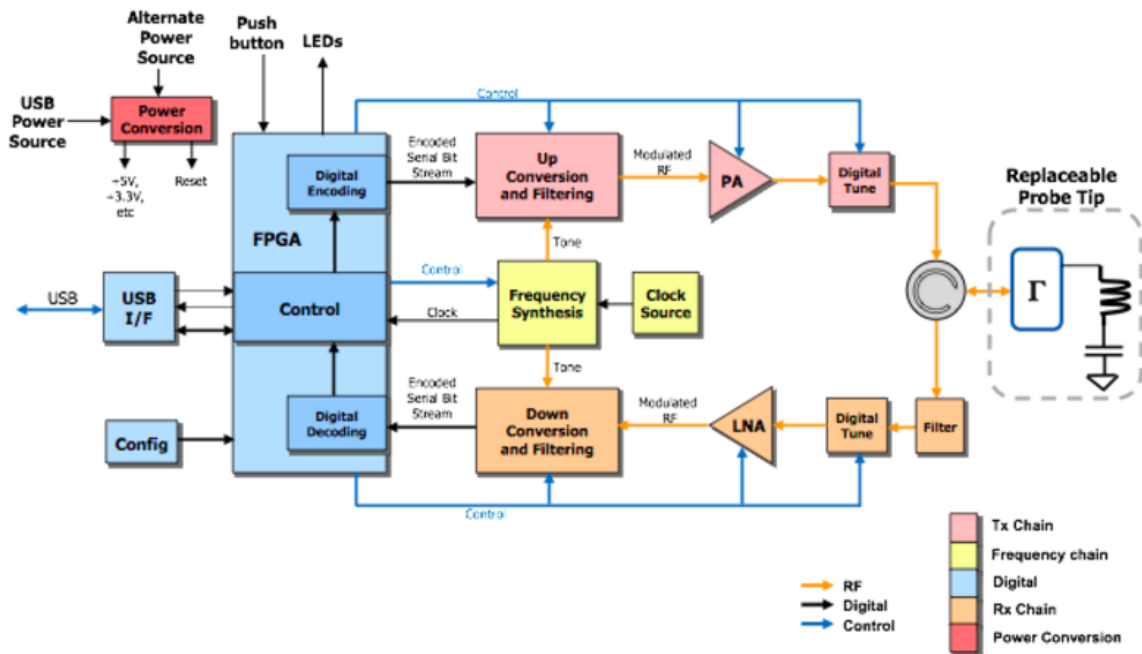


Figure 2: Probe Functional Diagram

2.1 FPGA

As seen in the diagram, the probe operates as a USB peripheral. It receives all control from and provides status to the USB port. The FPGA is used to perform all digital functions. Two of its main functions are the digital encoding and decoding of the RF baseband waveforms. The FPGA also provides control signals to upconverters, downconverters, amplifiers, and filters in the RF transmit and receive chain.

2.2 Frequency Synthesizer

Using a phase-locked loop (PLL), a 5.8GHz sine waveform can be generated to be used as the carrier signal for amplitude modulation and demodulation of the data signals.

2.3 Transmit Chain

The transmit chain consists of an up conversion and filtering block, power amplifier, and digital tuning block. An attenuator is also used to achieve 2.5dB amplitude shift keying (ASK) modulation scheme used for the 6.25kbps downlink data (probe-to-dielet). Low periods are 25% lower in voltage than high periods.

2.4 Receive Chain

The receive chain consists of a down conversion and filter block, low-noise amplifier, and digital tuning block. Additionally, an IQ demodulator is needed to perform in-phase and quadrature-phase amplitude demodulation of the uplink data (dielet-to-probe). The modulation scheme used is also 2.5dB ASK modulation. Then an analog-to-digital converter (ADC) is used to downconvert and digitize the signal. Because the receive signal is Miller encoded with $M = 16$, the signal is also phase modulated with binary phase shift keying (BPSK) scheme. This requires further processing to be done. The amplitude demodulated Miller sequence is further processed with digital functions on the FPGA. The phase demodulation is discussed further in later sections.

3 Digital TX Encoding and RX Decoding on FPGA

3.1 Manchester Encoding

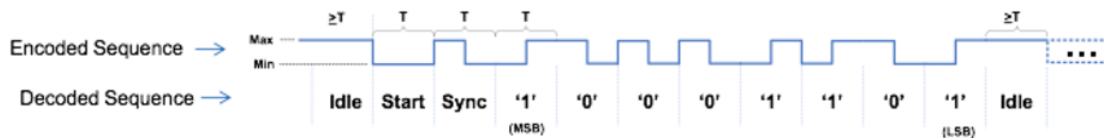


Figure 3: Transmitted Manchester Encoding

The transmitted signal is manchester encoded at 25kHz to support a downlink data rate of 25kbps. Data is sent 8 bits at a time preceded by a fixed sequence of “Start” and “Sync” bits. When no data is being sent “Idle” bits are transmitted instead. In Manchester encoding, each data symbol will always have one transition within the bit period; The bit is either low then high, or high then low, with high and low held for equal times. It is therefore a self-clocking encoding scheme. This allows the dielet to dynamically measure the bit period for each byte sent. The manchester encoded value is the exclusive or (XOR) of the clock signal and the data sequence. The manchester encoding for the transmit chain was implemented in Verilog using a finite state machine structure.

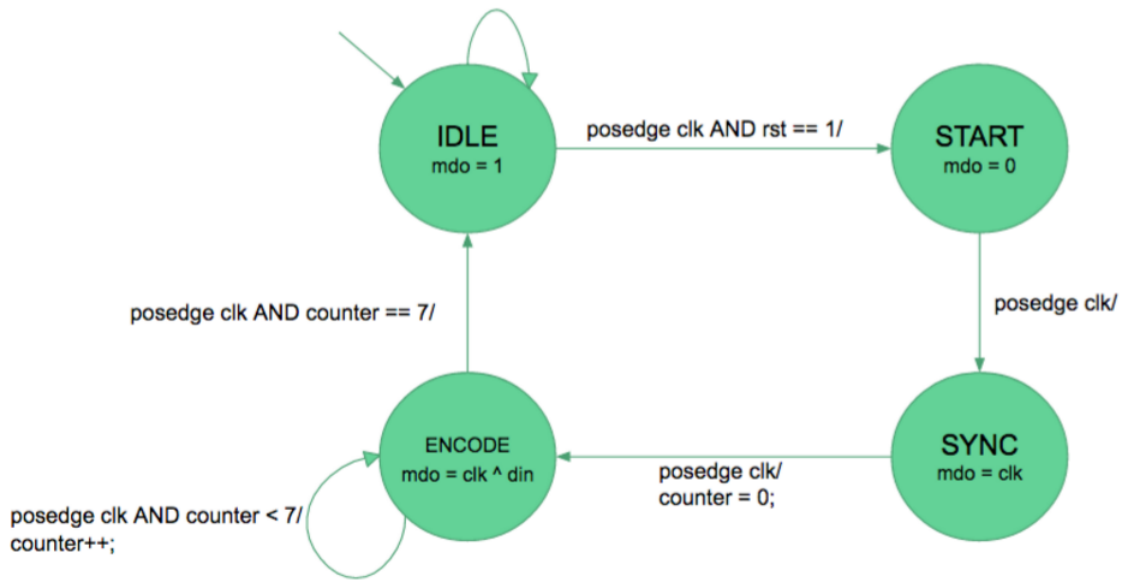


Figure 4: Manchester Encoding Finite State Machine

The state machine is initialized in the “IDLE” state and will take the default transition back to the “IDLE” state and output a high bit unless at the next positive edge of the clock, the reset signal goes high. Then it will transition into the “START” state and output a low bit. At the next positive edge of the clock, it transitions into the “SYNC” state and output high for half a period then low for half a period. Finally at the next positive edge of the clock, it transitions into the first “ENCODE” state and starts the Manchester encoding of 1 byte. The state machine stays in “ENCODE” for 8 clock periods to encode the 8 bits before transitioning back to “IDLE”.

Using the Xilinx Vivado design suite, simulations were run with a test input signal. Results were verified and the output waveforms correctly encoded using the Manchester scheme.

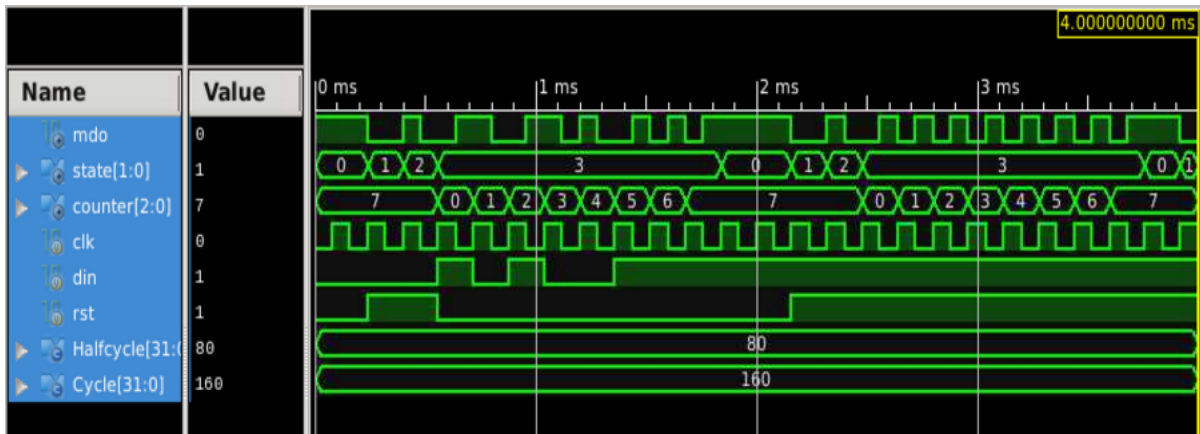


Figure 5: Manchester Encoding Verilog Simulations (States: 0 == IDLE, 1 == START, 2 == SYNC, 3 == ENCODE; Counter: to count how many bits to be encoded)

3.2 Miller Decoding

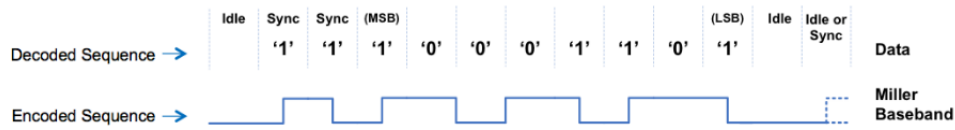


Figure 6: Received Miller Encoding

The Miller encoded waveform has an uplink data rate of 125kbps. Data is received 8 bits at a time preceded by a fixed sequence of two “Sync” bits. Miller encoding is also known as delay encoding because a delay of half a cycle period will be required to determine whether it is a data bit of 1 (baseband inverts) or 0 (baseband stays constant). The Miller decoding for the receive chain was also implemented in Verilog using a finite state machine structure.

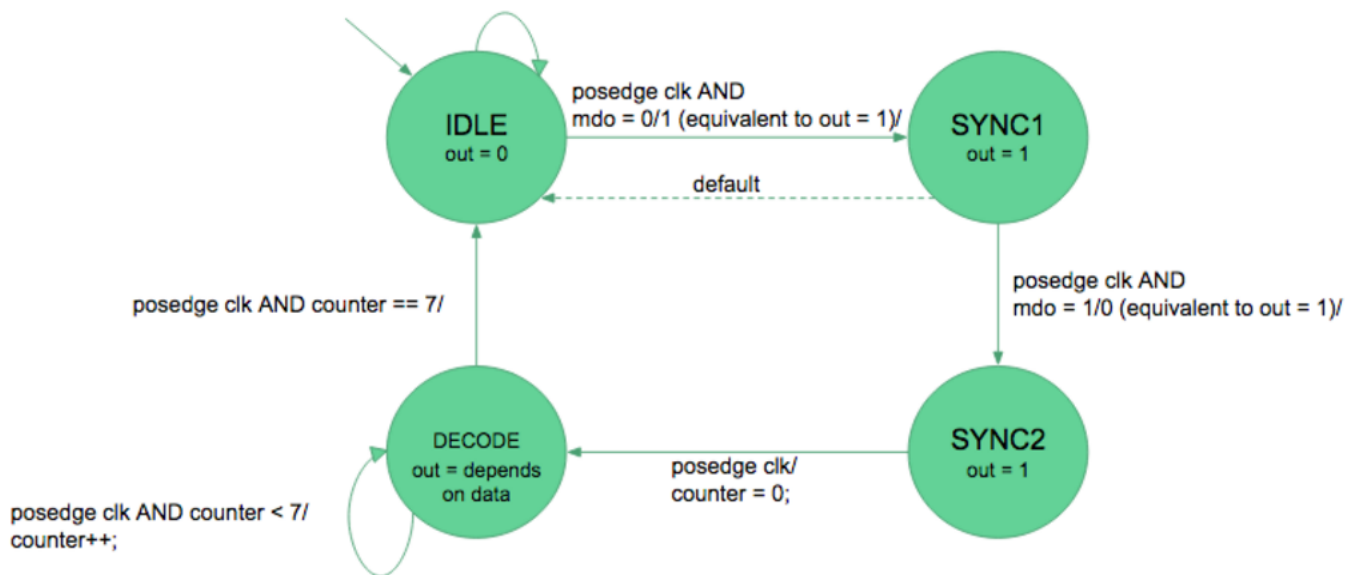


Figure 7: Miller Decoding Finite State Machine

Due to the nature of Miller encoding, even though the uplink data rate is 125kbps, a clock twice as fast (250kbps) is needed to detect transitions in the middle of the period (corresponds to a data bit of 1). The state machine is initialized in the “IDLE” state and will take the default transition back to itself unless at the next clock cycle the baseband transition from low to high in the middle of the clock period. It will output a high bit and transition to “SYNC1”. The state machine will either default back to “IDLE” or transition to “SYNC2” and output a high bit if it sees a transition from high to low in the middle of the next clock period. At the next positive edge of the clock it transitions into the first of eight “DECODE” states and starts the Miller decoding of the 1 byte of data received. A transition from high to low or low to high in the middle of the clock period is decoded as a 1 bit. A constant high or low value during the entire clock period is decoded as a 0 bit. After 8 clock periods, the state machine transitions back to “IDLE”.

Again, using the Xilinx Vivado design suite, simulations were run with a test input signal. Results were verified and the output waveforms correctly decoded using the Miller scheme.

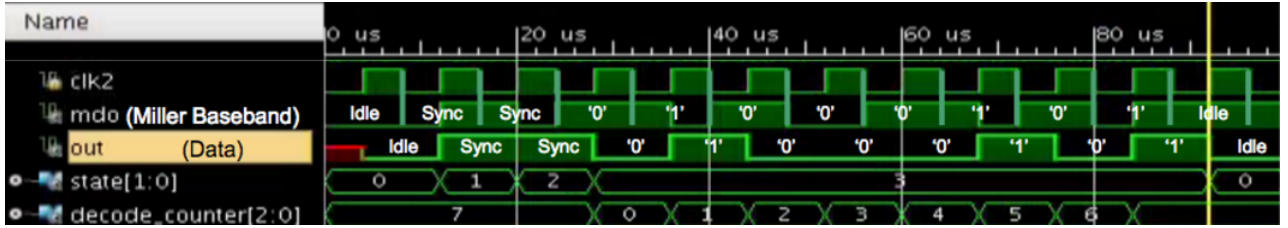


Figure 8: Miller Decoding Verilog Simulations (States: 0 == IDLE, 1 == SYNC1, 2 == SYNC2, 3 == DECODE; Counter: to count how many bits to be decoded)

4 RX Branch

The receive chain proved to pose a challenging problem because in order to demodulate and decode the receive signal, clock recovery and synchronization is necessary. The receive signal from the dielet is phase modulated (binary phase key shifting) to separate the baseband frequency from the square carrier frequency. The modulation rate from the dielet will range from 2-2.4MHz, $M=16$ times that of the uplink data rate of 125kbps.

4.1 Binary Phase Shift Keying

Binary Phase-Shift Keying (BPSK) is the most basic form of phase-shift keying, using just 2 phases separated by 180° . BPSK modulation is considered the more robust of modulation schemes in terms of noise immunity. It allows the highest level of distortion that can still successfully be demodulated. Hence, it is an ideal modulation scheme and widely used in wireless LAN, RFID and Bluetooth communication applications. A BPSK modulated signal is represented by a carrier signal shifted by 180° for one data symbol and not shifted for the other. In other words,

$$BPSK(t) = DATA(t) * \cos(2\pi ft) \quad (1)$$

where $DATA(t) \in \{-1, 1\}$ is the original data signal and the carrier frequency $w = 2\pi f \gg 2\pi/T$ where T is the bit clock period of the $DATA(t)$ signal.

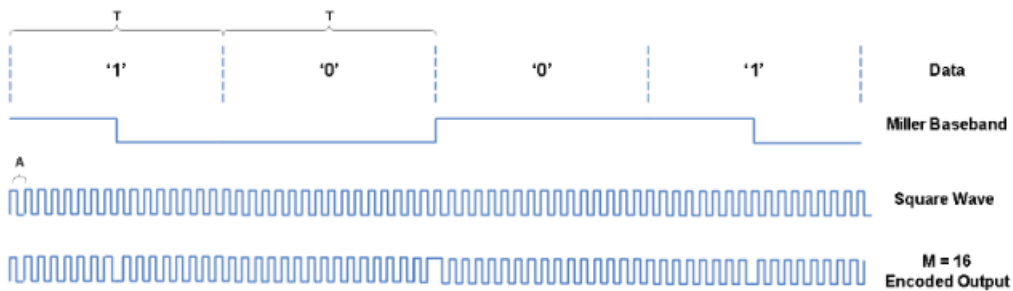


Figure 9: BPSK Phase Modulated Miller Encoding

Phase ambiguity poses a classic challenge in the demodulation of a BPSK signal. In incoherent demodulation, without knowing the exact carrier frequency, simply squaring and summing the in-phase and quadrature-phase signals, thresholding to digitize the signal, and counting the transitions that occur will not be precise and phase information may be lost in the process. Other techniques like using a training sequence with known data is often used but in the case of the dielet to probe communication, immediate data transmission is expected.

Another possible solution was to consider a threshold value that is the updated average of the BPSK modulated signal. The current value of the modulated signal is then compared against the threshold value. Considering the previous, current, and next bit, it may be possible to predict whether the signal stays constant, transitions, or a phase shift occurs. The idea is that counting the transitions will allow us to recover the carrier clock. A phase shift is then detected as a missed transition in the thresholded signal. A missed transition just means the length of the transition is longer than usual because it did

not take a transition that it would usually take. Implemented in MATLAB, this solution still misses phase shifts unless the length of transitions is considered to distinguish between phase shifts and constant signal. One issue is that it performs detection before filtering. Some feedback could be implemented, setting the digital bandpass filter at input after the subcarrier frequency has been detected.

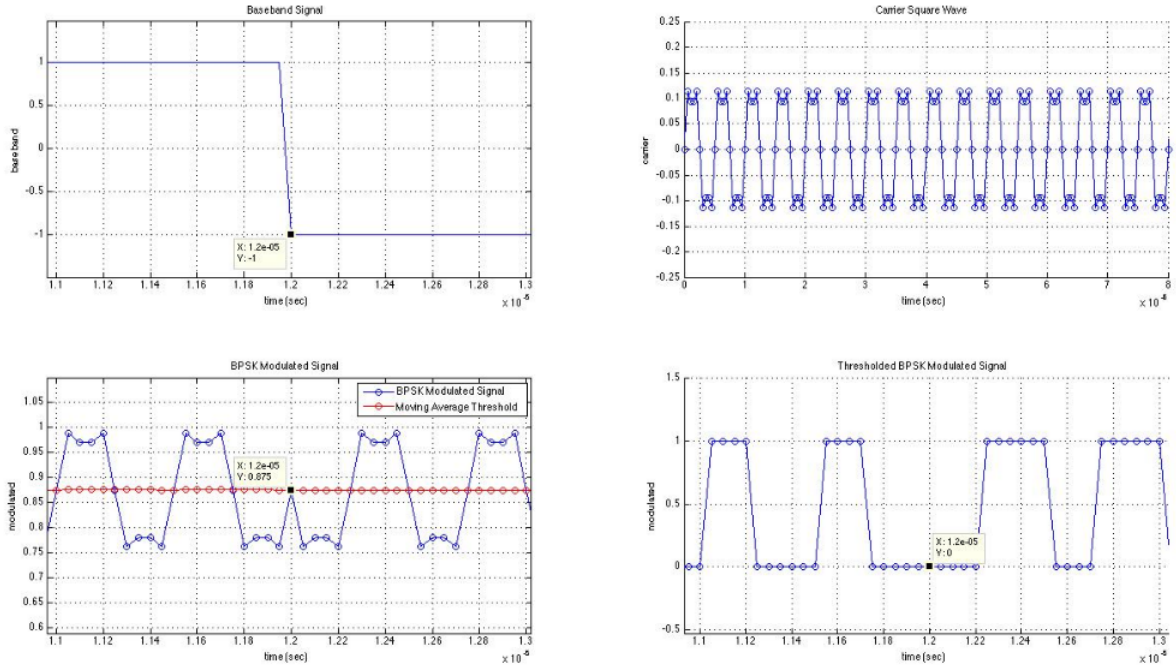


Figure 10: Threshold BPSK Demodulation

4.2 Demodulation

An ideal coherent demodulator determines the phase of the input RF signal to recover the data bit. This requires the demodulator to have a reference signal identical in frequency and phase to the original carrier. Multiplying the BPSK signal by the coherent carrier results in:

$$BPSK(t) * \cos(2\pi ft) = DATA(t) * \cos(2\pi ft) * \cos(2\pi ft) \quad (2)$$

$$= \frac{1}{2} DATA(t) [\cos(2 * 2\pi ft) + \cos(0)] \quad (3)$$

$$= \frac{1}{2} DATA(t) [\cos(2 * 2\pi ft) + 1] \quad (4)$$

Note the trigonometric identity:

$$\cos(u) * \cos(v) = \frac{1}{2} [\cos(u + v) + \cos(u - v)] \quad (5)$$

Applying an ideal low-pass filter removes the double-frequency term, leaving the DC term that consists of the $DATA(t)$ signal. The spectrum of the BPSK signal is symmetric along the carrier frequency. Upon demodulation, the upper and lower sidebands (which contain identical information) will coherently add together, but channel noise will randomly add, resulting in an inherent SNR of 3dB above that of the BPSK signal. Although this mathematical demodulation scheme is simple and straightforward, it requires close to ideal components to work. Furthermore, because the exact IF frequency of the receive signal varies from 2-2.4MHz in our problem, this scheme will not work.

4.3 Phase-Locked Loop

Instead we consider using a phase-locked loop (PLL) to recover the carrier frequency and phase. The PLL is a feedback control system that uses a voltage-controlled oscillator (VCO) to generate an output signal that matches the phase and frequency of the input signal. The voltage-controlled oscillator is constantly adjusted by a phase detector that compares the phase of the input signal to the phase of the output generated by the VCO. Similarly, the PLL can also track the frequency of the input signal. PLL's are often used to recover carrier signals to demodulate a signal because of its self-correcting nature. However, the greatest disadvantage is the loop settling time. The design of a PLL involves considerations about the noise and settling time tradeoffs.

4.4 Costas Loop

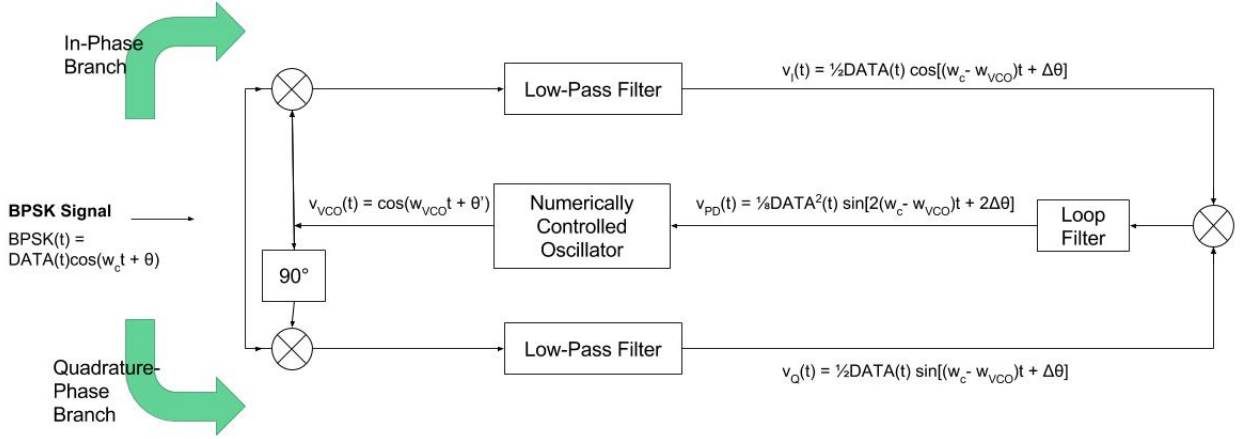


Figure 11: Costas Loop Function Diagram

A Costas loop is a PLL based circuit which consists of an in-phase and quadrature-phase branches. By using the Costas loop, the in-phase branch will yield the baseband signal when the loop locks while the quadrature-phase branch continues to track the frequency and phase of the carrier. The Costas loop offers better sensitivity than the PLL for small deviations because the loop error voltage is $\sin(2\Delta\theta)$ (in PLL, the error voltage is $\sin(\Delta\theta)$) where $\Delta\theta$ is the phase difference between the input signal and reference signal. The in-phase branch had been previously examined (equations 2-4). The quadrature-phase branch can be similarly analyzed in the case of a coherent carrier:

$$BPSK(t) * \sin(2\pi ft) = DATA(t) * \cos(2\pi ft) * \sin(2\pi ft) \quad (6)$$

$$= \frac{1}{2}DATA(t)[\sin(2 * 2\pi ft) + \sin(0)] \quad (7)$$

$$= \frac{1}{2}DATA(t)\sin(2 * 2\pi ft) \quad (8)$$

Note the trigonometric identity:

$$\cos(u) * \sin(v) = \frac{1}{2}[\sin(u + v) + \sin(u - v)] \quad (9)$$

Applying an ideal low-pass filter removes the double-frequency term, leaving the DC term of 0. Hence when the Costas loop is “locked” it operates with a maximum in-phase signal and minimum quadrature-phase signal (close to zero). The phase doubler is a multiplier that outputs the product of the in-phase and quadrature-phase signals. When the Costas loop has not “locked” (reference carrier does not match actual carrier yet) the phase error signal is:

$$\frac{1}{2}DATA(t)\cos(\Delta\theta) * \frac{1}{2}DATA(t)\sin(\Delta\theta) = \frac{1}{4}DATA^2(t) * \frac{1}{2}[\sin(2\Delta\theta) + \sin(0)] \quad (10)$$

$$= \frac{1}{8} DATA^2(t) \sin(2\Delta\theta) \quad (11)$$

Note the double phase difference which achieves double sensitivity. The phase doubler error signal is then passed through a loop filter. The loop filter should not significantly contribute to the loop response. The pole of the loop filter should be low enough to remove spurious noise components but high enough that it does not affect the phase in the loop bandwidth. If it contributed significantly to phase within the loop bandwidth, this may cause the loop to become unstable and start to oscillate.

When the Costas loop is “locked” ($w_c - w_{VCO} = 0$; $\Delta\theta = 0$), the in-phase signal $v_I(t)$ is the demodulated output signal.

$$v_I(t) = \frac{1}{2} DATA(t) \cos((w_c - w_{VCO})t + \Delta\theta) = \frac{1}{2} = \frac{1}{2} DATA(t) \quad (12)$$

The Costas loop has two stable locking points due to the doubled-sine phase detection response being 180° periodic. The two stable locking points occur at 180° and 0° phase error. This means that there is a 50% chance that the carrier is upside-down, resulting in an also flipped baseband when the phase error is $\Delta\theta = 180^\circ$

$$v_I(t) = \frac{1}{2} DATA(t) \cos((w_c - w_{VCO})t + \Delta\theta) = \frac{1}{2} = -\frac{1}{2} DATA(t) \quad (13)$$

However, Miller encoding is invariant to flipping as long as all bits have been consistently flipped. Hence, this issue does not affect the Miller decoding of the flipped baseband. Miller encoding is invariant to flipping because it is only concerned with whether the baseband is constant (decoded 0) or transitions in the middle of the data period (decoded 1).

4.5 Design Considerations

Performance measures of the Costas loop include noise performance, convergence time, and achievable lock range. In terms of noise performance, this is set by the low-pass filter responses in the in-phase and quadrature-phase branches. The Nyquist criterion requires that the LPF have a bandwidth at least half the symbol rate. A LPF with cutoff at half the symbol rate will remove the most noise possible without interfering with the desired $DATA(t)$ signal amplitude. For fast loop convergence, the loop gain should be eight times the baseband bandwidth. A higher value increases the frequency acquisition range. However, for better stability with a square wave IF, a lower value provides more stability. So the loop gain is set to 0.5 times the baseband bandwidth. For the loop filter which filters out noise in the double-phase detector signal, it should have its pole high enough that it does not cause too much instability and oscillations in the loop while low enough to remove as much noise as possible. A good rule-of-thumb is to set the loop filter’s pole to a minimum of four times that of the closed loop (eight times the LPF poles). For better stability, the loop filter’s pole is set at 12 times the baseband bandwidth.

4.6 Simulation Results

A digital Costas loop was simulated in MATLAB. Simulations utilized a sampling frequency of $f_s = 100MHz$. The input signal is the data signal with a nominal data rate of 125 kbps which corresponds to an IF carrier frequency of 2MHz. The maximum possible data rate is 150kbps which corresponds to an IF carrier frequency of 2.4MHz. The data signal is then Miller encoded and modulated with the carrier signal. The baseband bandwidth is double the data rate, with a range of 250-300 kHz. Gaussian white noise was also introduced to simulate noise performance.

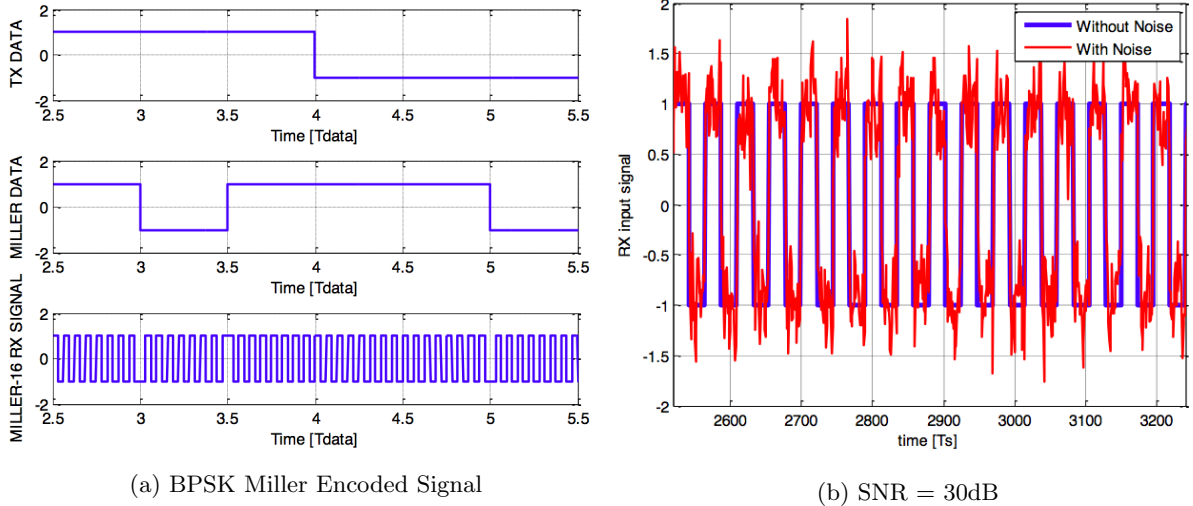


Figure 12: Costas Loop Input Signals

At the input, two additional filters were applied: a first-order low-pass filter and a wideband band-pass filter were used to simulate the transfer function of the analog baseband and for noise rejection. Both LPF's and the loop filter of the Costas loop are first-order to maintain phase performance. Anything larger than first-order would result in stability issues. The bandpass filter needed to operate in a frequency range such that as much noise is eliminated as possible. However, the tradeoff is bandlimiting the input signal. Cutting harmonics of the baseband signal results in phase transitions that are not as sharp, creating errors in the Costas loop. The bandwidth used was [1MHz - 3.5MHz].

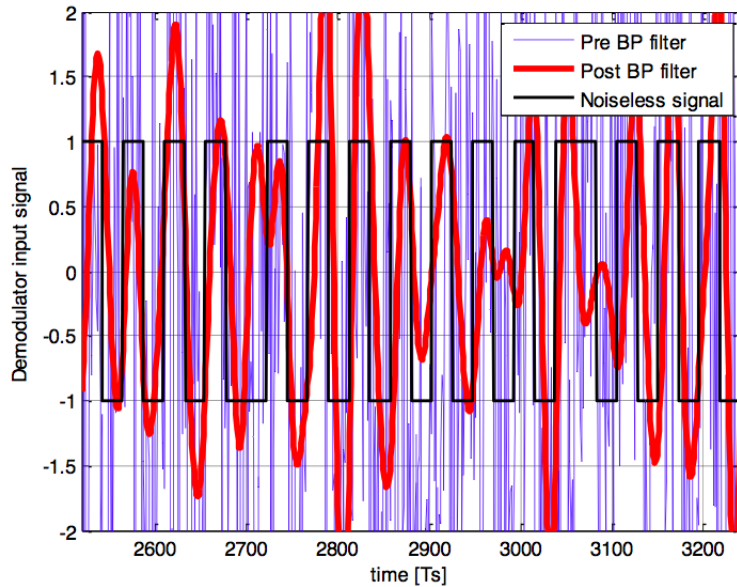


Figure 13: Bandpass Filtered Input Signal (SNR = 12dB; BW = 1-3.5MHz)

The Costas loop is initialized with a VCO free-running frequency of 2.2MHz, the average of the possible IF frequency range of 2-2.4MHz. The initial phase for the VCO is somewhat arbitrarily set at $\pi/3$. At high SNR, the Costas loop operates well and accurately demodulates the BPSK signal. The frequency tracking is also very fast, settling well within a data period. The bandlimited demodulated signal is then thresholded to get the RX data. The RX data is then Miller decoded to obtain the data bits.

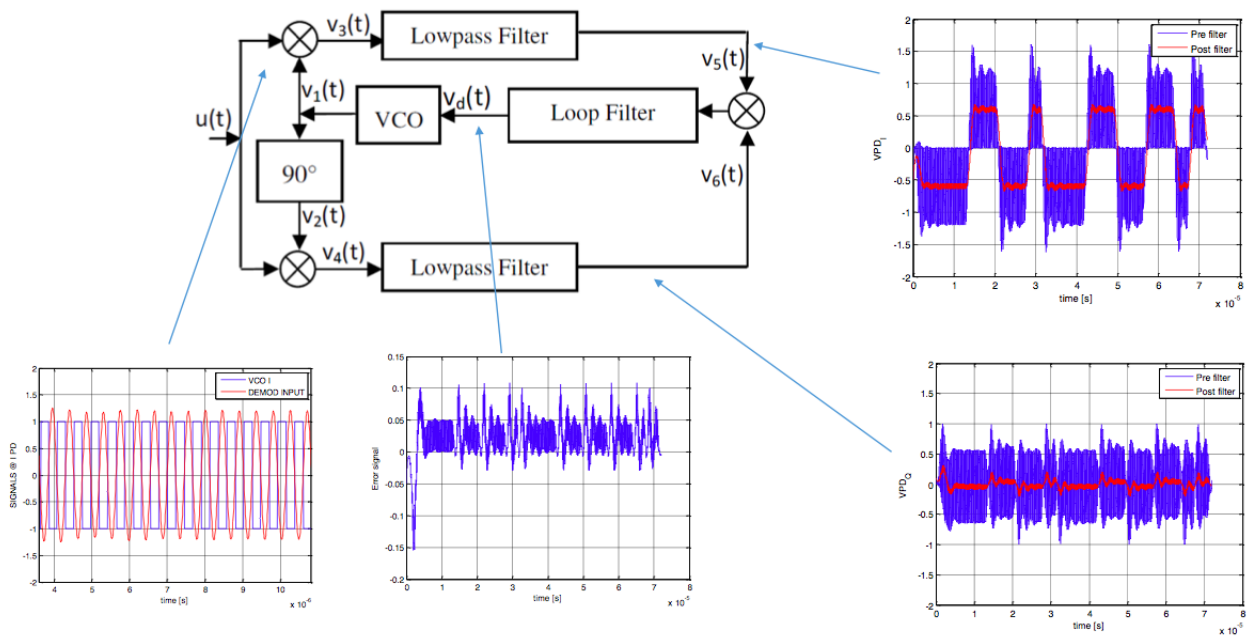


Figure 14: High SNR Costas Loop Operation

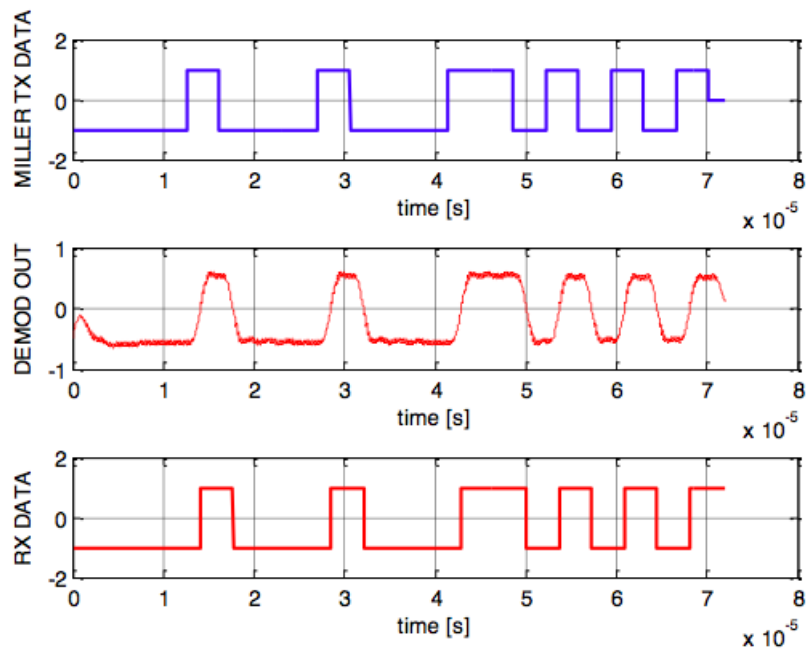


Figure 15: High SNR Costas Loop Demodulation Result

At an SNR of 20dB, however, performance is not as ideal, with the demodulated output missing some transitions in the received signal.

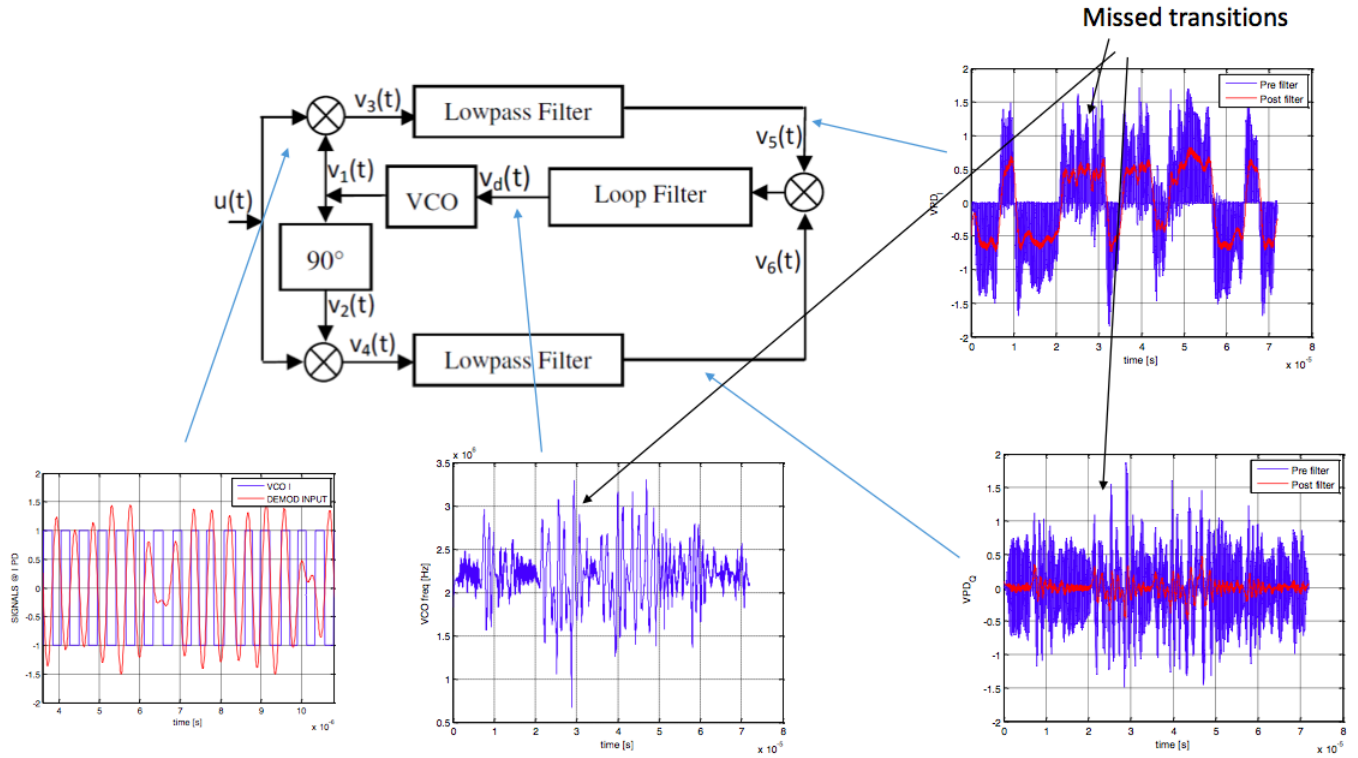


Figure 16: 20dB SNR Costas Loop Operation

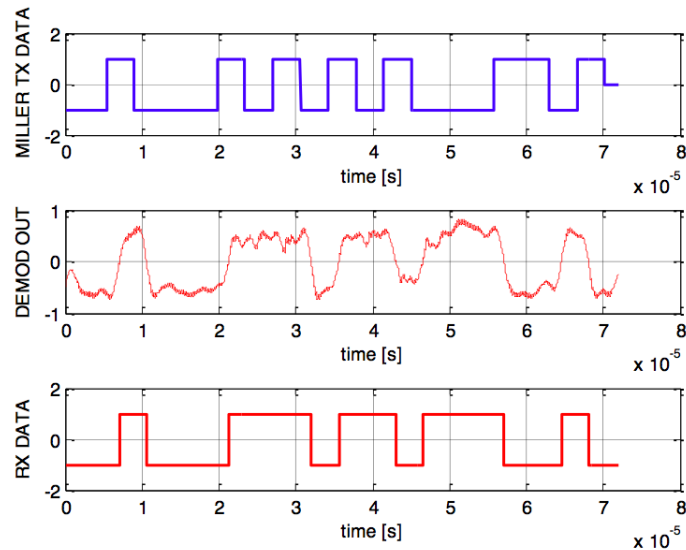


Figure 17: 20dB SNR Costas Loop Demodulation Result

With a smaller loop gain of 0.1 times the baseband bandwidth, the loop becomes more stable, especially for a square wave IF signal. The Costas loop accurately demodulates a signal with an IF frequency aligned with the initial VCO frequency. However, at 10% IF frequency mismatch, the Costas loop does not perform as well. Possible future solutions include adding a frequency acquisition loop with a frequency integrator. This additional loop however could contribute to the phase response causing the loop to become unstable. Parameters would need to be carefully calculated to achieve loop damping.

Looking at the Miller decoded data, the following BER vs. SNR information is obtained with 50000 data points. The curve matches the expected BER vs. SNR curve of a Costas loop but does not meet the 10^{-6} at 12dB bit error rate that we would like. Hence improvements can be made to reach that benchmark.

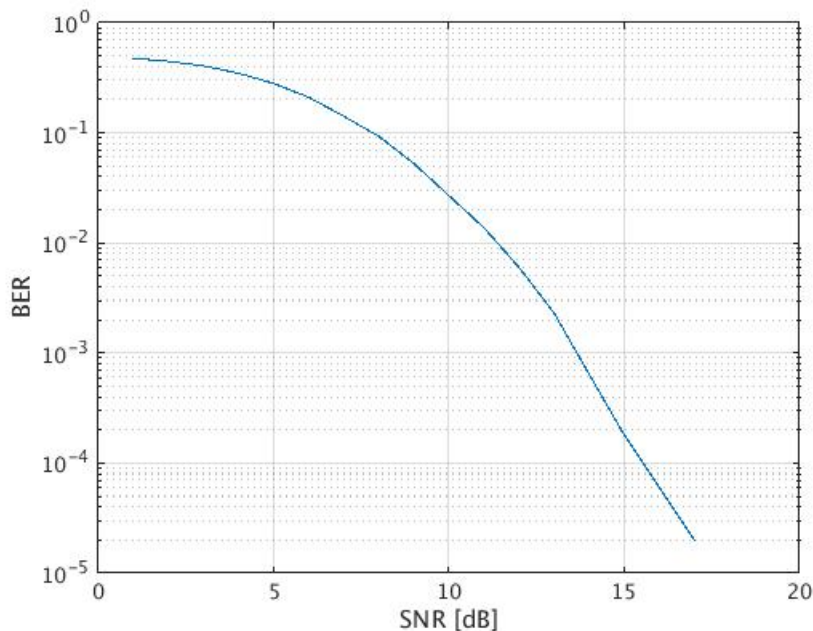


Figure 18: Bit Error Rate vs. SNR

5 Future Work & Conclusion

The next steps in this project would be to improve the BER vs. SNR performance by tweaking the filter and gain parameters of the Costas loop. Then simulations can be performed on Verilog to verify the MATLAB results. Finally, the FPGA can then be programmed with the Costas loop Verilog modules. Testing with a receive signal to verify functionality and accuracy can be done.

The Costas loop is an effective self-correcting demodulator as long as parameters for the channel and loop filters are chosen carefully. Specifically, first-order filters with appropriate pole placement will ensure noise removal while maintaining a stable loop. The Costas loop can also achieve quick convergence time, generating the carrier signal and demodulated signal in less than one data period. For fast loop convergence, the loop gain should be chosen to be about eight times the baseband bandwidth. However, for a square wave IF a lower value provides more stability. For this project's application, a smaller loop gain still achieves the IF frequency range necessary. Finally, the loop filter pole should be chosen such that it does not significantly contribute to the loop response. Its purpose is to remove high-frequency spurious noise components. To ensure stability, the loop filter pole was chosen to be set at 12 times the baseband bandwidth. The main challenge when designing a Costas loop is to ensure stability in the loop while balancing the noise contribution tradeoff.

6 References

- [1] K. McCaney, "New Technology Looks Deep to Identify Counterfeit Microchips", 2014.
- [2] D. Mutz, and K. George, "Costas Loop and FFT based BPSK Demodulation for Pulsed Radar Receivers," *2016 IEEE Aerospace Conference*
- [3] Roshna T. R. et al., "Design and Implementation of Digital Costas Loop and Bit Synchronizer in FPGA for BPSK Demodulation," *2013 International Conference on Control Communication and Computing*, Thiruvananthapuram, India, pp. 13 – 15, Dec. 2013.

- [4] X.Hu, H.Yuan and J.Huang, "Design and Implementation of Costas loop Based on FPGA", *IEEE Conference on Industrial Electronics and Applications*, ICIEA, 2008, pp.2383-2388.
- [5] C.J. Kikkert, and C. Blackburn, "Digitally Demodulating Binary Phase Shift Keyed Data Signals", Townsville, Qld, Australia.
- [6] J Feigin, "Practical Costas loop design," January 1, 2002, RF Design: 20–36.