

# Full-Duplex Shaped QPSK Modem

Data Sheet

July 23, 2003 ver. 1.7

## Target Applications:

Communications  
Digital Signal Processing



Nova Engineering, Inc.  
5 Circle Freeway Drive  
Cincinnati, OH 45246  
Tel. (513) 642-3000  
Fax (513) 642-3300  
E-mail sales@nova-eng.com  
WWW http://www.nova-eng.com

## Features

- Full-Duplex, low-latency design ( $< 700 \mu\text{s}$  round-trip delay) capable of burst processing of data
- Compliant with SCA v 2.0 to support Software Defined Radio Applications
- Designed to operate within 1.5 dB of theory at  $E_b/N_o$  of 4.0 dB and above.
- Designed to acquire carriers with frequency offset of  $\pm 1200$  Hz within 1 second.
- Data scrambling, differential encoding, and Forward Error Correction implemented as specified in MIL-STD-188-165
- PLD Provides ASIC Performance with Software Flexibility
- VHDL Source Code Modules are IEEE Std 1076-1993 Compliant
- Re-Programmable for Customized Interfaces and Processing Algorithms
- Parallel Processing to achieve computational efficiency

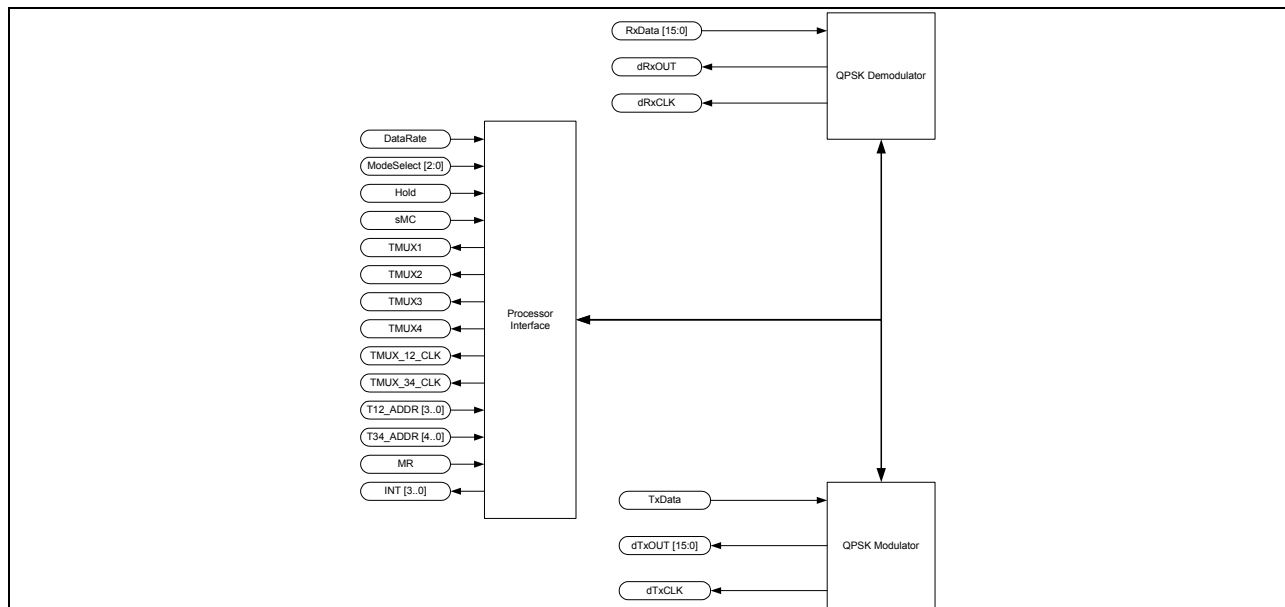


Figure I: Interface Diagram—Shaped QPSK Modem

## 1.0 General Description

The Shaped QPSK modem is designed to provide full duplex operation, with the ability to hold state information between bursts of information, thereby yielding an SCA compliant design capable of operating within an SDR (Software Defined Radio) framework. The VHDL design is constructed to perform Shaped QPSK modulation/demodulation with emphasis to support data rates of 64 kbps and 128 kbps in a PLD. VHDL is utilized when performing specific, computationally intensive tasks, thereby off-loading the systems software processor(s) in a manner similar to the classic co-processor.

The Shaped QPSK modem simultaneously executes multiple tasks providing true concurrent operation. Robust demodulation techniques are used in compatibility with MIL-STD-188-165. A self synchronizing Viterbi decoder is used to decode the convolutional code as specified in the standard. Furthermore, the demodulator contains an early/late gate detector, a frequency locked loop and a phase locked loop to efficiently and accurately acquire frequency, phase and time shifted signals. Additionally, the signal processing algorithms employ pipelining and parallelism to maximize power efficiency and processing speed.

## 2.0 Acronyms / Abbreviations

SQPSK	Shaped Quadrature Phase Shift Keying
kbps	Kilobits per second
ks <sub>ps</sub>	Kilosymbols per second
ks <sub>sps</sub>	Kilosamples per second
FEC	Forward Error Correction
LFSR	Linear Feedback Shift Register
AGC	Automatic Gain Control
E <sub>b</sub> /N <sub>0</sub>	Energy per Bit / Noise Energy
VHSIC	Very High Speed Integrated Circuit
VHDL	VHSIC Hardware Definition Language

### 3.0 Interface Description

The interfaces to the Shaped QPSK modulator/demodulator are described in Table 3-1. Signal types are defined as Input (I) and Output (O). Unused input pins on the FPGA should be pulled to ground. There is no performance impact concerning the state of unused pins.

**Table 3-1: Interface Descriptions**

SIGNAL	TYPE	DESCRIPTION
Peripheral Interfaces		
DataRate	I	Selects between 64 kbps and 128 kbps. 0 = 64 kbps, 1 = 128 kbps
TxModeSelect[2:0]	I	<b>Enables options on the encoder. (Refer to Table 3-2.)</b>
RxModeSelect[2:0]	I	<b>Enables options on the decoder. (Refer to Table 3-2.)</b>
System Clocks		
Tx_MC	I	<b>Transmit Master Clock.</b> Rate is 15 MHz.
Rx_MC	I	<b>Receive Master Clock.</b> Rate is 5.12 MHz for 64 kbps, FEC disabled. Rate is 10.24 MHz for 128 kbps, FEC disabled and for 64 kbps, FEC enabled. Rate is 20.48 MHz for 128 kbps, FEC enabled.
Data Interfaces		
TxData	I	<b>Transmit Data Input.</b> This is the data that is to be modulated for transmission.
TxFrameMarkIn	I	Transmit frame mark with clock period of 8 kHz, used for timing synchronization.
RxData[15:0]	I	<b>Receive Data Input.</b> The input sample stream is oversampled at a multiple of the data rate and synchronous with the Rx_MC.
RxFrameMarkIn	I	Receive frame mark with clock period of 32 kHz, 64 kHz or 128 kHz, used for timing synchronization.
dTxOUT[15:0]	O	<b>Transmit Data Output.</b> Processed receive data; synchronous to RxFramePulseOut.
TxFramePulseOut	O	Transmit frame pulse at the data rate.
dRxOUT	O	<b>Receive Data Output.</b> Modulated transmit data; synchronous to RxFramePulseOut.
RxFramePulseOut	O	Receive frame pulse at the data rate.
Control Interfaces		
Hold	I	Halts processing of data; system state is maintained and all input lines ignored until the hold signal returns to logic zero.
MR	I	<b>Master Reset (Active low).</b> Asynchronously resets the state of the device.

**Table 3-2: Operating Modes**

<b>MODESELECT</b>	<b>FEC ENABLED</b>	<b>DIFFERENTIAL ENCODING ENABLED</b>	<b>SCRAMBLING ENABLED</b>
0x0*			
0x1*			✓
0x2		✓	
0x3		✓	✓
0x4*	✓		
0x5*	✓		✓
0x6	✓	✓	
0x7	✓	✓	✓

\* If differential encoding is turned off, the demodulator may present data in an inverted format. There are four possible orientations in which the data would be presented.

1. All bits are inverted.
2. Odd and even numbered bits are swapped, and odd bits are inverted.
3. Odd and even numbered bits are swapped, and even bits are inverted.
4. Bits are received in the expected orientation.

The easiest way to mitigate these errors is to enable differential encoding at all times. However, an intelligently designed system can exploit known data patterns to detect when the modem locks onto the signal in a different phase orientation than expected. This problem is known as “phase ambiguity”, and must be handled at the system level if differential encoding is disabled.

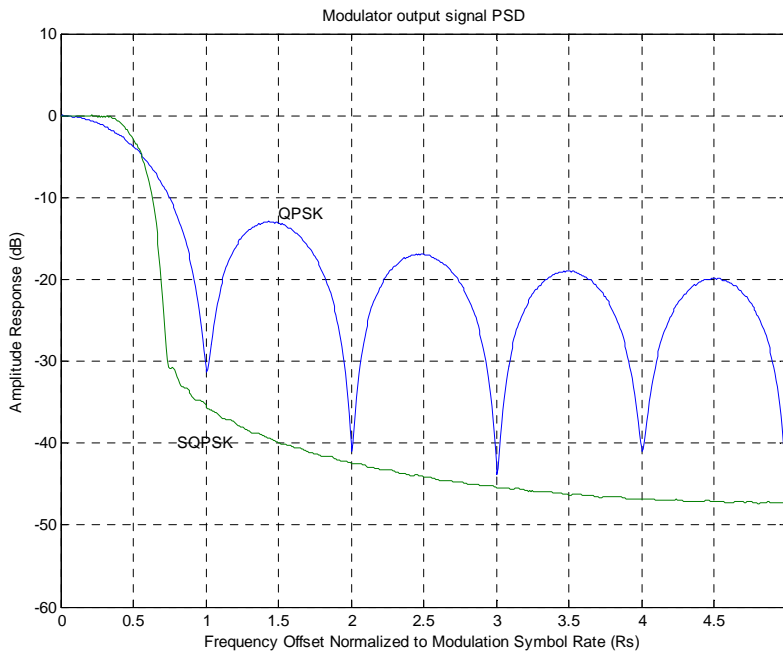
## 4.0 Functional Overview

### 4.1 Theory

The full-duplex Shaped QPSK modem consists of a modulator and demodulator. Modulating with a Shaped QPSK waveform offers superior spectral containment as compared to QPSK, while still maintaining the same data rate. This allows for efficient transmission of data, while simultaneously reducing the impact on any signal transmissions in neighboring frequency bands.

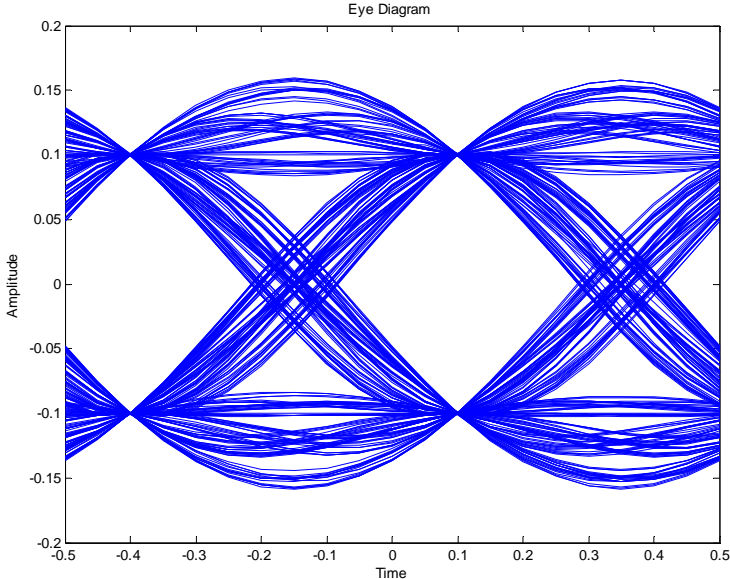
A Nyquist filter is used to shape the waveform. The impulse response of this filter is defined by the equation below.

$$h(t) = \frac{\sin(\pi \frac{t}{T}(1-\alpha)) + 4\alpha \frac{t}{T} \cos(\pi \frac{t}{T}(1+\alpha))}{\pi \frac{t}{T}(1-(4\alpha \frac{t}{T})^2)} \quad \begin{array}{l} T = \text{symbol period} \\ \alpha = 0.42 \text{ (in this application)} \end{array}$$



**Figure II: Spectral Containment**

The Nyquist filter used to obtain the spectral containment features above still maintains an open eye diagram, allowing for accurate symbol detection at the sampling instances.



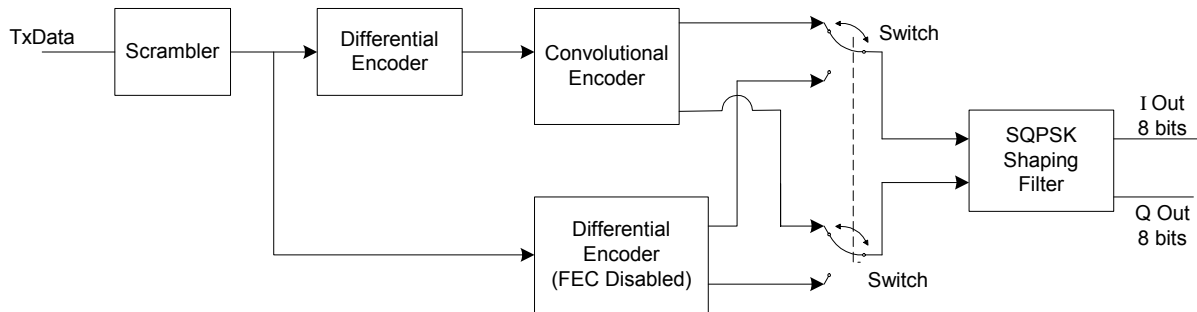
**Figure III: Eye Diagram**

## 4.2. Block Diagrams

The individual components of the modulating and demodulating system are illustrated below. Each of the functions is described further in the ensuing pages.

### 4.2.1. Modulator

Note: Scrambling and Differential Encoding operations act as pass-throughs when operating in a mode where these functions are disabled.



**Figure IV: Shaped QPSK Modulator**

### 4.2.2. Demodulator

Note: Descrambling and Differential Decoding operations act as pass-throughs when operating in a mode where these functions are disabled.

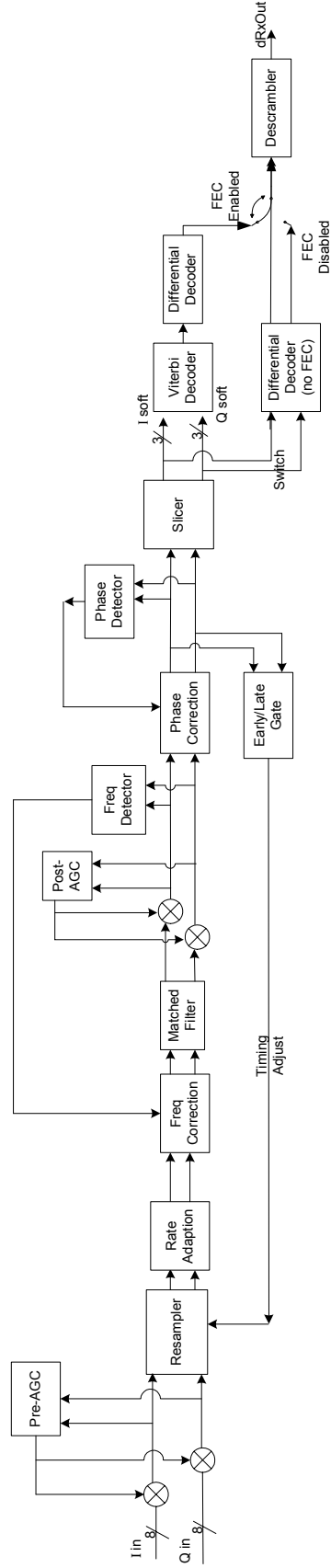


Figure V: Shaped QPSK Demodulator

### 4.3. Source Code List

#### 4.3.1. Packages & Shared Components

types\_pkg.vhd  
macro\_pkg.vhd  
rom.vhd

#### 4.3.2. Hex Files

fir\_lut\_ROM.hex  
resampler\_ROM\_c1.hex  
resampler\_ROM\_c2.hex  
resampler\_ROM\_c3.hex  
correction\_cos\_ROM.hex  
correction\_sin\_ROM.hex  
freq\_detector\_ROM.hex  
phase\_detector\_ROM.hex

#### 4.3.3. MDR Modem

mdr\_modem.vhd  
reset\_synch.vhd

#### 4.3.4. Modulator

mod\_timer.vhd  
scrambler.vhd  
diff\_enc\_fec\_en.vhd  
conv\_encoder.vhd  
diff\_enc\_fec\_dis.vhd  
fir\_lut.vhd  
fir\_filter.vhd  
sqpsk\_modulator.vhd

### 4.3.5. Demodulator

demod\_timer.vhd  
dual\_port\_ram\_a7\_d10.vhd  
pre\_agc\_scale\_gen.vhd  
saturating\_multiplier.vhd  
pre\_agc.vhd  
coefficient\_gen.vhd  
resample\_gen.vhd  
resampler.vhd  
dual\_port\_ram\_a3\_d16.vhd  
rate\_adaption.vhd  
freq\_correction.vhd  
symmetric\_filter.vhd  
matched\_filter.vhd  
post\_agc\_scale\_gen.vhd  
post\_agc.vhd  
phase\_correction.vhd  
divider2.vhd  
freq\_detector.vhd  
divider.vhd  
phase\_detector.vhd  
symbol\_trans\_detect.vhd  
early\_late\_gate.vhd  
slicer.vhd  
vit\_decoder.vhd  
viterbi\_decoder.vhd  
diff\_dec\_fec\_en.vhd  
diff\_dec\_fec\_dis.vhd  
descrambler.vhd  
sqpsk\_demodulator.vhd

### 4.4. Tree Diagram

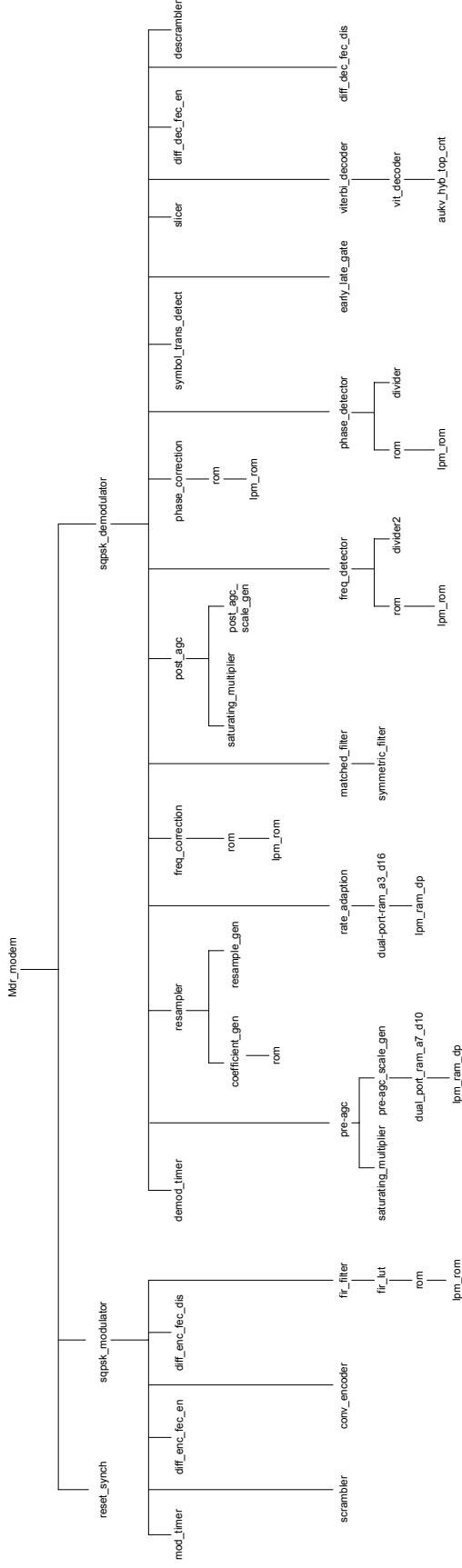


Figure VI: MDR Modem

## 5.0 Modulator Sub-block Functional Description

The following signal processing functions are implemented to modulate the Shaped QPSK waveform.

### 5.1. Modulator Timer

This module provides the timing for all components of the modulator. It runs off a 15 MHz clock where the incoming data rate is either 64 kbps or 128 kbps. Since 64 KHz and 128 KHz do not divide evenly into 15 MHz, the lowest factor of 128 KHz and 15 MHz is 8 KHz. Therefore, an 8 KHz frame mark is needed in order to be the basis for the synchronization of the clock and the internal component timing. An outgoing frame pulse is sent along with the outgoing modulated data.

**Table 5-1: Modulator Timer Interface Description**

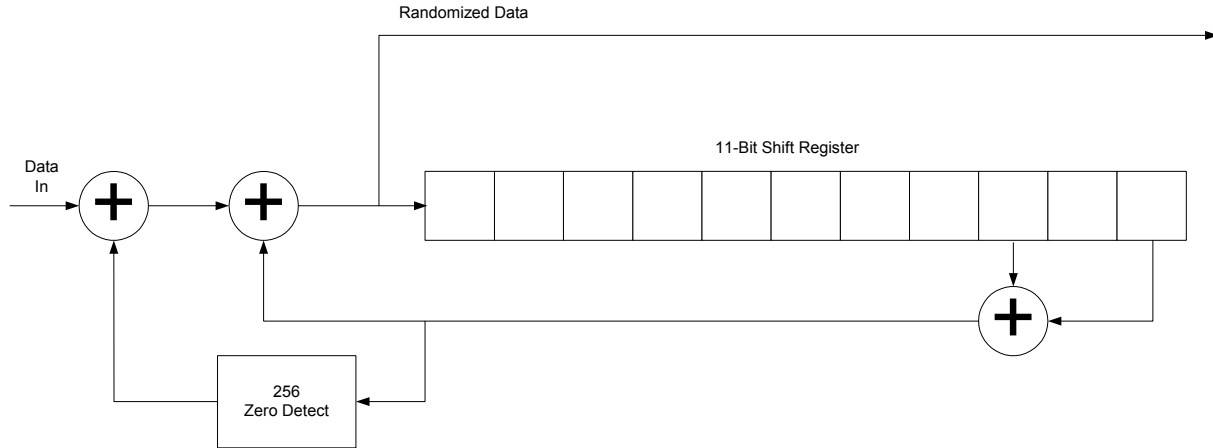
Signal	I / O	Description
CLOCK	I	15 MHz system clock.
RESET_N	I	Active low, synchronous reset.
SYSTEM_HOLD	I	System level hold input.
FRAME_MARK	I	8 kHz frame pulse used to synchronize timing.
RATE_SELECT	I	Selects between input data rate of 64 kbps or 128 kbps.
FEC_ENABLE	I	Enable for forward error correction.
MOD_HOLD_Z[6:0]	O	Hold signals for all modulator components to provide timing.
FRAME_Z	O	Frame pulse that is synchronous with outgoing transmit data.

### 5.2. Scrambler

This module is a simple linear feedback shift register used to scramble 1-bit data. The scrambler also has a second feedback path that is active if the main feedback of the LFSR is logic zero for 256 clock cycles. Scrambling can be enabled or disabled as desired. The circuit is described in Figure VII.

**Table 5-2: Scrambler Interface Description**

Signal	I / O	Description
CLOCK	I	15 MHz system clock.
RESET_N	I	Active low, synchronous reset.
MODULE_HOLD	I	Hold signal for this module at 64 kHz or 128 kHz.
SCRAM_ENABLE	I	Enables scrambling.
DATA	I	Incoming data at 64 kbps or 128 kbps.
SCRAM_DATA	O	Scrambled output data at 64 kbps or 128 kbps.



**Figure VII: Data Scrambling**

### 5.3. Differential Encoder, FEC Enabled

This module performs differential encoding when FEC is enabled. It takes a bit stream and encodes it using the equation  $C(k) = C(k-1) \text{ XOR } m(k)$ , where  $C(k)$  is the current differentially encoded bit,  $C(k-1)$  is the previous differentially encoded bit and  $m(k)$  is the current message bit. If differential encoding is disabled, encoding is bypassed and the output is the same as the input.

**Table 5-3: Differential Encoder, FEC Enabled Interface Description**

Signal	I / O	Description
CLOCK	I	15 MHz system clock.
RESET_N	I	Active low, synchronous reset.
MODULE_HOLD	I	Hold signal for this module at 64 kHz or 128 kHz.
DIFF_ENC_ENABLE	I	Enables differential encoding.
M_BIT	I	Current message bit at 64 kbps or 128 kbps.
ENC_BIT	O	Differentially encoded output data at 64 kbps or 128 kbps.

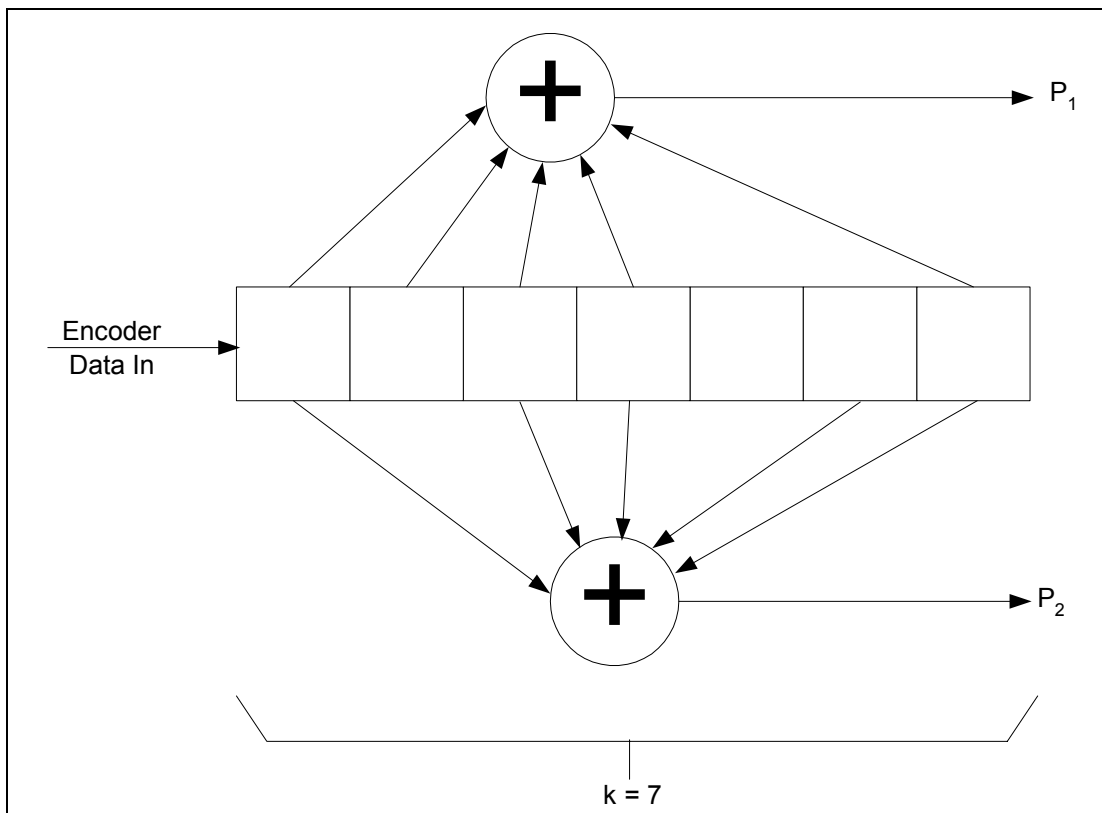
### 5.4. Convolutional Encoder

This module performs a rate 1/2 convolutional encoding on the data. The constraint length of the encoder is seven, as shown in Figure VIII. It takes in 1-bit data and encodes it into in-phase and quadrature components. The generating polynomials are of the following binary form:

P1 taps: 1111001, P2 taps: 1011011

**Table 5-4: Convolutional Encoder Interface Description**

Signal	I / O	Description
CLOCK	I	15 MHz system clock.
RESET_N	I	Active low, synchronous reset.
MODULE_HOLD	I	Hold signal for this module at 64 kHz or 128 kHz.
DATA	I	Incoming data at 64 kbps or 128 kbps.
ENC_DATA_I_Z	O	Outgoing convolutionally encoded in-phase data at 64 ks <sub>y</sub> ps or 128 ks <sub>y</sub> ps.
ENC_DATA_Q_Z	O	Outgoing convolutionally encoded quadrature data at 64 ks <sub>y</sub> ps or 128 ks <sub>y</sub> ps.



**Figure VIII: Rate 1/2 Convolutional Encoder**

## 5.5. Differential Encoder, FEC Disabled

This module performs differential encoding when FEC is disabled. It takes a bit stream and separates it into I and Q components. If differential encoding is enabled then the I and Q components are encoded using Table 5-6. If differential encoding is disabled, then the I and Q components are output without encoding. Data leaves this module at half the rate at which it comes in.

**Table 5-5: Differential Encoder, FEC Disabled Interface Description**

Signal	I / O	Description
CLOCK	I	15 MHz system clock.
RESET_N	I	Active low, synchronous reset.
MODULE_HOLD[1:0]	I	Hold signals for this module at 32 kHz or 64 kHz.
DIFF_ENC_ENABLE	I	Enables differential encoding.
M_BIT	I	Current message bit at 64 kbps or 128 kbps.
ENCODED_I	O	Differentially encoded in-phase output data at 32 ks <sub>y</sub> ps or 64 ks <sub>y</sub> ps.
ENCODED_Q	O	Differentially encoded quadrature output data at 32 ks <sub>y</sub> ps or 64 ks <sub>y</sub> ps.

**Table 5-6: Shaped QPSK Differential Encoding**

INPUT TO DIFFERENTIAL ENCODER		OUTPUT FROM DIFFERENTIAL ENCODER	
I	Q	I(k)	Q(k)
0	0	I(k-1)	Q(k-1)
0	1	$\overline{Q(k-1)}$	I(k-1)
1	0	$\overline{I(k-1)}$	$\overline{Q(k-1)}$
1	1	Q(k-1)	$\overline{I(k-1)}$

I = in-phase  
 Q = quadrature phase  
 (k) = current differentially encoded bit  
 (k-1) = previous differentially encoded bit

## 5.6. Finite Impulse Response Filter

This module is the finite impulse response filter for the in-phase and quadrature data. It consists of a pair of FIR look-up tables. It takes the data coming in and computes five sample values. Therefore, the output sample data rate is five times the input data rate. To generate the samples, the current data and the previous five data values are saved. Conceptually, a sum is computed by multiplying the data values by specifically-chosen filter coefficients.

**Table 5-7: FIR Filter Interface Description**

Signal	I / O	Description
CLOCK	I	15 MHz system clock.
RESET_N	I	Active low, synchronous reset.
MODULE_HOLD[1:0]	I	Hold signals for this module. Bit 1 is at 32 kHz, 64 kHz or 128 kHz. Bit 0 is at 160 kHz, 320 kHz or 640 kHz.
DATA_I	I	Current incoming in-phase data at 32 ks <sub>y</sub> ps, 64 ks <sub>y</sub> ps or 128 ks <sub>y</sub> ps.
DATA_Q	I	Current incoming quadrature data at 32 ks <sub>y</sub> ps, 64 ks <sub>y</sub> ps or 128 ks <sub>y</sub> ps.
SAMPLED_DATA_I[7:0]	O	Sampled outgoing in-phase data at 160 ks <sub>a</sub> ps, 320 ks <sub>a</sub> ps or 128 ks <sub>a</sub> ps.
SAMPLED_DATA_Q[7:0]	O	Sampled outgoing quadrature data at 160 ks <sub>a</sub> ps, 320 ks <sub>a</sub> ps or 128 ks <sub>a</sub> ps.

## 6.0 Demodulator Sub-block Functional Description

The following DSP functions are implemented to demodulate the Shaped QPSK waveform.

### 6.1. Demodulator Timer

This module provides the timing for all components of the demodulator. It runs off a clock that is variable depending on the mode selected. The incoming data rate is 160 kbps, 320 kbps or 640 kbps that have clock rates of 5.12 MHz, 10.24 MHz or 20.48 MHz respectively. At each clock rate, the lowest data rate necessary for a module is 160 clocks. Therefore, a 32 KHz (5.12 MHz / 160), 64 KHz (10.24 MHz / 160) or 128 KHz (20.48 MHz / 160) frame mark is needed for a corresponding data rate in order to be the basis for the synchronization of the clock and the internal component timing. An outgoing frame pulse is sent along with the outgoing descrambler data.

**Table 6-1: Modulator Timer Interface Description**

Signal	I / O	Description
CLOCK	I	5.12 MHz, 10.24 MHz or 20.48 MHz system clock.
RESET_N	I	Active low, synchronous reset.
SYSTEM_HOLD	I	System level hold input.
FRAME_MARK	I	32 kHz, 64 kHz or 128 kHz frame pulse used to synchronize timing.
FEC_ENABLE	I	Enable for forward error correction.
MOD_HOLD_Z[43:0]	O	Hold signals for all demodulator components to provide timing.
VIT_ENBL_Z	O	Enable signal for Viterbi Decoder.
FRAME_Z	O	Frame pulse that is synchronous with outgoing receive data.

### 6.2. Pre-AGC

This module performs automatic gain control manipulation to keep the average power fairly constant at a particular value. It determines the scale factor by which the data is to be multiplied. An average power is computed from the previous 128 scaled in-phase and quadrature data values. A scale factor is computed to adjust the incoming data to produce the desired power.

**Table 6-2: Pre-AGC Interface Description**

Signal	I / O	Description
CLOCK	I	5.12 MHz, 10.24 MHz or 20.48 MHz system clock.
RESET_N	I	Active low, synchronous reset.
MODULE_HOLD[12:0]	I	Hold signals for this module. All bits are at 160 kHz, 320 kHz or 640 kHz.
DATA_I	I	Current incoming in-phase data at 160 ks <sub>a</sub> ps, 320 ks <sub>a</sub> ps or 640 ks <sub>a</sub> ps.
DATA_Q	I	Current incoming quadrature data at 160 ks <sub>a</sub> ps, 320 ks <sub>a</sub> ps or 640 ks <sub>a</sub> ps.
AGC_DATA_I	O	Gain-controlled outgoing in-phase data at 160 ks <sub>a</sub> ps, 320 ks <sub>a</sub> ps or 640 ks <sub>a</sub> ps.
AGC_DATA_Q	O	Gain-controlled outgoing quadrature data at 160 ks <sub>a</sub> ps, 320 ks <sub>a</sub> ps or 640 ks <sub>a</sub> ps.

### 6.3. Resampler

This module operates in cadence with the Early/Late Gate to produce an output with an aligned peak sampling instance of the incoming oversampled data. It produces four valid resampled values for every five that enter. A valid indicator is used to inform the Rate Adaption FIFO whether or not a resample value is valid.

**Table 6-3: Resampler Interface Description**

Signal	I / O	Description
CLOCK	I	5.12 MHz, 10.24 MHz or 20.48 MHz system clock.
RESET_N	I	Active low, synchronous reset.
MODULE_HOLD[6:0]	I	Hold signals for this module. All bits are at 160 kHz, 320 kHz or 640 kHz.
ADJUST_STEP	I	Specifies that the step-size is to be adjusted.
STEP_DIRECTION	I	Specifies whether the step-size adjustment is positive or negative (logic zero -> negative, logic one -> positive)
DATA_I[7:0]	I	Current incoming in-phase data at 160 ks <sub>a</sub> ps, 320 ks <sub>a</sub> ps or 640 ks <sub>a</sub> ps.
DATA_Q[7:0]	I	Current incoming quadrature data at 160 ks <sub>a</sub> ps, 320 ks <sub>a</sub> ps or 640 ks <sub>a</sub> ps.
RESAMPLED_DATA_I[7:0]	O	Resampled outgoing in-phase data at 160 ks <sub>a</sub> ps, 320 ks <sub>a</sub> ps or 640 ks <sub>a</sub> ps.
RESAMPLED_DATA_Q[7:0]	O	Resampled outgoing quadrature data at 160 ks <sub>a</sub> ps, 320 ks <sub>a</sub> ps or 640 ks <sub>a</sub> ps.
VALID_INDICATOR	O	Indicates that the current resample value is valid at 160 ks <sub>a</sub> ps, 320 ks <sub>a</sub> ps or 640 ks <sub>a</sub> ps.

### 6.4. Rate Adaption FIFO

This module performs the rate adaption in going from five samples per symbol to four samples per symbol. A resampled value is written to the FIFO whenever it has been specified to be valid. A value is read out of the FIFO each time. Since the Resampler will normally indicate a valid sample four times for every five samples, the FIFO should not underflow or overflow.

**Table 6-4: Rate Adaption Interface Description**

Signal	I / O	Description
CLOCK	I	5.12 MHz, 10.24 MHz or 20.48 MHz system clock.
RESET_N	I	Active low, synchronous reset.
MODULE_HOLD[1:0]	I	Hold signals for this module. Bit 0 is at 160 kHz, 320 kHz or 640 kHz. Bit 1 is at 128 kHz, 256 kHz or 512 kHz.
WRITE_ENABLE	I	Indicates that the FIFO is to be written with the current data.
DATA_I[7:0]	I	Current incoming in-phase data at 160 ks <sub>a</sub> ps, 320 ks <sub>a</sub> ps or 640 ks <sub>a</sub> ps.
DATA_Q[7:0]	I	Current incoming quadrature data at 160 ks <sub>a</sub> ps, 320 ks <sub>a</sub> ps or 640 ks <sub>a</sub> ps.
RATE_ADAPTED_DATA_I[7:0]	O	Rate adapted outgoing in-phase data at 128 ks <sub>a</sub> ps, 256 ks <sub>a</sub> ps or 512 ks <sub>a</sub> ps.
RATE_ADAPTED_DATA_Q[7:0]	O	Resampled outgoing quadrature data at 128 ks <sub>a</sub> ps, 256 ks <sub>a</sub> ps or 512 ks <sub>a</sub> ps.

## 6.5. Frequency Correction

This module takes in-phase and quadrature data and rotates it according to an adjustment input from the Frequency Detector. Its purpose is to correct for any frequency offset within specification. It includes an NCO and a complex multiplier. It can correct up to plus or minus 1200 Hz worth of frequency offset.

**Table 6-5: Frequency Correction Interface Description**

Signal	I / O	Description
CLOCK	I	5.12 MHz, 10.24 MHz or 20.48 MHz system clock.
RESET_N	I	Active low, synchronous reset.
MODULE_HOLD[4:0]	I	Hold signals for this module. All bits are at 128 kHz, 256 kHz or 512 KHz.
I_DATA[7:0]	I	Current incoming in-phase data at 128 ks <sub>a</sub> ps, 256 ks <sub>a</sub> ps or 512 ks <sub>a</sub> ps.
Q_DATA[7:0]	I	Current incoming quadrature data at 128 ks <sub>a</sub> ps, 256 ks <sub>a</sub> ps or 512 ks <sub>a</sub> ps.
FREQ_ADJUST[16:0]	I	Amount of rotation needed to compensate for frequency offset at 32 ks <sub>y</sub> ps, 64 ks <sub>y</sub> ps or 128 ks <sub>y</sub> ps.
I_ADJUSTED[14:0]	O	Frequency corrected outgoing in-phase data at 128 ks <sub>a</sub> ps, 256 ks <sub>a</sub> ps or 512 ks <sub>a</sub> ps.
Q_ADJUSTED[14:0]	O	Frequency corrected outgoing quadrature data at 128 ks <sub>a</sub> ps, 256 ks <sub>a</sub> ps or 512 ks <sub>a</sub> ps.

## 6.6. Matched Filter

This module performs produces a signal of maximum magnitude at the sampling instance, which increases the likelihood of correct symbol decisions.

**Table 6-6: Matched Filter Interface Description**

Signal	I / O	Description
CLOCK	I	5.12 MHz, 10.24 MHz or 20.48 MHz system clock.
RESET_N	I	Active low, synchronous reset.
MODULE_HOLD[3:0]	I	Hold signals for this module. All bits are at 128 kHz, 256 kHz or 512 kHz.
DATA_I[7:0]	I	Current incoming in-phase data at 128 ks <sub>a</sub> ps, 256 ks <sub>a</sub> ps or 512 ks <sub>a</sub> ps.
DATA_Q[7:0]	I	Current incoming quadrature data at 128 ks <sub>a</sub> ps, 256 ks <sub>a</sub> ps or 512 ks <sub>a</sub> ps.
FILTERED_DATA_I[14:0]	O	Filtered outgoing in-phase data at 128 ks <sub>a</sub> ps, 256 ks <sub>a</sub> ps or 512 ks <sub>a</sub> ps.
FILTERED_DATA_Q[14:0]	O	Filtered outgoing quadrature data at 128 ks <sub>a</sub> ps, 256 ks <sub>a</sub> ps or 512 ks <sub>a</sub> ps.

## 6.7. Post-AGC

This module performs automatic gain control manipulation to keep the values fairly constant at a particular value. It determines the scale factor by which the data is to be multiplied. An average value is computed from the largest absolute value between the in-phase and quadrature data values. A scale factor is computed to adjust the incoming data to produce the desired values.

**Table 6-7: Matched Filter Interface Description**

Signal	I / O	Description
CLOCK	I	5.12 MHz, 10.24 MHz or 20.48 MHz system clock.
RESET_N	I	Active low, synchronous reset.
MODULE_HOLD[8:0]	I	Hold signals for this module. Bits 0-2 are at 128 kHz, 256 kHz or 512 kHz. Bits 3-8 are at 32 kHz, 64 kHz, 128 kHz.
DATA_I[14:0]	I	Current incoming in-phase data at 128 ks <sub>a</sub> ps, 256 ks <sub>a</sub> ps or 512 ks <sub>a</sub> ps.
DATA_Q[14:0]	I	Current incoming quadrature data at 128 ks <sub>a</sub> ps, 256 ks <sub>a</sub> ps or 512 ks <sub>a</sub> ps.
FILTERED_DATA_I[14:0]	O	Filtered outgoing in-phase data at 128 ks <sub>a</sub> ps, 256 ks <sub>a</sub> ps or 512 ks <sub>a</sub> ps.
FILTERED_DATA_Q[14:0]	O	Filtered outgoing quadrature data at 128 ks <sub>a</sub> ps, 256 ks <sub>a</sub> ps or 512 ks <sub>a</sub> ps.

## 6.8. Frequency Detector

This module takes in-phase and quadrature data and estimates the instantaneous frequency. It filters the instantaneous frequency values and produces an accumulated adjustment for the Frequency Correction module. Only the peak sampling instance contributes to the frequency estimation.

**Table 6-8: Frequency Detector Interface Description**

Signal	I / O	Description
CLOCK	I	5.12 MHz, 10.24 MHz or 20.48 MHz system clock.
RESET_N	I	Active low, synchronous reset.
MODULE_HOLD[6:0]	I	Hold signals for this module. All bits are at 32 kHz, 64 kHz or 128 kHz.
RATE_SELECT	I	Selects between input data rate of 64 kbps or 128 kbps.
FEC_ENABLE	I	Enable for forward error correction.
I_DATA[14:0]	I	Current incoming in-phase data at 128 ks <sub>a</sub> ps, 256 ks <sub>a</sub> ps or 512 ks <sub>a</sub> ps.
Q_DATA[14:0]	I	Current incoming quadrature data at 128 ks <sub>a</sub> ps, 256 ks <sub>a</sub> ps or 512 ks <sub>a</sub> ps.
FREQ_ADJUST[16:0]	O	Amount of rotation needed to compensate for frequency offset at 32 ks <sub>y</sub> ps, 64 ks <sub>y</sub> ps or 128 ks <sub>y</sub> ps.

## 6.9. Phase Correction

This module takes in-phase and quadrature data and rotates it according to an adjustment input from the Phase Detector. Its purpose is to correct for any phase offset within specification. It includes an NCO and a complex multiplier. It can correct for any amount of phase offset.

**Table 6-9: Phase Correction Interface Description**

Signal	I / O	Description
CLOCK	I	5.12 MHz, 10.24 MHz or 20.48 MHz system clock.
RESET_N	I	Active low, synchronous reset.
MODULE_HOLD[4:0]	I	Hold signals for this module. All bits are at 128 kHz, 256 kHz or 512 kHz.
I_DATA[7:0]	I	Current incoming in-phase data at 128 ks <sub>a</sub> ps, 256 ks <sub>a</sub> ps or 512 ks <sub>a</sub> ps.
Q_DATA[7:0]	I	Current incoming quadrature data at 128 ks <sub>a</sub> ps, 256 ks <sub>a</sub> ps or 512 ks <sub>a</sub> ps.
PHASE_ADJUST[14:0]	I	Amount of rotation needed to compensate for phase offset at 32 kbps, 64kbps or 128 kbps.
I_ADJUSTED[7:0]	O	Phase corrected outgoing in-phase data at 128 ks <sub>a</sub> ps, 256 ks <sub>a</sub> ps or 512 ks <sub>a</sub> ps.
Q_ADJUSTED[7:0]	O	Phase corrected outgoing quadrature data at 128 ks <sub>a</sub> ps, 256 ks <sub>a</sub> ps or 512 ks <sub>a</sub> ps.

## 6.10. Phase Detector

This module takes in-phase and quadrature data and estimates the phase error. The Phase Detector and Phase Correction modules comprise a first-order, closed loop synchronizer. The output of the phase detector is the difference, or phase error, between the input data stream and the desired angle of 45 degrees between I & Q. The phase error output from the phase detector is accumulated in an up-down counter, which increments and decrements according to the sign and magnitude of the phase error. Phase adjustments to the local clock phase are made in fixed step sizes. The step size, or magnitude, of the phase adjustment determines the loop acquisition time and data clock jitter. Large step sizes can be used to minimize acquisition times, since large phase steps can quickly correct large phase errors. Small step sizes can be used to minimize clock jitter when the loop is locked.

**Table 6-10: Phase Detector Interface Description**

Signal	I / O	Description
CLOCK	I	5.12 MHz, 10.24 MHz or 20.48 MHz system clock.
RESET_N	I	Active low, synchronous reset.
MODULE_HOLD[6:0]	I	Hold signals for this module. All bits are at 32 kHz, 64 kHz or 128 kHz.
FEC_ENABLE	I	Enable for forward error correction.
I_DATA[7:0]	I	Current incoming in-phase data at 128 ks <sub>a</sub> ps, 256 ks <sub>a</sub> ps or 512 ks <sub>a</sub> ps.
Q_DATA[7:0]	I	Current incoming quadrature data at 128 ks <sub>a</sub> ps, 256 ks <sub>a</sub> ps or 512 ks <sub>a</sub> ps.
PHASE_ADJUST[14:0]	O	Amount of rotation needed to compensate for phase offset at 32 ks <sub>y</sub> ps, 64 ks <sub>y</sub> ps or 128 ks <sub>y</sub> ps.

## 6.11. Symbol Transition Detect

This module determines whether a symbol transition has been encountered. A symbol transition is defined as a change in the sign bits of both the in-phase and quadrature data from the previous sign bits. Only the peak sampling instance is used to determine a symbol transition.

**Table 6-11: Symbol Transition Detect Interface Description**

Signal	I / O	Description
CLOCK	I	5.12 MHz, 10.24 MHz or 20.48 MHz system clock.
RESET_N	I	Active low, synchronous reset.
MODULE_HOLD	I	Hold signal for this module at 32 kHz, 64 kHz or 128 kHz.
DATA_I	I	Current incoming in-phase sign bit at 128 ks <sub>a</sub> ps, 256 ks <sub>a</sub> ps or 512 ks <sub>a</sub> ps.
DATA_Q	I	Current incoming quadrature sign bit at 128 ks <sub>a</sub> ps, 256 ks <sub>a</sub> ps or 512 ks <sub>a</sub> ps.
SYMBOL_TRANSITION_Z	O	Indicates that a symbol transition took place at 32 ks <sub>y</sub> ps, 64 ks <sub>y</sub> ps or 128 ks <sub>y</sub> ps.

## 6.12. Early/Late Gate

This module takes in-phase and quadrature data and estimates the time error. It drives the Resampler to produce data such that one of the four samples per symbol is the peak sampling instance. The output of the Resampler is advanced or retarded whenever the error accumulator exceeds a specified threshold. The error threshold is capable of being changed at compilation time and is used to control the bandwidth of the loop filter. The loop bandwidth can be narrowed by increasing the error threshold and widened by decreasing the error threshold. Small error thresholds allow the filter to respond to rapid changes in the timing alignment. Only the peak sampling instance contributes to the time error estimation.

**Table 6-12: Early/Late Gate Interface Description**

Signal	I / O	Description
CLOCK	I	5.12 MHz, 10.24 MHz or 20.48 MHz system clock.
RESET_N	I	Active low, synchronous reset.
MODULE_HOLD[5:0]	I	Hold signals for this module. All bits are at 32 kHz, 64 kHz or 128 kHz.
SYMBOL_TRANSITION	I	Indicates that a symbol transition took place at 32 kHz, 64 kHz or 128 kHz.
DATA_I[7:0]	I	Current incoming in-phase data at 128 ks <sub>a</sub> ps, 256 ks <sub>a</sub> ps or 512 ks <sub>a</sub> ps.
DATA_Q[7:0]	I	Current incoming quadrature data at 128 ks <sub>a</sub> ps, 256 ks <sub>a</sub> ps or 512 ks <sub>a</sub> ps.
ADJUST_STEP_Z	O	Specifies that the timing is to be adjusted.
STEP_DIRECTION_Z	O	Specifies whether the stepsize adjustment is positive or negative (zero -> negative, one -> positive)

### 6.13. Data Slicer

The slicer reduces the output of the Phase Correction module from 8-bit values down to 3-bit soft values. Only the peak sampling instance is processed.

**Table 6-13: Data Slicer Interface Description**

Signal	I / O	Description
CLOCK	I	5.12 MHz, 10.24 MHz or 20.48 MHz system clock.
RESET_N	I	Active low, synchronous reset.
MODULE_HOLD	I	Hold signal for this module at 32 kHz, 64 kHz or 128 kHz.
I_DATA[7:0]	I	Current incoming in-phase sign bit at 128 ks <sub>a</sub> ps, 256 ks <sub>a</sub> ps or 512 ks <sub>a</sub> ps.
Q_DATA[7:0]	I	Current incoming quadrature sign bit at 128 ks <sub>a</sub> ps, 256 ks <sub>a</sub> ps or 512 ks <sub>a</sub> ps.
SLICED_I_Z[2:0]	O	Sliced outgoing in-phase data at 32 ks <sub>y</sub> ps, 64 ks <sub>y</sub> ps or 128 ks <sub>y</sub> ps.
SLICED_Q_Z[2:0]	O	Sliced outgoing quadrature data at 32 ks <sub>y</sub> ps, 64 ks <sub>y</sub> ps or 128 ks <sub>y</sub> ps.

### 6.14. Viterbi Decoder (COTS)

The module decoder removes the convolutional encoding of the information bits, and uses the convolutional code to detect and reduce the instances of bit errors in making symbol decisions. The Viterbi decoder is self-synchronizing, detecting data in all possible phase rotations and locking onto the correct rotation.

**Table 6-14: Viterbi Decoder Interface Description**

Signal	I / O	Description
CLOCK	I	5.12 MHz, 10.24 MHz or 20.48 MHz system clock.
RESET_N	I	Active low, synchronous reset.
MODULE_HOLD	I	Hold signal for this module at 64 kHz or 128 kHz.
ENABLE	I	Enables the decoder to process data and adapts the decoder to decode at the correct data rate. This signal is at 64 kHz or 128 kHz.
DATA_I[2:0]	I	Current incoming in-phase soft bit data at 64 ks <sub>y</sub> ps or 128 ks <sub>y</sub> ps.
DATA_Q[2:0]	I	Current incoming quadrature sign bit at 64 ks <sub>y</sub> ps or 128 ks <sub>y</sub> ps.
DECODED_DATA_Z	O	Decoded outgoing data at 64 kbps or 128 kbps.

### 6.15. Differential Decoder, FEC Enabled

This module removes the encoding imparted on the signal by the FEC enabled differential encoder in the Modulator. It is necessary because the Viterbi Decoder cannot distinguish between two pairs of symbol rotations. If differential decoding is disabled, then the output data is the same as the input data.

**Table 6-15: Differential Decoder, FEC Enabled Interface Description**

Signal	I / O	Description
CLOCK	I	5.12 MHz, 10.24 MHz or 20.48 MHz system clock.
RESET_N	I	Active low, synchronous reset.
MODULE_HOLD	I	Hold signal for this module at 64 kHz or 128 kHz.
DIFF_DEC_ENABLE	I	Enables differential decoding.
ENC_BIT	I	Current incoming Viterbi decoded bit at 64 kbps or 128 kbps.
DEC_BIT	O	Differentially decoded output data at 64 kbps or 128 kbps.

### 6.16. Differential Decoder, FEC Disabled

This module removes the encoding imparted on the signal by the FEC disabled differential encoder in the Modulator. If differential decoding is disabled, then the output data is the same as the input data.

**Table 6-16: Differential Decoder, FEC Disabled Interface Description**

Signal	I / O	Description
CLOCK	I	5.12 MHz, 10.24 MHz or 20.48 MHz system clock.
RESET_N	I	Active low, synchronous reset.
MODULE_HOLD[1:0]	I	Hold signals for this module at 32 kHz or 64 kHz.
DIFF_DEC_ENABLE	I	Enables differential decoding.
ENCODED_I	I	Current incoming in-phase sign bit at 32 ks <sub>y</sub> ps or 64 ks <sub>y</sub> ps.
ENCODED_Q	I	Current incoming quadrature sign bit at 32 ks <sub>y</sub> ps or 64 ks <sub>y</sub> ps.
DECODED_DATA_Z	O	Differentially decoded output data at 64 kbps or 128 kbps.

## 6.17. Descrambler

This module unscrambles the data sequence, as was applied by the Scrambler in the Modulator.

**Table 6-17: Descrambler Interface Description**

Signal	I / O	Description
CLOCK	I	5.12 MHz, 10.24 MHz or 20.48 MHz system clock.
RESET_N	I	Active low, synchronous reset.
MODULE_HOLD	I	Hold signal for this module at 64 kHz or 128 kHz.
SCRAM_ENABLE	I	Enables descrambling.
DATA	I	Current incoming differentially decoded data at 64 kbps or 128 kbps.
DESCRAM_DATA	O	Descrambled output data at 64 kbps or 128 kbps.

## 7.0 Testing

All test vectors were generated from Matlab and were used to verify the C model and VHDL design. Modelsim was the simulator used to verify the VHDL. The following is a subset of test vectors that were executed to verify operation of the Full-Duplex Shaped QPSK modem. No sub-block test vectors have been provided. The test vector files are listed in the Appendix.

**Table 7-1: SQPSK Modulator Test Vector Settings**

Rate	Forward Error Correction	Differential Encoding	Scrambling
64kbps	OFF	OFF	OFF
128kbps	OFF	OFF	ON
64kbps	OFF	ON	OFF
128kbps	OFF	ON	ON
128kbps	OFF	OFF	OFF
64kbps	OFF	OFF	ON
128kbps	OFF	ON	OFF
64kbps	OFF	ON	ON
64kbps	ON	OFF	OFF
128kbps	ON	OFF	ON
64kbps	ON	ON	OFF
128kbps	ON	ON	ON
128kbps	ON	OFF	OFF

**Table 7-2: SQPSK Demodulator Test Vector Settings**

$E_b / N_o$	Frequency Offset	Phase Offset	Rate	Forward Error Correction	Differential Encoding	Scrambling	Acquisition Time	Bit Error Rate
10.0	0	0	64kbps	OFF	OFF	OFF	0 $\mu$ s	$5.5 \times 10^{-5}$
4.0	1200	$\pi/3$	64kbps	OFF	ON	OFF	700 $\mu$ s	$3.78 \times 10^{-2}$
4.0	-1200	$\pi/3$	64kbps	OFF	ON	OFF	700 $\mu$ s	$3.80 \times 10^{-2}$
4.0	1200	$2\pi/3$	64kbps	OFF	ON	OFF	500 $\mu$ s	$3.92 \times 10^{-2}$
4.0	-1200	$-2\pi/3$	64kbps	ON	ON	ON	550 $\mu$ s	0
4.0	1200	$\pi/4$	128kbps	OFF	ON	ON	700 $\mu$ s	$7.86 \times 10^{-2}$
4.0	-1200	$-\pi/4$	128kbps	OFF	ON	OFF	550 $\mu$ s	$3.86 \times 10^{-2}$
4.0	1200	$3\pi/4$	64kbps	ON	ON	OFF	725 $\mu$ s	0
4.0	-1200	$-3\pi/4$	128kbps	ON	ON	OFF	400 $\mu$ s	0
4.0	1200	0	64kbps	OFF	ON	OFF	700 $\mu$ s	$3.86 \times 10^{-2}$
4.0	-1200	$\pi/2$	64kbps	ON	ON	OFF	400 $\mu$ s	0
4.0	1200	$\pi$	128kbps	ON	ON	OFF	400 $\mu$ s	0

**Table 7-3: SQPSK Demodulator Doppler Testing**

$E_b/N_o$	Frequency Offset	Frequency Change	Phase Offset	Rate	Forward Error Correction	Differential Encoding	Scrambling	Acquisition Time	Bit Error Rate
4.5	1200	20 Hz/s	$3\pi/4$	64kbps	OFF	ON	OFF	475 $\mu$ s	$2.69 \times 10^{-2}$
4.5	1200	20 Hz/s	$3\pi/4$	128kbps	OFF	ON	OFF	410 $\mu$ s	$2.65 \times 10^{-2}$
4.5	-1200	20 Hz/s	$3\pi/4$	64kbps	OFF	ON	OFF	660 $\mu$ s	$2.75 \times 10^{-2}$
4.5	-1200	20 Hz/s	$3\pi/4$	128kbps	OFF	ON	OFF	430 $\mu$ s	$2.58 \times 10^{-2}$

## 8.0 Overall Performance Limitations

Acceptable Bit Rates: 64 kbps, 128 kbps

### 8.1. Modulator

System Latency:

- < 33 $\mu$ S for data rate of 64 kbps and FEC disabled
- < 17 $\mu$ S for data rate of 128 kbps, FEC disabled
- < 17 $\mu$ S for data rate of 64 kbps, FEC enabled
- < 10 $\mu$ S for data rate of 128 kbps, FEC enabled

### 8.2. Demodulator

System Latency:

- < 455 $\mu$ S for data rate of 64 kbps and FEC disabled
- < 230 $\mu$ S for data rate of 128 kbps, FEC disabled
- < 685 $\mu$ S for data rate of 64 kbps, FEC enabled
- < 345 $\mu$ S for data rate of 128 kbps, FEC enabled

$E_b/N_o$ : Operates within 1.5 dB of theory for  $E_b/N_o$  down to 4.0 dB.

Frequency Offset Limits: Up to plus or minus 1200 Hz

Phase Offset Limits: 0 to  $2\pi$

Time Offset Limits: None

## 9.0 Synthesis Results

Modulator Maximum Clock Rate: 65.02 MHz\*

Demodulator Maximum Clock Rate: 28.66 MHz\*

Logic Elements Consumed: 8,963 / 16,640 (53%)\*\*

Embedded System Blocks Consumed: 119,680 / 212,992 (56%)\*\*

Timing Constraints for Physical Synthesis: None

Layout and Placement Constraints: None

Clock Constraints: Tx\_MC is 15 MHz, Rx\_MC is 20.48 MHz

Critical Path Constraints: None

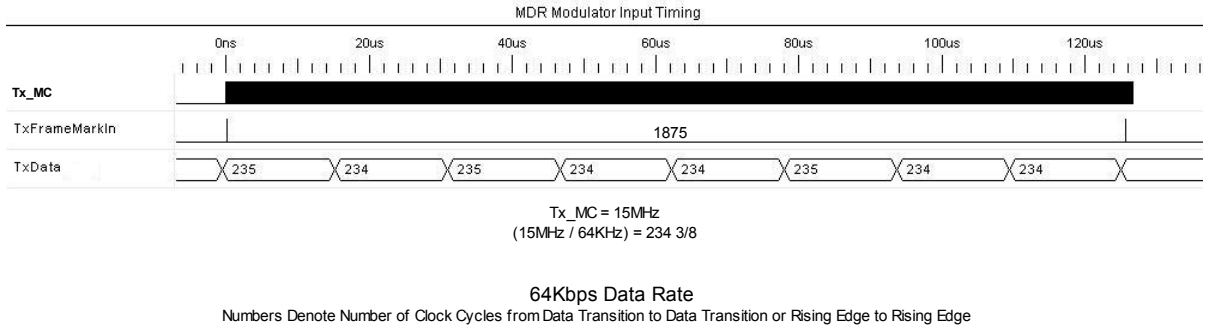
False Paths: None\*\*\*

\* No clock rates other than those specified in the Interface Description section can be used without modification to internal timing circuitry. These rates are a theoretical maximum according to Quartus II static timing analysis.

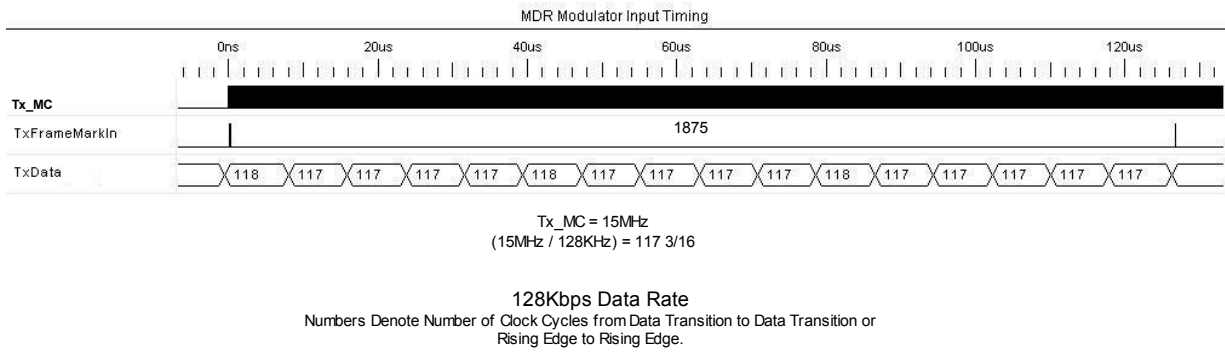
\*\* Results for Altera EP20K400EFC672-2X part.

\*\*\* The modem was synthesized with synchronous clocking of inputs. However, those inputs listed in the Peripheral and Control Interfaces sections of Table 3-1 can be asynchronous.

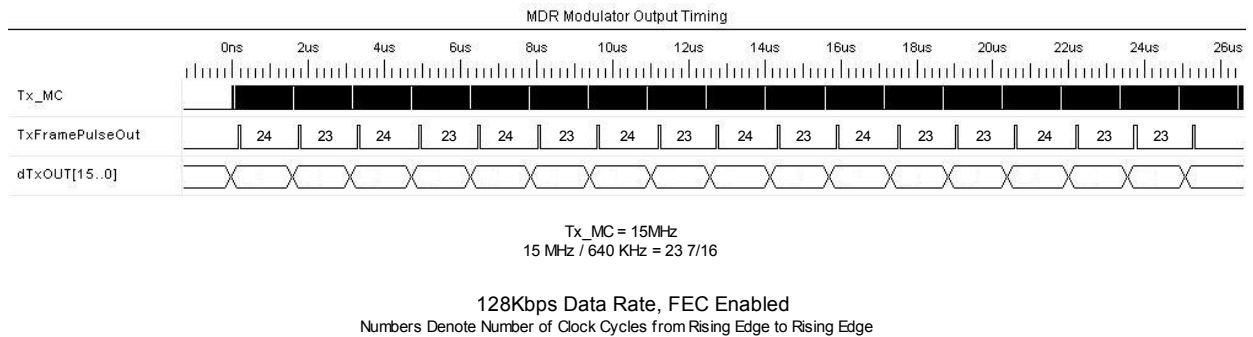
## 10.0 Timing Diagrams



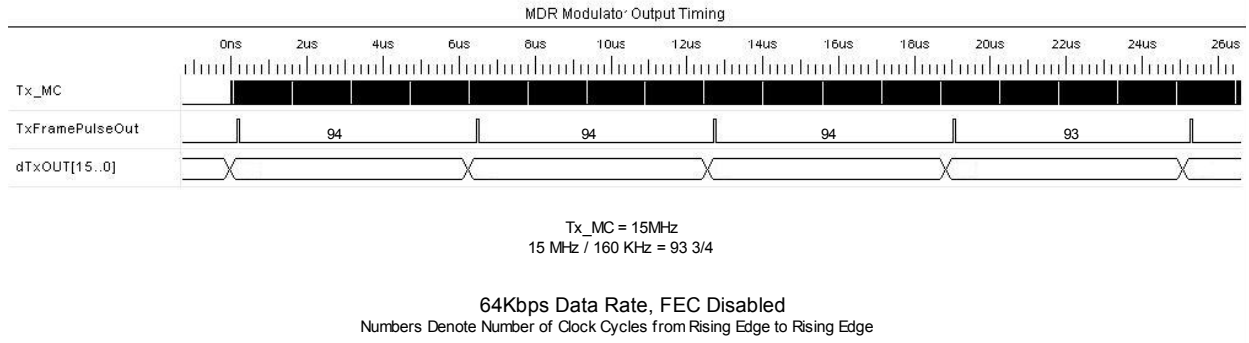
**Figure IX: Input Timing 64Kbps**



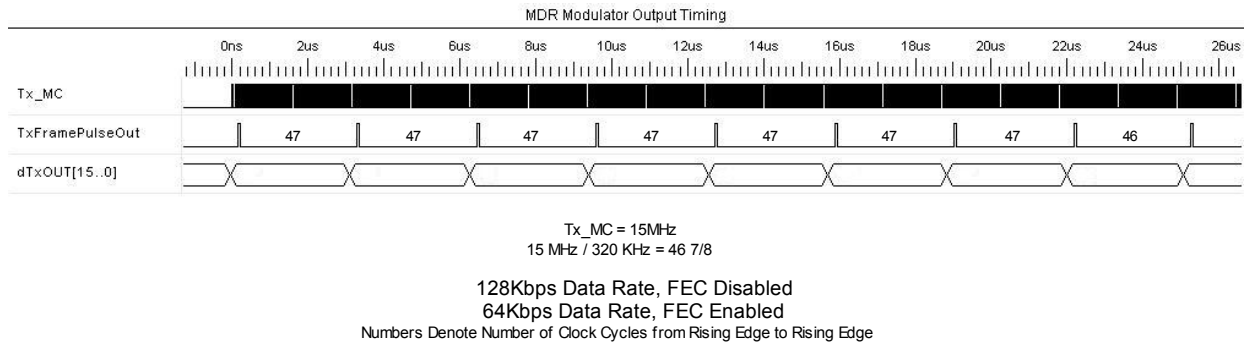
**Figure X: Input Timing 128Kbps**



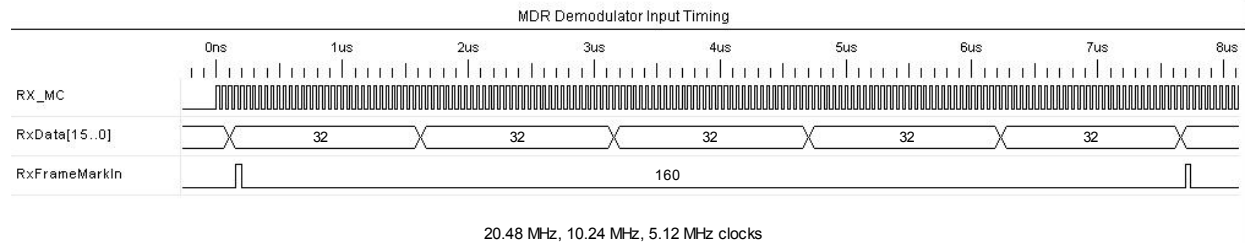
**Figure XI: Output Timing 128Kbps, FEC Enabled**



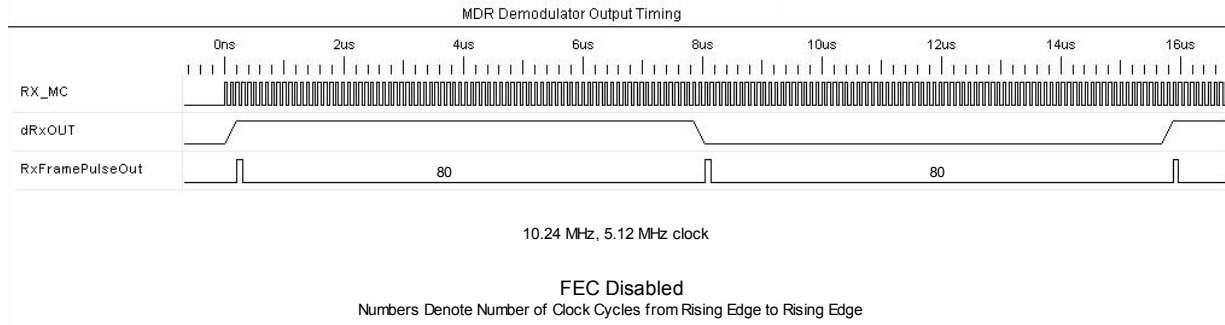
**Figure XII: Output Timing 64Kbps, FEC Disabled**



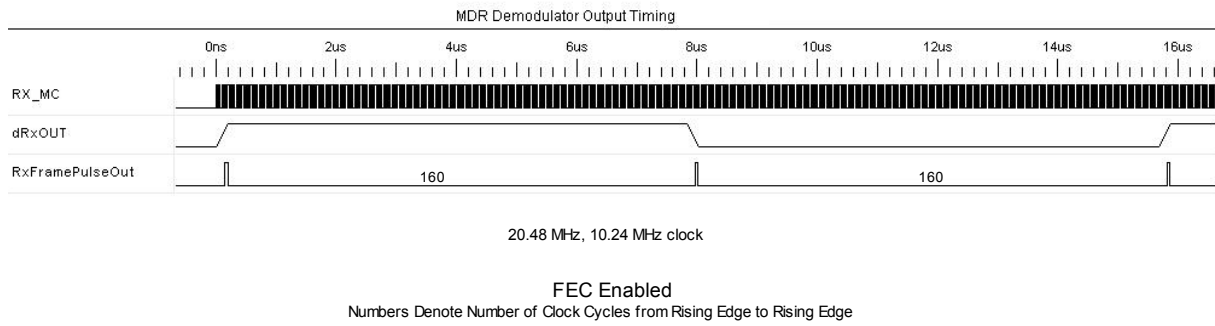
**Figure XIII: Output Timing 128Kbps, FEC Disabled & 64Kbps, FEC Enabled**



**Figure XIV: Demodulator Input Timing**



**Figure XV: Demodulator Output Timing, FEC Disabled**



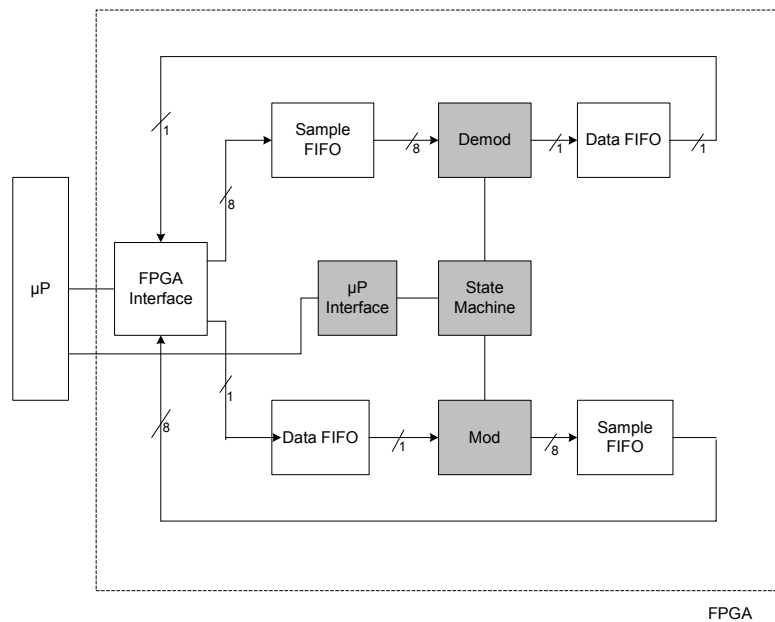
**Figure XVI: Demodulator Output Timing, FEC Enabled**

## 11.0 Test Features

A Matlab model of the modem has been constructed to generate test vectors. These test vectors are used to verify the performance of the system and each of the individual components.

## 12.0 Application Note

Figure XVII demonstrates a typical application of the Shaped QPSK modem. Shaded items are the primary modules in the Shaped QPSK modem. The other elements shown are necessary to utilize the Shaped QPSK modem, and are expected to exist on the target platform for this VHDL component.



**Figure XVII: Typical Shaped QPSK Modem Application**

The FIFOs shown in Figure 12-1 are retiming FIFOs. They are used to cross clock domains. The modem assumes that the data inputs are synchronous with their clocks. Therefore, these FIFOs are necessary if the data inputs are not synchronous with their clocks. There are no known constraints associated with these FIFOs; their only use is to cross the data onto the modem's clocks.

## Appendix:

Below are the lists of test vectors used to verify the modem. The parentheses after the file names signifies whether the file is an stimulus (I) file or an expect (E) file.

### 1.1 Modulator

Under each of the following directories are these files:

- data.txt (I)
- filter\_data\_i.txt (E)
- filter\_data\_q.txt (E)

- Modulator/Rate\_64K\_FEC\_0\_DiffEnc\_0\_Scr\_0/Matlab/
- Modulator/Rate\_64K\_FEC\_0\_DiffEnc\_0\_Scr\_1/Matlab/
- Modulator/Rate\_64K\_FEC\_0\_DiffEnc\_1\_Scr\_0/Matlab/
- Modulator/Rate\_64K\_FEC\_0\_DiffEnc\_1\_Scr\_1/Matlab/
- Modulator/Rate\_64K\_FEC\_1\_DiffEnc\_0\_Scr\_0/Matlab/
- Modulator/Rate\_64K\_FEC\_1\_DiffEnc\_0\_Scr\_1/Matlab/
- Modulator/Rate\_64K\_FEC\_1\_DiffEnc\_1\_Scr\_0/Matlab/
- Modulator/Rate\_64K\_FEC\_1\_DiffEnc\_1\_Scr\_1/Matlab/
- Modulator/Rate\_128K\_FEC\_0\_DiffEnc\_0\_Scr\_0/Matlab/
- Modulator/Rate\_128K\_FEC\_0\_DiffEnc\_0\_Scr\_1/Matlab/
- Modulator/Rate\_128K\_FEC\_0\_DiffEnc\_1\_Scr\_0/Matlab/
- Modulator/Rate\_128K\_FEC\_0\_DiffEnc\_1\_Scr\_1/Matlab/
- Modulator/Rate\_128K\_FEC\_1\_DiffEnc\_0\_Scr\_0/Matlab/
- Modulator/Rate\_128K\_FEC\_1\_DiffEnc\_0\_Scr\_1/Matlab/
- Modulator/Rate\_128K\_FEC\_1\_DiffEnc\_1\_Scr\_0/Matlab/
- Modulator/Rate\_128K\_FEC\_1\_DiffEnc\_1\_Scr\_1/Matlab/

### 1.2 Demodulator

Under each of the following directories are these files:

- Matlab/mesg\_bits.txt
- Manipulation/conv\_data\_i.txt (I)
- Manipulation/conv\_data\_q.txt (I)
- Matlab/demod\_output.txt (E)

- Demodulator/SN\_10.0\_Fr\_0\_Ph\_0\_Tm\_0.00/Rate\_64K\_FEC\_0\_DiffEnc\_1\_Scr\_0/
- Demodulator/SN\_4.0\_Fr\_1200\_Ph\_60\_Tm\_0.00/Rate\_64K\_FEC\_0\_DiffEnc\_1\_Scr\_0/
- Demodulator/SN\_4.0\_Fr\_-1200\_Ph\_60\_Tm\_0.00/Rate\_64K\_FEC\_0\_DiffEnc\_1\_Scr\_0/
- Demodulator/SN\_4.0\_Fr\_1200\_Ph\_120\_Tm\_0.00/Rate\_64K\_FEC\_0\_DiffEnc\_1\_Scr\_0/
- Demodulator/SN\_4.0\_Fr\_-1200\_Ph\_-120\_Tm\_0.00/Rate\_64K\_FEC\_1\_DiffEnc\_1\_Scr\_1/
- Demodulator/SN\_4.0\_Fr\_1200\_Ph\_45\_Tm\_0.00/Rate\_128K\_FEC\_0\_DiffEnc\_1\_Scr\_1/
- Demodulator/SN\_4.0\_Fr\_-1200\_Ph\_-45\_Tm\_0.00/Rate\_128K\_FEC\_0\_DiffEnc\_1\_Scr\_0/
- Demodulator/SN\_4.0\_Fr\_1200\_Ph\_135\_Tm\_0.00/Rate\_64K\_FEC\_1\_DiffEnc\_1\_Scr\_0/
- Demodulator/SN\_4.0\_Fr\_-1200\_Ph\_-135\_Tm\_0.00/Rate\_128K\_FEC\_1\_DiffEnc\_1\_Scr\_0/
- Demodulator/SN\_4.0\_Fr\_1200\_Ph\_0\_Tm\_0.00/Rate\_64K\_FEC\_0\_DiffEnc\_1\_Scr\_0/
- Demodulator/SN\_4.0\_Fr\_-1200\_Ph\_90\_Tm\_0.00/Rate\_64K\_FEC\_1\_DiffEnc\_1\_Scr\_0/