COM-1804SOFT CCSDS TM Modem
VHDL source code overview / IP core

Overview

The COM-1804SOFT is a Telemetry modem fully compliant with the CCSDS standard for implementation in FPGA, SoC or ASIC. It is written in generic portable VHDL.

The entire **VHDL source code** is deliverable. It is portable to a variety of FPGA targets.

This IP core is available as transmitter-only, receiver-only or bundled tx/rx.

Key Features

- Continuous-mode modem for BPSK/QPSK modulation. Programmable symbol rate, up to 0.4*ADC sampling frequency
- Convolutional FEC encoding, Viterbi decoding
- Interleaving/Deinterleaving
- Reed Solomon FEC encoding / decoding
- ASI interface (TM output)
- Demodulator performance:
  - BER: < 0.5 dB implementation losses w.r.t. theory
  - Programmable frequency acquisition range.
  - Demodulator acquisition threshold (uncoded) Eb/No = 1dB
- Frequency acquisition range > +/- 12% of symbol rate. Tracking symbol rates over +/- 50ppm around nominal setting.
- Built-in tools: PRBS-11 pseudo-random test sequence, BER tester, AWGN generator, internal loopback mode, carrier frequency error measurement.

CCSDS TM standard compliance

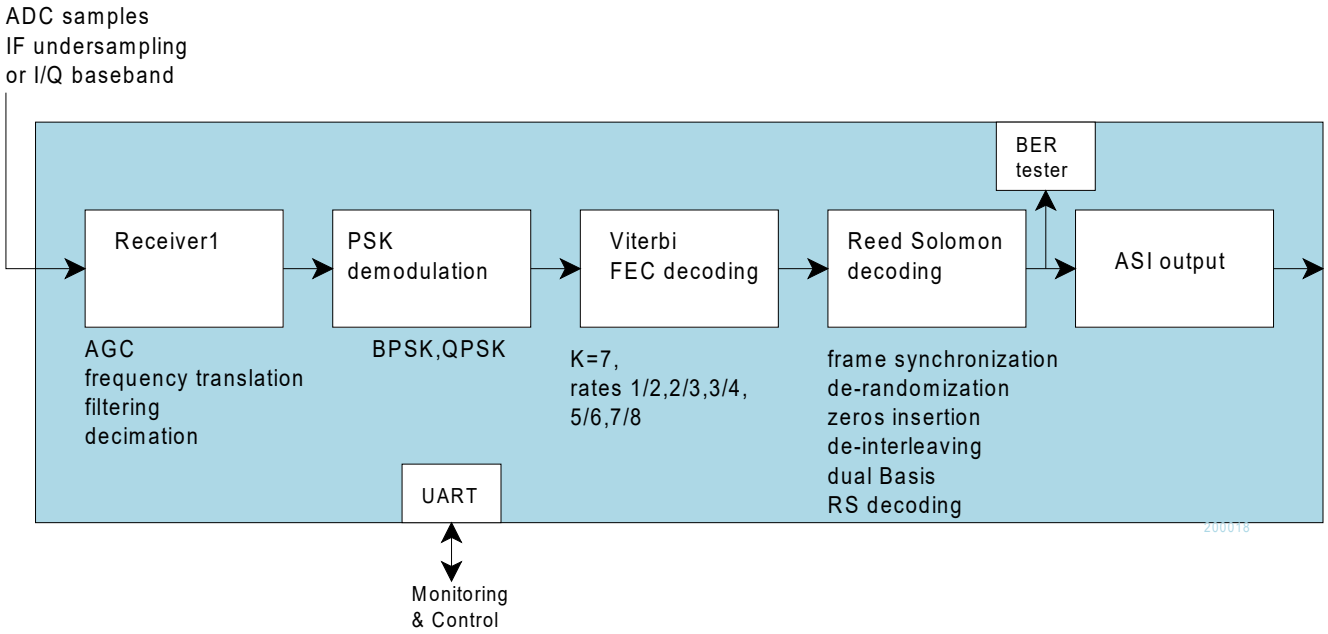
Subsystem	CCSDS reference
Modulator Demodulator	[2] Section 5.2.1.1.a [3] Section 2.4.10
Convolutional encoder Viterbi decoder	[1] Section 3
Attached Sync Marker	[1] Section 9.1.2, 9.2, 9.3.1, 9.3.5
Randomizer De-Randomizer	[1] Section 10
Transfer Frame Length	[1] Sections 11, 11.6
Interleaver De-Interleaver	[1] Sections 4, 4.3.5, 11.6
Reed-Solomon encoder / decoder	[1] 2.2.3, 3.2.3, 4.2.2
ASI output	[4] Section B.3

Portable VHDL code

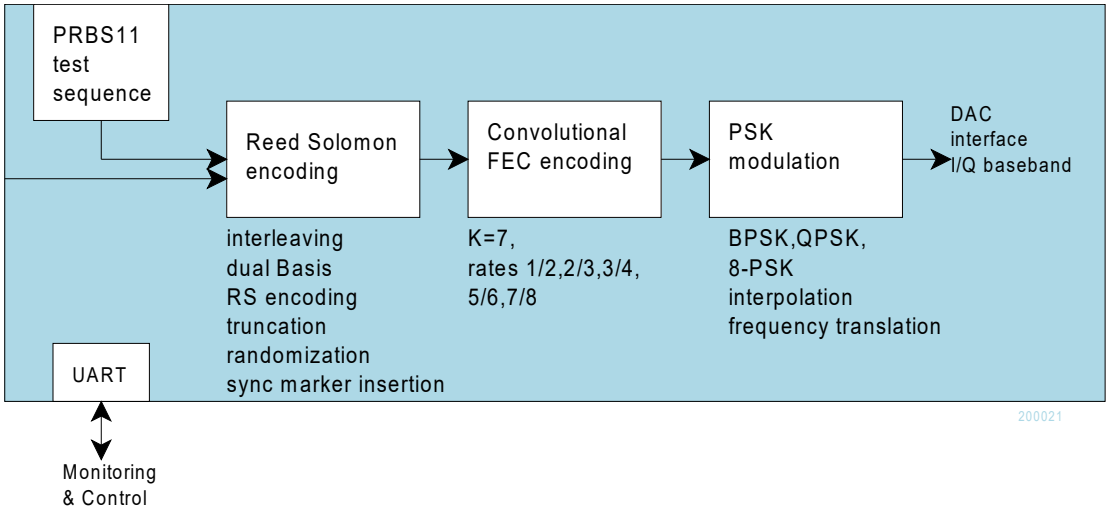
The code is written in generic standard VHDL and is thus portable to a variety of FPGAs. The code was developed on a Xilinx 7-series FPGA but is expected to work similarly on other targets. No manufacturer-specific primitive is used.

MSS • 845 Quince Orchard Boulevard Ste N • Gaithersburg, Maryland 20878-1676 • U.S.A.
Telephone: (240) 631-1111 www.ComBlock.com
 Issued 6/20/2023

Block Diagram



TM receiver block diagram



TM transmitter block diagram

Configuration

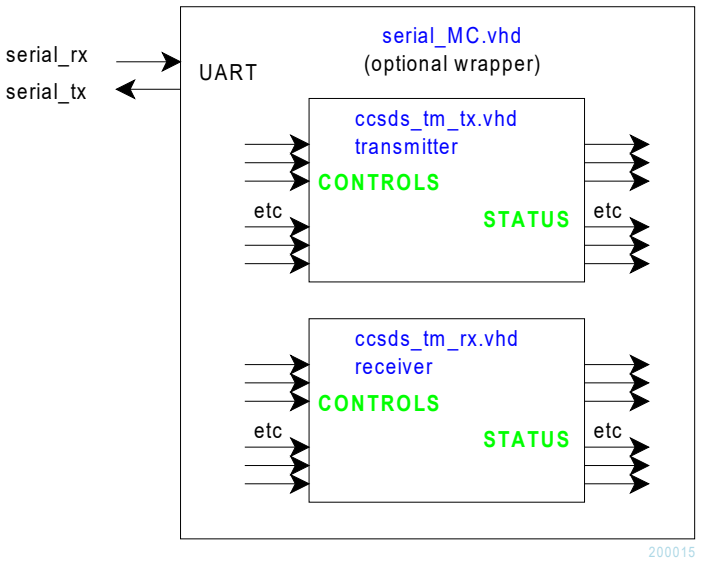
Pre-Synthesis configuration parameters

The following constants are user-defined in the component generic section prior to synthesis. These parameters generally affect the size of the embodiment.

Synthesis-time configuration parameters	
TM transmitter	
RS_ENC_INST	'1' to instantiate the Reed-Solomon encoder circuit and associated functions (frame synchronization, derandomization, deinterleaving)
CONV_ENC_INST	'1' to instantiate the convolutional FEC encoder circuit.
AWGN_EN	'1' to instantiate an Additive White Gaussian Noise generator. '0' during operational conditions to save space in FPGA (and to increase clock speed)
TM receiver	
DYNAMIC_PROFILE_INST	'1' to instantiate the dynamic (Doppler) profile
CONV_DEC_INST	'1' to instantiate the convolutional (Viterbi) FEC decoder circuit.
RS_DEC_INST	'1' to instantiate the Reed-Solomon decoder circuit and associated functions (frame synchronization, derandomization, deinterleaving)
BER_INST	'1' to instantiate a Bit Error Rate Tester.

Runtime dynamic configuration

The transmitter and receiver can be configured dynamically at runtime, in parallel (using the VHDL components input parameters), or serially via a UART.



The top-level components for parallel (I/O) configuration are:
ccsds_tm_tx.vhd
ccsds_tm_rx.vhd

The top-level components for serial (UART) configuration are:
ccsds_tm_tx_serialmc.vhd
ccsds_tm_rx_serialmc.vhd

Serial M&C

M&C is performed by exchanging 8-bit control and status registers through UARTs. The built-in UART is configured for 115.2 Kbaud, 8-N-1. Electrical levels are defined by the FPGA pins (generally 3.3V or less, but NOT RS-232C levels).

The three transactions are:

1. @WXXYY :
   (write control register XX with value YY),
2. @RXX
   (read back control register XX),
3. @GXX
   (get status register XX)

where @,W,R,G,X,Y are ASCII characters (upper cases only)

Register addresses XX and values YY are expressed in hexadecimal form

The number of control and status registers are defined in the package SERIAL_MC_PKG.

Control registers (TM receiver)

CREG designates the 8-bit control register when using the serial interface for configuration. **SIGNAL** designates the I/O signal when configuring the VHDL component directly at its interface.

f_{clk_adc} is the ADC sampling frequency
f_{clk_p} is the FPGA processing clock frequency

Demodulator (controls must be synchronous with CLK_RXg)	
Parameters	**Configuration**
Nominal input center frequency (f_c)	The nominal center frequency is a fixed frequency offset applied to the input samples. It is used for fine frequency corrections, for example to correct clock drifts. 32-bit signed integer (2's complement representation) expressed as $f_c * 2^{32} / f_{clk\_adc}$ CREG(128) (LSB) – CREG(131) (MSB) **RECEIVER_CENTER_FREQ(31:0)**
Nominal symbol rate f_{symbol_rate}	Nominal symbol rate, defined as $f_{symbol\_rate} * 2^{32} / f_{clk\_adc}$ CREG(132) (LSB) – CREG(135) (MSB) **DEMOD_SYMBOL_RATE(31:0)**
External AGC response time	Users can to optimize AGC response time while avoiding instabilities (depends on external factors such as gain signal filtering at the RF front-end and modulation symbol rate). The AGC_DAC gain control signal is updated as follows 0 = every symbol, 1 = every 2 input symbols, 2 = every 4 input symbols, 3 = every 8 input symbols, etc…. 10 = every 1000 input symbols. Valid range 0 to 14. CREG(136)(4:0) **RECEIVER_AGC_RESPONSE(4:0)**
CIC_R	Receiver decimation factor from f_{clk_adc} to 4* f_{symbol_rate}. Valid range 1 - 16384 CREG(137) (LSB) – CREG(138) (MSB) **CIC_R(15:0)**
Modulation type	0 = BPSK 1 = QPSK CREG(139)(5:0) **DEMOD_CONTROL(5:0)**
Spectrum inversion	Invert Q bit

Demodulator (controls must be synchronous with CLK_RXg)	
	0 = off 1 = on CREG(139(6) **DEMOD_CONTROL(6)**

Convolutional FEC decoder (controls must be synchronous with CLK_P)	
Parameters	**Configuration**
Convolutional (Viterbi) FEC decoding	'1' enabled, '0' bypassed CREG(141)(0) **FEC_DEC_EN**
Viterbi decoding constraint length K and rate R	1011 = (K = 7, R=1/2, CCSDS) 1100 = (K = 7, R=2/3, CCSDS/DVB) 1101 = (K = 7, R=3/4, CCSDS/DVB) 1110 = (K = 7, R=5/6, CCSDS/DVB) 1111 = (K = 7, R=7/8, CCSDS/DVB) CREG(141)(4:1) **FEC_DEC_R1/ FEC_DEC_R2**
Viterbi decoding: Differential Decoding	0 = disabled 1 = enabled CREG(141)(5) **FEC_DEC_DIFF_ENC_ON**

Reed Solomon FEC decoder (controls must be synchronous with CLK_P)	
Parameters	**Configuration**
RS decoding	'1' enabled, '0' bypassed (encompasses frame synchronization, de-randomization, deinterleaving, dual-basis, RS decoding proper) CREG(142)(7) **RS_DEC_EN**
RS code	Standard selection 5 = CCSDS (255,223,16) 6 = CCSDS (255,239,8) CREG(142)(3:0) **RS_CODE(3:0)**

RS decoder controls	bit 0: detect and remove sync word 0x1ACFFC1D (1) or ignore (0) bit 2: de-randomization on (1) /off(0) Other bits set to zero. CREG(143) **RS_DEC_REG1(7:0)**
RS interleaving	number of interleaved blocks CCSDS valid values 1 (no interleaving),2,3,4,5,8 CREG(144)(3:0) **RS_I(3:0)**
Shortened frame	Uncoded blocks can be shortened by RS_SHORT Bytes. RS_SHORT Bytes (zeroes) are inserted before the payload data prior to encoding. They are not sent over the transmission channel. In effect, the shortened payload size in a frame is RS_I*(RS_K - RS_SHORT). This setting must be consistent between transmitter and receiver. CREG(145) **RS_SHORT(7:0)**

Default values for the configuration registers at power-up or reset can be defined in serial_MC.vhd component, process CREG_WRITE_001a

Multi-Byte control words are generally enacted upon writing to the last control register CREG(N_CREG_MAX) over the serial link.

Control registers (TM transmitter)

Reed Solomon FEC encoder (controls must be synchronous with CLK_P)	
Parameters	**Configuration**
Transmitter input selection	"00" for serial input "01" internal PRBS11 test sequence "10" zeros "11" no input CREG(0)(5:4) **TX_INPUT_SEL**
RS encoding	'1' enabled, '0' bypassed (encompasses frame marker insertion, randomization, interleaving, dual-basis, RS encoding proper) CREG(0)(7) **RS_ENC_EN**
RS code	Standard selection 5 = CCSDS (255,223,16) 6 = CCSDS (255,239,8) CREG(0)(3:0) **RS_CODE (3:0)**
RS encoder controls	bit 0: insert periodic sync marker 0x1ACFFC1D (1) or ignore (0) bit 2: randomization on (1) /off(0) Other bits set to zero. CREG(1) **RS_REG1(7:0)**
RS interleaving	number of interleaved blocks CCSDS valid values 1 (no interleaving),2,3,4,5,8 CREG(2)(3:0) **RS_I(3:0)**
Shortened frame	Uncoded blocks can be shortened by RS_SHORT Bytes. RS_SHORT Bytes (zeroes) are inserted before the payload data prior to encoding. They are not sent over the transmission channel. In effect, the shortened payload size in a frame is RS_I*(RS_K - RS_SHORT). This setting must be consistent between transmitter and receiver. CREG(3) **RS_SHORT(7:0)**

Convolutional FEC encoder (controls must be synchronous with CLK_P)	
Parameters	**Configuration**
Convolutional FEC encoding	'1' enabled, '0' bypassed CREG(4)(0) **CONV_ENC_EN**
Convolutional FEC encoding constraint length K and rate R	1011 = (K = 7, R=1/2, CCSDS) 1100 = (K = 7, R=2/3, CCSDS/DVB) 1101 = (K = 7, R=3/4, CCSDS/DVB) 1110 = (K = 7, R=5/6, CCSDS/DVB) 1111 = (K = 7, R=7/8, CCSDS/DVB) CREG(4)(4:1) **CONV_ENC_KR**
Differential convolutional FEC encoding:	0 = disabled 1 = enabled CREG(4)(5) **CONV_ENC_DIFF_ENC_ON**

Modulator (controls must be synchronous with CLK_TXg)	
Parameters	**Configuration**
Symbol rate f_{symbol_rate}	The modulator symbol rate is in the form $f_{symbol\ rate\ tx} = f_{clk\_tx} / 2^n$ where n ranges from 1 (2 samples per symbol) to 15 (symbol rate = f_{clk_tx} / 32768). n is defined in CREG(5)(3:0) **MOD_SYMBOL_RATE_NDIV (31:0)**
Output Center frequency (f_{c_tx})	Fine tuning of center frequency. Typically 0 Hz. 32-bit signed integer (2's complement representation) expressed as $f_{c\_tx} * 2^{32} / f_{clk\_tx}$ For a clean output waveform, we recommend keeping the maximum frequency (center frequency + ½ symbol rate) below 1/10th of the processing clock f_{clk_tx}. Note: as the AWGN noise samples are not frequency translated, noise tests should only be performed while the center frequency translation is smaller than the modulation bandwidth. CREG(6) (LSB) – CREG(9)(MSB)

Modulator (controls must be synchronous with CLK_TXg)	
Parameters	**Configuration**
Symbol rate f_{symbol_rate}	The modulator symbol rate is in the form $f_{symbol\ rate\ tx} = f_{clk\_tx} / 2^n$ where n ranges from 1 (2 samples per symbol) to 15 (symbol rate = f_{clk_tx} / 32768). n is defined in CREG(5)(3:0) **MOD_SYMBOL_RATE_NDIV (31:0)**
	MOD_CENTER_FREQ(31:0)

Digital Signal gain	16-bit amplitude scaling factor for the modulated signal. The maximum level should be adjusted to prevent saturation. The settings may vary slightly with the selected symbol rate. Therefore, we recommend checking for saturation at the D/A converter when changing either the symbol rate or the signal gain. Enacted upon writing the MSB. CREG(10) (LSB) – CREG(11)(MSB) **MOD_SIGNAL_SCALING (15:0)**
Additive White Gaussian Noise gain	16-bit amplitude scaling factor for additive white Gaussian noise. Because of the potential for saturation, please check for saturation at the D/A converter when changing this parameter. CREG(12) (LSB) – CREG(13)(MSB) **AWGN_SCALING (15:0)**
Modulation type	0 = BPSK 1 = QPSK CREG(14)(5:0) **MOD_CONTROL(5:0)**
Spectrum inversion	Invert Q bit. (Inverts the modulated spectrum only, not the subsequent frequency translation) 0 = off 1 = on CREG(14)(6) **MOD_CONTROL(6)**

Default values for the configuration registers at power-up or reset can be defined in serial_MC.vhd component, process CREG_WRITE_001a

Status registers (TM receiver)

Summary	
Signal presence detection (from FFT)	0 = not present 1 = present SREG(128)(0) **DEMOD_SIGNAL_PRESENT**
Carrier lock status	0 = unlocked 1 = locked (poor reliability) SREG(128)(1) **CARRIER_LOCK**
Viterbi decoder synchronization status	0 = not synchronized 1 = synchronized SREG(128)(2) **FEC_DEC_CONFIRMED_LOCK**
SOF locked	Detected periodic synchronization sequences. This is the single most reliable status regarding the receive signal reception. 0 = not synchronized 1 = synchronized SREG(128)(3) **SOF_LOCK**

Demodulator (status synchronous with CLK_RXg)	
RF/IF/analog front-end AGC gain level	gain control for the external analog/IF/RF front-end. May need to be inverted depending on the analog front-end. Format: 12-bit unsigned. FFF represents the minimum gain, 000 the maximum gain SREG(143)(LSB) - SREG(144)(MSB) **AGC_DAC(11:0)**
Inverse SNR	A measure of noise over signal power. 0 represents a noiseless signal. Valid only when demodulator is locked. SREG(129)(7:0) **SNR(7:0)**
Carrier frequency offset1	Residual frequency offset with respect to the nominal carrier frequency. Part 1/2. Includes receiver frequency scanning and carrier tracking loop. 32-bit signed integer expressed as fcerror * 2^{32} / f_{clk_adc}

Demodulator (status synchronous with CLK_RXg)	
	SREG(130) (LSB) – SREG(133) (MSB) **CARRIER_FREQUENCY_ERROR1**
Carrier frequency offset2	Residual frequency offset with respect to the nominal carrier frequency. Part 2/2. Includes FFT-based frequency measurement (fixed after acquisition) 32-bit signed integer expressed as fcerror * 2^{31} / f_{symbol_rate} SREG(134) (LSB) – SREG(137) (MSB) **CARRIER_FREQUENCY_ERROR2**
Received signal amplitude	A measure of the received signal (+ noise) AMPLITUDE in a frequency band of 4*symbol rate. This information can be helpful in assessing the strength of a very low-level received signal, when all front-end AGCs are mostly tracking wideband noise. Format: 14-bit unsigned SREG(138) (LSB) – SREG(139) (MSB) **RX_SIGNAL_AMPLITUDE(13:0)**

FEC decoders (status synchronous with CLK_P)	
Viterbi decoder input BER estimate	The Viterbi decoder computes the BER on the received (encoded) data stream irrespective of the transmitted bit stream. Encoded stream bit errors detected over a 1000-bit measurement window. SREG(140) (LSB) – SREG(142) (MSB) **FEC_DEC_BER_COUNT (23:0)**

Multi-Byte status words are generally latched to status registers SREG() upon readingthe first status register SREG(128) for the TM receiver or SREG(0) for the TM transmitter over the serial link.

Status registers (TM transmitter)

TBD

I/Os

Receiver inputs

ADC_DATA_I/Q_IN(15:0): input samples from one or two external ADCs. (one in the case of IF undersampling, two for near-zero frequency complex inputs). If the ADCs have fewer than 16-bit precision, align the most significant bit with ADC_DATA_IN(15). Format: 2's complement (signed).

ADC_SAMPLE_CLK_IN: '1' when **ADC_DATA_I/Q_IN** is valid. Generally fixed at '1' when input is connected directly to ADCs.

AGC_DAC(11:0): output to an external DAC to control an external AGC. Gain control for the external analog/IF/RF front-end. May need to be inverted depending on the analog front-end. 12-bit unsigned. FFF represents the minimum gain, 000 the maximum gain.
Read when **AGC_DAC_SAMPLE_CLK** is '1'

The above signals are clock-synchronous with ADC sampling clock CLK_RXg.

Receiver output (1/2)

DATA_OUT(7:0): output data is sent out one Byte at a time.

DATA_OUT_VALID: 1 clock-wide pulse indicating that **DATA_OUT** is valid.

SOF_OUT: output Start Of Frame. 1 clock-wide pulse. The SOF is aligned with **DATA_OUT_VALID**. Note that there is no need for an end of frame as the frame size is determined by the **RS_CODE** selection.

The output signals are synchronous with the CLK_P processing clock.

Receiver output (2/2)

ASI_OUT: Asynchronous Serial Interface. 8b/10b encoded serial stream as per [4] standard. The baseline code includes a LVDS output driver which can be easily substituted by other I/O standards.

BAUD_CLK_OUT: optional synchronous clock. Generally unused as the ASI receiving end recovers the baud clock from the serial data.

The baud rate can be either fixed prior to synthesis (most common case) or dynamically programmed at run-time, depending on the baud rate clock generator commented/uncommented in the code. Use *CLKGEN7_MMCM_ADJ.vhd* for fixed baud rate or *CLKGEN7_MMCM_DYNAMIC.vhd* for programmable baud rate.

Transmitter inputs

DATA_IN(7:0): input data is read one Byte at a time.

DATA_IN_VALID: 1 clock-wide pulse indicating that **DATA_IN** is valid.

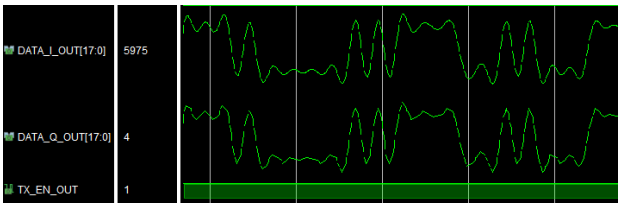
SOF_IN: optional Start Of Frame. 1 clock-wide pulse. The SOF is aligned with **DATA_IN_VALID**. The spacing between Start-Of-Frame pulses is expected to match the uncoded RS frame length (see **RS_CODE**)

CTS_OUT: "Clear-To-Send" output flow-control signal. The data source should stop sending new Bytes when CTS_OUT = '0';

Transmitter outputs

DATA_I/Q_OUT(17:0): Modulated baseband output samples (I = in-phase, Q = quadrature). One output sample every clock. Format: 2's complement (signed)

TX_EN_OUT: goes low to turn off an external power amplifier when the modulator is active. It includes a timing margin at the start/end of burst.



Operation

Clocks

The transmitter uses two clocks:
CLK_TXg, the DAC interface sampling clock, and CLK_P, the processing clock and user-interface. Both are global clocks.

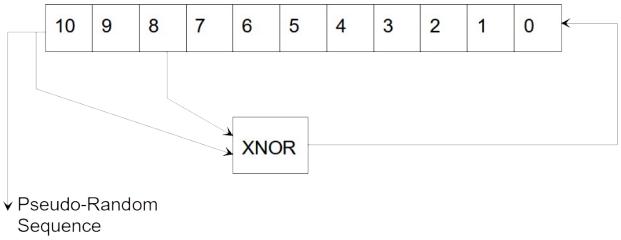
CLK_TXg frequency is a function of the modulation symbol rate. $f_{CLK\_TXg} = f_{symbol\_rate} * 2^{\wedge n}$, where n is an integer.

The receiver also uses two clocks:
CLK_RXg, the ADC interface sampling clock, and CLK_P, the processing clock and user-interface. Both are global clocks.

The CLK_RXg frequency can be set at a fixed value which must be at least twice the maximum symbol rate.

Pseudo-Random Bit Stream (Test Pattern)

A periodic pseudo-random sequence can be used as modulator source instead of the input data stream. A typical use would be for end-to-end bit-error-rate measurement of a communication link. The sequence is 2047-bit long maximum length sequence generated by a 11-tap linear feedback shift register:



The first 100 bits of the PN sequence are as follows:
0000000000 0111111111 0011111110 0001111100 1100111000 0000010011 1111010001 1110110100 1101001100 0011000001

Format Conversion

Serial to parallel conversion occurs at the interface between the modem and the application. The general rule is that the first received bit is placed at the MSb position in the byte.

Bit Timing Tracking

A first order loop is capable of acquiring and tracking bit timing differences between the transmitter and the receiver of at least ± 50 ppm.

Receiver AGC

To maintain linearity throughout the receive path, several AGC loops control the signal level. While most AGC loops are internal, an additional AGC loop is dedicated to controlling an external RF/IF/analog front-end.

The purpose of this AGC is to prevent saturation at the external A/D converter(s) while making full use of the A/D converter(s) dynamic range. The controlling signal **AGC_DAC(11:0)** can be read from the receiver status or can be connected directly to an external auxiliary DAC.

The AGC responsiveness can be adjusted using the **RECEIVER_AGC_RESPONSE(4:0)** control.

Frequency profile table

In applications with large Doppler based on a known orbital dynamics, users can pre-position the receiver center frequency dynamically as a function of time. The compensation is table-based.

First, instantiate the dynamic profile circuit by setting the generic parameter
DYNAMIC_PROFILE_INST = '1';

Users can declare the expected receiver frequency variation with time by uploading a frequency profile table. The frequency table consists of a 32-bit start time followed by up to 4096 32-bit frequency samples.
Each sample represents a nominal center frequency expressed in units of $\mathbf{f}_{\mathbf{clk\_adc}} / 2^{32}$ (about 37 mHz steps @160MHz ADC sampling clock), sampled at 1s intervals.

The table is uploaded one Byte at a time, MSB first.

The frequency table is read (played-back) every second starting at the specified start time. The receiver interpolates linearly 64x between successive 1s samples so as to minimize discontinuities. This ensures phase and frequency continuity. This frequency bias is removed from the receiver input samples for the playback duration, irrespective of the demodulator lock status.

Table playback is mutually exclusive with table upload. Uploading a new table will immediately stop any playback in progress.

Because the table is quite small (131Kbits max), the upload time (a few ms) is insignificant relative to the playback duration (68 minutes max).

The user must provide a 32-bit system time **SYSTEM_TIME** (expressed in seconds) to the TM receiver.

To upload the frequency table:
1. Set **FT_UPLOAD** <= '1'; for the duration of the table upload.

2. Wait at least 1 **CLK_RXg** period

3. Enter the start time and the follow-on 4096 frequency samples, one Byte at a time, MSB first
FT_DATA_IN <= data Byte
FT_DATA_IN_VALID <= '1' pulse for each input Byte.

The start time is a 32-bit number expressed in seconds.

The frequency is a signed 32-bit number (2's complement), expressed in units of $\mathbf{f}_{\mathbf{clk\_adc}} / 2^{32}$

4. Clear **FT_UPLOAD** <= '0'; when the table upload is complete.

The frequency table playback will automatically start when **SYSTEM_TIME** equals the start time. When the playback reaches the last (4096th) entry, the Doppler correction is unchanged, unless a new frequency table is uploaded.

Software Licensing

The COM-1804SOFT is supplied under the following key licensing terms:

1. A nonexclusive, nontransferable license to use the VHDL source code internally, and
2. An unlimited, royalty-free, nonexclusive transferable license to make and use products incorporating the licensed materials, solely in bit stream format, on a worldwide basis.

The complete VHDL/IP Software License Agreement can be downloaded from
http://www.comblock.com/download/softwarelicense.pdf

Configuration Management

The current software revision is 062023

Directory	Contents
/doc	Specifications, user manual, implementation documents
/src	.vhd source code,.pkg packages, .xdc constraint files (Xilinx) One component per file.
/sim	VHDL test benches
/matlab	Matlab .m file for simulating the encoding and decoding algorithms, for generating stimulus files for VHDL simulation and for end-to-end BER performance analysis at various signal to noise ratios

Project files:

Xilinx Vivado v2020 project file:
project_1v2020.xpr
tcl file

VHDL development environment

The VHDL software was developed using the following development environment:

(a) Xilinx Vivado 2020 for synthesis, place and route and VHDL simulation

Device Utilization Summary

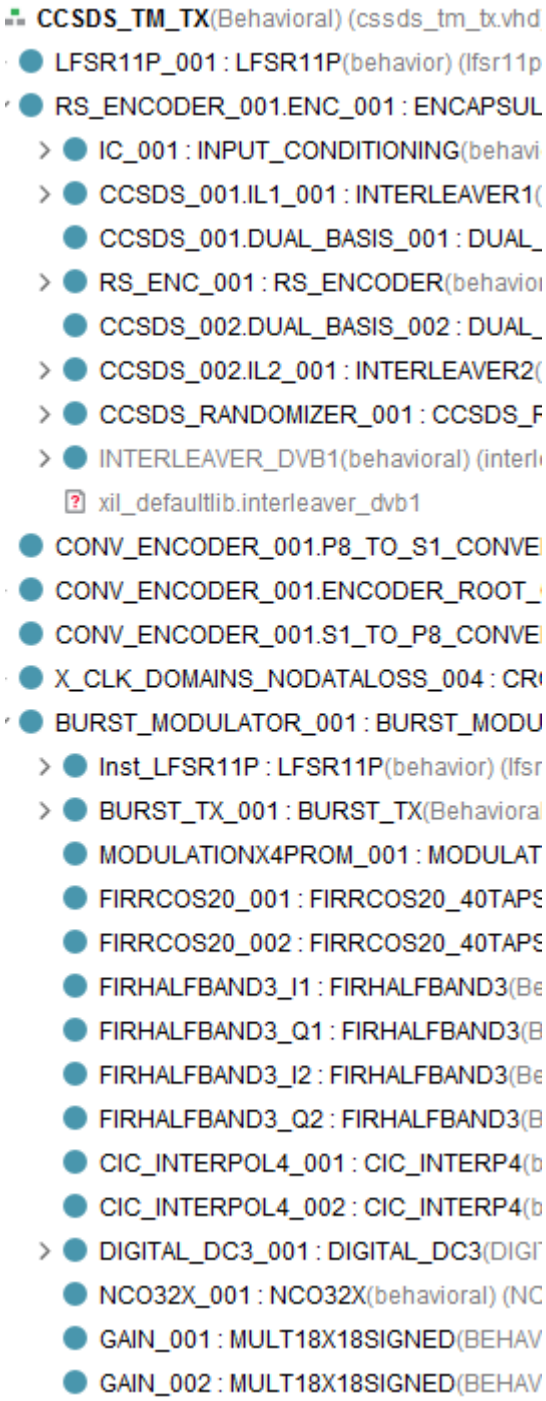
RS encoder and convolutional encoder enabled.
AWGN disabled.

TM transmitter		Artix7-100T utilization
LUTs	6282	9.9%
Registers	7227	5.7%
Block RAM/FIFO 36Kb	7.5	5.6%
DSP	6	2.5%
GCLKs	2	6.3%

TM receiver		Artix7-100T utilization
LUTs	12881	20.3%
Registers	14041	11.1%
Block RAM/FIFO 36Kb	18	13.3%
DSP	37	15.4%
GCLKs	2	6.3%

VHDL components overview

TM transmitter top level



ENCAPSULATED_RS_ENCODER.vhd includes all encoding functions on the transmit side: RS encoding, shortened frame, interleaving/de-interleaving, dual-basis conversion T and T^{-1}, sync marker insertion and randomization. This component buffers the input Byte stream and computes the parity bits for each input frame. If a Start Of Frame marker is not supplied, the component will segment the input Byte stream into frames. Both inputs and outputs are 8-bit parallel.

The *INPUT_CONDITIONING.vhd* component performs the following tasks:
1. Short (16 Byte) input elastic buffer
2. Segment input Byte stream into frames. Insert SOF/EOF markers
3. Insert all zeros Bytes prior to the payload when the frame is shortened ([1] Section 4.3.7).
4. Report inconsistency between internal segmentation and external SOF_IN marker

BRAM_DP2.vhd is a generic dual-port memory, used as input and output elastic buffers. Memory is inferred for code portability (no primitive is used).

The *INTERLEAVER1.vhd* component consists of a I-rows by K-columns interleaver prior to RS encoding, as per CCSDS specifications [1] Sections 4.3.5, 4.4.2. *INTERLEAVER2.vhd* performs the inverse de-interleaving.

DUAL_BASIS.vhd transforms Bytes between Berlekamp and Conventional representations. See [1] Annex F.

The heart of the encoder is *RS_ENCODER.vhd*. The encoder processes one block at a time. It supports multiple standards, encompassing two distinct Galois fields with primitive polynomials $p(x) = x^8+x^7+x^2+x+1$ (391)

RS_ENC_SPECIFICATIONS.vhd describes the supported standards in terms of primitive polynomial to generate the Galois field, uncoded block size K, encoded block size N and code generator polynomials G(x).

GF_MULT_391.vhd multiplies two numbers over GF(2^8) that are in polynomial tuple representation. The component assumes that the polynomial representation is based on the primitive polynomial $x^8+x^7+x^2+x+1$.

SHIFT_REGISTER.vhd implements the shift registers and multipliers part of the polynomial division and remainder computation.

CCSDS_RANDOMIZER.vhd is a pseudo-randomizer to increase the bit transition density prior to modulation. It also inserts a periodic sync marker. The implementation follows [1] Section 10.

BURST_MODULATOR.vhd implements the digital modulation and spectrum shaping. Key controls include modulation symbol rate and output signal amplitude. In this application, only BPSK and QPSK modulation are supported, although the framework allows for fairly simple upgrades to higher-order modulations.

The *BURST_TX.vhd* component stores input data in an elastic input buffer, then packs input bits into symbols (1 or 2 bits/symbol) at the specified symbol rate.

FIRCOS20_40TAPS.vhd implements a root raised cosine filter with 20% rolloff to shape the output spectrum. It is implemented as an FIR filter with 40 coefficients. I/O sampling rate is 2 samples/symbol (meaning that the maximum modulation rate could be up to f_{CLK_TXg} /2)

FIRHALFBAND3.vhd implements a half-band FIR filter used to interpolate by 2. In includes 20-taps, nearly half of them are zero. 60 dB rejection. For lower symbol rates, two half-band filters are concatenated for a x4 interpolation to 8 samples/symbol.

The last interpolation of the modulated waveform is performed by *CIC_INTERPOL4.vhd*. up to the DAC sampling rate f_{CLK_TXg}.The interpolation factor CIC_R must be a power of 2.

Small adjustments in the transmitted signal center frequency are implemented by *DIGITAL_DC3.vhd* as a vector rotation. The sine and cosine values are read from ROM in *SIGNED_SIN_COS_TBL3.vhd*. The translation frequency/phase rotation is generated by *NCO32X.vhd*.

Prior to the DAC, the digital waveform amplitude is adjusted by digital multipliers *MULT18X18SIGNED.vhd*.

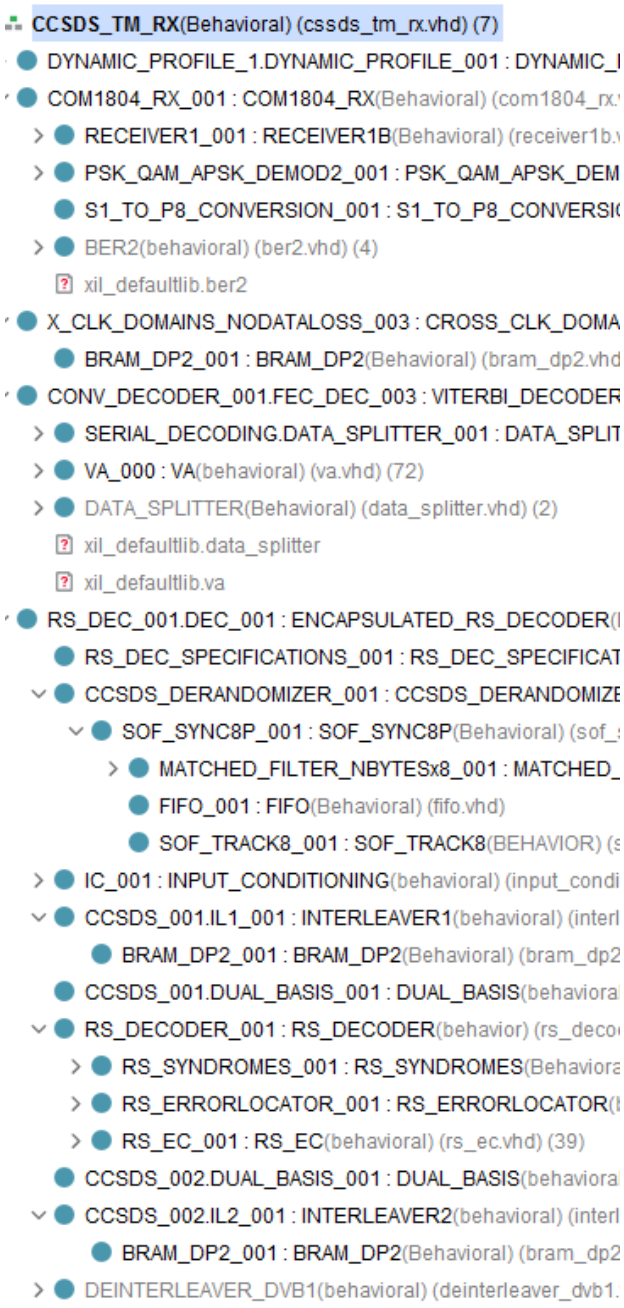
Ancillary components

LFSR11P.vhd is a pseudo-random sequence generator used for test purposes. It generates a PRBS11 test sequence commonly used for bit error rate testing at the receiving end of a transmission channel.

AWGN.vhd generates a precise Additive White Gaussian Noise. The noise bandwidth is 2*symbol rate.

SIM2OUTFILE.vhd writes three 12-bit data variables to a tab delimited file which can be subsequently read by Matlab (load command) for plotting or analysis.

TM receiver top level



RECEIVER1B.vhd is the front-end digital receiver which processes digital samples from the A/D converter(s). It performs non modulation-specific tasks, including fixed frequency translation to (near-zero) baseband, AGC, variable decimation (CIC) filters and one half-band filter for image rejection. Input digital samples can be complex (in the case of baseband input samples) or real (in the case of IF undersampling). This generic component is not modulation-specific.

BURST_PSK_QAM_APSK_DEMOD2.vhd performs the demodulation, including carrier tracking, symbol timing tracking and AGC. It is currently limited to BPSK/QPSK demodulation.

VITERBI_DECODER.vhd is the Viterbi decoder top component. It's role is to segment the continuous input stream of encoded (hard or soft-quantized) bits into fixed-length blocks which can be processed by **NDEC** parallel decoders. The decoded bit stream is then reassembled into a seamless output stream. This component also adjusts the alignment of the input encoded bit stream with the coding and puncturing periodic pattern. Indeed, the follow-on Viterbi algorithm (*VA.vhd*) can only work properly if a parity bit encoded by generator polynomial Gx in the transmitter is decoded with the same generator polynomial at the receiver.

DATA_SPLITTER.vhd implements the continous input stream segmentation.

BER3.vhd synchronizes with the received bit stream and counts the number of bit error when a PRBS-11 sequence is being transmitted.

ENCAPSULATED_RS_DECODER.vhd includes all decoding functions on the receive side: RS decoding, shortened frame, interleaving/de-interleaving, dual-basis conversion T and T^{-1}, sync marker detection, frame synchronization and derandomization. A Start-Of-Frame is required if the input Byte stream does not include a periodic sync marker. Both inputs and outputs are 8-bit parallel.

The *CCSDS_DERANDOMIZER.vhd* component detects and removes the periodic sync markers, reconstructs the start of frame and end of frame pulses and descrambles the received soft-quantized bit stream. It complies with sections 9 and 10 of the specifications [1].

The *SOF_SYNC8P.vhd* component detects, confirms and removes the periodic sync markers. It includes a fly-wheel mechanism to reconstruct the frame structure in the event of high bit errors. It also reports and corrects the input symbols bit to Byte packing alignment. Finally, it monitors the bit error rate within the received sync markers. I/Os are 8-symbols in parallel.

MATCHED_FILTER_NBYTESx8.vhd: a 64-bit matched filter operating on 8-parallel 1-bit hard-quantized input symbols. The matched filter detects a match 'on-the-fly' on all 8 possible bits/Byte alignments. It also report inverted sequences. Default detection threshold is 10 mismatches out of 64 (15.6% BER). The threshold can be adjusted through the DETECT_THRESHOLD generic parameter.

SOF_TRACK8.vhd: Confirmation circuit for the frame synchronization. It generates a reliable SOF_LOCK_DETECT status based on the detection of the periodic sync marker at the expected time.

The *INTERLEAVER1.vhd* component consists of a I-rows by N-columns interleaver prior to RS decoding, as per CCSDS specifications [1] Sections 4.3.5, 4.4.2. *INTERLEAVER2.vhd* performs the inverse de-interleaving.

DUAL_BASIS.vhd transforms Bytes between Berlekamp and Conventional representations. See [1] Annex F.

Ancillary components

AD936xIF.vhd is a stub to interface with AD936x chip, receive path, LVDS. See UG-673 Figure 76 for details.

BER2.vhd is a bit error rate tester expecting to receive a PRBS11 test sequence. It synchronizes with the received bit stream and count errors over a user-defined window. It can be placed immediately after the demodulator, or after the error correction.

SERIAL_MC.vhd component: through the UART interface, users can write to control registers and read from control or status registers. The UART is configured for 115.2 Kbaud, 8-N-1

ASI_TX.vhd converts the decoded Bytes into an ASI serial stream (typically for transmission over coax. Asynchronous clock (baud rate) is available if needed, although generally not used.

Specifications

[1] CCSDS "TM Synchronization and Channel Coding", CCSDS 131.0-B-3, September 2017

[2] CCSDS "Flexible Advanced Modulation and Coding Scheme for High-Rate Telemetry Applications", CCSDS 131.2-B-1, March 2012

[3] CCSDS "Radio Frequency and Modulation Systems, Part 1", CCSDS 401.0-B-30, February 2020

[4] Asynchronous Serial Interface, EN 50083-9 standard 2002, Section B.3

Acronyms

Acronym	Definition
ADC	Analog to Digital Converter
ASI	Asynchronous Serial Interface
AWGN	Additive White Gaussian Noise
BRAM	Dual-port Block RAM
BER	Bit Error Rate
CCSDS	Consultative Committee for Space Data Systems
DAC	Digital to Analog Converter
DVB	Digital Video Broadcast
FPGA	Field Programmable Gate Arrays
LSb	Least Significant bit
LSB	Least Significant Byte
M&C	Monitoring and Control
MSb	Most Significant bit
MSB	Most Significant Byte
N/A	Not Applicable
PRBS-11	Pseudo-Random Binary Sequence, 2047-bit period
SoC	System on Chip
TC	Telecommand
TM	Telemetry
UART	Universal Asynchronous Receiver/Transmitter

ComBlock Ordering Information

COM-1804SOFT CCSDS Telemetry modem, ,
VHDL source code / IP core

- transmit-only
- receive-only
- tx/rx bundle

ECCN: EAR99

MSS • 845 Quince Orchard Boulevard Ste N•
Gaithersburg, Maryland 20878-1676 • U.S.A.
Telephone: (240) 631-1111
E-mail: sales@comblock.com