

# COM-1202/1203 PSK / QAM / APSK modem VHDL SOURCE CODE OVERVIEW

#### Overview

The COM-1202/1203 ComBlock Module comprises two pieces of software:

- VHDL code to run within the FPGA for all signal processing functions.
- C/Assembly code running within the ATMega8515L microprocessor for non application-specific monitoring and control functions.

The VHDL code interfaces to the monitoring and control functions by exchanging byte-wide registers on the Atmel microcontroller 8-bit data bus. The control and monitoring registers are defined in the specifications [1].

The Atmel microprocessor code is generic (i.e. non application specific), not user-programmable and functionally transparent to the user. It is thus not described here.

The COM-1202 VHDL code runs on the generic COM-1200 hardware platform. The schematics [2] for this platform are available in this CD.

# Reference documents

[1] specifications: com1202.pdf

[2] hardware schematics: com\_1200schematics.pdf

[3] VHDL source code in directory com-1202\_rev\src

[4] Xilinx ISE project files com-1202\_*rev*\com1202\_ISE82.npl com-1202\_*rev*\com1202\_ISE91.npl

[5] .ucf constraint file com-1202\_*rev*\src\COM1202.ucf

[6] .mcs FPGA bit files com-1202\_*rev*\bin\com1202A\_*rev*.mcs com-1202\_*rev*\bin\com1202B\_*rev*.mcs com-1202\_rev\bin\com1202D\_rev.mcs com-1202\_rev\bin\com1202E\_rev.mcs

where *rev* is the current revision number.

# **Configuration Management**

The current software revision is 16, release 1.

# **Configuration Options**

In order to provide configuration flexibility without unduly increasing the hardware complexity, some features require generating different firmware versions. In particular, the channel filter (root raised cosine square root) rolloff can take four distinct values: 20%, 25%, 35% and 40%.

Four versions of the *raised\_cos5x* root raised filters are included in the source code .src directory. To change the filter:

- (a) change the OPTION constant in the *com1202.vhd* top level file so that the resulting bit file can later be correctly identified.
- (b) Change the RAISED\_COS5x statements in three places within the *psk\_demod.vhd* file: one declaration and two instantiations.

# VHDL development environment

The VHDL software was developed using two development environments:

- (a) Xilinx ISE 8.2 with XST as synthesis tool
- (b) Xilinx ISE 9.2 with XST as synthesis tool.

#### Target Hardware

The modem code is written in generic VHDL so that it can be ported to a variety of FPGAs. The modem code was developed on a Xilinx Spartan-3 XC3S2000-4FG676 FPGA.

The modem throughput is related primarily to the processing FPGA technology. The VHDL code is

MSS • 18221 Flower Hill Way #A • Gaithersburg, Maryland 20879 • U.S.A. Telephone: (240) 631-1111 Facsimile: (240) 631-1676 www.ComBlock.com © MSS 2000-2008 Issued 8/7/2008 designed for a maximum modulation symbol rate of <sup>1</sup>/<sub>4</sub> of the FPGA processing clock CLK\_P. In other words, the processing is performed with 4 samples per symbol. In practice, CLK\_P is limited to about 90 MHz for a Xilinx Spartan-3 or 140 MHz for a Xilinx Virtex-4 (mostly because of the hardware multiplier latency). The CLK\_P frequency is user-selectable through a digital clock manager.

At the modem periphery, three interface components are hardware-specific: the USB, LAN, and Analog Front-End interface. These drivers are written for specific external integrated circuits:

- *usb20.vhd*: interfaces with an external USB 2.0 PHY over a standard UTMI interface. (for example, the SMSC USB3250 IC).
- *tcpip3.vhd*: SMSC LAN91C111 or compatible LAN MAC/PHY
- *afe.vhd*: Analog Devices AD9860/9862 dual A/D and D/A converters

The USB20, TCPIP3, AFE driver components are supplied as VHDL source code to facilitate porting to other hardware platforms.

#### Xilinx-specific code

The VHDL source code is written in generic VHDL with few Xilinx primitives. No Xilinx CORE is used. The Xilinx primitives are:

- IBUF
- IBUFG
- BUFG (global clocks)
- DCM (digital clock management, DLL)
- Various RAM block components (RAMB16\_S16\_S16, RAMB16\_S9, RAMB16\_S9\_S9, RAMB4\_S8\_S8, etc.)

# Top-Level VHDL hierarchy

xc3s2000-4fg676

😘 🏪 COM1202 - Behavioral (SRC/COM1202.vhd)
Inst_com1200_1 - com1200_1 - BEHAVIORAL (SRC/com1200_1.vhd)
强 Inst_com1200_2 - com1200_2 - BEHAVIORAL (src/com1200_2.vhd)
Inst_com1200_3 - com1200_3 - BEHAVIORAL (src/com1200_3.vhd)
👜 😘 AFE_INTERFACE_001 - AFE - behavioral (src/afe.vhd)
- 😘 ELASTIC_BUFFER_FLEX_001 - ELASTIC_BUFFER_FLEX - behavioral (SRC/PSK_QAM_/
👜 强 MODULATOR_001 - PSK_QAM_APSK_MODULATOR - behavioral (SRC/PSK_QAM_APS
👜 🜇 TRANSMITTER_001 - TRANSMITTER_1 - Behavioral (SRC/TRANSMITTER/transmitter_1
👜 😘 FRONT_END_AGC_002 - AGC10 - behavioral (SRC/PSK_QAM_APSK_DEMOD/AGC10.v
- 🙀 PWM_001 - PWM - Behavioral (SRC/PSK_QAM_APSK_DEMOD/pwm.vhd)
👜 🜇 DEMOD1 - PSK_QAM_APSK_DEMOD - behavioral (SRC/PSK_QAM_APSK_DEMOD/psk,
👜 😘 BER_ROOT_001 - BER_ROOT - behavioral (src/ber/ber_root.vhd)
👜 😘 USB20_001 - USB20 - behavioral (src/usb20/usb20.VHD)
👜 🜇 TCPIP3_001 - TCPIP3 - behavioral (SRC/TCPIP/TCPIP3.VHD)
👜 🜇 COMSCOPE_001 - COMSCOPE - behavioral (src/COMSCOPE.VHD)
- 🔁 COM1202 ucf (SBC/COM1202 ucf)

The code is stored with one, and only one, component per file.

The root entity (highlighted above) is *com1202.vhd*. It includes the following components:

- The modulator *psk\_qam\_apsk\_modulator.vhd* generates a baseband (zero center frequency) complex modulated signal at 4 samples/symbol.
- The following *transmitter1.vhd* interpolates the modulator samples and translates the output to a non-zero frequency.
- On the receive side, demod\_psk\_qam\_apsk.vhd performs all demodulator functions, including decimation, channel filtering, gain control, carrier tracking, symbol timing tracking and phase ambiguity resolution.
- The demodulated bit stream can be fed to the *ber\_root.vhd* component which counts bit errors when the PRBS-11 pseudorandom test sequence is transmitted.
- Another built-in test tool is *comscope.vhd* which captures internal signals in real-time to be displayed on a host PC using the ComBlock Control Center (supplied). Please note that the built-in test tools are optional and can be removed once debugging is complete.
- The *tcpip3.vhd* driver provides a conduit for sending and receiving data over a highspeed TCP-IP network connection provided the FPGA is connected to a SMSC LAN91C111 or compatible integrated circuit.
- The *usb20.vhd* driver allows one to connect the modulator input and demodulator output to an external PC over a high-speed USB 2.0 link. An external USB PHY with standard UTMI-compliant interface is required.
- The *afe.vhd* driver configures the AD9860/2 integrated circuit which implements analog-to-digital and digital-to-analog conversion.
- *acg10.vhd* is the first receiver AGC. Its goal is to maximize the input signal amplitude while preventing saturation at the A/D converters. The gain actuation is external.
- *pwm.vhd* generates a pulse-width modulated gain control signal for an external receive amplifier under the control of *acg10.vhd*.

*elastic\_buffer\_flex.vhd* is another ancillary component to convert bit-wise or byte-wise input data stream into symbol-wide samples.

#### Clock / Timing

The software uses two main clocks:

- the main processing clock CLK\_P, typically 90 MHz on a Xilinx Spartan-3.
- CLK\_IN is the input sample clock on the left connector. Typical maximum frequency 105 MHz on a Xilinx Spartan-3.

Other secondary clocks include:

- USB\_CLK60G, a clean low-jitter 60 MHz clock from the USB PHY. It is used as a frequency reference to generate CLK\_P, AFE\_CLK64 and CLK\_IO.
- AFE\_CLK64, the 64 MHz clock driving the AD9860/2 analog front-end.
- CLK\_IO is the 40 MHz output sample clock.



(\*) denotes edge-trigger signal Baseline clock architecture Yellow = 60 MHz reference clock Green = processing zone 90 MHz Dark Blue = output clock(40/90 MHz) Light Blue = external input clock Brown = 64 MHz I/O zone

# Modulator

The modulator is implemented as two top-level components: *psk\_qam\_apsk\_modulator.vhd* and *transmitter\_1.vhd*. The first component implements the modulation at baseband with 4 samples per symbol. The second component interpolates the output and translates the signal to the desired center frequency.

*psk\_qam\_apsk\_modulator.vhd* first selects the input data stream. The input data stream can be an external stream (DATA\_IN) or an internal pseudo-random binary (PRBS-11) test sequence generated by the *lfsr11c.vhd* component.

Periodic frame markers can be inserted in the modulated stream to help with the demodulation. Two methods are supported:

- (a) Method 1: insert a 32-bit synchronization pattern, called unique word, every FRAME\_LENGTH bits, where FRAME\_LENGTH is a user-specified constant. The unique word and the payload bits are subsequently modulated using a common user-specified modulation.
- (b) Method 2 (preferred) : insert a 32-symbol synchronization pattern every SYMBOL\_COUNT\_MODULO symbols, where SYMBOL\_COUNT\_MODULO is a user-specified constant. This synchronization sequence is modulated using a simple BPSK modulation, irrespective of the modulation selected for the payload data. This method is preferred as it simplifies the task of removing phase ambiguities at the demodulator.

#### Modulator VHDL hierarchy

😑 😘 MODULATOR_001 - PSK_QAM_APSK_MODULATOR - behavioral (SRC/PSK_QAM_APSK_MOD/SRC/psk_gam_apsk_modulator.vhd)
EFSR11_001 - LFSR11C - behavior (SRC/PSK_QAM_APSK_MOD/SRC/LFSR11C.VHD)
ELASTIC_BUFFER_00A · ELASTIC_BUFFER · behavioral (SRC/PSK_QAM_APSK_MOD/SRC/elastic_buffer.vhd)
ELASTIC_BUFFER_001 · ELASTIC_BUFFER · behavioral (SRC/PSK_QAM_APSK_MOD/SRC/elastic_buffer.vhd)
ELASTIC_BUFFER_002 · ELASTIC_BUFFER · behavioral (SRC/PSK_QAM_APSK_MOD/SRC/elastic_buffer.vhd)
🔤 🔚 RAISED_COS_001 - RAISED_COS5E - behavior (SRC/PSK_QAM_APSK_DEMOD/raised_cos5e.vhd)
- 🔚 RAISED_COS_002 - RAISED_COS5E - behavior (SRC/PSK_QAM_APSK_DEMOD/raised_cos5e.vhd)
- 🔚 AMPLITUDE_CONTROL_001 - MULT18×18SIGNED - BEHAVIOR (SRC/PSK_QAM_APSK_MOD/SRC/mult18×18signed.vhd)
MAMPLITUDE_CONTROL_002 · MULT18X18SIGNED · BEHAVIOR (SRC/PSK_QAM_APSK_MOD/SRC/mult18x18signed.vhd)
📮 😘 TRANSMITTER_001 - TRANSMITTER_1 - Behavioral (SRC/TRANSMITTER/transmitter_1.vhd)
CIC_INTERPOL_001 - CIC_INTERPOL3 - behavioral (SRC/TRANSMITTER/CIC_Interpol3.VHD)
CIC_INTERPOL_002 - CIC_INTERPOL3 - behavioral (SRC/TRANSMITTER/CIC_Interpol3.VHD)
🖨 强 DIGITAL_DC2_001 - DIGITAL_DC2 - DIGITAL_DC_ARCH (SRC/PSK_QAM_APSK_DEMOD/digital_dc2.vhd)
🛄 SIN_COS001 - SIGNED_SIN_COS_TBL2 - BEHAVIOR (SRC/PSK_QAM_APSK_DEMOD/signed_sin_cos_tbl2.vhd)
🔤 🛄 CCF_NCO_001 - NCO32 - behavioral (SRC/PSK_QAM_APSK_DEMOD/NCO32.vhd)



# Block Diagram (PSK / QAM / APSK Digital Modulator)

# Modulator VHDL Simulation

Representative simulation screens for salient internal signals are captured and discussed below.

All constellations for all modulation types are stored inside table MOD\_TABLE\_001, implemented as a readonly block RAM RAMB16\_S18\_S18. Each constellation point is expressed as a complex (DATA1\_I, DATA1\_Q) coordinate, where DATA1\_x are 18-bit precision signed numbers.

For example, the 32-APSK modulation is shown below when transmitting a (pseudo-)random sequence:



32-APSK constellation at the output of the modulation look-up table

Please note that the MOD\_TABLE\_001 look-up table has plenty of room available to store other (custom) modulations, whether PSK, QAM or APSK.

The exact same table is used at the demodulator (see *symbol\_decoding.vhd*) to re-encode the decoded symbols. (for decision-directed algorithms).

The ideal constellation is then converted to impulses (DATA2\_I, DATA2\_Q) before undergoing root raised cosine (RRC) filtering to generate (DATA3A\_I, DATA3A\_Q), as illustrated by the time-domain plot below:



RRC filter input impulses(red) and output impulse response (blue). 32-APSK

The RRC filter is implemented as a 30-tap FIR filter with 4 samples per symbol. The latency between input impulse and peak of the impulse response output is thus 3.75 symbols.

A final interpolation is implemented in *CIC\_Interpol3.vhd* so as to reduce aliasing, as illustrated below. The interpolation factor CIC\_R is limited to minimize the intersymbol interferences.



# Demodulator VHDL hierarchy

🖃 🔡 DB	EMOD1 - PSK_QAM_APSK_DEMOD - behavioral (SRC/PSK_QAM_APSK_DEMOD/psk_gam_apsk_demod.vhd)
···· ¥	BIAS_REMOVAL_001 - BIAS_REMOVAL - behavioral (SRC/PSK_QAM_APSK_DEMOD/bias_removal.vhd)
🗐 · 🕌	DIGITAL_DC2_001 - DIGITAL_DC2 - DIGITAL_DC_ARCH (SRC/PSK_QAM_APSK_DEMOD/digital_dc2.vhd)
	🔚 SIN_COS001 - SIGNED_SIN_COS_TBL2 - BEHAVIOR (SRC/PSK_QAM_APSK_DEMOD/signed_sin_cos_tbl2.vhd)
···· 🖌	CCF_NCD_001 - NC032 - behavioral (SRC/PSK_QAM_APSK_DEMOD/NC032.vhd)
···· 🖌	CIC_FILTER_001 - CIC - behavioral (SRC/PSK_QAM_APSK_DEMOD/CIC.vhd)
···· 🖌	CIC_FILTER_002 - CIC - behavioral (SRC/PSK_QAM_APSK_DEMOD/CIC.vhd)
···· 🖌	RESAMPLING_001 - RESAMPLING - behavioral (SRC/PSK_QAM_APSK_DEMOD/resampling.vhd)
···· 🖌	RAISED_COS_001 - RAISED_COS5E - behavior (SRC/PSK_QAM_APSK_DEMOD/raised_cos5e.vhd)
···· 🖌	RAISED_COS_002 - RAISED_COS5E - behavior (SRC/PSK_QAM_APSK_DEMOD/raised_cos5e.vhd)
···· 🖌	MULT18X18_001 - MULT18X18SIGNED - BEHAVIOR (SRC/PSK_QAM_APSK_MOD/SRC/mult18x18signed.vhd)
···· 🖌	MULT18X18_002 - MULT18X18SIGNED - BEHAVIOR (SRC/PSK_QAM_APSK_MOD/SRC/mult18x18signed.vhd)
···· 🖌	POLAR_001 - POLAR3 - Behavior (SRC/PSK_QAM_APSK_DEMOD/POLAR3.vhd)
···· 🖌	AGC14_001 - AGC14 - BEHAVIOR (SRC/PSK_QAM_APSK_DEMOD/AGC14.vhd)
🖨 🖌	SYMBOL_TIMING_LOOP2_001 - SYMBOL_TIMING_LOOP2 - BEHAVIOR (SRC/PSK_QAM_APSK_DEMOD/symbol_timing_loop2.vhd)
	🔀 POLAR_001 - POLAR3 - Behavior (SRC/PSK_QAM_APSK_DEMOD/POLAR3.vhd)
···· 🖌	SYMBOL_DECODING_001 - SYMBOL_DECODING - BEHAVIOR (SRC/PSK_QAM_APSK_DEMOD/symbol_decoding.vhd)
···· 🖌	CARRIER_TRACKING_002 · CARRIER_TRACKING · behavioral (SRC/PSK_QAM_APSK_DEMOD/carrier_tracking.vhd)
<u> </u>	SOF_SYNC2_001 - SOF_SYNC2 - BEHAVIOR (SRC/PSK_QAM_APSK_DEMOD/sof_sync2.vhd)
E	MF001 - MATCHED_FILTER32 - BEHAVIOR (SRC/UW codesnippet/matched_filter32.vhd)
	📾 🔚 PC_16A - PC_16 - BEHAVIOR (SRC/UWcodesnippet/pc_16.vhd)
	📾 🔚 PC_16B - PC_16 - BEHAVIOR (SRC/UWcodesnippet/pc_16.vhd)
	📾 🔚 PC_16C - PC_16 - BEHAVIOR (SRC/UWcodesnippet/pc_16.vhd)
	😑 🔚 PC_16D - PC_16 - BEHAVIOR (SRC/UWcodesnippet/pc_16.vhd)
	🔚 FA01 - FA - BEHAVIOR_FA (src/ber/fa.vhd)
	FA02 - FA - BEHAVIOR_FA (src/ber/fa.vhd)
	FA03 - FA - BEHAVIOR_FA (src/ber/fa.vhd)
	FA04 - FA - BEHAVIOR_FA (src/ber/fa.vhd)
	FA05 - FA - BEHAVIOR_FA (src/ber/fa.vhd)
	FA09 - FA - BEHAVIOR_FA (src/ber/fa.vhd)
	FA10 - FA - BEHAVIOR_FA (src/ber/fa.vhd)
	FA11 - FA - BEHAVIOR_FA (src/ber/fa.vhd)
	HA12 - HA - BEHAVIOR_HA (src/ber/ha.vhd)
	HA13 - HA - BEHAVIOR_HA (src/ber/ha.vhd)
	FA14 - FA - BEHAVIOR_FA (src/ber/fa.vhd)
	HA15 - HA - BEHAVIOR_HA (src/ber/ha.vhd)
	FA17 - FA - BEHAVIOR_FA (src/ber/fa.vhd)
	AIS FA18 - HA - BEHAVIOR_HA (src/ber/ha.vhd)
	HA03 - HA - BEHAVIOR_HA (src/ber/ha.vhd)
_	HAU4 - FA - BEHAVIUR_FA (src/ber/ta.vhd)
Ŀ	MFUU2 · MATCHED_FILTER32 · BEHAVIUR (SRC/UW codesnippet/matched_hiter32.vhd)
	SUF_TRACKZ_UUT - SUF_TRACK2 - BEHAVIUR (SRC/PSK_UAM_APSK_DEMUD/sof_track2.vhd)
_	The BEH_UUT - BEH_MEASUREMENTZ - BEHAVIUR (SRU/UW codesnippet/ber_measurementZ.vhd)



# Block Diagram (PSK / QAM / APSK Digital Demodulator)

# Demodulator VHDL Simulation #1

Conditions: QPSK, period 1080ns, noiseless channel, 45deg initial phase error.



Demodulated I-channel symbols DATA3\_I\_D2 (red) vs nominal constellation NOMINAL\_SYMBOL\_I(blue). Illustrates the combined effect of the AGC, Carrier tracking and Symbol timing tracking loops.



Carrier tracking (CCF\_PHASE). Response to a 45deg initial phase offset. Stabilizes in about 250 symbols. No significant overshoot.



Symbol timing tracking (NCO\_TIMING\_CONTROL\_ACC) Response to a 0.16 symbol delay. Stabilizes in about 850 symbols.



#### **Demodulator VHDL Simulation #2**

Conditions: 16-QAM, period 1080ns, noiseless channel, 45deg initial phase error.



Illustrates loops convergence (AGC, carrier tracking, symbol timing) DATA6\_I: I-channel at the output of the root raised cosine filter, blue DATA3\_I\_D2: I-channel sampled at the center of the symbol, red dot



Response to a 45deg initial phase offset. Stabilizes in about 500 symbols. No significant overshoot.



Symbol timing tracking (NCO\_TIMING\_CONTROL\_ACC) Response to a 0.16 symbol delay. Stabilizes in about 850 symbols.



Demodulated I-channel symbols DATA3\_I\_D2 (blue) vs nominal constellation NOMINAL\_SYMBOL\_I(red). Illustrates the combined effect of the AGC, Carrier tracking and Symbol timing tracking loops. A small amount of intersymbol interference is visible (due to the CIC interpolation and decimation filters)



Demodulated constellation (DATA3\_I\_D2, DATA3\_Q\_D2)

# **Demodulator VHDL Simulation #3**

Conditions: BPSK, period 1080ns, noiseless channel, 0deg initial phase error.



Demodulated I-channel symbols DATA3\_I\_D2 (blue) vs nominal constellation NOMINAL\_SYMBOL\_I(red). Illustrates the combined effect of the AGC, Carrier tracking and Symbol timing tracking loops.

#### **FPGA** Occupancy

Design Summary					
Logic Utilization:					
Total Number Slice Registers:	10,022 out of	40,960	24%		
Number used as Flip Flops:	10,0	013			
Number used as Latches:		9			
Number of 4 input LUTs:	13,987 out of	40,960	34%		
Logic Distribution:					
Number of occupied Slices:		10,203	out of	20,480	49%
Number of Slices containing or	nly related logic	c: 10,20	03 out of	10,203	100%
Number of Slices containing u	nrelated logic:		0 out of	10,203	0 응
*See NOTES below for an exp.	lanation of the e	effects (	of unrela	ted logic	2
Total Number 4 input LUTs:	15,244 out of	40,960	37%		
Number used as logic:	13,987				
Number used as a route-thru:	977				
Number used as Shift registers:	280				
Number of bonded IOBs:	166 out of	489	33%		
IOB Flip Flops:	110				
IOB Latches:	1				
Number of Block RAMs:	20 out of	40	50%		
Number of MULT18X18s:	16 out of	40	40%		
Number of GCLKs:	8 out of	8 2	100%		
Number of DCMs:	3 out of	4	75%		

Total equivalent gate count for design: 1,616,179

# **Contact Information**

MSS • 18221-A Flower Hill Way • Gaithersburg, Maryland 20879 • U.S.A. Telephone: (240) 631-1111 Facsimile: (240) 631-1676 E-mail: info@comblock.com