COM-1402 PSK / QAM / APSK modulator
VHDL SOURCE CODE OVERVIEW

Overview

The COM-1402 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-1402 VHDL code runs on the generic COM-1400 hardware platform. The schematics [2] for this platform are available in this CD.

Reference documents

[1] specifications: com1402.pdf

[2] hardware schematics: com_1400schematics.pdf

[3] VHDL source code in directory
com-1402_rev\src

[4] Xilinx ISE project files
com-1402_*rev*\com1402_ISE82.npl
com-1402_*rev*\com1402_ISE91.npl

[5] .ucf constraint file
com-1402_*rev*\src\COM1402.ucf

[6] .mcs FPGA bit files
com-1402_*rev*\bin\com1402A_*rev*.mcs
com-1402_*rev*\bin\com1402B_*rev*.mcs
com-1402_*rev*\bin\com1402D_*rev*.mcs
com-1402_*rev*\bin\com1402E_*rev*.mcs

where *rev* is the current revision number.

Configuration Management

The current software revision is 6.

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 *com1402.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.

MSS • 18221-A Flower Hill Way • Gaithersburg, Maryland 20879 • U.S.A.
Telephone: (240) 631-1111 Facsimile: (240) 631-1676 www.ComBlock.com

Target Hardware

The modulator code is written in generic VHDL so that it can be ported to a variety of FPGAs. The modulator code was developed on a Xilinx Spartan-3 XC3S400-4FT256 FPGA.

The maximum achievable modulation rate depends primarily on the processing FPGA technology. The VHDL code is designed for a maximum modulation symbol rate of ¼ 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.

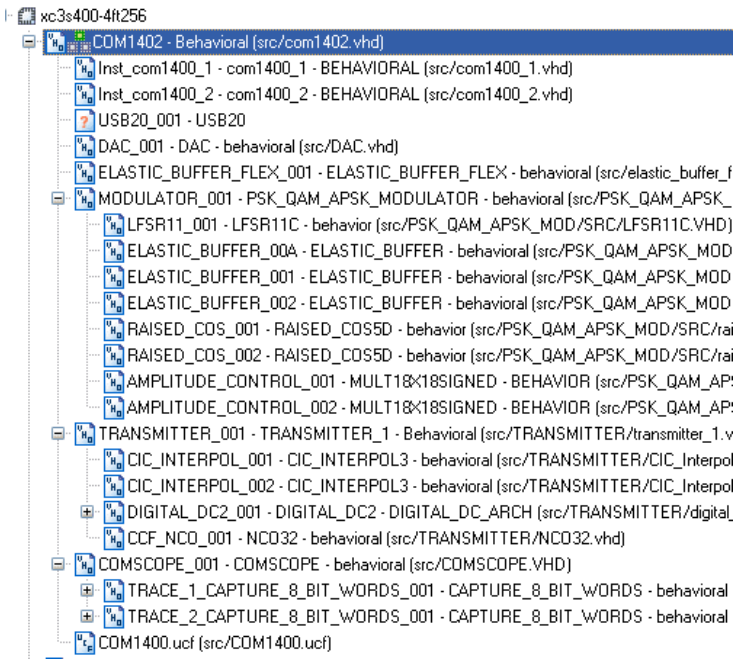
At the modulator periphery, the *usb20.vhd*: interfaces with any external USB 2.0 PHY over a standard UTMI interface. (for example, the SMSC USB3250 IC).

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



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

The root entity (highlighted above) is *com1402.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.
- 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 *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.
- *dac.vhd* sets an external AD5611 DAC (unused in this project) in tri-state / idle mode.

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

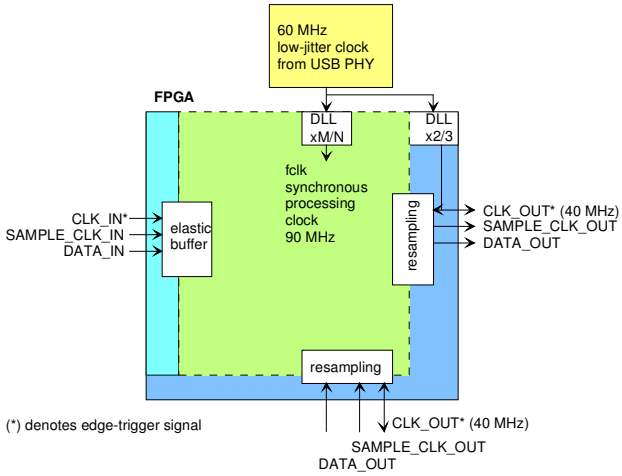
Clock / Timing

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.

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 and CLK_IO.
- CLK_IO is the 40 MHz output sample clock.



Baseline clock architecture
Yellow = 60 MHz USB PHY reference clock
Green* = CLK_P *processing zone 90 MHz
Dark Blue = output clock(40/90 MHz)
Light Blue = external input clock

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.

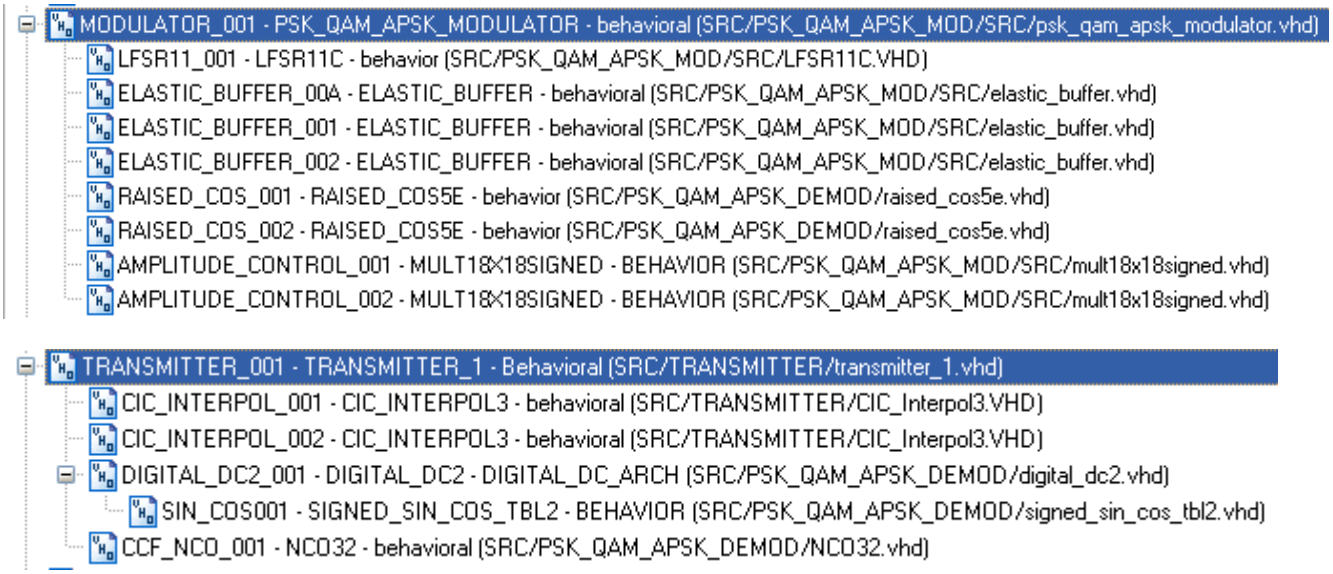
psk_qam_apsk_modulator.vhd first selects the input data stream. The input data stream can be an external stream (DATA_IN) or an internal pseudo-random binary (PRBS-11) test sequence generated by the *lfsr11c.vhd* component.

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

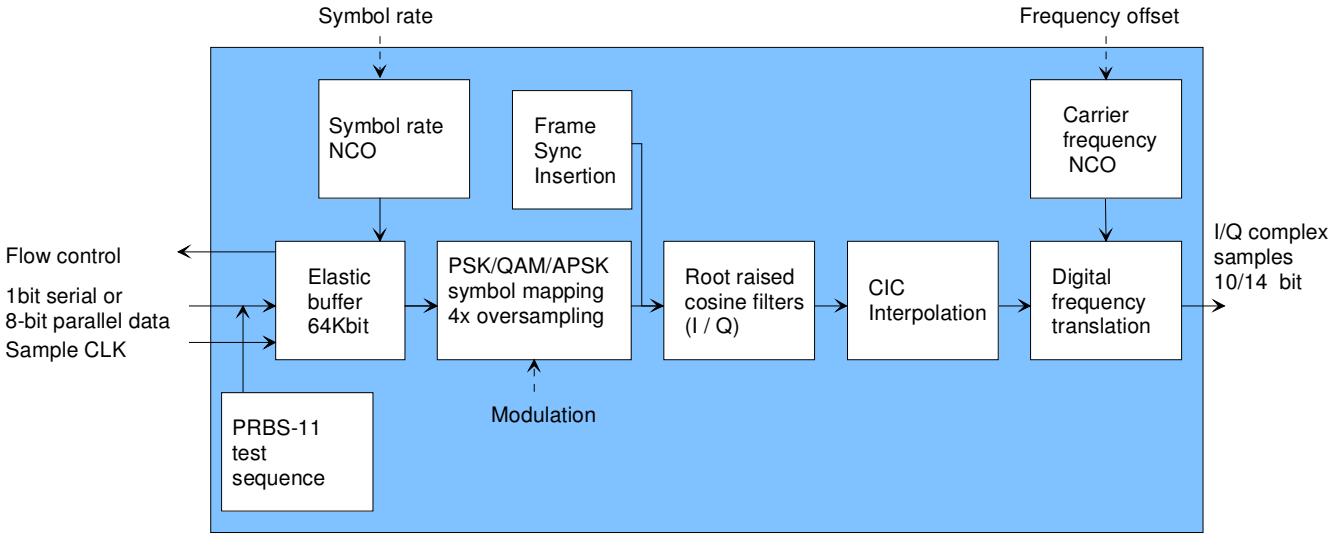
(a) Method 1: insert a 32-bit synchronization pattern, called unique word, every FRAME_LENGTH bits, where FRAME_LENGTH is a user-specified constant. The unique word and the payload bits are subsequently modulated using a common user-specified modulation.

(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



Block Diagram (PSK / QAM / APSK Digital Modulator)

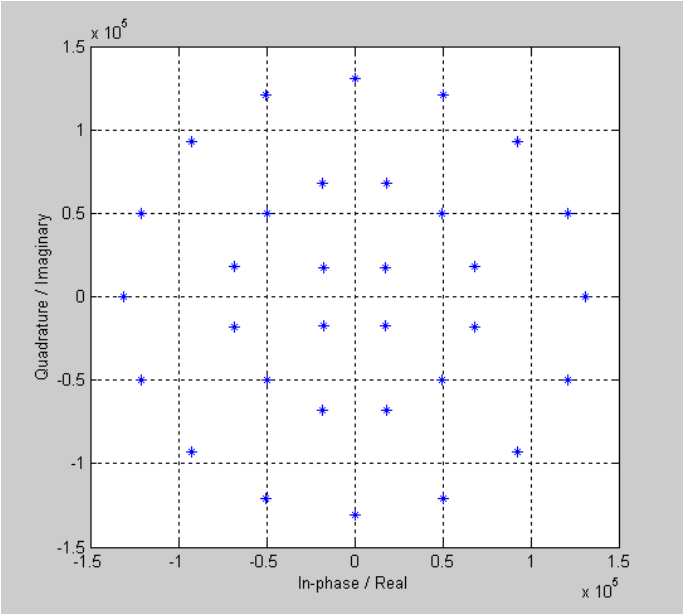


Modulator VHDL Simulation

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 read-only 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:

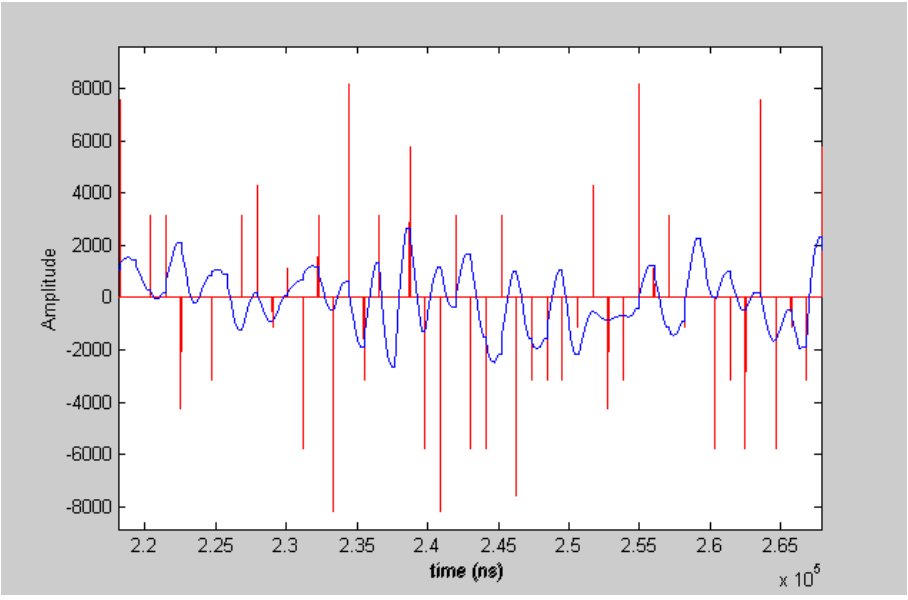


32-APSK constellation at the output of the modulation look-up table

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).

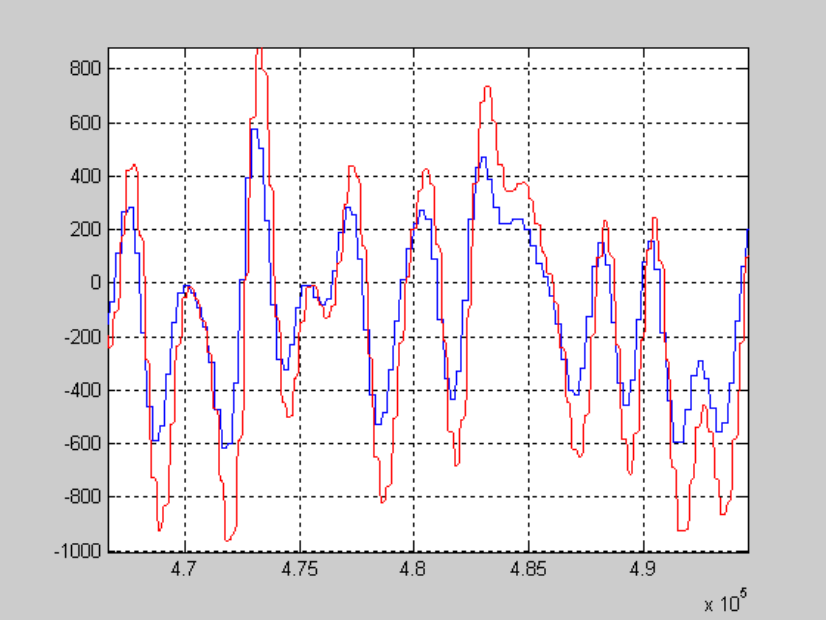
The ideal constellation is then converted to impulses (DATA2_I, DATA2_Q) before undergoing root raised cosine (RRC) filtering to generate (DATA3A_I, DATA3A_Q), as illustrated by the time-domain plot below:



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.



Interpolation input (blue) and output (red). 32-APSK

FPGA Occupancy

```
Design Summary
--------------
Logic Utilization:
  Number of Slice Flip Flops:       3,807 out of   7,168   53%
  Number of 4 input LUTs:           4,989 out of   7,168   69%
Logic Distribution:
  Number of occupied Slices:                      3,271 out of   3,584   91%
  Number of bonded IOBs:              157 out of     173   90%
    IOB Flip Flops:                    38
  Number of Block RAMs:               13 out of      16   81%
  Number of MULT18X18s:                6 out of      16   37%
  Number of GCLKs:                     8 out of       8  100%
  Number of DCMs:                      2 out of       4   50%

Total equivalent gate count for design:  969,910
```

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