COM-1019 DIRECT-SEQUENCE SPREAD-SPECTRUM MODULATOR 20 Mchip/s VHDL SOURCE CODE OVERVIEW

Overview

The COM-1019 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 Atmel AT90S8515 or 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 COM-1019 VHDL code runs on the generic COM-8000 hardware platform. The schematics [2] for this platform are available in this CD.

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.

Reference documents

[1] specifications: com1019.pdf

[2] hardware schematics: com_8000schematics.pdf

[3] VHDL source code in directory com-1019_012\src

[4] .ucf constraint file com-1019_012\src\root_mod.ucf

[5] .mcs FPGA bit files
com-1019_012\com1019A_012.mcs
com-1019_012\com1019B_012.mcs
com-1019_012\com1019D_012.mcs
com-1019_012\com1019E_012.mcs

Configuration Management

The current software revision is 12.

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_cos4x* root raised filters are included in the source code .src directory. To change the filter:

(a) change the OPTION constant in the *root_mod.vhd* file so that the resulting bit file can later be correctly identified.
(b) Change the RAISED_COS4x statements in three places within the *ds_ss_modulator.vhd* file: declaration and two instantiations.

VHDL development environment

The VHDL software was developed using two development environments:

(a) Xilinx ISE 4.1 with Synopsys FPGA Express 3.6 as synthesis tool.
(b) Xilinx ISE 6.3 with XST as synthesis tool.

Target FPGA

The VHDL code was synthesized for the Xilinx Spartan-IIE XC2S300E-6PQ208 FPGA.

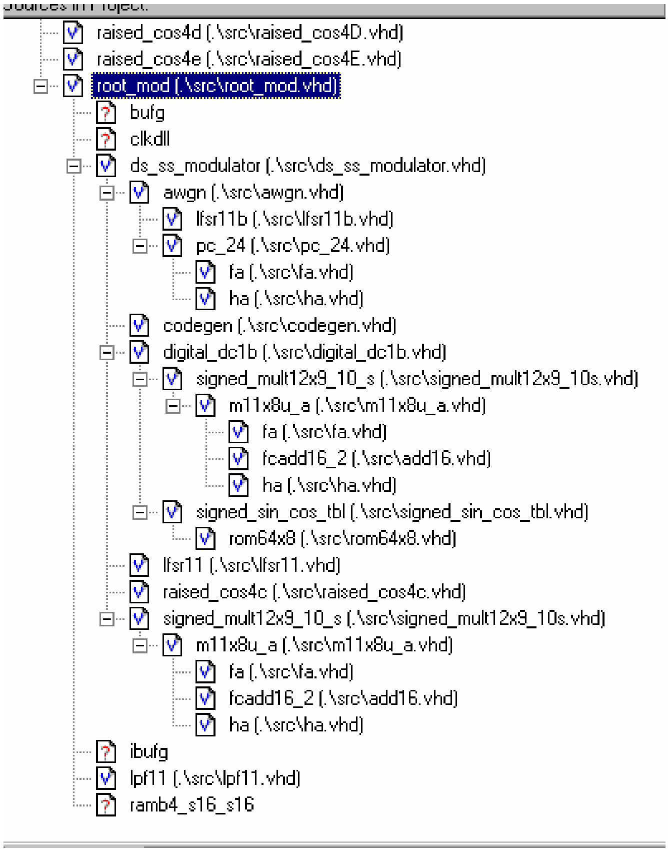
Xilinx-specific code

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

- BUFG
- IBUFG

- CLKDLL (x2)
- RAMB4_S16_S16

VHDL software hierarchy



```
Sources in Project:
    raised_cos4d (.\src\raised_cos4D.vhd)
    raised_cos4e (.\src\raised_cos4E.vhd)
  ⊟ root_mod (.\src\root_mod.vhd)
      bufg
      clkdll
    ⊟ ds_ss_modulator (.\src\ds_ss_modulator.vhd)
      ⊟ awgn (.\src\awgn.vhd)
          lfsr11b (.\src\lfsr11b.vhd)
        ⊟ pc_24 (.\src\pc_24.vhd)
            fa (.\src\fa.vhd)
            ha (.\src\ha.vhd)
        codegen (.\src\codegen.vhd)
      ⊟ digital_dc1b (.\src\digital_dc1b.vhd)
        ⊟ signed_mult12x9_10_s (.\src\signed_mult12x9_10s.vhd)
          ⊟ m11x8u_a (.\src\m11x8u_a.vhd)
              fa (.\src\fa.vhd)
              fcadd16_2 (.\src\add16.vhd)
              ha (.\src\ha.vhd)
        ⊟ signed_sin_cos_tbl (.\src\signed_sin_cos_tbl.vhd)
            rom64x8 (.\src\rom64x8.vhd)
        lfsr11 (.\src\lfsr11.vhd)
        raised_cos4c (.\src\raised_cos4c.vhd)
      ⊟ signed_mult12x9_10_s (.\src\signed_mult12x9_10s.vhd)
        ⊟ m11x8u_a (.\src\m11x8u_a.vhd)
            fa (.\src\fa.vhd)
            fcadd16_2 (.\src\add16.vhd)
            ha (.\src\ha.vhd)
      ibufg
      lpf11 (.\src\lpf11.vhd)
      ramb4_s16_s16
```

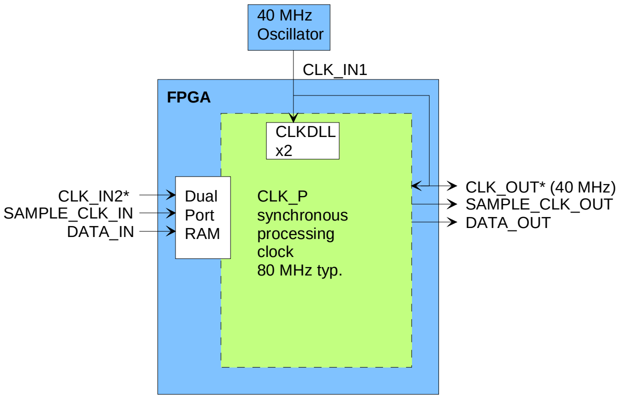
The code is stored with one, and only one, entity per file as shown above.

The root program (highlighted) is *root_mod.vhd*.

Clock / Timing

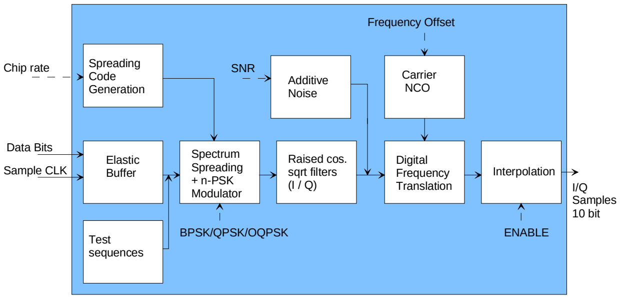
The software uses two different clocks:

- external clock CLK_IN2 which serves synchronous clock for the input data stream.
- CLK_IN1 is generated by a 40 MHz oscillator on the COM-1019 module. It is used as reference for the output clock and for the double-frequency processing clock. The code is written to meet the timing requirements on the target FPGA at a speed of at least 80 MHz.



(*) denotes edge-trigger signal

Block Diagram



The hierarchical nature of the VHDL code reflects the block diagram above:

- *root_mod* is the root program which includes the modulator *ds_ss_modulator*, the concatenated interpolation filters *lpf11* and ancillary monitoring and control functions (interface with microprocessor).
- the main Direct-Sequence Spread-Spectrum functions are encapsulated within *ds_ss_modulator*.
- entity *Codegen* implements all spreading codes.
- the frequency translation is implemented within *digital_dc1b*. The frequency translation is realized in the form of a complex vector rotation, using sine/cosine lookup tables (*signed_sin_cos_tbl*) and pipeline multipliers (*signed_mult12x9_10_s*) made of half adders *ha* and full adders *fa*.
- spectrum shaping is made by means of two root raised cosine filters *raised_cos4x*, one for each complex axis.

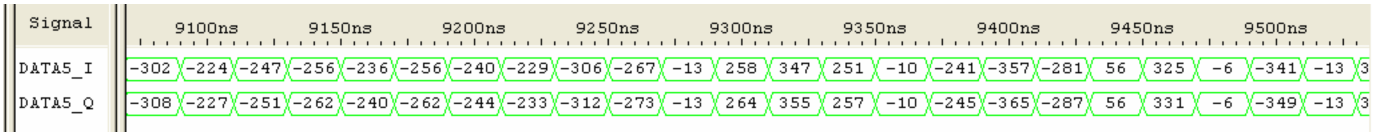
- the PRBS-11 pseudo-random test pattern is generated within the *lfsr11* entity.
- an approximation of additive white gaussian noise is implemented by summing several nearly independent random sequences within the *awgn* entity.

VHDL Simulation

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

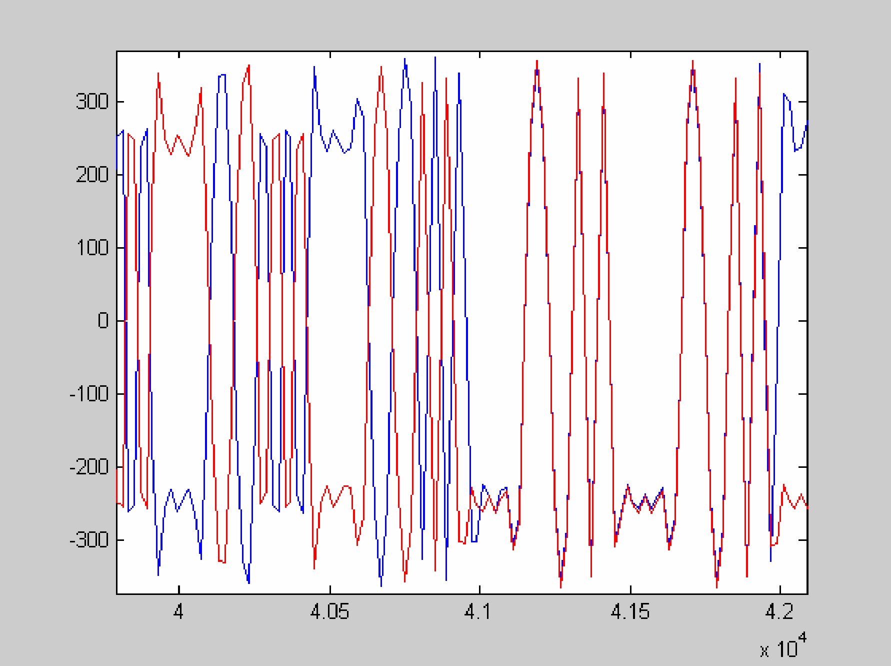
Ouput waveform

VHDL simulation is used to capture the output modulated waveform while the COM-1019 DSSS modulator is configured as 19.9 Mchips/s, Barker code, code length 13, PRBS-11 internal data source, QPSK modulation, noiseless, maximum amplitude, no frequency offset.
NCO_CHIP_RATE := x”3FAE14”
SPREADING_FACTOR := x"00000D"
CODE_SEL := "011"
NCO_CENTER_FREQUENCY := x"000000"
SIGNAL_SCALING := "011111111"
AWGN_SCALING := "000000000"
REG18 := x"96"
REG19 := x"0C"
REG20 := x"00"



```
Signal    9100ns      9150ns      9200ns      9250ns      9300ns      9350ns      9400ns      9450ns      9500ns
DATA5_I   -302 -224 -247 -256 -236 -256 -240 -229 -306 -267 -13 258 347 251 -10 -241 -357 -281 56 325 -6 -341 -13 3
DATA5_Q   -308 -227 -251 -262 -240 -262 -244 -233 -312 -273 -13 264 355 257 -10 -245 -365 -287 56 331 -6 -349 -13 3
```

The waveform is capture and plotted:



Modulator output samples. I-channel (blue) and Q-channel (red).

Because of the high chip rate, the output is sampled with only 4 samples / chip.

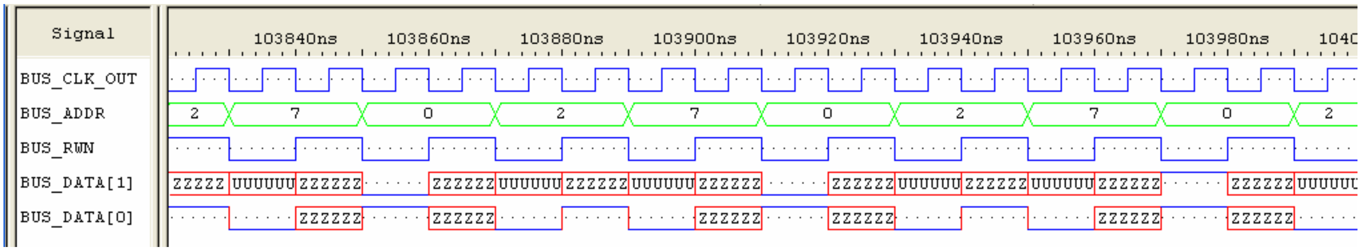
Input Data Bus

A data bus format is one of the several input formats supported by the COM-1019 modulator.
A VHDL simulation of the COM-8004 -> COM-1019 interface over a shared data bus is discussed here.
Interface definition:

Input Module Interface Bus connection, REG19(4) = '1'	**Definition**
BUS_CLK_IN	40 MHz input reference clock for use on the synchronous bus.
BUS_ADDR[3:0]	Bus address. Input (since this module is a bus slave). Designates which slave module is targeted for this read or write transaction. All 1's indicates that the write data is to be broadcasted to all receiving slave modules. Read at the rising edge of BUS_CLK_IN
BUS_RWN	Read/Write#. Input (since this module is a bus slave). Indicates whether a read (1) or write (0) transaction is conducted. Read at the rising edge of BUS_CLK_IN. Read and Write refer to the bus master's perspective.
BUS_DATA[15:0]	Bi-directional data bus. Input when BUS_RWN='0'. Output when BUS_RWN='1'. Read data latency is 2 clock periods after the read command. Functional definition during write: • bit 0 SAMPLE_CLK_IN. '1' when DATA_IN is available • bit 1 DATA_IN data stream to modulator. • bits(15:2) undefined Functional definition during read: • bit 0 SAMPLE_CLK_IN_REQ requests data from the source. Used for flow control. • bits(15:1) undefined

The COM-1019 is a bus 'slave', i.e. it read the BUS_CLK_IN, BUS_ADDR, BUS_RWN signals. The 16-bit BUS_DATA is bi-directional. The simulation below assumes that three modulators are available to receive stream 0,2 and 7 respectively. For simplicity, only one COM-1019 modulator, responding to the bus address 0, is simulated. The COM-1019 supplies the read data with a 2 clock latency.

In this simulation, for each stream, a write transaction is immediately followed by a read transaction. During the write transaction, BUS_DATA(1) conveys the data if available [DATA_IN], and BUS_DATA(0) indicates whether BUS_DATA(1) includes data [SAMPLE_CLK_IN]. During the read transaction, BUS_DATA(0) indicates whether the recipient modulator is ready to accept further input data bits [SAMPLE_CLK_IN_REQ].



Data bus simulation. COM-8004 -> COM-1019 DSSS modulator

FPGA Occupancy

```
Design Summary
--------------
   Number of errors:      0
   Number of warnings:   25
   Number of Slices:              3,070 out of  3,072   99%
   Number of Slices containing
      unrelated logic:              154 out of  3,070    5%
   Number of Slice Flip Flops:    4,113 out of  6,144   66%
   Total Number 4 input LUTs:     3,688 out of  6,144   60%
      Number used as LUTs:                      3,646
      Number used as a route-thru:                 42
   Number of bonded IOBs:            50 out of    142   35%
      IOB Flip Flops:                              25
   Number of Block RAMs:              3 out of     16   18%
   Number of GCLKs:                   4 out of      4  100%
   Number of GCLKIOBs:                4 out of      4  100%
   Number of DLLs:                    1 out of      4   25%
Total equivalent gate count for design:  118,593
Additional JTAG gate count for IOBs:  2,592
```

Contact Information

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