COM-1418 DIRECT SEQUENCE SPREAD-SPECTRUM DEMODULATOR 22 Mchip/s. VHDL SOURCE CODE OVERVIEW

Overview

The COM-1418 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 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-1418 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: com1418.pdf

[2] hardware schematics: com_1400schematics.pdf

[3] VHDL source code in directory
com-1418_*rev* \src

[4] Xilinx ISE project files
com-1418_*rev* \com1418_ISE82.npl
com-1418_*rev* \com1418_ISE91.npl

[5] .ucf constraint file
com-1418_*rev* \src\com1418.ucf

[6] .mcs FPGA bit files
com-1418_*rev* \com1418A_*rev*.mcs
com-1418_*rev* \com1418B_*rev*.mcs
com-1418_*rev* \com1418D_*rev*.mcs
com-1418_*rev* \com1418E_*rev*.mcs

where *rev* is the current revision number.

Configuration Management

The current software revision *rev* is 0.

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 *com1418.vhd* file so that the resulting bit file can later be correctly identified.

(b) Change the RAISED_COS4x statements in three places within the *demod.vhd* file: declaration and two instantiations.

VHDL development environment

The VHDL software was developed with Xilinx ISE 10.1 with XST as synthesis tool.

Target FPGA

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 XC3S400-4FT256 FPGA.

The demodulator throughput is related primarily to the target FPGA technology. The VHDL code is designed for a maximum chip rate of ¼ of the FPGA processing clock CLK_P. In other words, the processing is performed with 4 samples per chip. 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 demodulator periphery, two interface components are hardware-specific: the USB and D/A converter 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).
- *dac.vhd*: Analog Devices AD5611D/A converter

Xilinx-specific code

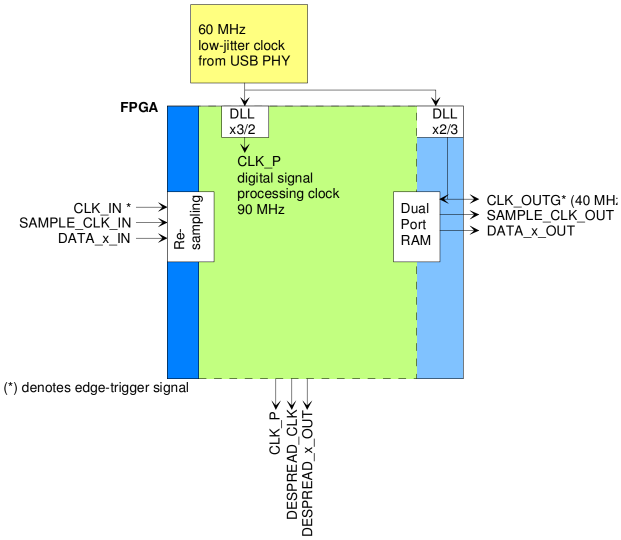
The VHDL source code is written in generic VHDL with few Xilinx primitives. No Xilinx CORE is used. The Xilinx primitives are:

- 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.)

Clock / Timing

The software uses three different clocks:

- external clock CLK_IN which serves synchronous clock for the input data stream.
- a processing clock CLK_P used for most of the digital signal processing. This processing clock is generated by a DCM locked onto a 60 MHz reference from the external USB PHY (USB_CLK60G).
- A 40 MHz output clock CLK_OUTG, also generated by a DCM locked onto a 60 MHz reference from the external USB PHY.



(*) denotes edge-trigger signal

Block Diagram

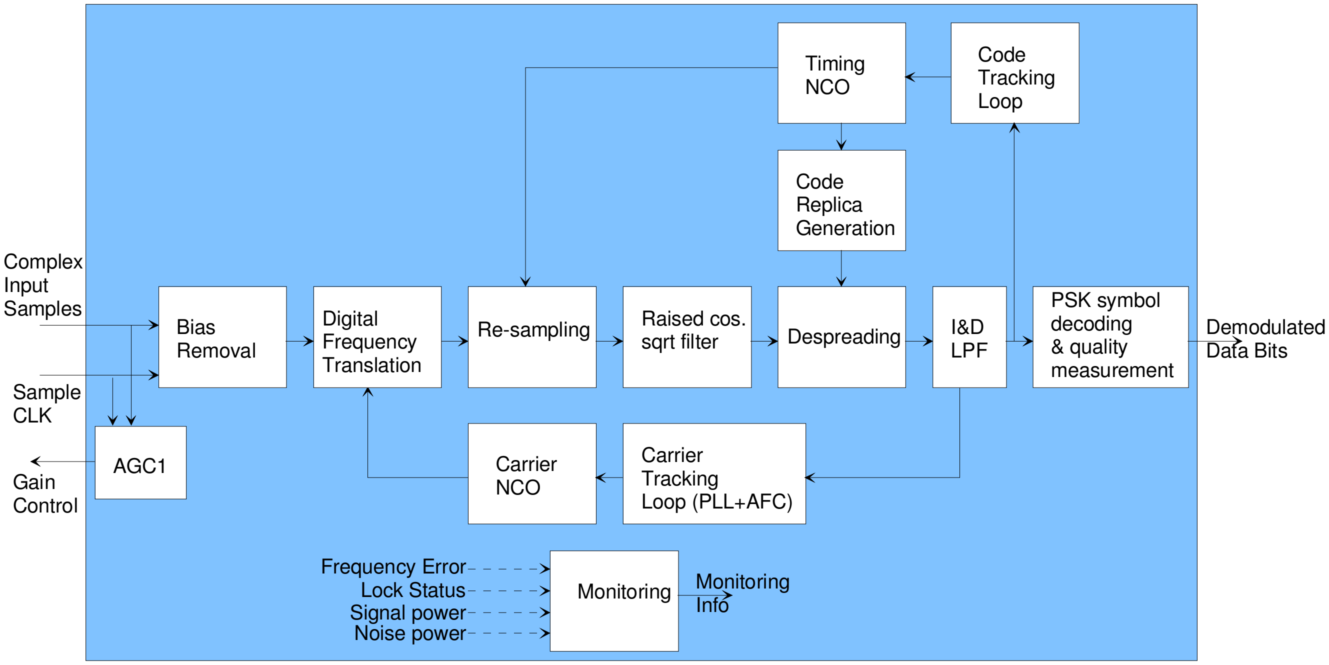
The hierarchical nature of the VHDL code reflects the block diagram below:

- *com1418.vhd* is the root program which includes the demodulator *dsss_demod*, the USB 2.0 driver *usb20*, the gain control DAC driver *dac* and ancillary functions such as monitoring and control functions (interface with microprocessor) and *comscope* to capture and display internal signals.
- the *usb20.vhd* driver allows one to connect the 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 main Direct-Sequence Spread-Spectrum functions are encapsulated within *dsss_demod*, including agc, channel filtering, despreading, carrier acquisition and tracking, code acquisition and tracking.
- the first processing step is removing any unwanted bias from the input signal in *bias_removal*.
- *acg10A* prevents saturation at the external A/D converter.
- entity *codegen* implements all spreading codes.
- the frequency translation is implemented within *digital_dc2*. The frequency translation is realized in the form of a complex vector rotation, using sine/cosine

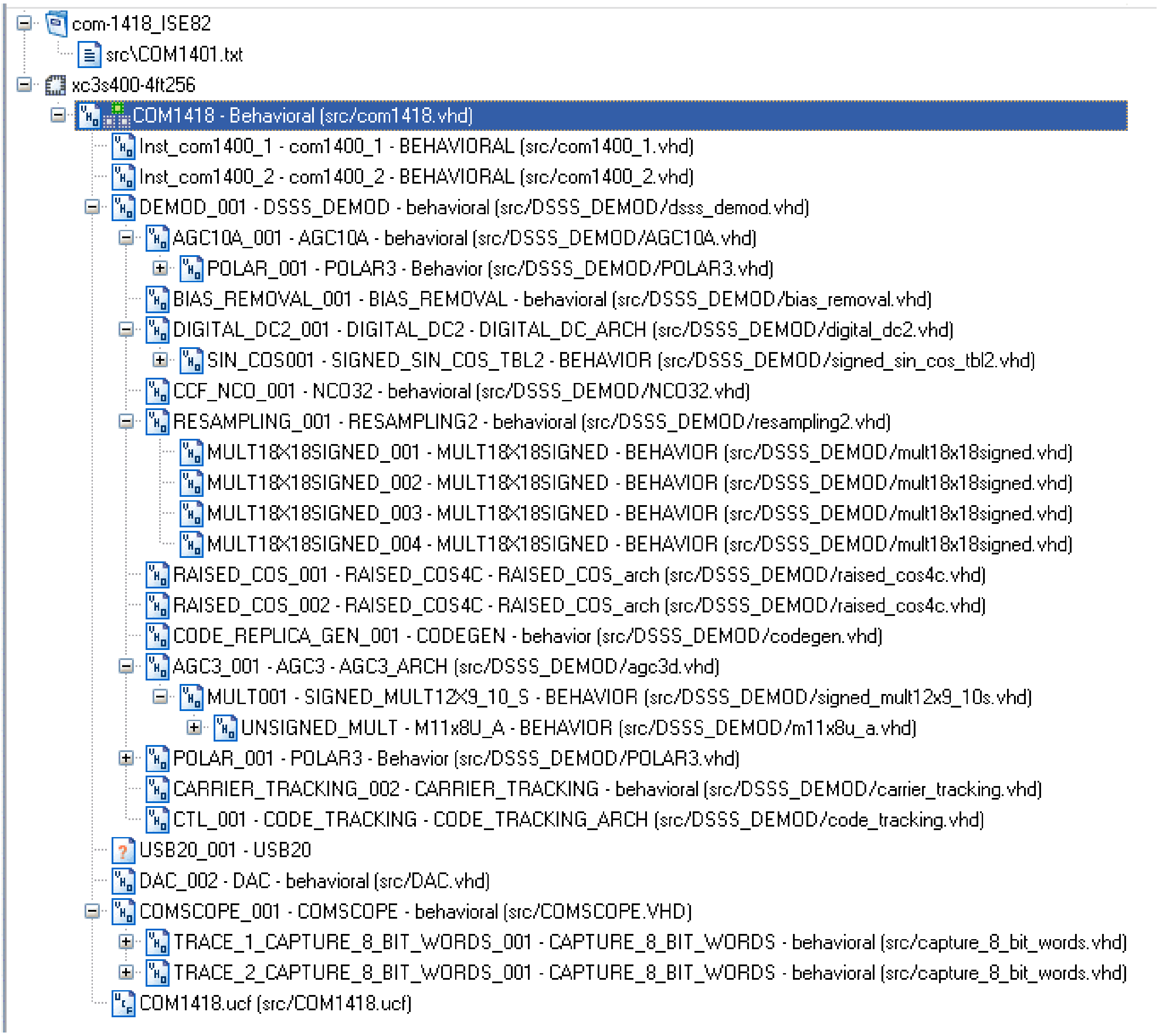
lookup tables (*signed_sin_cos_tbl2*) and hardware multipliers.

- noise rejection (channel filtering) is implemented by two root raised cosine filters *raised_cos4x*, one for each complex axis.
- a conventional early-late gate delay lock loop is implemented in *code_tracking*. This entity includes only the error computation and the loop gain settings. Currently, a first-order loop is implemented. The 2nd order is commented out. The NCO and integrate & dump functions are part of the *dsss_demod* entity.
- a digital AGC *agc3* is used to normalize the despread signal and to condition the resulting signal in a variety of formats: 4-bit unsigned (soft-quantization), 8-bit signed (for carrier tracking loop processing).
- most carrier acquisition, carrier tracking and AFC functions are implemented within the *carrier_tracking* entity. The actual carrier NCO is part of the *dsss_demod* entity.

Block Diagram



VHDL software hierarchy



```
com-1418_ISE82
    src\COM1401.txt
xc3s400-4ft256
    COM1418 - Behavioral (src/com1418.vhd)
        Inst_com1400_1 - com1400_1 - BEHAVIORAL (src/com1400_1.vhd)
        Inst_com1400_2 - com1400_2 - BEHAVIORAL (src/com1400_2.vhd)
        DEMOD_001 - DSSS_DEMOD - behavioral (src/DSSS_DEMOD/dsss_demod.vhd)
            AGC10A_001 - AGC10A - behavioral (src/DSSS_DEMOD/AGC10A.vhd)
                POLAR_001 - POLAR3 - Behavior (src/DSSS_DEMOD/POLAR3.vhd)
            BIAS_REMOVAL_001 - BIAS_REMOVAL - behavioral (src/DSSS_DEMOD/bias_removal.vhd)
            DIGITAL_DC2_001 - DIGITAL_DC2 - DIGITAL_DC_ARCH (src/DSSS_DEMOD/digital_dc2.vhd)
                SIN_COS001 - SIGNED_SIN_COS_TBL2 - BEHAVIOR (src/DSSS_DEMOD/signed_sin_cos_tbl2.vhd)
            CCF_NCO_001 - NCO32 - behavioral (src/DSSS_DEMOD/NCO32.vhd)
            RESAMPLING_001 - RESAMPLING2 - behavioral (src/DSSS_DEMOD/resampling2.vhd)
                MULT18X18SIGNED_001 - MULT18X18SIGNED - BEHAVIOR (src/DSSS_DEMOD/mult18x18signed.vhd)
                MULT18X18SIGNED_002 - MULT18X18SIGNED - BEHAVIOR (src/DSSS_DEMOD/mult18x18signed.vhd)
                MULT18X18SIGNED_003 - MULT18X18SIGNED - BEHAVIOR (src/DSSS_DEMOD/mult18x18signed.vhd)
                MULT18X18SIGNED_004 - MULT18X18SIGNED - BEHAVIOR (src/DSSS_DEMOD/mult18x18signed.vhd)
            RAISED_COS_001 - RAISED_COS4C - RAISED_COS_arch (src/DSSS_DEMOD/raised_cos4c.vhd)
            RAISED_COS_002 - RAISED_COS4C - RAISED_COS_arch (src/DSSS_DEMOD/raised_cos4c.vhd)
            CODE_REPLICA_GEN_001 - CODEGEN - behavior (src/DSSS_DEMOD/codegen.vhd)
            AGC3_001 - AGC3 - AGC3_ARCH (src/DSSS_DEMOD/agc3d.vhd)
                MULT001 - SIGNED_MULT12X9_10_S - BEHAVIOR (src/DSSS_DEMOD/signed_mult12x9_10s.vhd)
                    UNSIGNED_MULT - M11x8U_A - BEHAVIOR (src/DSSS_DEMOD/m11x8u_a.vhd)
            POLAR_001 - POLAR3 - Behavior (src/DSSS_DEMOD/POLAR3.vhd)
            CARRIER_TRACKING_002 - CARRIER_TRACKING - behavioral (src/DSSS_DEMOD/carrier_tracking.vhd)
            CTL_001 - CODE_TRACKING - CODE_TRACKING_ARCH (src/DSSS_DEMOD/code_tracking.vhd)
        USB20_001 - USB20
        DAC_002 - DAC - behavioral (src/DAC.vhd)
        COMSCOPE_001 - COMSCOPE - behavioral (src/COMSCOPE.VHD)
            TRACE_1_CAPTURE_8_BIT_WORDS_001 - CAPTURE_8_BIT_WORDS - behavioral (src/capture_8_bit_words.vhd)
            TRACE_2_CAPTURE_8_BIT_WORDS_001 - CAPTURE_8_BIT_WORDS - behavioral (src/capture_8_bit_words.vhd)
        COM1418.ucf (src/COM1418.ucf)
```

The code is stored with one, and only one, entity per file as shown above.
The root program (highlighted) is *com1418.vhd*.

VHDL Simulation

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

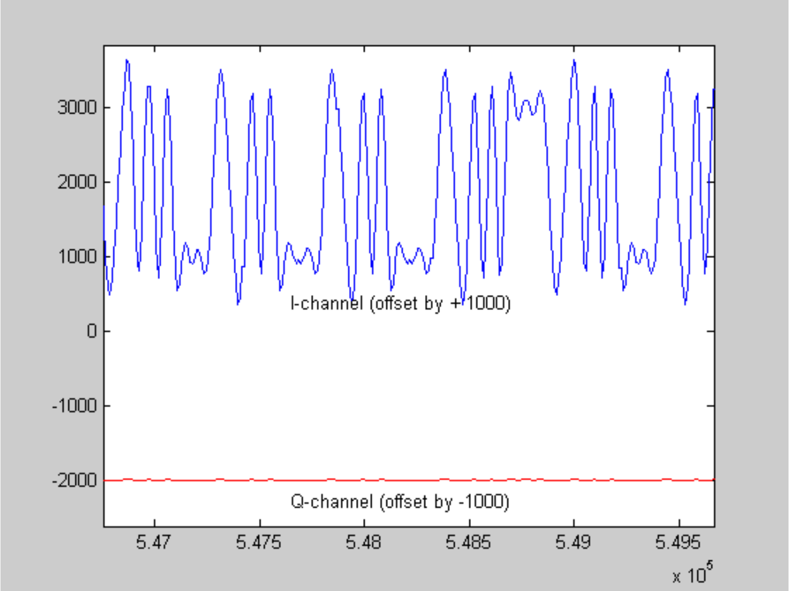
Signal Processing

In this section, we capture the key signals during VHDL simulation and plot them. The simulation is for back-to-back modulator-demodulator in ideal conditions (noiseless, no frequency offset, stable gain). The plots refer to the internal VHDL signal names. The internal clock is set at 100 MHz for simplicity (actual clock is 90 MHz).

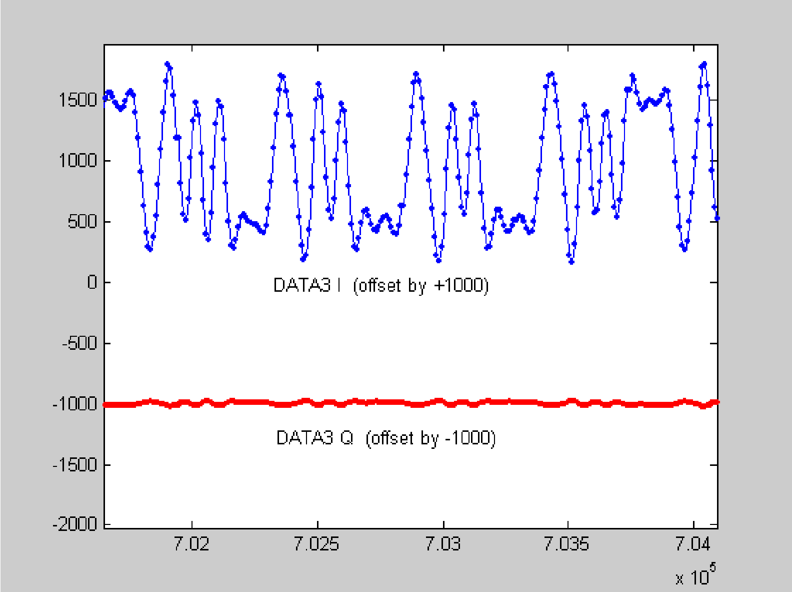
For this simulation, the DSSS modulator and demodulators are configured as 22 Mchips/s, Barker code, code length 13, PRBS-11 internal data source, BPSK modulation, noiseless, maximum amplitude, no frequency offset.

NCO_CHIP_RATE := x”3E93E93E”
SPREADING_FACTOR := x"000D"
CODE_SEL := "011"
NCO_CENTER_FREQUENCY := x"00000000"
SIGNAL_SCALING := "011111111"
AWGN_SCALING := "000000000"
MOD_CONTROL := x"0C93"
DEMOD_CONTROL := x"2000"

Input baseband samples DSSS_DEMOD/DATA_x_IN
Sampling rate: CLK_P

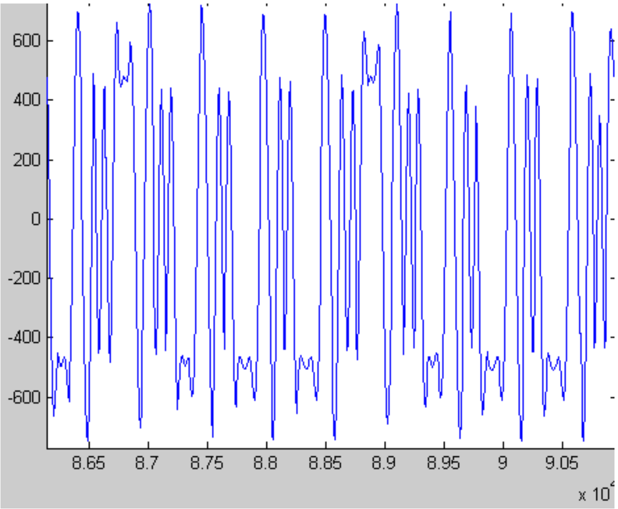


Baseband samples DSSS_DEMOD/DATA3_x after bias removal, frequency translation to baseband, x2 interpolation, decimation to 4 samples / chip and channel filtering in root raised cosine filter:

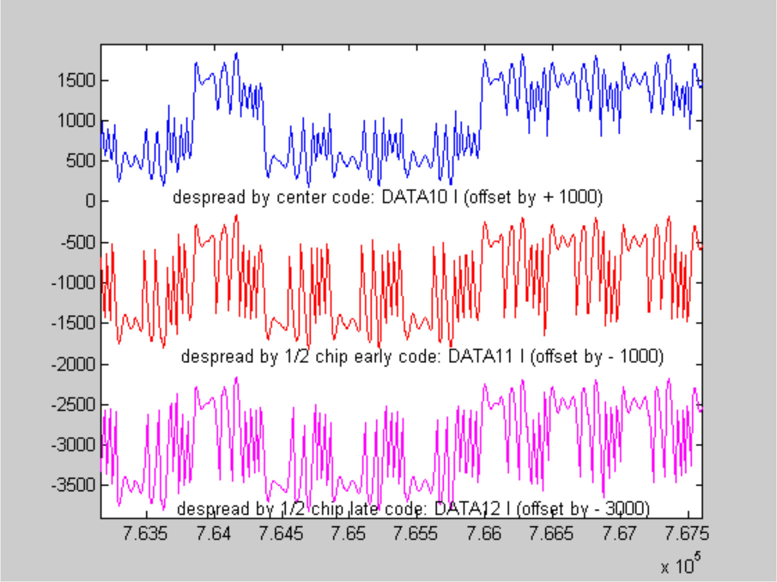


Baseband samples after phase/frequency compensation, root raised cosine filtering. The sampling rate is 4 samples / chip. Once the code tracking loop is locked, the amplitude at the center of chip is constant (amplitude

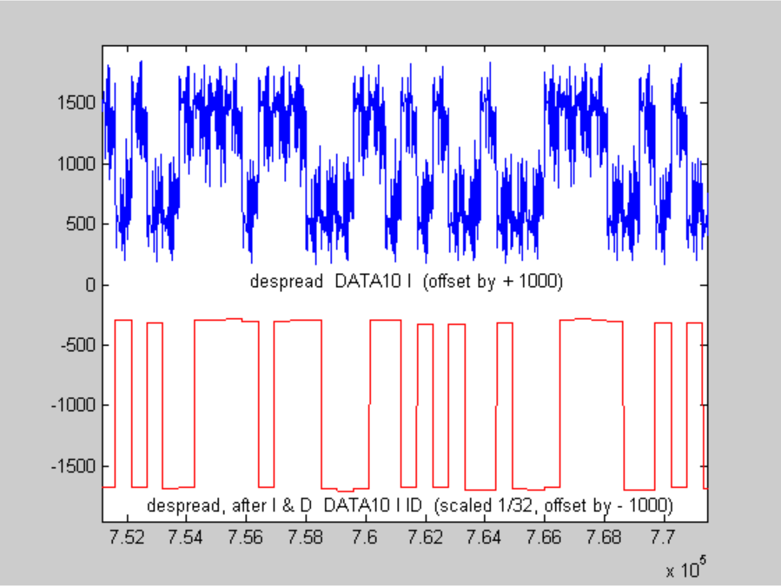
+500 or -500 approximately) as the combined transmit/receive filters do not theoretically cause any intersymbol interference. This is a plot of DEMOD/DATA3_I below:



After despreading with the code replica (center, early, late):



The despread signal is then subjected to low-pass filtering by an integrate and dump circuit synchronized with the symbol clock. The result is DEMOD/DATA10_I_ID:

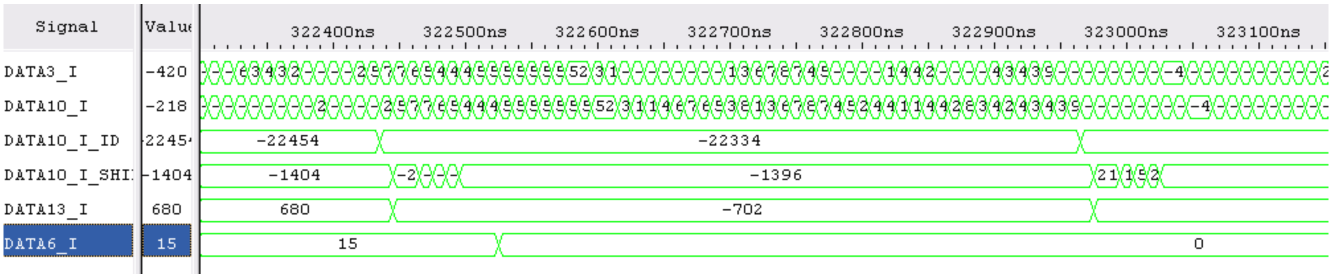


The demodulated signal is then normalized in two steps:

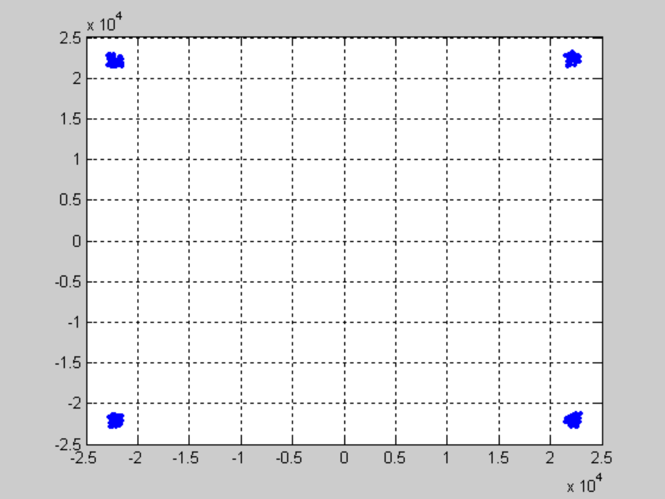
(a) a first coarse gain adjustment based on the spreading factor (by shifting DATA10_ID n times to the right, where 2^n-1 is the code length).

(b) the resulting DATA13_x complex sample is then converted to a phase DATA15_P by the *polar3* component which performs a simple cartesian to polar coordinates conversion. The resulting phase DATA15_P is used for extracting the demodulated data, the quality of the demodulated data (i.e. the distance to the nearest ideal phase constellation point) and the phase error for carrier tracking purposes.

Alternatively (both codes are included) DATA13_x undergoes fine gain adjustment through the AGC3 module. The 4-bit soft quantized output is named DATA6_I. The information bit is the most significant bit of DATA6_I. The lower three bits indicate the signal quality. Thus “0000” indicates a strong zero, “1111” a strong one, “0111” a weak zero, etc



Below is an example of DATA13_x in the case of QPSK modulation and noiseless channel:

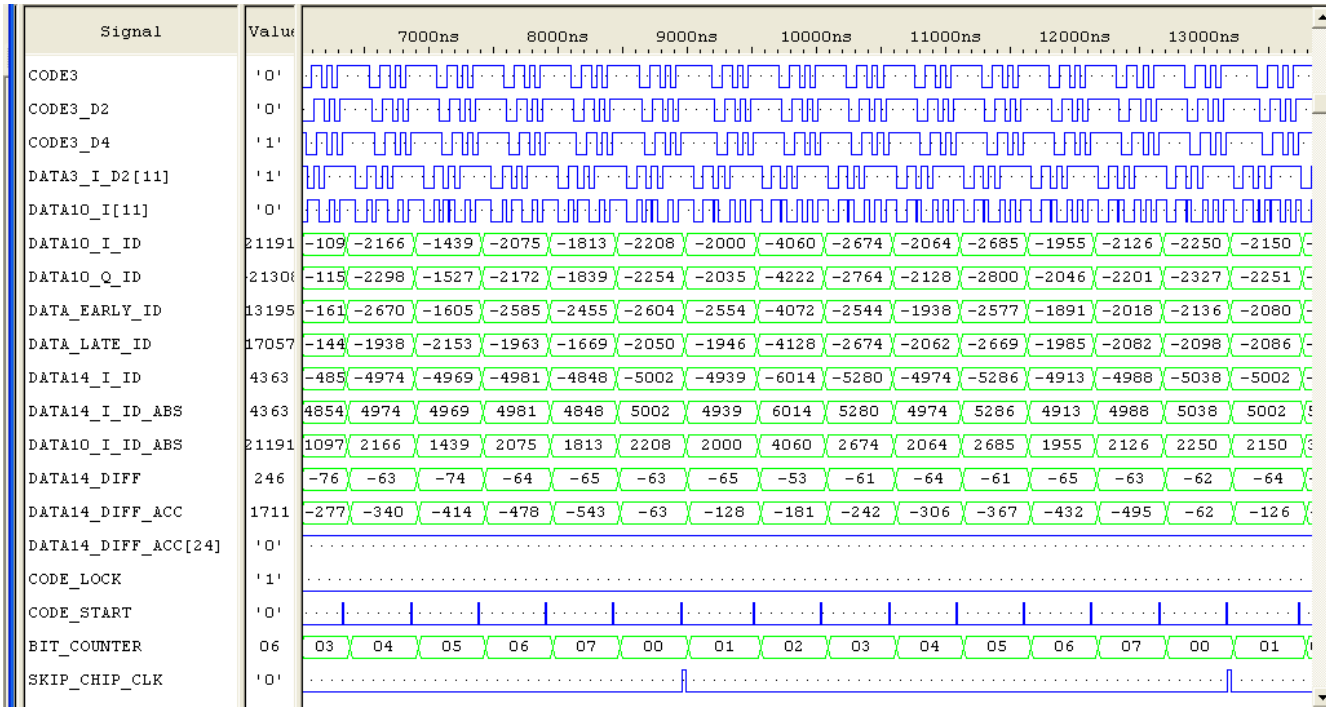


Code Acquisition

The code acquisition algorithm is a text-book sequential search: the code replica CODE3_D2 is shifted to the right by steps of ½ chip every N symbol periods, where the dwell time N is specified in control register REG18(6:4). This simulation specified a dwell time of 8 symbols.

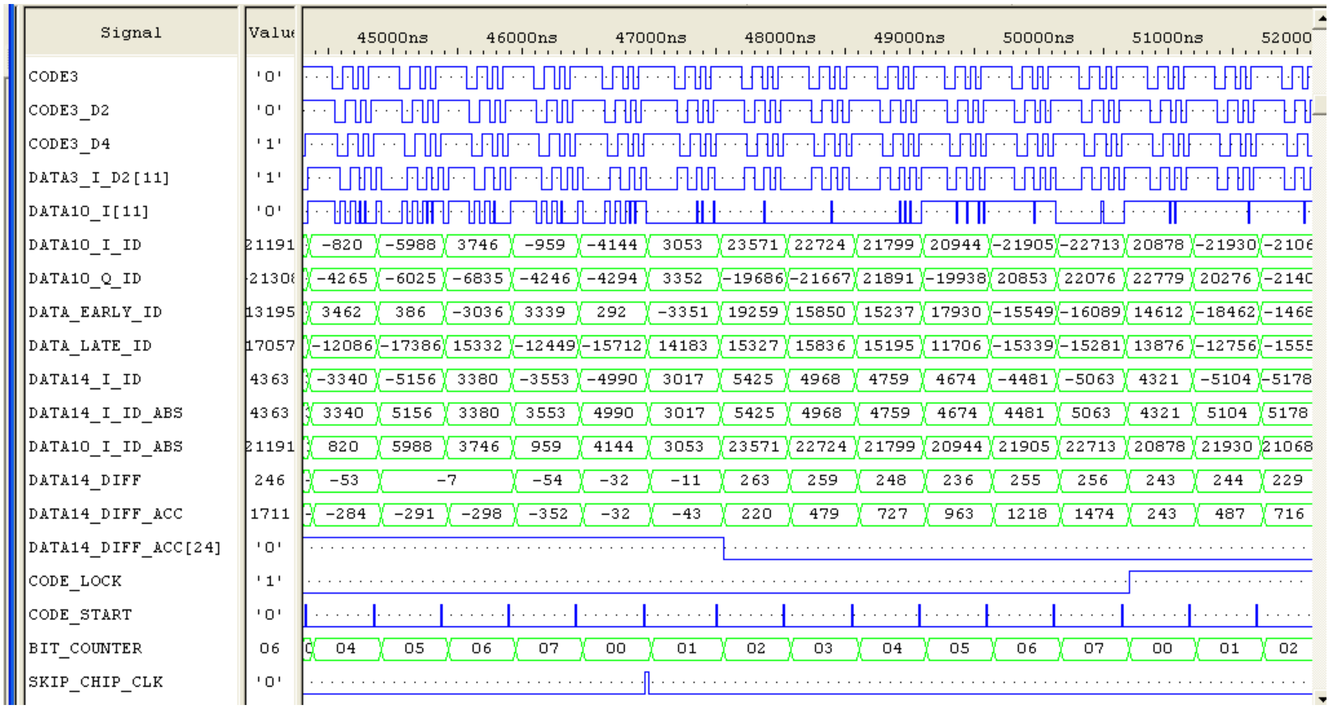
During the dwell time, the input signal DATA3_I is despread by the code replica CODE3_D2, resulting in DATA10_I. The despread signal is subsequently subject to an integrate and dump low-pass filter. The I&D output is DATA10_I_ID.

The salient signals are shown below while the demodulator is in code acquisition mode:



Signal	Value															
CODE3	'0'															
CODE3_D2	'0'															
CODE3_D4	'1'															
DATA3_I_D2[11]	'1'															
DATA10_I[11]	'0'															
DATA10_I_ID	21191	-109	-2166	-1439	-2075	-1813	-2208	-2000	-4060	-2674	-2064	-2685	-1955	-2126	-2250	-2150
DATA10_Q_ID	21308	-115	-2298	-1527	-2172	-1839	-2254	-2035	-4222	-2764	-2128	-2800	-2046	-2201	-2327	-2251
DATA_EARLY_ID	13195	-161	-2670	-1605	-2585	-2455	-2604	-2554	-4072	-2544	-1938	-2577	-1891	-2018	-2136	-2080
DATA_LATE_ID	17057	-144	-1938	-2153	-1963	-1669	-2050	-1946	-4128	-2674	-2062	-2669	-1985	-2082	-2098	-2086
DATA14_I_ID	4363	-485	-4974	-4969	-4981	-4848	-5002	-4939	-6014	-5280	-4974	-5286	-4913	-4988	-5038	-5002
DATA14_I_ID_ABS	4363	4854	4974	4969	4981	4848	5002	4939	6014	5280	4974	5286	4913	4988	5038	5002
DATA10_I_ID_ABS	21191	1097	2166	1439	2075	1813	2208	2000	4060	2674	2064	2685	1955	2126	2250	2150
DATA14_DIFF	246	-76	-63	-74	-64	-65	-63	-65	-53	-61	-64	-61	-65	-63	-62	-64
DATA14_DIFF_ACC	1711	-277	-340	-414	-478	-543	-63	-128	-181	-242	-306	-367	-432	-495	-62	-126
DATA14_DIFF_ACC[24]	'0'															
CODE_LOCK	'1'															
CODE_START	'0'															
BIT_COUNTER	06	03	04	05	06	07	00	01	02	03	04	05	06	07	00	01
SKIP_CHIP_CLK	'0'															

Code acquisition signals. Acquisition mode



Signal	Valu
CODE3	'0'
CODE3_D2	'0'
CODE3_D4	'1'
DATA3_I_D2[11]	'1'
DATA10_I[11]	'0'
DATA10_I_ID	21191
DATA10_Q_ID	-2130
DATA_EARLY_ID	13195
DATA_LATE_ID	17057
DATA14_I_ID	4363
DATA14_I_ID_ABS	4363
DATA10_I_ID_ABS	21191
DATA14_DIFF	246
DATA14_DIFF_ACC	1711
DATA14_DIFF_ACC[24]	'0'
CODE_LOCK	'1'
CODE_START	'0'
BIT_COUNTER	06
SKIP_CHIP_CLK	'0'

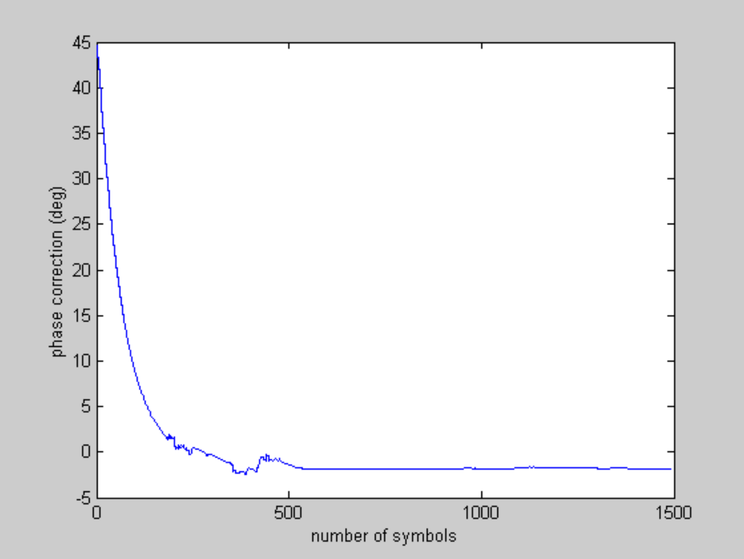
Code acquisition signals. At the time of acquisition.

The last code replica phase adjustment (SKIP_CHIP_CLK) results in a near-perfect alignment of the code replica (CODE3_D2) and the received signal DATA3_I_D2 spreading code. The despread signal DATA10_I is no longer noise-like. Consequently, the energy in the DATA10_I_ID_ABS received signal after dispreading/integrate & dump/ absolute value is greater than the noise-based variable threshold 1.25*DATA14_I_ID_ABS. This difference is integrated over the 8-bit integration period before confirming code lock (reason: the algorithm has to be sturdy in presence of noise).

Carrier Acquisition & Tracking

When the input signal is despread with the correct code replica (correct epoch), the resulting signal follows a simple BPSK or QPSK constellation. The *polar3* component performs a Cartesian to polar coordinate conversion. The output DATA15_P is the detected carrier phase for each symbol. The *carrier_tracking* component then computes the carrier NCO control signal based on a second order Costas loop (when tracking) or an AFC loop (during carrier acquisition).

The demodulator response to a 45 deg initial phase offset between the modulator and demodulator is show below. Displayed is CCF_PHASE, the phase correction at the *digital_dc2* frequency translator component.



The carrier lock status is determined after computing the standard deviation of the phase error. When the rms phase error, averaged over a window of 512 symbols, is less than 25 deg, the demodulator is deemed to track the carrier and the CARRIER_LOCK flag is set.

Quality

For each demodulated symbol one computes the quality of the demodulated information. For BPSK or QPSK, the quality is tied to the distance (phase error that is) between the demodulated symbol and the nearest nominal symbol in the PSK constellation.

The symbol quality is appended to each demodulated information bit to form a 4-bit soft quantized output.

FPGA Occupancy

```
Design Summary
--------------
Logic Utilization:
  Number of Slice Flip Flops:       3,973 out of   7,168   55%
  Number of 4 input LUTs:           4,680 out of   7,168   65%
Logic Distribution:
  Number of occupied Slices:                       3,582 out of   3,584   99%
    Number of Slices containing only related logic:   3,561 out of   3,582   99%
    Number of Slices containing unrelated logic:         21 out of   3,582    1%
      *See NOTES below for an explanation of the effects of unrelated logic
Total Number 4 input LUTs:          5,089 out of   7,168   70%
  Number used as logic:              4,680
  Number used as a route-thru:         343
  Number used as Shift registers:       66
  Number of bonded IOBs:              157 out of     173   90%
    IOB Flip Flops:                    49
    IOB Latches:                        7
  Number of Block RAMs:               10 out of      16   62%
  Number of MULT18X18s:                9 out of      16   56%
  Number of GCLKs:                     6 out of       8   75%
  Number of DCMs:                      2 out of       4   50%

Total equivalent gate count for design:  782,097
```

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