Overview

The COM-1518SOFT is a digital direct-sequence spread-spectrum demodulator written in VHDL, for intermediate frequency or baseband input signals. It is designed to be embodied within a single low-cost FPGA such as the Xilinx Spartan-6 LX45.

The entire VHDL source code is provided.

Key features and performance:

- Direct-Sequence Spread-Spectrum (DSSS) demodulation
- Continuous mode operation (i.e. Burst mode is not supported)
- Maximum processing gain: 33 dB
- Spreading factor: 3 to 2047
- Code period can be (significantly) longer than symbol period:
  - Maximal code period: 65535
- Maximum chip rate: 50% of processing clock frequency.
  - 78 Mchips/s Xilinx Spartan-6 -2
  - 99 Mchips/s Xilinx Kintex-7 -2
- Parallel code search for fast code acquisition. The number of parallel search circuits instantiated within can be selected by the user prior to synthesis.
- False code lock prevention.
- Accurate time of arrival pulse generated once per code period (can be used for round-trip delay measurement for example).
- Built-in Bit Error Rate measurement.

Target Hardware

The code is written in generic VHDL so that it can be ported to a variety of FPGAs. The code was developed and tested on a Xilinx Spartan-6 FPGA.

It can be easily ported to any Xilinx Kintex7, Virtex-6, Spartan-6, Virtex-5 FPGAs and other FPGAs.

Device Utilization

Device: Xilinx Spartan-6 XC6SLX45-2

<table>
<thead>
<tr>
<th>Resource</th>
<th>30 parallel code acquisition circuits</th>
<th>% utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flip Flops</td>
<td>9221</td>
<td>16%</td>
</tr>
<tr>
<td>LUTs</td>
<td>9414</td>
<td>34%</td>
</tr>
<tr>
<td>RAMB16BWERs</td>
<td>5</td>
<td>4%</td>
</tr>
<tr>
<td>DSP48E1s</td>
<td>20</td>
<td>34%</td>
</tr>
<tr>
<td>GCLKs</td>
<td>1</td>
<td>6%</td>
</tr>
<tr>
<td>DCMs/PLLs</td>
<td>0</td>
<td>0%</td>
</tr>
</tbody>
</table>

Device: Xilinx Kintex-7 XC7K70T-2

<table>
<thead>
<tr>
<th>Resource</th>
<th>30 parallel code acquisition circuits</th>
<th>% utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flip Flops</td>
<td>8999</td>
<td>10%</td>
</tr>
<tr>
<td>LUTs</td>
<td>12195</td>
<td>29%</td>
</tr>
<tr>
<td>RAMB16BWERs</td>
<td>4</td>
<td>2%</td>
</tr>
<tr>
<td>DSP48E1s</td>
<td>20</td>
<td>8%</td>
</tr>
<tr>
<td>GCLKs</td>
<td>1</td>
<td>6%</td>
</tr>
<tr>
<td>DCMs/PLLs</td>
<td>0</td>
<td>0%</td>
</tr>
</tbody>
</table>

Clock speed

This design uses a single global clock, namely the ADC sampling clock. Typical maximum clock frequencies for various FPGA families are listed below:

<table>
<thead>
<tr>
<th>Device family</th>
<th>f_{clk_{rx}}</th>
<th>Max. chip rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xilinx Kintex 7 -2</td>
<td>198 MHz</td>
<td>99 Mchips/s</td>
</tr>
<tr>
<td>Xilinx Virtex-6 -2</td>
<td>156 MHz</td>
<td>78 Mchips/s</td>
</tr>
<tr>
<td>Xilinx Spartan-6 -3</td>
<td>177 MHz</td>
<td>88 Mchips/s</td>
</tr>
</tbody>
</table>
This component’s interface comprises four distinct groups:

(a) the receiver interface typically connects to one (IF) or two (baseband) analog-to-digital converters (ADC). Use both I and Q inputs in the case of near-zero center frequency signal. Use the I input only in the case of IF input signal, while the Q input is set to zero.

(b) The demodulator output consists of 4-bit soft-quantized output. The most significant bit DATA_OUT(3) represents the demodulated bit, while the remaining bits DATA_OUT(2:0) rate its quality.

(c) Controls can be changed at any time. In most cases, the component should be reset with a SYNC_RESET pulse after a configuration change.

(d) Monitoring
**Monitoring and Control**

**Configuration**

The key configuration parameters are brought to the interface so that the user can change them dynamically at run-time. Other, more arcane, parameters are fixed at the time of VHDL synthesis.

**Pre-synthesis configuration parameters**

The following configuration parameters are set as constants prior to synthesis:

<table>
<thead>
<tr>
<th>Configuration parameters in DSSS_DEMOD2</th>
<th>Description</th>
</tr>
</thead>
</table>
| Number of parallel code acquisition circuits | NACQ  
The larger this number, the faster the code acquisition time. The FPGA occupancy increases significantly as well.  
Limitation: NACQ must be selected to be less than half the spreading factor. |
| Non-coherent Integration and Dump (I&D) period | N_NCID  
Number of successive coherent I&D results accumulated non-coherently (power-wise) to improve the SNR before making a detection decision.  
Trade-off acquisition speed versus acquisition SNR threshold. |

**Run-time configuration parameters**

The user can set and modify the following controls at run-time. All controls are synchronous with the user-supplied global CLK. Most configuration changes should be followed by a SYNC_RESET pulse.

The FPGA processing clock equals the ADC sampling clock: \( f_{\text{clk}_{\text{rx}}} \)

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Configuration</th>
</tr>
</thead>
</table>
| 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 chip rate). The AGC_DAC gain control signal is updated as follows  
0 = every chip,  
1 = every 2 input chips,  
2 = every 4 input chips,  
3 = every 8 input chips, etc….  
10 = every 1000 input chips.  
Valid range 0 to 14. |
| Nominal Center frequency \( f_{f_{\text{IF}}} \) | The nominal center frequency can be null (in the case of a baseband input signal) or non-zero in the case of an IF input signal. If the IF center frequency is sufficiently greater than the modulation bandwidth (chip rate), the Q input can be ignored and force to zero, thus saving an ADC.  
This field can also be used for fine frequency corrections, for example to correct clock drifts.  
32-bit signed integer (2’s complement representation) expressed as  
\( f_{f_{\text{IF}}} \cdot 2^{32} / f_{\text{clk}_{\text{rx}}} \) |
| Chip rate \( f_{\text{chip rate}} \) | 32-bit integer expressed as  
\( f_{\text{chip rate}} \cdot 2^{32} / f_{\text{clk}_{\text{rx}}} \).  
The maximum practical chip rate is \( f_{\text{clk}_{\text{rx}}} / 2 \).  
Example for a 150 MHz \( f_{\text{clk}_{\text{rx}}} \):  
60 Mchips/s: 0x66666666  
The maximum allowed error between transmitted and received chip rate is +/- 100ppm. |

**RECEIVER_CENTER_FREQ(31:0)**

**NOMINAL_CHIP_RATE(31:0)**
| Code period | In chips.  
Valid range 3 – 65535  
Can be less than the natural length of the selected code. In which case, the code is truncated.  
CODE_PERIOD(31:0) |
| --- | --- |
| Code selection | 1 = Gold code  
2 = Maximal length sequence  
3 = Barker code (lengths 11 or 13 only)  
4 = GPS C/A codes  
(Case G2 as GPS PRN number)  
CODE_SEL(2:0) |
| Gold sequence / Maximal Length Sequence generator polynomial G1 | 24-bit. Describes the taps in the linear feedback shift register 1:  
Bit 0 is the leftmost tap (2^n in the polynomial). The largest non-zero bit is the polynomial order. n determines the code period 2^n – 1.  
Example:  
G1 = 1 + x + x^4 + x^5 + x^6  
is represented as 0x000039  
This field is used only if Gold code or Maximal length sequences are selected.  
G1(23:0) |
| Gold code generator polynomial G2 | 24-bit. Describes the taps in the linear feedback shift register 2: Same format as G1 above.  
This field is used only if Gold codes are selected.  
G2(23:0) |
| Gold code G1/G2 phase offset | A Gold code is generated by adding two maximal length sequences (as defined by their generator polynomials G1 and G2). A set of orthogonal Gold codes can be created by changing the phase offset between the two maximal length sequences.  
G2OFFSET(23:0) |
| GPS satellite ID | GPS signals from different satellites are designated by a PRN signal number in the range 1 – 37.  
This field is used only if GPS C/A codes are selected.  
G2(5:0) |
| Symbol rate f_{symbol_rate} | The symbol rate can be set independently of the spreading code period as  
f_{symbol_rate} = 2^{32} / f_{clk_tx}  
Limitation: the symbol rate must be higher than chip rate / 2047.  
NOMINAL_SYMBOL_RATE(31:0) |
| Spreading factor (Processing gain) | Approximate (i.e truncated) ratio of chip rate / symbol rate  
Range: 3 – 2047  
Note: to effectively achieve this processing gain, the code period must be longer than one symbol duration.  
SPREADING_FACTOR(10:0) |
| Spectrum inversion | Invert Q bit. (Inverts the modulated spectrum only, not the subsequent frequency translation)  
0 = off  
1 = on  
DEMOD_CONTROL(0) |
| BPSK / QPSK decoding | Note: the modulation symbol transitions are not necessarily aligned with the chip transitions.  
0 = BPSK  
1 = QPSK  
DEMOD_CONTROL(1) |
| Carrier frequency loop gain | 00 = nominal  
01 = 2x loop gain  
10 = 4x loop gain  
11 = 8x loop gain  
DEMOD_CONTROL(3:2) |
| AFC enable | The automatic frequency control circuit extends the frequency acquisition over +/- 10% of the symbol rate. When disabled, the receiver only means of carrier acquisition is the carrier frequency tracking loop which is inherently limited to approximately 1% of the symbol rate. The AFC should only be active during acquisition as it interferes with the Costas Loop operation.  
00 = automatic AFC selection.  
01 = force AFC disabled. Carrier tracking loop only  
10 = force AFC enabled.  
11 = reserved (test).  
DEMOD_CONTROL(5:4) |
| Internal vs external AGC | The code can act as a level sensor for an external gain control actuator (for example RF or IF receiver gain control) to prevent saturation at the external A/D converter. If so, configure as “external” AGC.  
Otherwise, an internal AGC will prevent saturation from occurring within the FPGA.  
0 = internal  
1 = external  
DEMOD_CONTROL(6) |
The built-in PSK demodulator exhibits an inherent phase ambiguity (modulo 180 deg for BPSK, 90 deg for QPSK). Subsequent circuits such as FEC decoders can detect when the carrier tracking loop is locked onto the wrong carrier phase. By generating a 1 CLK-wide pulse, this demodulator will jump 180 deg (BPSK) or 90 deg (QPSK) to the next modulo.

### Monitoring

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carrier lock</td>
<td>Carrier tracking loop status, based on the measured RMS phase error. May wrongly indicate lock when no input signal and no input noise is present.</td>
</tr>
<tr>
<td>Code lock</td>
<td>CODE LOCK_OUT</td>
</tr>
<tr>
<td>Carrier frequency error</td>
<td>Measured frequency error relative to the nominal center frequency. 20-bit signed integer (2’s complement representation) expressed as $f_{\text{error}} \times 2^{20} / f_{\text{symbol rate}}$.</td>
</tr>
<tr>
<td></td>
<td>CARRIER FREQUENCY ERROR(19:0)</td>
</tr>
<tr>
<td>Bit error rate</td>
<td>Monitors the BER (number of bit errors over 1,000,000 received bits) when the modulator is sending a PRBS-11 test sequence. Valid only when BER analyzer is synchronized.</td>
</tr>
<tr>
<td></td>
<td>BER ERROR_COUNT(31:0)</td>
</tr>
<tr>
<td>BER analyzer synchronized</td>
<td>‘1’ when synchronized with the received PRBS-11 test sequence.</td>
</tr>
<tr>
<td></td>
<td>BER SYNC</td>
</tr>
<tr>
<td>Byte error</td>
<td>One pulse for each byte error. Helpful in visualizing the bit error distribution (random? Burst?) with an oscilloscope.</td>
</tr>
<tr>
<td></td>
<td>BYTE_ERROR</td>
</tr>
</tbody>
</table>
The ADC samples are first processed in the `RECEIVER1.vhd` component as illustrated above.

The AGC detects saturation or near saturation and adjust the gain control signal accordingly. This gain control is sent to either the internal or external gain control stage as per the configuration. The reaction time of the AGC loop is controlled by the `AGC_RESPONSE` parameter.

After DC bias removal, the real (IF) or complex (near baseband) input samples are frequency translated to baseband, as per the `RECEIVER_CENTER_FREQ` parameter.

The resulting baseband signal undergoes up to two types of low-pass filtering (CIC decimation and Half-Band filter), depending on the modulation bandwidth (or more specifically, depending on the ratio of the sampling rate over the chip rate). These low-pass filters are used for (a) image-rejection, in the case of IF inputs, (b) anti-aliasing prior to decimation.

Resulting baseband complex samples are further sent to the `DSSS_DEMOD2.vhd` component, as illustrated below.
The *DSSS_DEMOD2.vhd* component conceptually comprises two distinct and mostly independent sections: (a) a spread-spectrum receiver (code acquisition and despreading) followed by (b) a regular PSK demodulator. Architecturally, this means that the PSK modulation could be replaced with other modulation types without much effort.

**NACQ** parallel code acquisition circuits are used for faster code acquisition. Each circuit consists of a code replica delay, despreading, coherent integration and dump (I&D) over a period of approximately half a symbol period, and non-coherent (i.e. power) I&D over a period of approximately $\frac{N_{NCID}}{2}$ symbols.

During the code acquisition phase, these parallel circuits are used to search **NACQ** code epochs spaced $\frac{1}{2}$ a chip apart.

During the code tracking phase, three parallel circuits are used to compare the early/on-time/late code epochs for code tracking. The remaining parallel circuits are used to search for false code lock (a consistently better code epoch).
Software Licensing

The COM-1518SOFT 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 bitstream 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 1.

<table>
<thead>
<tr>
<th>Directory</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>/doc</td>
<td>Specifications, user manual, implementation documents</td>
</tr>
<tr>
<td>/src</td>
<td>.vhd source code, .ucf constraint files One component per file.</td>
</tr>
<tr>
<td>/sim</td>
<td>Test benches, Matlab .m signal generation</td>
</tr>
<tr>
<td>/bin</td>
<td>n/a</td>
</tr>
</tbody>
</table>

Key files:
Xilinx ISE project file: com-1518_ISE141.xise

VHDL development environment

The VHDL software was developed using the following development environment:

(a) Xilinx ISE 14.1 with XST as synthesis tool
(b) Xilinx ISE Isim as VHDL simulation tool

The entire project fits within a Xilinx Spartan-6 LX45. Therefore, the ISE project can be processed using the free Xilinx WebPack tools.
Xilinx-specific code
The VHDL source code was written in generic VHDL with few low-level Xilinx primitives. No Xilinx CORE is used. The Xilinx primitives are:
- RAMB16_Sx_Sx

VHDL software hierarchy

Comparison with Previous Version

<table>
<thead>
<tr>
<th>Key Improvements with respect to COM-1418 Direct-Sequence Spread-Spectrum Demodulator</th>
</tr>
</thead>
<tbody>
<tr>
<td>- 2x faster: maximum chip rate is fclk/2</td>
</tr>
<tr>
<td>- Faster acquisition: parallel acquisition instead of sequential search.</td>
</tr>
<tr>
<td>- Symbol duration and alignment are independent of the spreading code period (2 independent tracking loops for code and symbol timing)</td>
</tr>
<tr>
<td>- Better performance through reduced dependencies between loops: code acquisition is less dependent on center frequency error.</td>
</tr>
<tr>
<td>- Independent AGCs before and after despreading.</td>
</tr>
<tr>
<td>- Phase ambiguity resolution under control of external FEC decoder.</td>
</tr>
</tbody>
</table>

ComBlock Ordering Information

COM-1518SOFT  DSSS DEMODULATOR

ECCN: 5E001.b.4

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

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