Key Design Features

- Synthesizable, technology independent IP Core for FPGA, ASIC and SoC
- Supplied as human readable VHDL (or Verilog) source code
- 16-bit signed input data samples
- Automatic carrier acquisition with no complex setup required
- User specified carrier frequency, symbol rate and sample rate
- Practical symbol rates of up to 10 Mbits/s
- Typical FPGA sample rates of up to 200 MHz

Applications

- Software radio
- Short to medium-range telemetry
- IF, SRD and ISM band devices
- Robust, low bandwidth radio applications for small FPGA devices
- Low-cost radio links over a few 100 meters using either wireless or cable - e.g. coax or twisted pair
- Applications where fast carrier acquisition is essential - e.g. where data packets are transmitted in short discrete bursts

Generic Parameters

Generic name	*Description*	*Type*	*Valid range*
sym_period	Symbol period in sample clocks	integer	0 to 65535
sym_polarity	Swaps symbol polarity i.e. 1 ↔ 0	boolean	TRUE / FALSE

Pin-out Description

Pin name	*I/O*	*Description*	*Active state*
clk	in	Sample clock	rising edge
reset	in	Asynchronous reset	low
en	in	Clock enable	high
x_in [15:0]	in	BPSK 16-bit signed input	data
mag_thresh [15:0]	in	Bit decision level threshold for a 0 /1 (16-bit unsigned)	data
mag_out [15:0]	out	Magnitude of symbol values at the decoder (16-bit signed)	data
psk_val	out	PSK bit valid strobe	high
psk_out	out	PSK bit out	data

Block Diagram

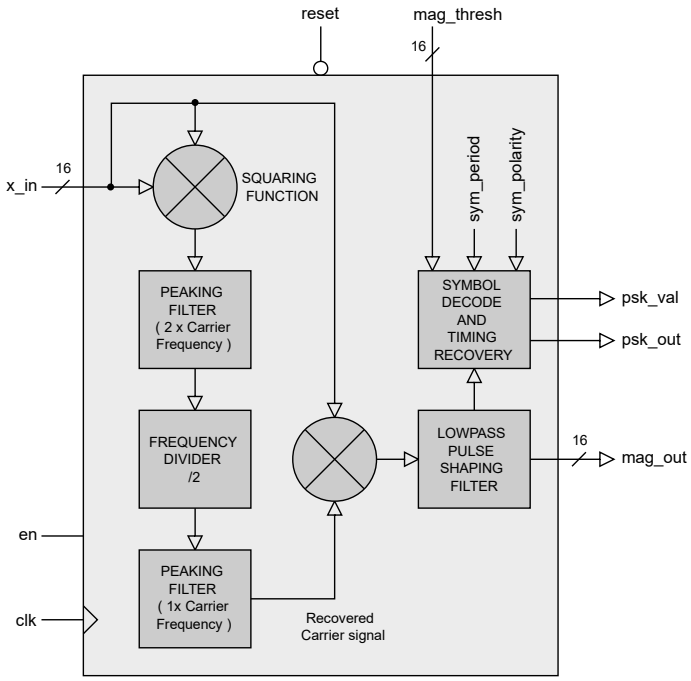

Figure 1: Simplified BPSK Demodulator architecture

General Description

The BPSK_DEMOD IP Core (Figure 1) is a 16-bit resolution Binary-PSK demodulator based on a multiply-filter-divide architecture. The design is robust and flexible and allows easy connectivity to an external ADC.

As the carrier recovery circuit is open-loop, there is no feedback path or loop-filter to configure. This results in an extremely simple circuit with a very fast carrier acquisition time. The only requirement is that the user must set the desired symbol period and a suitable threshold level for the bit decisions at the symbol decoder. The other design parameters including carrier frequency, symbol rate and sampling frequency should be specified by the user before delivery of the IP Core[1].

The input data samples are 16-bit signed (2's complement) values that are synchronous with the system/sample clock. Input values are sampled on the rising edge of *clk* when *en* is high.

Figure 1 shows the basic architecture in more detail. The input signal is first squared in order to generate a harmonic at twice the carrier frequency and zero phase-shift. This squared signal is then filtered and divided in frequency to recover the original carrier. A second filter is employed to isolate a clean carrier signal which is used to demodulate the original input signal. The demodulated input signal then goes through a low-pass pulse-shaping filter before a bit-decision is made at the symbol decoder.

The demodulated BPSK bit-stream appears at the output *psk_out*. Bits are valid on the rising edge of *clk* when both *psk_val* and *en* are high.

1 Please contact Zipcores first to discuss your design parameters. We can also provide a subset of programmable design parameters if necessary.

Peaking Filters

The signal path is filtered using a series of peaking filters in order to recover the carrier signal. These filters are precision 2ND order IIR filters with fully configurable 16-bit coefficients. The coefficients are specified at compile time and should be set correctly for the chosen sample rate and carrier frequency. The characteristics of these filters will largely determine how fast the carrier signal is acquired and how the system responds to jitter and frequency drift between the transmitter and receiver clocks.

Figures 2 and 3 below show some example filter responses for a system with a carrier frequency set to 5 MHz. The first peaking filter isolates the 10 MHz tone after squaring. The second peaking filter is used to recover the 5 MHz carrier signal.

In both cases (Figure 2 and 3) the sample frequency is set to 125 MHz.

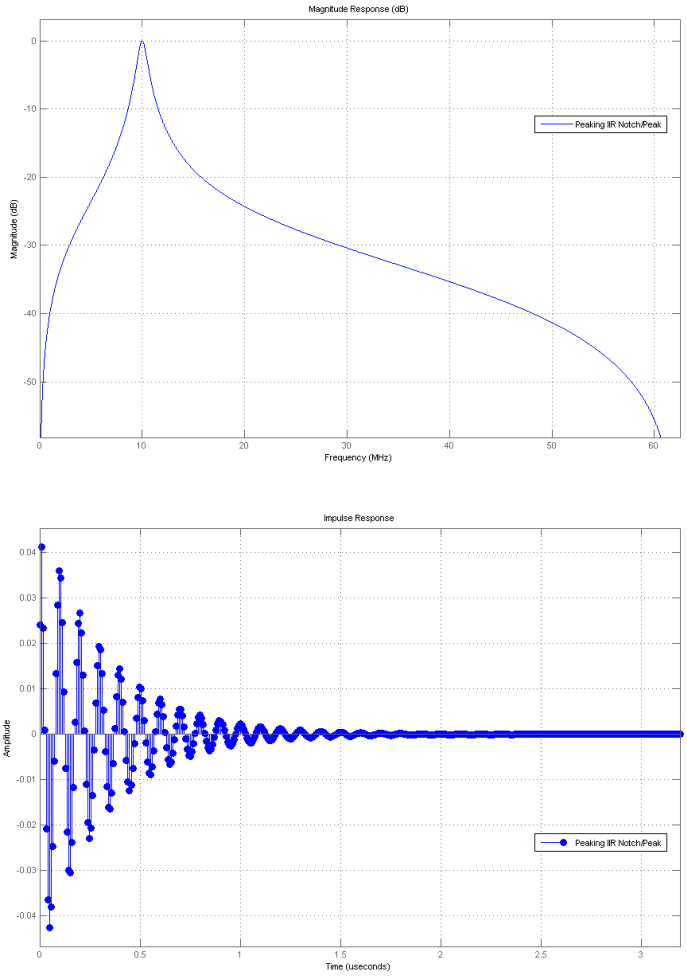

Figure 2: Magnitude and Impulse responses for a 10 MHz peaking filter (2 x carrier frequency)

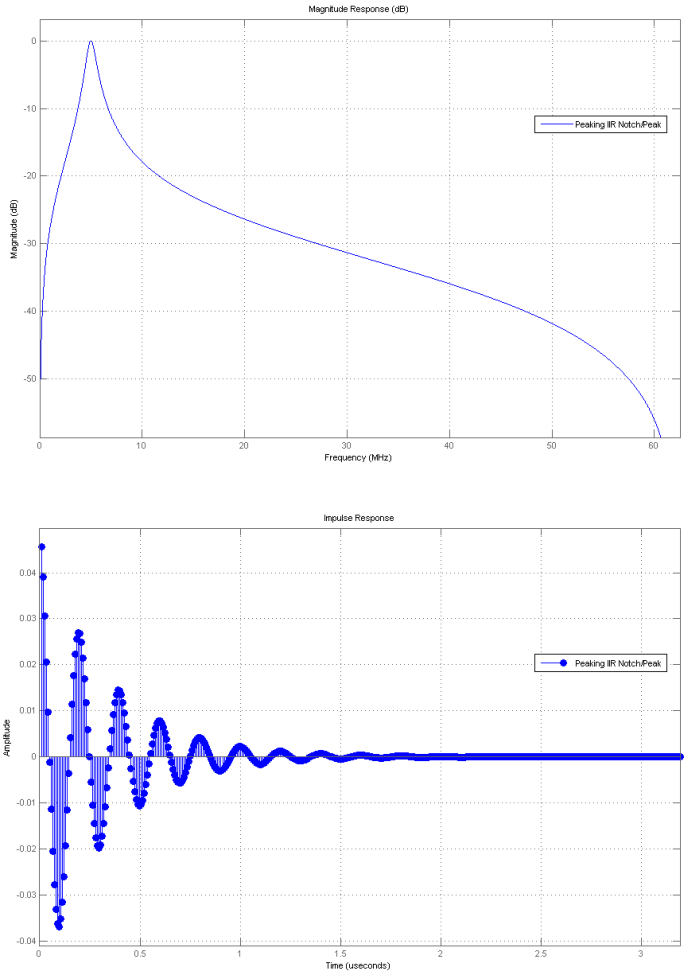

Figure 3: Magnitude and Impulse responses for a 5 MHz peaking filter (1 x carrier frequency)

Low-pass Pulse-shaping Filter

After demodulation, the signal is low-pass filtered using a raised-cosine FIR filter. This filter serves to remove any unwanted frequency components above the symbol frequency. The FIR filter is also acts as a pulse-shaping filter with the impulse response *exactly* tuned to the symbol period[2]. In this way, the recovered symbols are extremely clean and exhibit a signal-to-noise ratio of up to 80 dBs in simulation.

As an example, the following plot in Figure 4 shows the magnitude and impulse responses for an FIR raised-cosine filter with the symbol period tuned to 0.8 us (symbol rate 1.25 Mbps).

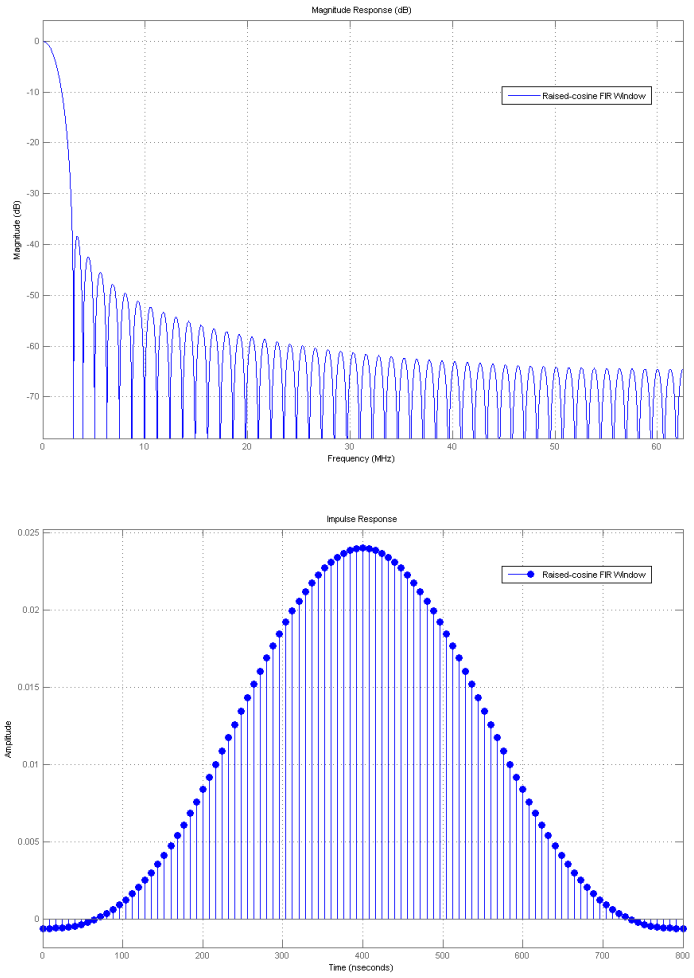

Figure 4: Magnitude and Impulse responses for a raised cosine filter with the symbol period set to 0.8us

2 Please contact Zipcores to discuss your desired symbol rate, carrier frequency and system sample rate so we may characterize the filters correctly for your application.

Symbol rate, Carrier frequency and Sample rate

In order to recover the phase information correctly from the modulated signal, there must be at least one full wave of the carrier signal per symbol. Figure 5 shows this relationship pictorially.

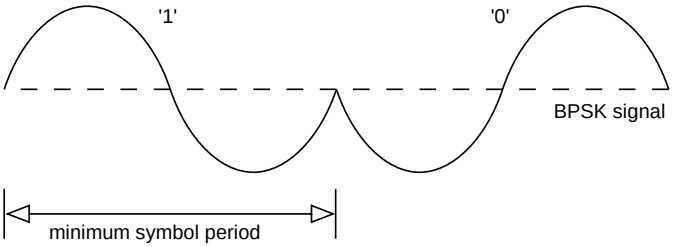

Figure 5: Relationship between carrier sinusoid and the symbol period

In other words, the symbol period must be no smaller than one complete cycle of the carrier signal. As a general rule, the maximum symbol period is given by:

$$Symbol\ period \geq 1 \,/\, Carrier\ frequency$$

For example. If the carrier frequency is set to 2 MHz, then the BPSK symbol period should be set to 0.5 us or greater. This equates to a symbol rate of 2 Mbps or less.

In addition, it is observed by experiment that the most reliable results are achieved when the system sampling frequency is at least 10 times the carrier frequency of the BPSK source signal. This ensures there are enough samples for the clean recovery of the squared carrier signal.

$$Sampling\ frequency \geq Carrier\ frequency * 10$$

Symbol decoding and Timing recovery

The symbol decoder block extracts the symbol timing information and symbol values from the received BPSK signal. In order for the symbol decoder to function correctly, the design parameters *sym_period* and *mag_threshold* must be set appropriately. Note that the symbol period is set via a fixed generic parameter at compile time. The threshold decision level is programmable and may be modified during operation.

The symbol period is specified as an integer number of clock cycles for the chosen sampling frequency. The threshold is a relative amplitude, and is used to determine the presence of a symbol (0 or 1) at the decoder. Decreasing the threshold increases the sensitivity of the decoder. Increasing the threshold decreases sensitivity.

Setting the threshold too high or too low may result in incorrect bit decisions. The best threshold level is dependent on a large number of system variables and is best determined by experiment. In order to help with this, the output signal *mag_out* may be examined. The *mag_out* signal is a signed 16-bit value that comes directly from the output of the pulse shaping filter (shown graphically in Figure 6).

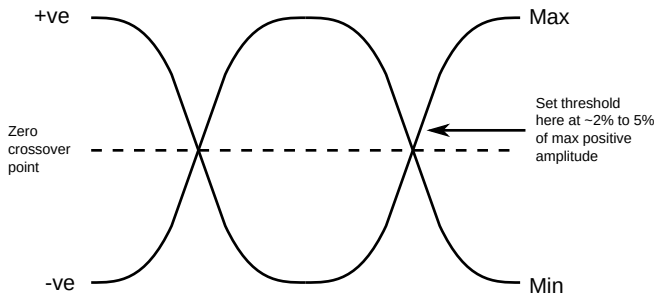

Figure 6: Symbol decoder bit threshold measurement and calculation

By experiment it has been found that setting the bit decision threshold to between 2% and 5% of the average maximum *positive* amplitude gives the best results. For instance if the average max and min values of *mag_out* are +5200 and -5100 respectively, then threshold should be set to +104 for the 2% level.

In addition, the *mag_out* signal may also be used to plot a precise eye-diagram of the symbol out of the pulse shaping filter. Figure 7 shows an example eye-diagram plot with the BPSK demodulator set up for a 5 MHz carrier frequency, 125 MHz sample rate and 0.8 us symbol period. All values were captured in simulation and are relative.

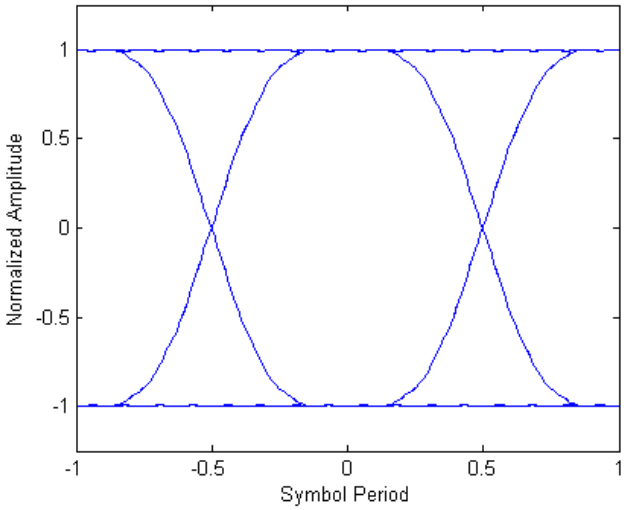

Figure 7: Example eye-diagram plot for a complete symbol as captured from simulation

Functional Timing

Figure 8 shows the operation of the BPSK demodulator during normal operation. The clock-enable signal has been de-asserted for one clock cycle to demonstrate what happens when there's a stall in the pipeline.

All inputs and outputs are sampled on the rising edge of *clk* when *en* is high. The demodulated output bits *psk_out* are valid when *psk_val* is high.

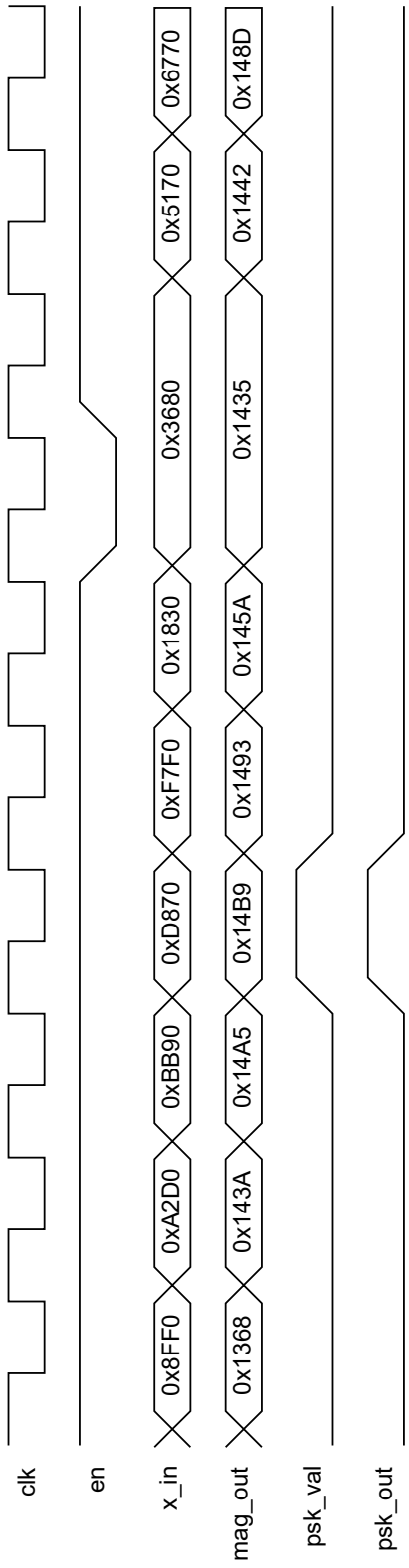

Figure 8: BPSK demodulator timing waveforms

Source File Description

All source files are provided as text files coded in VHDL. The following table gives a brief description of each file.

Source file	*Description*
sincos16.vhd	SIN/COS look-up table
dds16.vhd	16-bit DDS component
iir_biquad.vhd	Basic IIR filter component
iir_peaking.vhd	IIR peaking filter
fir_shaping_pack.vhd	FIR pulse-shaping filter package
fir_shaping_madl.vhd	FIR filter mult-add block odd tap
fir_shaping_mad.vhd	FIR filter mult-add block
fir_shaping.vhd	FIR pulse-shaping filter top
bpsk_sym_dec.vhd	Symbol decoder
bpsk_sym_gen.vhd	Random symbol generator
bpsk_demod.vhd	Top-level component
bpsk_demod_bench.vhd	Top-level test bench

Functional Testing

An example VHDL testbench is provided for use in a suitable VHDL simulator. The compilation order of the source code is as follows:

1. sincos16.vhd
2. dds16.vhd
3. iir_biquad.vhd
4. iir_peaking.vhd
5. fir_shaping_pack.vhd
6. fir_shaping_madl.vhd
7. fir_shaping_mad.vhd
8. fir_shaping.vhd
9. bpsk_sym_dec.vhd
10. bpsk_sym_gen.vhd
11. bpsk_demod.vhd
12. bpsk_demod_bench.vhd

The VHDL testbench instantiates the demodulator component and also a separate DDS that is used to generate a BPSK source signal.

In the example test provided, the system clock period is set to 125 MHz which is the sampling frequency for the simulation. The carrier frequency is set to 5 MHz and the symbol rate is set to 1.25 Mbps. This represents a symbol period of 0.8 us or 100 sample clock cycles.

During the course of the test, the component 'bpsk_sym_gen.vhd' generates a randomized sequence of 1's and 0's which are used to modulate the 5 MHz carrier. The simulation must be run for at least 5 ms during which time the input bit stream and demodulated output bit stream are captured in the files *bpsk_demod_in.txt* and *bpsk_demod_out.txt*. These two files may be compared to verify that the bits have been demodulated correctly.

In addition, the output magnitudes at the symbol sample points are captured in the file *bpsk_demod_mag.txt*. The *bpsk_demod_mag_eye.txt* file captures the output magnitudes at every sampling clock edge. This file is useful for plotting a precise eye diagram of the symbols out of the pulse-shaping filter as per Figure 7.

Synthesis and Implementation

The files required for synthesis and the design hierarchy is shown below: There are no special physical synthesis constraints required for implementation.

- bpsk_demod.vhd
  - iir_peaking.vhd
    - iir_biquad.vhd
  - fir_shaping_pack.vhd
  - fir_shaping.vhd
    - fir_shaping_madl.vhd
    - fir_shaping_mad.vhd
  - bpsk_sym_dec.vhd

The VHDL IP Core is designed to be technology independent. However, as a benchmark, synthesis results have been provided for the Xilinx® 7-series FPGAs. Synthesis results for other FPGAs and technologies can be provided on request.

Note that depending on the exact specification of the internal filters, the total number of hardware multipliers in the design may be slightly higher or lower for different implementations. In addition, the type of multipliers used in the design may (typically) be controlled by the use of compiler directives in the source code. For instance, the attribute 'use_dsp' may be used in Xilinx implementations to control the use of DSP hardware blocks.

The IIR peaking filters benefit more from the DSP resources as they are more timing critical. The FIR pulse-shaping filter benefits from using simple shift/add logic as many logic optimizations can be made during synthesis.

Trial synthesis results are shown with the generic parameters set to: sym_period = 100 and sym_polarity = true. The table below shows resource usage after place and route of the design.

XILINX® 7-SERIES FPGAS

Resource type	*Artix-7*	*Kintex-7*	*Virtex-7*
Slice Register	2682	2682	2681
Slice LUTs	1675	1647	1653
Block RAM	0	0	0
DSP48	65	65	65
Occupied Slices	787	904	807
Clock freq. (approx)	150 MHz	200 MHz	250 MHz

Revision History

Revision	*Change description*	*Date*
1.0	Initial revision	15/09/2011
1.1	Modified synthesis results in line with minor source-code changes	03/10/2011
1.2	Added dev-board testing description including scope traces	29/02/2012
1.3	Added symbol polarity generic and optimized the IIR filters for speed	26/02/2015
2.0	Major architecture revision. New design has programmable bit threshold and high performance pulse-shaping filters	09/04/2019