

Key Design Features

- Synthesizable, technology independent VHDL Core
- 16-bit signed input data samples
- Accepts either complex or real inputs
- Programmable mark / space frequencies
- Choice of low pass filter responses
- Carrier separation ~ symbol rate or greater
- Practical symbol rates of up to 1 Mb/s¹
- Baseband or passband operation
- Typical FPGA sample rates 150 MHz or less²
- Compatible with a wide range of third party ICs such as quadrature down-converters, mixers and ADCs

Applications

- Software-based radio
- Short range telemetry
- SRD and ISM band devices
- Low cost RF applications for small FPGA devices

Block Diagram

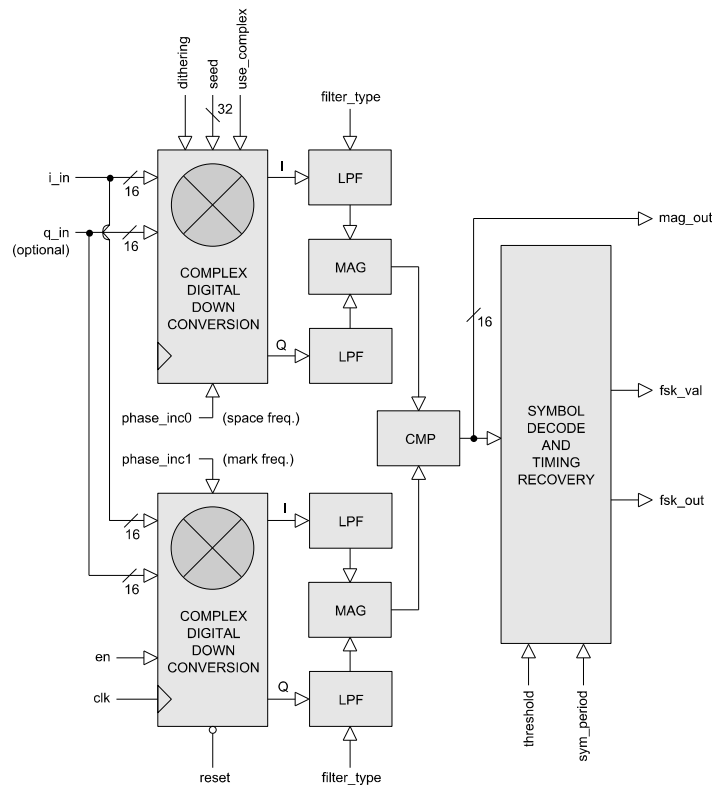


Figure 1: FSK demodulator architecture

Generic Parameters

Generic name	Description	Type	Valid range
dithering	Enable phase dithering in DDS component	boolean	TRUE/FALSE
seed	Seed for random number generator in DDS component	std_logic vector	$0 < \text{seed} < 2^{32}$
use_complex	Enable complex or real data samples	boolean	TRUE: use ports <i>i_in</i> and <i>q_in</i> FALSE: use port <i>i_in</i> only
filter_type	Low-pass filter response type	integer	0: Narrow BW 1: Medium BW 2: Wide BW
threshold	Mark/Space level threshold	integer	0 to 65535
sym_period	Symbol period in sample clocks	integer	0 to 65535

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
phase_inc0 [31:0]	in	Phase increment as an unsigned 32-bit number (f_0 centre frequency)	data
phase_inc1 [31:0]	in	Phase increment as an unsigned 32-bit number (f_1 centre frequency)	data
i_in [15:0]	in	Real (In-phase) inputs as 16-bit signed	data
q_in [15:0]	in	Imaginary (Quadrature) inputs as 16-bit signed (optional)	data
mag_out[15:0]	out	Magnitude of symbol values at the decoder	data
fsk_val	out	FSK bit valid strobe	high
fsk_out	out	FSK bit out	data

1 For higher symbol rates please consult Zipcores
 2 Xilinx Virtex 5 FPGA used as a benchmark

General Description

FSK_DEMOD is a precision Binary-FSK Demodulator based on a non-coherent receiver design. The demodulator is fully programmable, allowing for a varied range of symbol rates and tone frequencies. Input data samples may be either complex or real and support for both passband and baseband signals is provided. The module allows easy connectivity to an external quadrature down-converter, mixer, or directly to an ADC.

Figure 1 shows the basic architecture in more detail. The mark and space tone frequencies are generated by a pair of local oscillators. Each oscillator is implemented as a DDS with an SFDR of better than 80 dBs and a theoretical SNR of approximately 100 dBs.

After mixing, the I and Q signal paths for each tone are filtered to remove components above the mark and space centre frequencies. The characteristics of these filters may be changed depending on the desired FSK signal bandwidth and symbol rate.

A power function is used to compute the relative magnitudes of the mark/space tones after filtering. These magnitudes are then compared and passed to the symbol decode and timing recovery circuit. The demodulated FSK bit-stream appears at the output *fsk_out*. Bits are valid on the rising edge of *clk* when *fsk_val* is high.

Mark and Space Centre Frequencies

The frequencies of the mark and space tones are controlled by the signals *phase_inc0* and *phase_inc1*. The phase increment may be calculated using the formula:

$$\Phi_{inc} = (F_{out} * 2^{32}) / F_s$$

Where F_{out} is the desired oscillator frequency and F_s is the sampling frequency. When the desired oscillator frequency is negative, then the formula becomes:

$$\Phi_{inc} = ((F_s - F_{out}) * 2^{32}) / F_s$$

Note that an *integer* value for the phase increment must be used. As an example, consider a 100 MHz sample clock with a desired local oscillator frequency of 6.197 MHz. The phase increment would be calculated as $(6.197 * 2^{32}) / 100 = 266159123$. The minimum and maximum local oscillator frequencies are given by the following formulas:

$$F_{min} = F_s / 2^{32}, \quad F_{max} = F_s / 2$$

As an example, a 100 MHz sample clock would allow a minimum local oscillator frequency of 0.0233 Hz. Conversely, the maximum frequency the local oscillator can generate is given by the Nyquist-Shannon sampling theorem ($F_s/2$).

Low pass I/Q filters

The I and Q signal paths are filtered using a bank of low pass IIR filters. In total, there are three separate filter responses that may be selected using the generic parameter *filter_type*. Figure 2 shows the different filter responses available.

Note that it is important that the full 16-bit dynamic range of the demodulator inputs are used in order for the low-pass filters to function optimally. For input samples with lower numbers of significant bits, the data samples should be left-shifted so that the full 16-bit dynamic range is utilized.

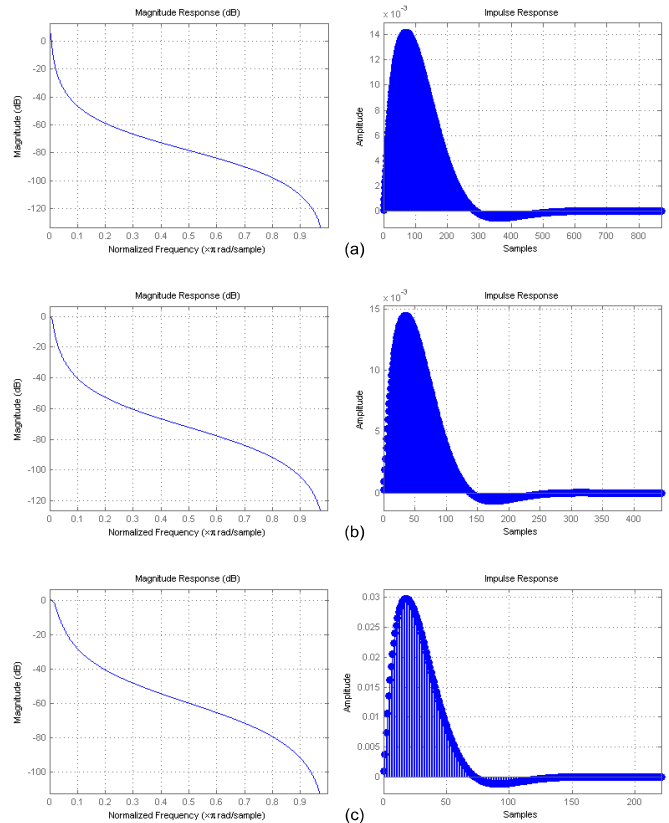


Figure 2: Low-pass filter responses.
 The -3dB cutoff points are:
 (a) 0.005, (b) 0.01 and (c) 0.02 rads/sample

Filter (a) is characterized by a very narrow bandwidth and a long impulse response time. Conversely, filter (c) has a wider bandwidth but a shorter impulse response time. The table below outlines the different filter characteristics in more detail³.

Fig.	Filter type	-3dB cutoff frequency	Approximate Response time
(a)	0	$0.005 * (F_s / 2)$	300 samples
(b)	1	$0.01 * (F_s / 2)$	150 samples
(c)	2	$0.02 * (F_s / 2)$	75 samples

³ A range of different filter responses are available on request. Please contact ZIPcores for more details.

Symbol rate and FSK tone Separation

The bandwidth of the chosen low-pass filter will effect the minimum allowable separation between FSK tones. In addition, the filter impulse response time will effect the the rate at which the filter can respond to a change in input symbol. Note that choosing a lower symbol rate and a wider separation between FSK tones will limit the effects of Inter-Symbol-Interference (ISI). The table below outlines the range of practical tone separation frequencies and symbol rates for the three different filter types.

Filter type	Minimum FSK tone separation	Maximum FSK Symbol rate
0	$0.005 * F_s$	$F_s / 200$
1	$0.01 * F_s$	$F_s / 100$
2	$0.02 * F_s$	$F_s / 50$

As a general rule-of-thumb, and for the most reliable operation, the symbol period and carrier separation is given by:

$$\text{Symbol Period} \geq 1 / \text{Carrier Separation}$$

Symbol decoding and Timing recovery

The symbol decoder block extracts the symbol timing information and symbol values from the received FSK signal. In order for the symbol decoder to function correctly, the generic parameters *sym_period* and *threshold* must be set appropriately.

The symbol period is specified as an integer number of clock cycles for the chosen sampling frequency. The threshold is a relative magnitude, 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 number of factors such as the dynamic range of the input signal, the chosen filter-type, the carrier spacing and the symbol period. The output signal *mag_out* (Figure 3) may be used to determine the best threshold level to set for a given set of parameters. This is usually 1/4 of the way between the 0 cross-over point and the maximum positive value of *mag_out*.

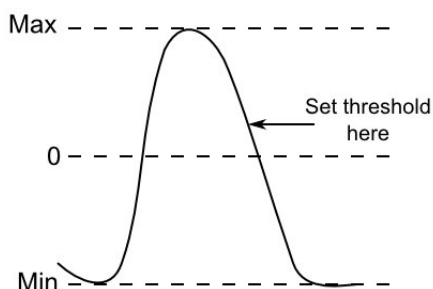


Figure 3: Setting the threshold level for a bit decision

Functional Timing

Figure 4 shows the operation of the FSK demodulator during normal operation. In this particular example, *use_complex* has been set to *false* meaning that only the 'I' signal (real) path is used with 'Q' unused.

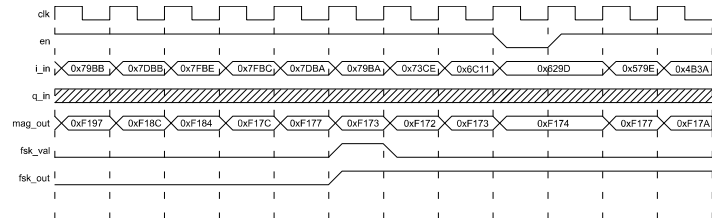


Figure 4: Binary FSK demodulator timing waveforms

FSK inputs and outputs are sampled on the rising edge of *clk* when *en* is high. FSK output bits are valid when *fsk_val* is high.

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
sym_gen_rand.vhd	Random symbol generator
sincos16.vhd	SIN/COS look-up table
dds16.vhd	16-bit DDS component
ddc16.vhd	16-bit Digital Down Converter
iir_biquad.vhd	IIR filter
lpf.vhd	Dual-channel low pass I/Q filter
tone_dec.vhd	Tone decoder
fsk_sym_dec.vhd	Symbol decoder
fsk_demod.vhd	Top-level component
fsk_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. sym_gen_rand.vhd
2. sincos16.vhd
3. dds16.vhd
4. ddc16.vhd
5. iir_biquad.vhd
6. lpf.vhd
7. tone_dec.vhd
8. fsk_sym_dec.vhd
9. fsk_demod.vhd
10. fsk_demod_bench.vhd

The VHDL testbench instantiates the demodulator component and also a separate DDS that provides the FSK source input signal.

In the example test provided, the demodulator is configured to use filter type '0' with the magnitude threshold set to 512 and the symbol period set to 200 sample clocks. The system clock period is set to 100MHz which is the sampling frequency for the simulation. The mark and space tone frequencies are set to 2.0 and 1.0 MHz respectively.

During the course of the test, the component 'sym_gen_rand.vhd' generates a randomized sequence of 1's and 0's which are used to modulate a source FSK signal with centre frequencies of 1.0 and 2.0 MHz. The resultant output waveforms from the demodulator are shown in Figure 5.

The simulation must be run for at least 10 ms during which time the input bit stream and demodulated output bit stream are captured in the files *fsk_demod_in.txt* and *fsk_demod_out.txt*. These two files may be compared to verify that the bits have been demodulated correctly. In addition, the comparator output magnitudes at the symbol sample points are captured in the file *fsk_demod_mag.txt*. These magnitudes may be used to calculate the Signal-to-Noise ratio (SNR) at the symbol decoder.

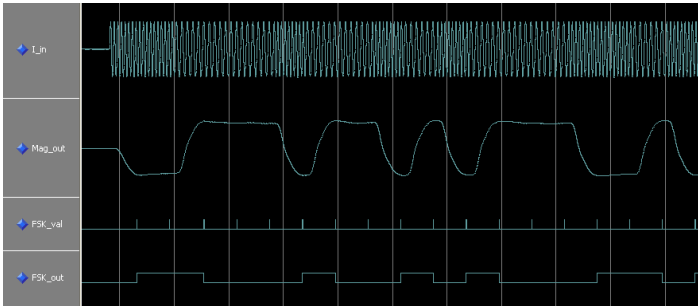


Figure 5: B-FSK Demodulator simulation output

Performance

The output signal *mag_out* was captured over a series of 10,000 symbols for the three different filter types set in their nominal configurations. The symbol rates were respectively set to 200, 100 and 50 samples for each filter. In all three cases, the input FSK signal was generated using a 16-bit DDS component, utilizing the 16-bit dynamic range available at the demodulator inputs.

The signal amplitudes in the *mag_out* capture file were overlaid over 2 symbol periods in order to realize the eye diagrams in Figure 6. The eye diagrams were noted for the size of the 'eye-openings' and the time variations at the zero crossing points.

In all three test cases, the size of the eye-openings were seen to be excellent and the time variations at the zero crossing points were observed to be minimal.

In addition, the SNR at the symbol sampling points (the eye mid-points) were measured. For comparison, the SNR was measured for each filter with different FSK tone separations. In all three test cases, the Bit Error Rate (BER) was observed to be Zero over the 10,000 symbols.

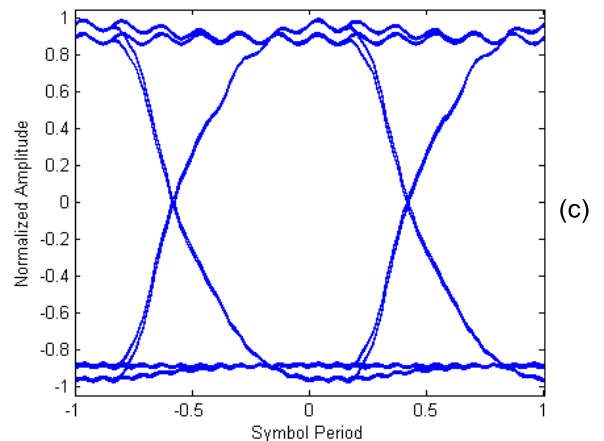
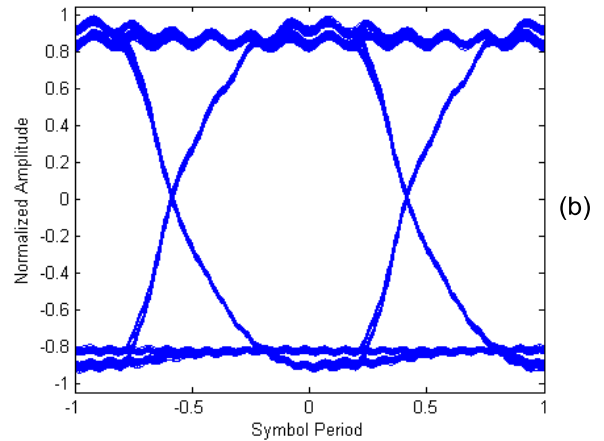
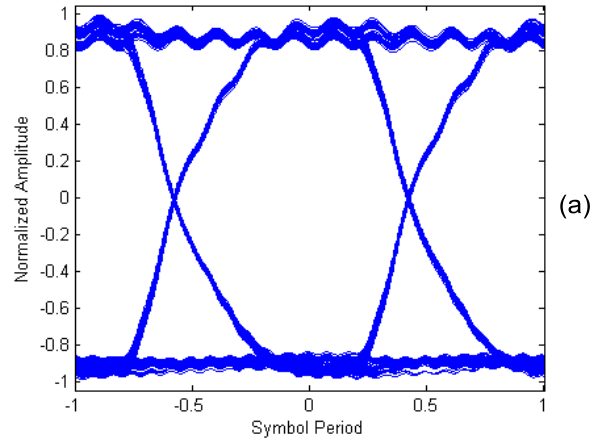


Figure 6: Eye diagrams for symbols at the decoder with the three filter-type settings

The SNR at the symbol sample points was calculated using the following formula:

$$SNR = 20 \log \frac{\bar{A}_1 - \bar{A}_0}{\sqrt{\sigma_1^2 + \sigma_0^2}}$$

Where values A_1 and A_0 signify the mean signal amplitudes at the logic '1' and logic '0' levels. Values σ_1 and σ_0 are the standard deviations from the mean at the logic '1' and '0' levels. The results for the different filter types for different FSK bandwidths are shown in the table below.

Filter type	FSK tone separation	SNR at decoder
0	0.005 * F_s 0.01 * F_s	29 dB 31 dB
1	0.01 * F_s 0.02 * F_s	28 dB 29 dB
2	0.02 * F_s 0.04 * F_s	30 dB 31 dB

Synthesis

The files required for synthesis and the design hierarchy is shown below:

- fsk_demod
 - fsk_sym_dec
 - tone_dec.vhd
 - ddc16.vhd
 - dds16.vhd
 - sincos16.vhd
 - lpf.vhd
 - iir_biquad.vhd

The VHDL core is designed to be technology independent. However, as a benchmark, synthesis results have been provided for the Xilinx Virtex 5 and the Altera Stratix III series of FPGA devices. The lowest and highest speed grade devices have been chosen in both cases for comparison.

Note that setting the parameter *use_complex* to 'false' will result in a saving of hardware multiplier components.

Trial synthesis results are shown with the generic parameters set to: dithering = true, seed = 0x14FFDE78, use_complex = false, filter_type = 0, threshold = 512, sym_period = 200.

Resource usage is specified after Place and Route.

VIRTEX 5

Resource type	Quantity used
Slice register	839
Slice LUT	1659
Block RAM	2
DSP48	16
Clock frequency (worst case)	100 MHz
Clock frequency (best case)	150 MHz

STRATIX III

Resource type	Quantity used
Register	681
ALUT	878
Block Memory bit	45296
DSP block 18	36
Clock frequency (worse case)	100 MHz
Clock frequency (best case)	150 MHz

Revision History

Revision	Change description	Date
1.0	Initial revision	04/11/2009
1.1	Added description of how to set threshold values	15/10/2010
1.2	Modified the DDS LUT to use 12-bits internally in order to reduce Block RAM usage. Updated synthesis results	29/12/2011