Key Design Features

- Technology independent IP Core for FPGA and ASIC
- Supplied as human readable VHDL (or Verilog) source code
- 16-bit signed input data samples
- Accepts either complex (I/Q) or real input samples
- Programmable mark / space frequencies
- Choice of low pass filter responses
- Carrier separation ~ symbol rate or greater
- Practical symbol rates of up to 10 Mbits/s
- Baseband or passband operation
- Typical FPGA sample rates of up to 200 MHz[1]
- Connects directly to an external ADC

Applications

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

Block Diagram

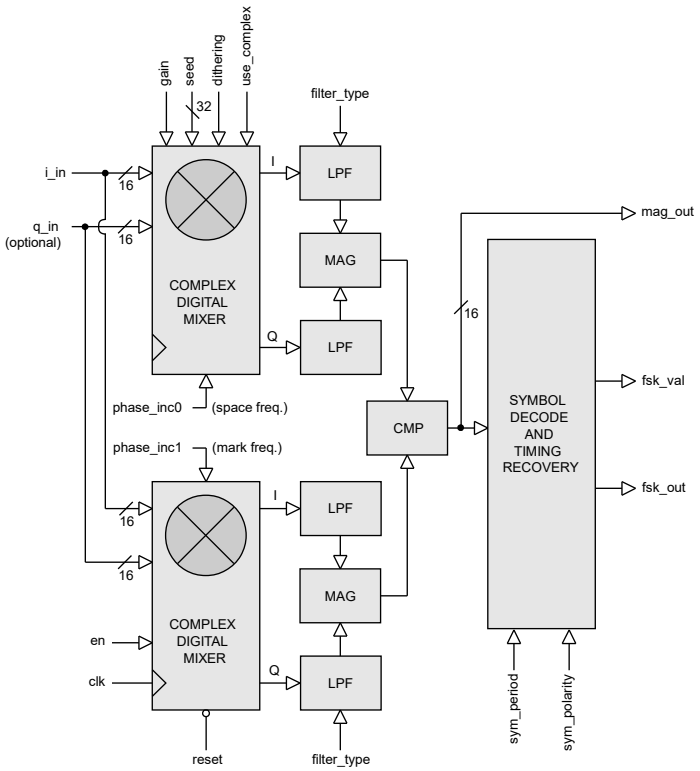

Figure 1: FSK demodulator architecture

Generic Parameters

Generic name	*Description*	*Type*	*Valid range*
gain	Internal gain setting (compensates for low amplitude input signals)	integer	0: x 1 1: x 2 2: x 4 3: x 8
seed	Seed for random number generator in DDS component	std_logic vector	0 < seed < 2^{32}
dithering	Enable phase dithering in DDS	boolean	TRUE / FALSE
use_complex	Enable complex or real data samples	boolean	TRUE: use both ports *i_in* and *q_in* FALSE: use port *i_in* only
filter_type	Low-pass filter response type	integer	0: min B/W 3: max B/W
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
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) input as 16-bit signed	data
q_in [15:0]	in	Imaginary (Quadrature) input as 16-bit signed (optional input)	data
mag_out[15:0]	out	Magnitude of symbols after compare block as 16-bit signed	data
fsk_val	out	FSK bit valid strobe	high
fsk_out	out	FSK bit out	data

1 Xilinx® 7-series used as a benchmark

General Description

BFSK_DEMOD is a precision Binary-FSK Demodulator IP Core based on a non-coherent receiver design. The demodulator is fully programmable, allowing for a varied range of symbol rates and mark/space tone frequencies. Input data samples may be either complex or real for support of either passband or baseband signals. The module allows easy connectivity to an external ADC with up to 16-bit signed input samples.

Figure 1 shows the basic architecture in more detail. The mark and space tone frequencies are generated by a pair of precision 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.

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 signal *mag_out* is a 16-bit signed output that allows the user to monitor the magnitude of the output symbols before decode and timing recovery.

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:

$$\Phi_{INC} = \left(F_{OUT} * 2^{32}\right) / F_S$$

Where F_{OUT} is the desired oscillator frequency and F_S is the sampling frequency. When the desired oscillator frequency is negative (e.g. for baseband operation) then the formula becomes:

$$\Phi_{INC} = \left(\left(F_S - F_{OUT}\right) * 2^{32}\right) / F_S$$

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:

$$F_{MIN} = F_S / 2^{32} \quad , \quad F_{MAX} = F_S / 2$$

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 (Fs/2).

Low pass I/Q filters

The I and Q signal paths are filtered using a pair of low pass IIR filters. In total, there are four separate filter responses that may be selected using the generic parameter *filter_type*.

Note that for the best possible results, it is important that the full 16-bit dynamic range of the IIR filters is used. In the case of low amplitude input signals, the generic parameter *gain* may be adjusted to increase the amplitudes of the I and Q signals out of the DDC. Figure 2 shows the different filter responses available.

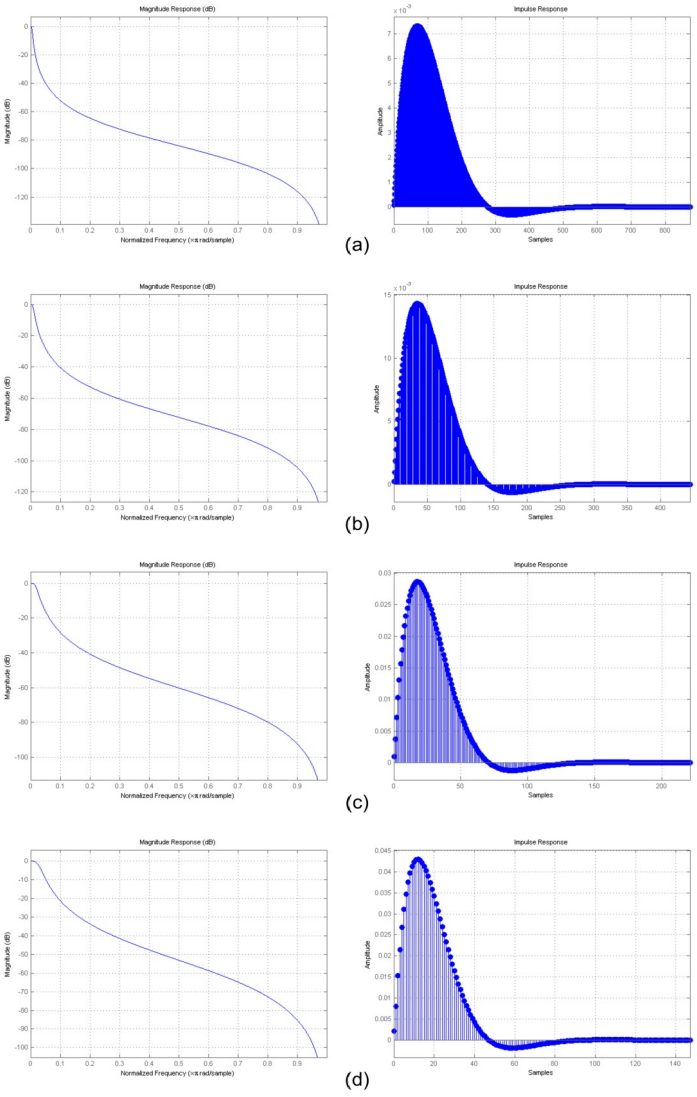

Figure 2: Low-pass filter responses. The -3dB cutoff points are: (a) 0.005, (b) 0.01, (c) 0.02 and (d) 0.03 rads/sample

Filter (a) is characterized by a very narrow bandwidth and a long impulse response time. Conversely, filter (d) has the widest bandwidth but a shorter impulse response time. The table below outlines the different filter characteristics in more detail[2].

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

Fig.	*Filter type*	*-3dB cutoff frequency*	*Approximate Response time*
(a)	0	0.005 * (F_S / 2) (narrow bandwidth)	300 samples (slow response)
(b)	1	0.01 * (F_S / 2)	150 samples
(c)	2	0.02 * (F_S / 2)	75 samples
(d)	3	0.03 * (F_S / 2) (wide bandwidth)	50 samples (fast response)

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 response time will effect the the rate at which the filter can respond to a change in input symbol. Choosing a lower symbol rate and a wider separation between FSK tones will limit the effects of Inter-Symbol-Interference (ISI).

As a general rule-of-thumb, and for the most robust designs, the carrier separation should be greater or equal to the symbol rate:

$$Carrier\ separation\ (Hz) \geq Symbol\ rate\ (bits/s)$$

Note that in practice the choice of sample rate, symbol rate, centre frequency and carrier separation will largely be determined by the telecommunications standard in use[3].

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 parameter *sym_period* must be set appropriately.

The symbol period is specified as an integer number of clock cycles for the chosen sampling frequency. The value is calculated as follows:

$$sym\_period\ (clocks) = \frac{(System\ clock\ frequency)}{(Symbol\ rate)}$$

The polarity of the decoded symbol is set using the *sym_polarity* parameter. Setting the *sym_polarity* to *True* will leave the decoded bit unchanged. Setting *sym_polarity* to *False* will invert the bit so that a '1' becomes '0' and vice-versa.

The output signal *mag_out* is a 16-bit signed value that shows the magnitude of the samples before the symbol decoder and is useful for system debug (e.g. plotting eye-diagrams, checking signal quality etc.)

3 Please contact Zipcores if you require assistance in characterizing your FSK demodulator system.

Functional Timing

Figure 3 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.

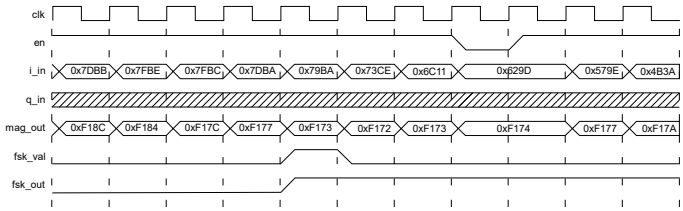

Figure 3: Binary FSK demodulator timing waveforms

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
bfsk_sym_dec.vhd	Symbol decoder
bfsk_demod.vhd	Top-level component
bfsk_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. bfsk_sym_dec.vhd
9. bfsk_demod.vhd
10. bfsk_demod_bench.vhd

The VHDL testbench instantiates the demodulator component and also a separate DDS that provides the FSK source input signal[4]. In the example test provided, the system sample frequency is set to 33.33 MHz. The demodulator is configured to use filter type '3' with a symbol period set to 32 sample clocks which equates to 0.96 us. The mark and space tone frequencies are set to 10 MHz with a deviation of +/- 312.5 kHz. Figure 4 below shows this graphically:

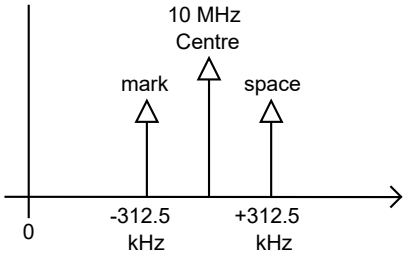

Figure 4: Spectrum of the FSK signal used in the test setup

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 the source FSK signal . 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 *bfsk_demod_in.txt* and *bfsk_demod_out.txt*. These two files may be compared to verify that the bits have been demodulated correctly.

In addition, the *mag_out* values are captured at each sample period and saved to the file *bfsk_demod_mag.txt*. These values may be used to plot an eye-diagram as shown in Figure 5 below:

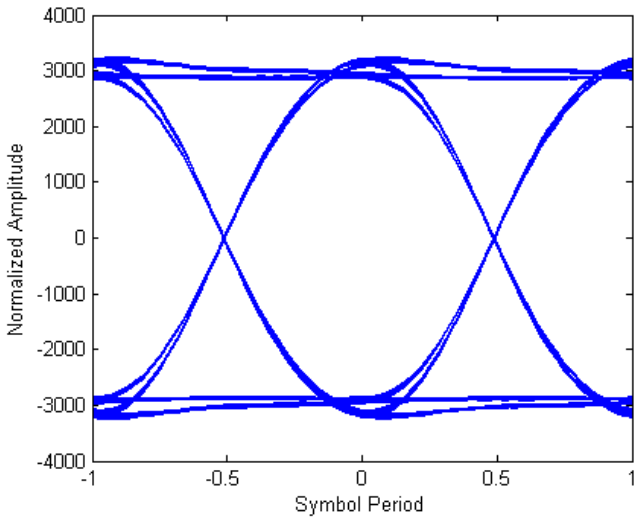

Figure 5: Eye-diagram for the example simulation

The *mag_out* values can also be used to calculate the Signal-to-Noise ratio (SNR) at the symbol decoder 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 resulting SNR for the test was calculated as 29 dB.

Synthesis and Implementation

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

- bfsk_demod
  - bfsk_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® 7-series FPGAs. Synthesis results for other FPGAs and technologies can be provided on request.

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: gain = 0, seed = 0x14FFDE78, dithering = true, use_complex = false, filter_type = 3, sym_period = 32 and sym_polarity = true. Resource usage is specified after place and route.

XILINX® 7-SERIES FPGAS

Resource type	***Artix-7***	***Kintex-7***	***Virtex-7***
Slice Register	778	778	778
Slice LUTs	2850	2850	2850
Block RAM	0	0	0
DSP48	36	36	36
Occupied Slices	1418	1348	1334
Clock freq. (approx)	100 MHz	150 MHz	200 MHz

4 Note: the test is set up to emulate the design parameters for an aircraft UAT ADS-B transceiver (RTCA DO-282B).

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	29/12/2011
1.3	Updated synthesis results in line with minor source code changes	21/02/2012
1.4	Optimized internal IIR filters for speed. Added additional filter response type. Added DDC gain function for improved SNR. Updated synthesis results.	24/02/2015
1.5	Updated synthesis results for Xilinx® 7-series FPGAs	03/03/2016

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for Development Software ***category:***

Click to view products by Zipcores ***manufacturer:***

Other Similar products are found below :

SRP004001-01 SW163052 SYSWINEV21 WS01NCTF1E W128E13 SW89CN0-ZCC IP-UART-16550 MPROG-PRO535E AFLCF-08-LX-CE060-R21 WS02-CFSC1-EV3-UP SYSMAC-STUDIO-EIPCPLR 1120270005 SW006021-2H ATATMELSTUDIO 2400573 2702579 2988609 SW006022-DGL 2400303 88970111 DG-ACC-NET-CD 55195101-101 55195101-102 SW1A-W1C MDK-ARM SW006021-2NH B10443 SW006021-1H SW006021-2 SW006022-2 SW006023-2 SW007023 MIKROE-730 MIKROE-2401 MIKROE-499 MIKROE-722 MIKROE-724 MIKROE-726 MIKROE-728 MIKROE-732 MIKROE-734 MIKROE-736 MIKROE-738 MIKROE-744 MIKROE-928 MIKROE-936 1120270002 1120270003 1120275015 NT-ZJCAT1-EV4