

Key Design Features

- Synthesizable, technology independent VHDL Core
- Fully pipelined architecture for highest possible sample rates
- 16-bit signed input and output samples
- 10-bit fixed-point coefficients and scaling factor
- Output saturation and overflow detection
- SOS coefficient matrix maps directly to filter coefficients

Applications

- Basic building block for higher order IIR filters – simply cascade SOS blocks in series.
- General purpose IIR filters, low-pass, high-pass, all-pass, group delay equalization, arbitrary magnitude
- IIR filtering in higher sample-rate applications

Pin-out Description

Pin name	I/O	Description	Active state
clk	in	Sampling clock	rising edge
reset	in	Asynchronous reset	low
en	in	Clock enable	high
coeff_s0 [9:0]	in	10-bit Scale factor as a signed fixed-point number in [10 7] format	data
coeff_b0 [9:0]	in	Coefficient b0 as a signed fixed-point number in [10 7] format	data
coeff_b1 [9:0]	in	Coefficient b1 as a signed fixed-point number in [10 7] format	data
coeff_b2 [9:0]	in	Coefficient b2 as a signed fixed-point number in [10 7] format	data
coeff_a1 [9:0]	in	Coefficient a1 as a signed fixed-point number in [10 7] format	data
coeff_a2 [9:0]	in	Coefficient a2 as a signed fixed-point number in [10 7] format	data
x_in [15:0]	in	Filter input samples as a signed number	data
y_out [15:0]	out	Filter output samples as a signed number	data
overflow	out	Output overflow flag	high

Block Diagram

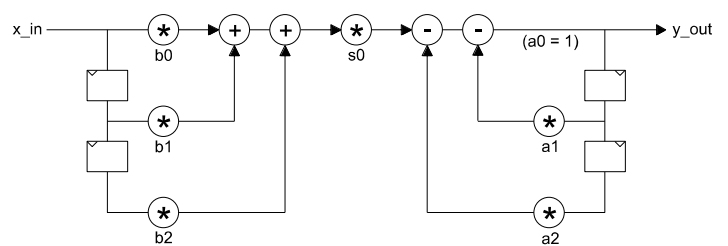


Figure 1: IIR filter SOS (Simplified)

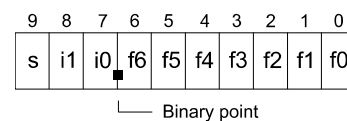
General Description

IIR_SOS is a 2nd order IIR filter sometimes referred to as a 'Bi-quad'. Internally, it has a fully pipelined architecture permitting the highest possible sample rates for IIR filtering. The IIR block is modular and any number of SOS blocks may be joined in series to implement higher order filters. Mathematically, the filter implements the difference equation:

$$y[n] = s_0(b_0x[n] + b_1x[n-1] + b_2x[n-2]) - a_1y[n-1] - a_2y[n-2]$$

In the above equation, the input signal is $x[n]$, the output signal is $y[n]$ and b_0, b_1, b_2, a_1, a_2 represent the filter coefficients. The value s_0 is a scale factor that is often required in stable IIR implementations to prevent the filter oscillating.

Filter coefficients are 10-bits wide and defined as signed fixed-point numbers in [10 7] format – i.e. 1 sign-bit, 2 integer bits and 7 fraction bits. The scale factor is also specified in the same format. The diagram below shows this pictorially:



Inputs and output samples are signed 16-bit values (their format is purely relative). During normal operation of the filter, if the output samples are too large to be accommodated within a 16-bit signed number, then the output samples will saturate to the largest positive or negative number. The *overflow* flag will be asserted in the event of an overflow condition. Generally, the system should be designed so that an overflow does not occur – otherwise the filter may become unstable.

Values are sampled on the rising clock-edge of *clk* when *en* is high. The latency of the IIR filter between the first input sample and the first output sample is 7 clock cycles. Note that it is recommended that, before normal operation begins, the filter be reset by asserting the *reset* signal low for at least one clock cycle. This ensures that the feedback paths of the filter are initialized to zero.

Functional Timing

Figure 2 shows a sequence of input samples for the IIR 2nd order section. Output samples appear 7 clock cycles later.

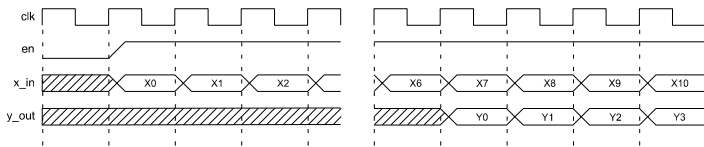


Figure 2: IIR SOS filter input/output samples

Figure 3 demonstrates an example of a positive overflow condition where the output sample saturates to the maximum positive number (0x7FFF) for a signed 16-bit value.

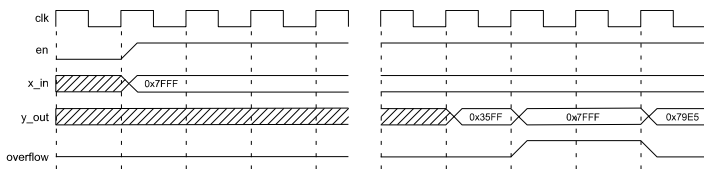


Figure 3: IIR filter positive overflow and output saturation

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
iir_sos.vhd	Top-level block
iir_sos_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. iir_sos.vhd
2. iir_sos_bench.vhd

The VHDL testbench instantiates the IIR_SOS filter component. The user may modify the coefficients as required in order to implement the desired filter response. The test provided is configured for a 2nd order low-pass Butterworth filter. Coefficients for a high-pass Chebyshev and an all-pass arbitrary group-delay filter are also provided as examples.

The simulation must be run for at least 100 us during which time the impulse response and step response of the filter is tested.

The simulation generates a text file called: iir_sos_out.txt. This file contains the output samples captured during the course of the test. Figures 4 and 5 respectively contain the impulse response and step response outputs of the low-pass Butterworth example.

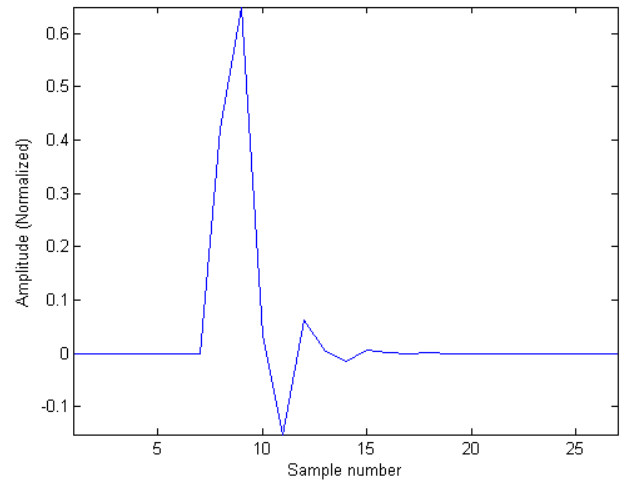


Figure 4: 2nd order low-pass Butterworth : Impulse response

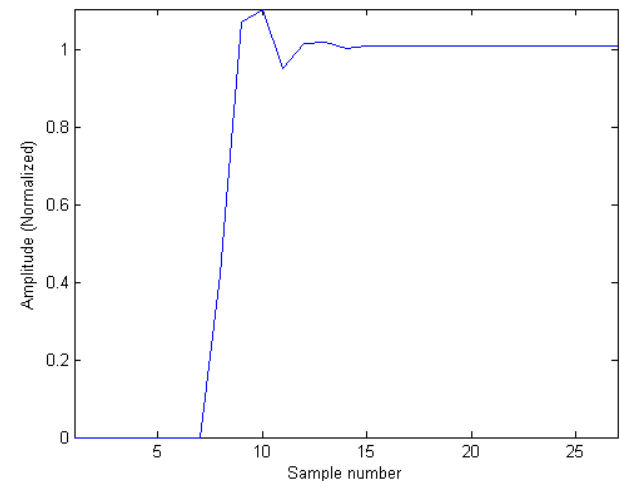


Figure 5: 2nd order low-pass Butterworth: Step response

Synthesis

The source file 'iir_sos.vhd' is the only required for synthesis. There are no sub-modules in the design.

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.

Resource usage is specified after Place and Route.

VIRTEX 5

Resource type	Quantity used
Slice register	191
Slice LUT	328
Block RAM	0
DSP48	6
Clock frequency (worst case)	110 MHz
Clock frequency (best case)	146 MHz

STRATIX III

Resource type	Quantity used
Register	203
ALUT	168
Block Memory bit	0
DSP block 18	12
Clock frequency (worse case)	105 MHz
Clock frequency (best case)	139 MHz

Revision History

Revision	Change description	Date
1.0	Initial revision	20/05/2008