Biquad Infinite Impulse Response Filter

IP Core

April 3, 2001

Author: Chuck Cox
Chuck@100home.com

Table of Contents

1.0 Introduction

The biquad filter core is an implementation of an infinite impulse response (IIR) filter with two poles and two zeros. The IIR filter can be compared to another type of filter called the finite impulse response (FIR) filter. When selecting a filter for a particular application a designer must choose between these two types of filter implementations. For an equivalent amount of silicon running at the same clock speed the IIR filter will provide better out of band attenuation than the FIR filter. However, the IIR filter causes phase distortion to the signal while a properly designed FIR filter will not.

Phase distortion basically means that different frequencies have different delays through the IIR filter. Some filtering applications are more sensitive to phase distortion than others. Audio signals are generally not sensitive to the phase distortion while phase modulated signals such as employed in modems may be very sensitive to it. System designers need to evaluate the sensitivity of their particular application to phase distortion prior to selecting a filter implementation.

The biquad filter core can be put in series with additional biquad filter cores to implement filters with more than two poles and zeros. Adding more poles and zeros to a filter allows the filter designer to achieve more abrupt transitions from the pass band to the stop band. Furthermore, phase distortion is often only bad enough to be of concern in the transition band of the filter. So selection of filters with large numbers of poles and zeros minimizes the extent of the phase distortion by minimizing the transition region.

The biquad filter core can be used to implement low pass filters, band pass filters, high pass filters, or band reject filters. The type of filter implemented depends upon the selection of the filter coefficients. The filter coefficients are inputs to the biquad filter module. The design of a particular set of filter coefficients is generally done using analog filter design techniques. The poles and zeros of the resulting analog filters are then mapped over to the discreet time domain using the bilinear transformation. Figure 1 is the frequency response of a low pass filter using two poles and two zeros.

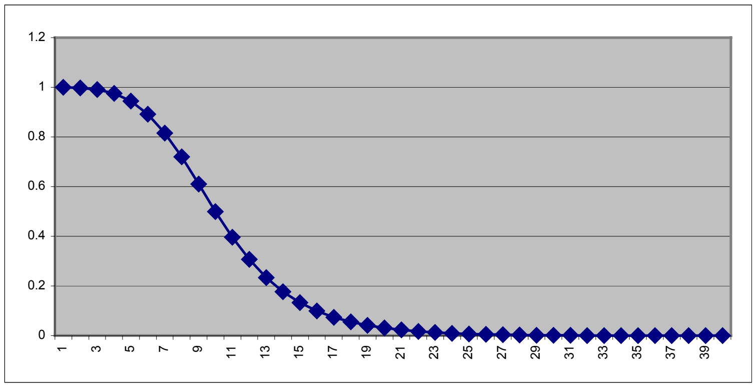

Figure 1: Freq. Response of a Low Pass Filter Implemented with Biquad Filter

2.0 Functional Description

The difference equation of the biquad filter core is given in figure 2. The implementation of this difference equation is illustrated in figure 3. This filter has 5 coefficients: b10, b11, b12, a11, and a12. Each of these filter coefficients can be up to 16 bits in length. The number of bits used to represent the filter coefficients can be modified by changing the COEFWIDTH parameter in the Verilog code.

$$Y[n] = b10*x[n] + b11*x[n-1] + b12*x[n-2] + a11*y[n-1] + a12*y[n-2]$$

Figure 2: Filter Difference Equation

The number of bits that can be used to represent the data is unrestricted. The data word length can be modified by changing the DATAWIDTH parameter in the Verilog code. Both the input and the output samples of the filter are DATAWIDTH bits wide. The multipliers inside this filter core will create a product that is DATAWIDTH + COEFWIDTH bits wide. The multipliers will typically be the critical path for the purpose of timing analysis. Selection of smaller data words and smaller coefficients will result in faster clock speeds at the expense of filter response.

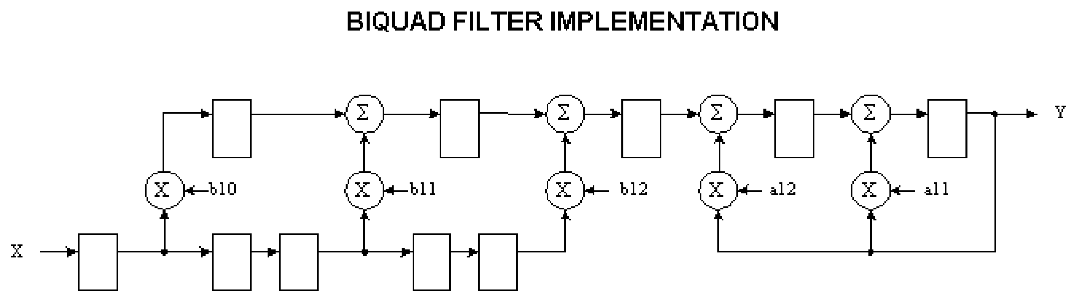

Figure 3: Filter Block Diagram

The biquad filter core provides a data valid input that indicates to the core that valid data is present on the data input port. This valid signal can be used to implement a filter with multi-cycle multipliers. For example, if the multiplier requires 4 clock cycles to produce a result then valid input data should be put in to the filter only once every 4th clock. When using this scheme the output is valid only when the input is valid (ie every 4 clocks). This can be a useful technique when high precision coefficient and high precision data requirements make the multipliers large and potentially slow. Other design constraints that might require the use of this feature might be the requirement to use a low cost and slow FPGA or PLD.

The numerical mapping between binary and decimal representation for the data is fractional twos complement. In this mapping scheme the msb is the sign bit the next msb represents 2^{-1}, the next msb represents 2^{-2}, and so on. This numbering scheme is illustrated below for the case of 4 bits:

0000 = 0	0001 = .125	0010 = .25	0011 = .375	0100 = .5	0101 = .625
0110 = .75	0111 = .875	1000 = -1	1001 = -.875	1010 = -.75	1011 = -.625
1100 = -.5	1101 = -.375	1110 = -.25	1111 = -.125		

The numerical mapping between binary and decimal for the filter coefficients is slightly different because filter coefficients as large as 2.0 (poles and zeros on or inside unit circle) need to be accommodated. The mapping used for filter coefficients is twos complement with the msb representing the sign bit, the next msb representing 2^{0}, the next msb representing 2^{-1}, and so on. This numbering scheme is illustrated below for the case of 4 bits:

0000 = 0	0001 = .25	0010 = .5	0011 = .75	0100 = 1.0	0101 = 1.25
0110 = 1.5	0111 = 1.75	1000 = -2	1001 = -1.75	1010 = -1.5	1011 = -1.25
1100 = -1.0	1101 = -.75	1110 = -.5	1111 = -.25		

3.0 Interfaces

The interfaces to this module consist of the standard wishbone interface signals for reading and writing filter coefficients as well as the data input, the data output, the DSP clock input, the data valid input, and the DSP reset input. The wishbone interface signals consist of clk_i, rst_i, we_i, stb_i, ack_o, dat_i, dat_o, and adr_i. The functions of these signals are exactly as described in revision B.1 of the "*Wishbone System-On-Chip (SOC) Interconnection Architectures for Portable IP Cores*" specification. This specification is available from Silicore Corporation (http://www.silicore.net). The biquad core is a 16 bit wishbone slave with 16 bit granularity.

The wishbone interface allows the 5 filter coefficients to be written and read. Each filter coefficient register is 16 bits in length. The number of bits used to represent the filter coefficients is determined by the value of the COEFWIDTH parameter in the Verilog code. When COEFWIDTH is less than 16 bits the most significant COEFWIDTH bits are taken from each 16 bit filter coefficient register and used to represent the filter coefficients.

The data input and data output ports are DATAWIDTH bits wide. As previously mentioned DATAWIDTH is a parameter in the Verilog code than can be modified by the user. The data input is clocked into the core on rising edge DSP clock when the valid input is high. The data output is clocked out on the rising edge of DSP clock. The DSP clock input clocks the registers inside the biquad core. The DSP reset input is an active low asynchronous reset that resets all filter functions. This reset is independent of the wishbone reset signal. Similarly, the DSP clock is independent of the wishbone clock.

4.0 Core Registers

The 5 registers of the biquad core are as shown below. These registers are used for the filter coefficients.

ADDRESS	**DESCRIPTION**	**WIDTH**	**DEFAULT VALUE**
0x0	a11	16 bits	0x0000
0x1	a12	16 bits	0x0000
0x2	b10	16 bits	0x0000
0x3	b11	16 bits	0x0000
0x4	b12	16 bits	0x0000