Trade-Off Between FPGA Resource Utilization and Roundoff Error in Optimized CSD FIR Digital Filters

Antonio E. de la Serna and Michael A. Soderstrand
Electrical and Computer Engineering Department
University of California, Davis
Davis, California 95616

Abstract

In this paper, we investigate the trade off between filter order and bits of coefficient precision in fixed-coefficient FIR digital filters utilizing *canonical signed digit* (CSD) coefficient representation. We demonstrate that the use of optimized CSD coefficients is often the only method for which many practical FIR filters can be prototyped on a single FPGA in today's technology. Due to finite FPGA resources, our resulting analysis of filter-length, word-length, and CSD bit optimization provides an indication of whether a desired filter performance can be obtained with a specific FPGA logic capacity. We develop a MATLAB algorithm for determining the optimum trade-off between FIR filter length and bits of precision of the coefficients in FIR digital filters.

1 Introduction

Over a decade ago Kodek and Steiglitz investigated the integer approximation problem for direct-form realization of FIR digital filters and concluded that for a given number of bits "b" there is an optimum order N_{opt} for the filter approximation [1]-[5]. Based upon this optimum order, they investigated the complexity of an FIR direct-form filter by plotting Nb as a function of b in the assumption that Nb is proportional to the complexity of the filter hardware [1].

Recently *Canonical signed digit* (CSD) multipliers have been shown to provide an efficient method for constant fixed-point multiplication by utilizing the redundancy of signed digit code [6, 7]. CSD is a radix 2 signed-digit representation for coefficients for which the permissible digit set is {-1,0,1}. Thus, CSD representation permits subtraction, as well as addition, of shifted data in accomplishing multiplication. The feature of redundancy in this representation allows a coefficient implementation to be selected which in general requires fewer adders/subtracters, and thus yields a faster more compact multiplier. This results in the complexity of the direct-form FIR filter being one-half of that using standard binary [6].

In another effort to reduce complexity in direct-form FIR digital filters, a number of authors have looked at designing filters with coefficients that are *powers-of-two* [8]-[11]. An even more drastic reduction in hardware complexity has been suggested in which the FIR filter taps are weighted only by $+1$, -1, or 0 [12, 13]. While these techniques can reduce hardware complexity, they introduce significant approximation errors. In the case of the powers-of-two technique, this is compensated for by sub-band coding and in the case of the design procedure with taps weights of $+1$, -1, 0, an accumulator is used as a digital integrator to compensate for the large approximation error [12, 13].

In this paper, we go back to the original approximation of Kodek and Steiglitz [1] and show that it does not apply to the direct approximation of ideal filters with limited precision coefficients. We shall show that there exists an optimum filter order N_{opt} which is significantly different from that predicted by Kodek and Steiglitz and we develop a MATLAB algorithm to calculate the optimum filter length. The result is a major reduction in hardware complexity for FIR filters.

2 Kodek-Steiglitz Theorem

The importance of the *Kodek-Steiglitz Theorem* is that it establishes the fact that an optimum filter order N_{opt} exists for which the maximum deviation of the approximation from the ideal response is dependent only upon the number of bits in the coefficient of the filter and not upon the order N of the filter for $N \geq N_{opt}$. However, the theorem assumes that the ideal filter can be realized with sufficient coefficient wordlength. However, the more common problem is

1058-6393/95 $4.00

to attempt to approximate an ideal filter characteristic such as a *"brick-wall"* or *"double-jump"* filter. When such ideal filters are approximated with finite word length coefficients, the N_{opt} of Kodek-Steiglitz is not useful. In this case, we will show that a new definition of N_{opt} will yield radically reduced hardware complexity in the design of FIR digital filters.

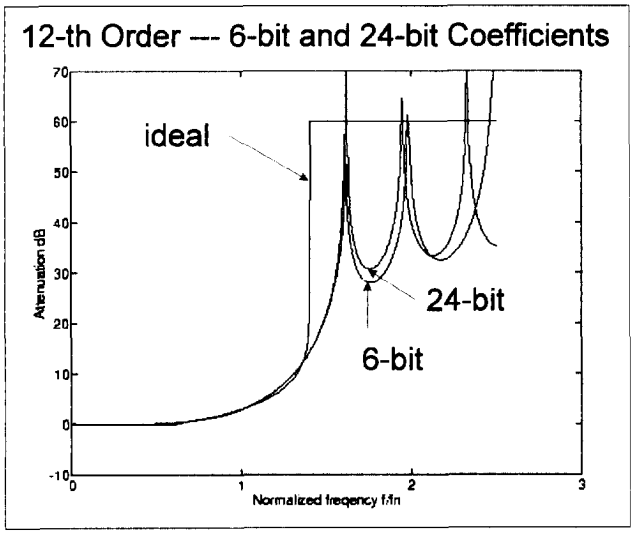

Figure 1: Effect of Bit in Coefficient

3 Approximating Ideal Filters with FPGA or ASIC Designs

It is often the case that a filter designer must design a filter to match as well as possible to an ideal filter that would require $N_{opt} = \infty$. We will show that in this type of design, it is not only possible but very desirable to increase the order of the filter and reduce the number of bits in the coefficients in order to meet the specifications with a design small enough to fit onto an Field-Programmable-Gate-Array (FPGA) or Application Specific Integrated Circuit (ASIC). In particular, we will show that it is advantages to use the minimum coefficient size possible to implement the maximum order filter.

3.1 Example Filter

Our interest in designing high-order FIR filters comes from our desire to implement filters required for the FQPSK-KF base-band modulator [14]-[20] which has important applications in wireless local area computer networks and mobile digital communications [18]. In the FQPSK-KF base-band modulator, it can be shown that a filter that approximates an ideal double-jump filter closely over the region below 20db

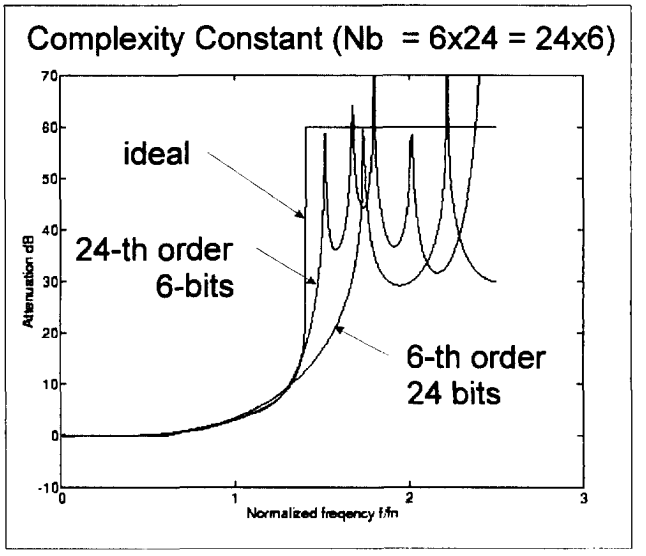

Figure 2: Effect of Bits of Coefficients

attenuation will have nearly an optimum power spectral density and bit error rate [19, 20]. An ideal double-jump filter is a filter that has no attenuation (zero db) from DC to some specified frequency ω_1, then has attenuation that varies directly with frequency (linear ramp) between frequencies ω_1 and ω_2, and then has a constant attenuation α_s for all frequencies above ω_2. This ideal double-jump filter may be approximated using the FIR2 function in MATLAB [21]. We have elected to apply a Kaiser window with $\beta = 1.0$. Figure 1 shows the ideal double-jump filter, a 12-th order approximation with 24-bit coefficients, and a 12-th order approximation with 6-bit coefficients. It should not be surprising to see from Figure 1 that reducing the bits in the coefficients primarily effects the stop-band attenuation, not the area of the filter of interest to us.

Figure 2 shows the MATLAB approximations for two filters with essentially the same hardware complexity according to the definition of Kodek and Steiglitz (ie: filter order times bits $N \cdot b$). The first approximation is 24-th order, but only 6 bits. The second approximation is 6-th order, but 24-bits. Clearly, the 24-th order approximation with only 6-bit coefficients is much superior to the other approximation.

3.2 FIR Filters With CSD Multipliers

Figure 3 shows a transpose-direct form realization of an FIR digital filter. For CSD implementation, we make all of the weights a_i positive and use a subtracter rather than an adder to implement the negative

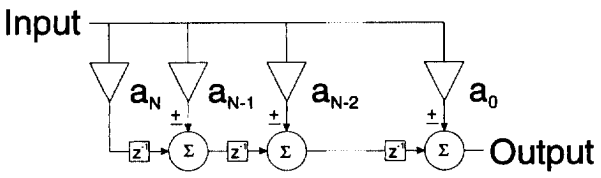

Figure 3: Transpose-Direct Form FIR Filter

weight. If the first weight (a_N in Figure 3) is negative and the second weight (a_{N-1}) is positive, we simple reverse the inputs to the first subtracter. However, if both a_N and a_{N-1} are negative, we use an adder rather than a subtracter and pass the sign on to the next unit. Eventually we will reach a true subtraction (ie: a unit that is adding a positive to a negative) and simply set the inputs on this subtracter to effect the correct operation. Thus we never need to implement a negative weight a_i.

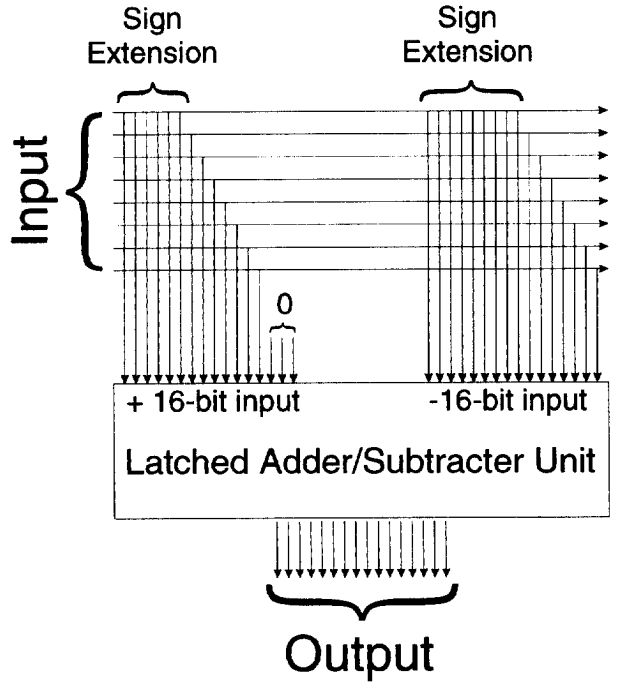

Figure 4: Multiplication by 7 in CSD

Fixed-coefficient CSD multipliers for the positive weights $|a_i|$ can be constructed simply by properly wiring the input to one or more *Latched Adder/Subtracter* (LAS) units [6]. Assuming our fixed coefficients α_i have been scaled to integers, any integer α_i that is a power of two can be implemented by wiring directly to the adder/subtracter units in Figure 3 without any extra LAS's. Integers that are not powers-of-two will be expressed in CSD. For example, multiplication of the input "x" by the fixed integer seven is accomplished in Figure 4 with a single LAS. In CSD, $7x = 8x - x$, so we form $7x$ by subtracting x from $8x$. Both x and $8x$ can be obtained from the input by proper wiring as shown in Figure 4. Notice that we generate a 16-bit product from the two's-compliment 8-bit input by using sign extension to expand the input to 16 bits. Here $N_c = 16$ bits is the computational word size. We will see later that our CSD is only 5.673 bits, thus with an 8-bit input we would only need 13.6 (thus 14) bits for N_c. However, our hard macros in Xilinx support either 8-bit or 16-bit LAS's. It is possible to use $N_c < 13.6$ bits, in which case we would generate round-off noise in the filter. However, in many instances the round-off noise generated is a small price to pay in order to get the size of the filter hardware small enough to fit onto a single FPGA or ASIC.

3.3 Maximum-Order Filters

In a Xilinx-type FPGA there are a finite number of *Configurable Logic Blocks* (CLB's) available on the chip [22]. In the XC4010 chip we will be using in our example there are 400 CLB's [23]. As noted before, our CSD implementation of the FIR filter will need only latched adder/subtracter units to construct the entire filter. Using the Xilinx *hard macros* we can construct an 8-bit LAS with 6 CLB's and a 16-bit LAS with 10 CLB's. Thus there are sufficient CLB's on the XC4010 to support 66 8-bit LAS's and 40 16-bit LAS's. In practice, however, we would only be able to get about 80% of this (50 8-bit LAS's or 32 16-bit LAS's) before it would become impossible to route the chip (ie: connect all the devices on the chip).

In the above discussion, the 8-bit LAS would be used for $N_c = 8$ and the 16-bit LAS would be used for $N_c = 16$. In standard CSD representation, an N-th order FIR filter with b-bits per coefficient would require approximately $N \cdot b/2$ LAS's [6]. This is quite discouraging in that $N \cdot b/2 < 50$ for 8-bit LAS's would limit the order $N \approx 12$ for 8-bit LAS's and $N \approx 8$ for 16-bit LAS's using typical 8-bit coefficients.

However, an interesting property of CSD representation is that for small integer numbers the number of LAS's required is zero. If we scale the coefficients to the smallest possible integer value, we have the potential for drastic reduction in the number of LAS's required in our CSD implementation. Assuming linear-phase FIR filters (ie: $\alpha(i) = \alpha(N - i)$, the minimum

b, b_{min}, is the number of bits necessary to represent the closest integer to $\alpha_{max}/\alpha(1)$. This will define a scaling factor $K_s = 2^{b_{max}}/\alpha_{max}$ which will scale the maximum coefficient α_{max} to $2^{b_{min}}$ and the first and last coefficient to $\alpha(1) = \pm 1$. (b_{min} need not be integer.)

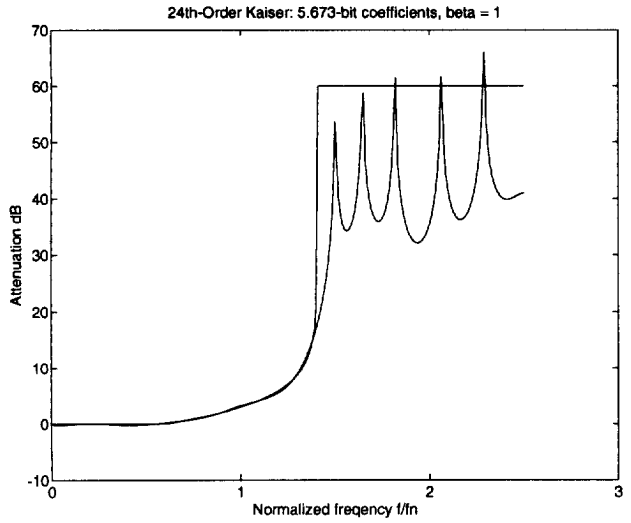

Figure 5: Optimum 24-th Order Filter

3.4 Minimum-Bit Coefficients

The minimum number of bits required for the coefficients depends not only on the filter order, but also on the actual coefficients of the filter. Thus we cannot give a general formula for calculating this minimum. Instead, we have written a MATLAB program that calculates the minimum number of bits along with the coefficients for the FIR2 approximation to the double-jump filter. Table 1 shows the scaling factor K_s, maximum coefficient α_{max} (note: minimum coefficient is ± 1), and minimum number of bits b_{min} for each order approximation to the double-jump filter. Table 2 gives the integer coefficients for each of the minimum-bit realizations. Notice that most of the coefficients are ± 1, powers-of-two, or zero. None of these require LAS's. The multiplication is accomplished simply by proper wiring.

4 Conclusion

We have chosen to implement the 24-th order filter with 5.673-bit coefficients using 16-bit LAS's and CSD multipliers. Figure 5 gives the frequency response of this filter. Looking at the filter coefficients in Table 2, we see that multipliers are required only for the coefficients -7, 34, and 51. Two LAS's are needed for $7x = 8x - x$ (note: we move the negative sign to the adder/subtracter) and $34x = 32x + 2x$. Although these coefficients appear twice in the $\alpha(i)$'s, they require only one multiplier each for a total of two LAS's. The single number 51 requires 3 LAS's ($51x = 32x + 16x + 2x + 1$), but only occurs once in the $\alpha(i)$'s. Thus the multipliers require only $2 + 3 = 5$ LAS's. We have two zero coefficients, so we need only $N - 2 = 24 - 2 = 22$ LAS's for the delay line. Hence, the total number of LAS's for the entire 24-th order FIR filter is 27. Using 16-bit LAS's this requires 270 CLB's which will fit very nicely on a single XC4010 FPGA.

Order	K_s	α_{max}	b_{min}
4	8.5504	9	3.096
6	4.4389	4	2.15
8	30.0023	30	4.907
9	16.8304	17	4.073
12	37.1683	37	5.216
16	26.0642	26	4.704
18	39.0162	39	5.286
24	51.0203	51	5.673

Table 1: Minimum Bits as a Function of Filter Order

Order	α_0	α_1	α_2	α_3	α_4	α_5	α_6
4	1	5	9				
6	-1	0	3	4			
8	-1	-4	2	20	30		
9	1	-2	-2	6	17		
12	-1	1	-1	-5	3	24	37
16	1	-1	0	1	-1	-4	2
	17	26					
18	1	1	-1	-1	2	-1	-5
	3	26	39				
24	1	0	-1	1	1	-1	-1
	2	-1	-7	4	34	51	

Table 2: Coefficients of Minimum-Bit Filter

References

[1] D. Kodek and K. Steiglitz, "Filter-length word-length tradeoffs in FIR digital filter design," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, pp. 739–44, Dec. 1980.

[2] D. Kodek, "Design of optimal finite wordlength FIR digital filters using integer programming techniques," *IEEE Transactions on Acoustics,*

Speech and Signal Processing, vol. 28, pp. 304–8, June 1980.

[3] W. Niedringhaus, K. Steiglitz, and D. Kodek, "An easily computed performance bound for finite wordlength direct-form FIR digital filters," *IEEE Transactions on Circuits and Systems*, vol. 29, pp. 191–3, Mar. 1982.

[4] D. Kodek, "An algorithm for the design of optimal finite word-length FIR digital filters," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, (Denver, CO), pp. 73–6, Apr. 1980.

[5] D. Kodek and K. Steiglitz, "Comparison of optimal and local search methods for designing finite wordlength FIR digital filters," *IEEE Transactions on Circuits and Systems*, vol. 28, pp. 28–32, Jan. 1981.

[6] R. Hartley, "Optimization of canonic signed digit multipliers for filter design," in *Proceedings IEEE Internationl Symposium on Circuits and Systems*, (Singapore), pp. 1992–1995, June 1991.

[7] K. Hwang, *Computer Arithmetic : Principles, Architecture, and Design*. New York: Wiley, 1979.

[8] D. Ait-Boudaoud and R. Cemes, "Modified sensitivity criterion for the design of powers-of-two FIR filters," *Electronics Letters*, vol. 29, pp. 1467–1469, Aug. 1993.

[9] B.-R. Horng, H. Samueli, and J. Wilson, A.N., "The design of low-complexity in linear-phase FIR filter banks using powers-of-two coefficients with an application to subband image coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 1, pp. 318–324, Dec. 1991.

[10] A. Mahmood, "An improved iterative design for powers-of-two coefficient FIR filters," in *IEEE International Conference on Systems Engineering*, ((Cat. No.91CH3051-0) Dayton, OH, USA), pp. 375–378, Aug. 1991.

[11] Z. Jiang, "Fir filter design and implementation with powers-of-two coefficients," in *ICASSP-89: 1989 International Conference on Acoustics, Speech and Signal Processing*, ((IEEE Cat. No.89CH2673-2), Glasgow, UK), pp. 1239–1242, May 1989.

[12] H. M. Kim, "FIR linear phase filter design using coefficients +1, 0, -1 and multiple integrator," *Journal of the Korean Institute of Telematics and Electronics*, vol. 26, pp. 151–159, Dec. 1989.

[13] J. Noonan and D. Marquis, "New algorithm for the design of linear phase fir filters with +1, -1 and 0 coefficients," *Signal Processing*, vol. 17, pp. 81–85, May 1989.

[14] K. Feher, "Modems for emerging digital cellular-mobile radio systems," *IEEE Transactions on Vehicular Technology*, vol. 40, pp. 355–365, May 1991.

[15] C. Liu and K. Feher, "Bit error-rate performance of $\pi/4$-DQPSK in a frequency-selective fast rayleigh fading channel," *IEEE Transactions on Vehicular Technology*, vol. 40, pp. 558–568, Aug. 1991.

[16] U. Guo and K. Feher, "Modem/radio IC architectures for ISM-band wireless applications," *IEEE Transactions on Consumer Electronics*, vol. 39, pp. 100–106, May 1993.

[17] C. Palmer and K. Feher, "Performance of $\pi/4$-SQAM in a hard-limited channel in the presence of AWGN," *IEEE Transactions on Broadcasting*, vol. 39, pp. 301–306, June 1993.

[18] P. Leung and K. Feher, "FQPSK: A superior modulation technique for mobile and personal communications," *IEEE Transactions on Broadcasting*, vol. 39, pp. 288–294, June 1993.

[19] M. Soderstrand, W. Chan, H. Choi, R. Strandberg, R. Atienza, and K. Feher, "DS-SS and higher-speed FH-SS modem VLSI implementations," *Document No. IEEE P802.11-94/06*, Jan. 1994.

[20] R. Atienza, W. Chan, W. Gao, M. Soderstrand, and K. Feher, "Experimental evaluation of DQPSK and FQPSK for the DS-SS and IR applications," *Document No. IEEE P802.11-94/52*, Mar. 1994.

[21] The Math Works, Inc., 24 Prime Park Way, Natick, MA 01760, *MATLAB User's Guide*, Aug. 1992.

[22] N. Sawyer, "Logic synthesis techniques for FPGAs," *Electronic Product Design*, vol. 13, no. 10, pp. 43–44, 1992.

[23] Xilinx, Inc., 2100 Logic Drive, San Jose, CA, *XACT Development System Reference Guide*, 1993.