CANONICAL SIGNED DIGIT REPRESENTATION FOR FIR DIGITAL FILTERS

Reid M. Hewlitt
Elec. and Comp. Engr. Dept.
University of Texas at Austin
Austin, TX 78712

Earl S. Swartzlander, Jr.
Elec. and Comp. Engr. Dept.
University of Texas at Austin
Austin, TX 78712

Abstract - In this paper, it is shown that the use of a canonical signed digit (CSD) representation of the filter coefficients can significantly reduce the complexity of the hardware implementation of digital FIR filters. This paper presents an example filter design that shows the error involved in limiting the number of allowable non-zero CSD coefficients for a real FIR bandpass filter. If not done carefully, brute force limiting can lead to large errors in the frequency response. The error is evaluated for varying numbers of non-zero CSD coefficients. Lastly, a system level architecture with a multiplier utilizing the properties of the CSD number representation system is proposed.

PROBLEM

Canonical signed digit (CSD) representation of filter coefficients can reduce the complexity of the hardware required to realize a FIR digital filter to less than the complexity of a 2's complement representation implementation. Often, only a few non-zero CSD digits are allowed to represent each coefficient which may introduce magnitude errors in the frequency response unless the filter coefficients can be decomposed into a few powers-of-two or sums of powers-of-two. The problem addressed in this paper is to determine the error in frequency response as the number of non-zero CSD digits are reduced from the maximum down to two allowable digits. The magnitude response is only affected since FIR filter coefficient quantization does not impact the phase response assuming a symmetric filter kernel.

INTRODUCTION

In 1961, Avizienis [5] described the signed digit number system (used since the 1950s [1]) in order to improve speed in arithmetic computation. The Canonical Signed Digit (CSD) number system is a signed digit number system that minimize the number of non-zero digits and thus can reduce the number of partial product additions in a hardware multiplier. The encoding scheme uses a digit set that is ternary and each digit can be either -1, 0, or +1. Adjacent CSD digits are never both non-zero, i.e., $\mathbf{c_i * c_{i-1} = 0}$. This property implies that for an n-bit number, there are at most $\lfloor n/2 \rfloor$ non-zero digits. For a 2's complement number, there can be n non-zero digits for a n-bit number. It can also be shown that the probability of a

0-7803-6488-0/00/$10.00 © 2000 IEEE

digit being zero is roughly 2/3 for CSD and exactly 1/2 for 2's complement. The following table shows an example for n = 3 where $\bar{1}$ represents -1. As can be seen in Table 1, for negative numbers the number of non-zero digits is less for the CSD representation than the 2's complement representation, 9 versus 12.

Table 1. Three Digit Canonical Signed Digit Numbers.

Number	2's Complement	Canonical Signed Digit
3	011	10$\bar{1}$
2	010	010
1	001	001
0	000	000
-1	111	00$\bar{1}$
-2	110	0$\bar{1}$0
-3	101	$\bar{1}$01
-4	100	$\bar{1}$00

CSD encoding is similar to Booth encoding and can be accomplished by analyzing pairs of adjacent digits from the LSB end to the MSB+1 end. If the number is negative, the MSB+1 (x^*_n) is 1, otherwise, it is a 0. This can be implemented as a copy of the MSB. Table 2 (from [1]) and the flow chart of Figure 1 provide a method to convert 2's complement numbers to CSD numbers. This method was used to create Table 1. The digits x_i and x_{i+1} are adjacent digits of the 2's complement number and the digits, c_i, are the CSD digits. The flow chart shows the algorithm to convert 2's complement filter coefficients, X, to CSD representation of the filter coefficient, C, where $X = x^*_n x_{n-1} x_{n-2} x_{n-3} \ldots x_3 x_2 x_1 x_0$ and $C = c_{n-1} c_{n-2} c_{n-3} \ldots c_3 c_2 c_1 c_0$.

Table 2. Conversion of 2's Complement Numbers to CSD Numbers

carry-in	x_{i+1}	x_i	carry-out	c_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	1	-1
1	0	0	0	1
1	0	1	1	0
1	1	0	1	-1
1	1	1	1	0

Digital FIR filters can take advantage of the CSD representation. FIR filters are computationally more expensive than an equivalent IIR filter having the same set of frequency response specifications, but FIR filters exhibit linear phase response which is useful in many applications such as decimation filters for Delta-Sigma oversampling analog-to-digital converters. With CSD representation, the multiplier reduces to only a few summations of partial products.

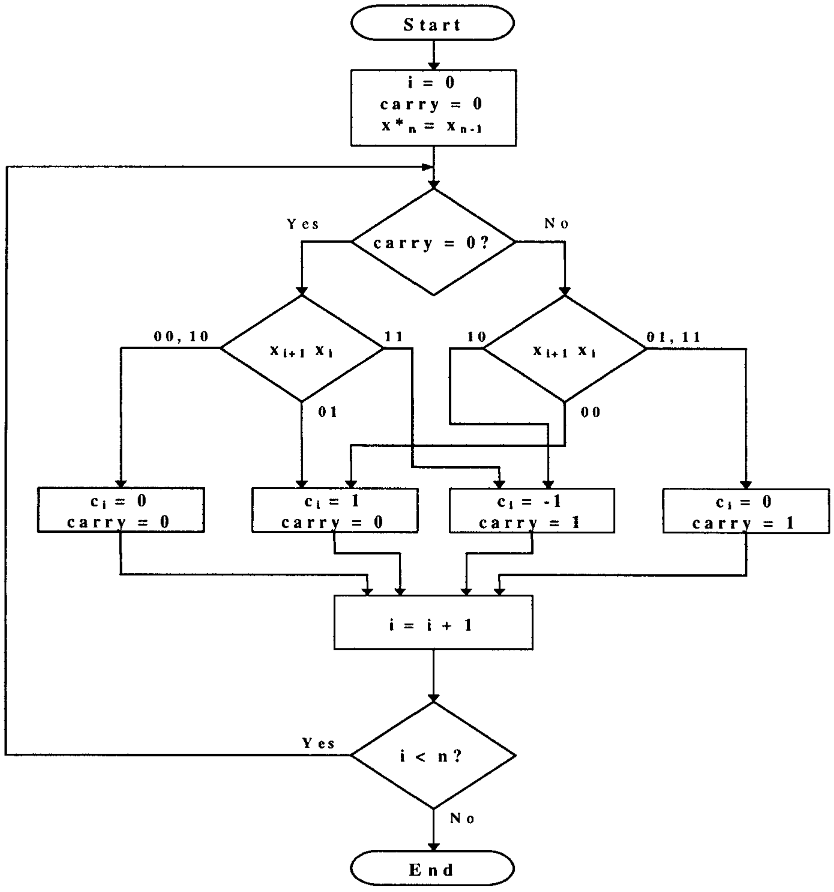

Figure 1. Conversion of 2's complement to CSD

APPROACH

A bandpass FIR filter was designed by using the Chebyshev approximation criterion and implemented by means of the Remez exchange algorithm. This passband filter is a three-band filter with passband edge frequencies of 0.2 and 0.35 and stopband frequencies of 0.1 and 0.425. The error weighting function is selected as 10, 1, and 10 with desired response of 0, 1, and 0 for each band respectively. Thus, the passband will have unity gain as shown in Figure 2. The magnitude frequency response of the filter is shown in Figure 2. This is the original filter with no coefficient digit reduction. The output of the filter is given by the following equation.

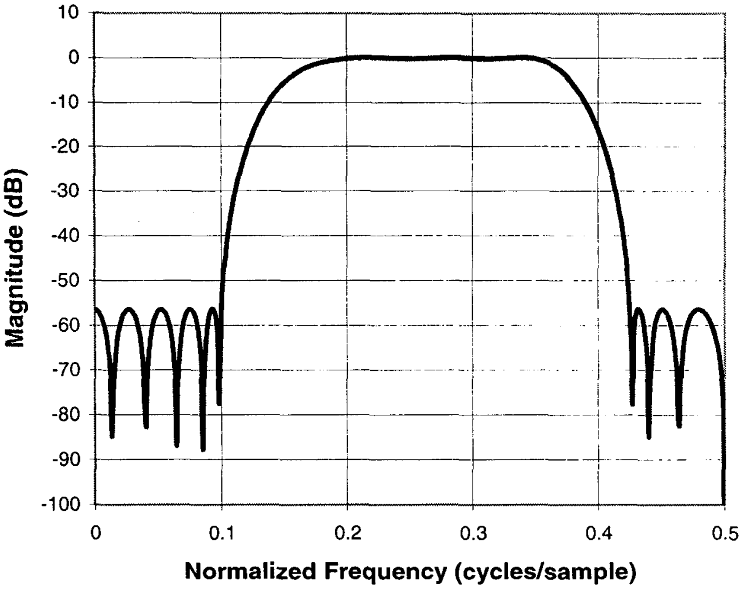

Figure 2. Frequency Response of M = 32 FIR digital filter.

$$y(k)=\sum_{i=0}^{M-1}h(i)x(k-i).$$

The filter coefficients are symmetrical (i.e., h(i) = h(31-i) for i = 0, 1, 2,...15). Thus like terms can be grouped for efficient hardware utilization.

$$y(k) = h(0)[x(k) + x(k-31)] + h(1)[x(k-1) + x(k-30)] + \ldots$$
$$+ h(16)[x(k-15) + x(k-16)].$$

The length or order of the filter is M = 32. The filter coefficients, h(k), as output from the Remez exchange algorithm are shown in Tables 4a and 4b. The coefficients are then scaled by multiplying each by 10^5 and rounded to the nearest integer. The scaling constant was chosen so that the largest coefficient is just less than 2^{15} -1, the largest, 16-bit, 2's complement number. For this scaling, the largest coefficient is 30411. The 2's complement coefficients are then converted to CSD coefficients. Tables 4a and 4b show the coefficients in 2's complement and CSD representation. Also shown in Tables 4a and 4b are the total number of additions which drops from 220 to 147. This amounts to a roughly 33% reduction in the number of partial product additions/subtractions.

The filter can be realized by the architecture shown in Figure 3. This architecture works for both 2's complement as well as CSD number representations; however, the multiplier can be implemented in different ways. A design for the well-known radix-4 Booth multiplier is shown in Figure 4 which performs the 2's complement multiplication. An alternate CSD multiplier is shown in Figure 5. This CSD multiplier skips the zeros in the CSD coefficients. The selection and skip logic is

Table 3. Number of Adds/Subtracts per Filter

Multiplier Type	Number of Adds/Subtracts
Booth, full 16 bits	128
CSD, 7 non-zero digits	112
CSD, 6 non-zero digits	96
CSD, 5 non-zero digits	80
CSD, 4 non-zero digits	64
CSD, 3 non-zero digits	48
CSD, 2 non-zero digits	32

Table 4a. Filter Coefficients and 2's Complement Number Representation

k	h(k)	h(k) Scaled	h(k) Rounded	h(k) 16-bit, 2's complement	Total Adds
0	-0.0057534026	-575.34026	-575	1111 1101 1100 0001	6
1	0.00099026691	99.026691	99	0000 0000 1100 0011	4
2	0.0075733471	757.33471	757	0000 0010 1111 0101	8
3	-0.0065141204	-651.41204	-651	1111 1101 0111 0101	10
4	0.013960509	1396.0509	1396	0000 0101 0111 0100	8
5	0.0022951644	229.51644	230	0000 0000 1110 0110	4
6	-0.019994041	-1999.4041	-1999	0000 0111 1100 1111	4
7	0.0071369656	713.69656	714	0000 0010 1100 1010	8
8	-0.039657373	-3965.7373	-3966	1111 0000 1000 0010	6
9	0.011260066	1126.0066	1126	0000 0100 0110 0110	6
10	0.066233635	6623.3635	6623	0001 1001 1101 1111	6
11	-0.010497202	-1049.7202	-1050	1111 1011 1110 0110	6
12	0.08513616	8513.616	8514	0010 0001 0100 0010	8
13	-0.12024988	-12024.988	-12025	1101 0001 0000 0111	8
14	-0.2967858	-29678.58	-29679	1000 1100 0001 0001	8
15	0.30410913	30410.913	30411	0111 0110 1100 1011	10
16	0.30410913	30410.913	30411	0111 0110 1100 1011	10
17	-0.2967858	-29678.58	-29679	1000 1100 0001 0001	8
18	-0.12024988	-12024.988	-12025	1101 0001 0000 0111	8
19	0.08513616	8513.616	8514	0010 0001 0100 0010	8
20	-0.010497202	-1049.7202	-1050	1111 1011 1110 0110	6
21	0.066233635	6623.3635	6623	0001 1001 1101 1111	6
22	0.011260066	1126.0066	1126	0000 0100 0110 0110	6
23	-0.039657373	-3965.7373	-3966	1111 0000 1000 0010	6
24	0.0071369656	713.69656	714	0000 0010 1100 1010	8
25	-0.019994041	-1999.4041	-1999	0000 0111 1100 1111	4
26	0.0022951644	229.51644	230	0000 0000 1110 0110	4
27	0.013960509	1396.0509	1396	0000 0101 0111 0100	8
28	-0.0065141204	-651.41204	-651	1111 1101 0111 0101	10
29	0.0075733471	757.33471	757	0000 0010 1111 0101	8
30	0.00099026691	99.026691	99	0000 0000 0110 0011	4
31	-0.0057534026	-575.34026	-575	1111 1101 1100 0001	6
				TOTAL ADDS	**220**

Table 4b. Filter Coefficients and CSD Number Representation

k	h(k)	h(k) Rounded	h(k) Canonical Signed Digit	Adds	Subtracts	Total
0	-0.0057534026	-575	0000 00$\bar{1}$0 0$\bar{1}$00 0001	1	2	3
1	0.00099026691	99	0000 0001 0$\bar{1}$00 010$\bar{1}$	2	2	4
2	0.0075733471	757	0000 010$\bar{1}$ 000$\bar{1}$ 0101	3	2	5
3	-0.0065141204	-651	0000 00$\bar{1}$0 $\bar{1}$00$\bar{1}$ 0101	2	3	5
4	0.013960509	1396	0000 10$\bar{1}$0 $\bar{1}$00$\bar{1}$ 0100	2	3	5
5	0.0022951644	230	0000 0001 00$\bar{1}$0 10$\bar{1}$0	2	2	4
6	-0.019994041	-1999	0000 1000 0$\bar{1}$01 000$\bar{1}$	2	2	4
7	0.0071369656	714	0000 010$\bar{1}$ 0$\bar{1}$00 1010	3	2	5
8	-0.039657373	-3966	000$\bar{1}$ 0000 1000 0010	2	1	3
9	0.011260066	1126	0000 0100 10$\bar{1}$0 10$\bar{1}$0	3	2	5
10	0.066233635	6623	0010 $\bar{1}$010 00$\bar{1}$0 000$\bar{1}$	2	3	5
11	-0.010497202	-1050	0000 0$\bar{1}$00 00$\bar{1}$0 10$\bar{1}$0	1	3	4
12	0.08513616	8514	0010 0001 0100 0010	4	0	4
13	-0.12024988	-12025	0$\bar{1}$01 0001 0000 100$\bar{1}$	3	2	5
14	-0.2967858	-29679	$\bar{1}$001 0$\bar{1}$00 0001 0001	3	2	5
15	0.30410913	30411	1000 $\bar{1}$00$\bar{1}$ 0$\bar{1}$01 0$\bar{1}$0$\bar{1}$	2	5	7
16	0.30410913	30411	1000 $\bar{1}$00$\bar{1}$ 0$\bar{1}$01 0$\bar{1}$0$\bar{1}$	2	5	7
17	-0.2967858	-29679	$\bar{1}$001 0$\bar{1}$00 0001 0001	3	2	5
18	-0.12024988	-12025	0$\bar{1}$01 0001 0000 100$\bar{1}$	3	2	5
19	0.08513616	8514	0010 0001 0100 0010	4	0	4
20	-0.010497202	-1050	0000 0$\bar{1}$00 00$\bar{1}$0 10$\bar{1}$0	1	4	5
21	0.066233635	6623	0010 $\bar{1}$010 00$\bar{1}$0 000$\bar{1}$	2	3	5
22	0.011260066	1126	0000 0100 10$\bar{1}$0 10$\bar{1}$0	3	2	5
23	-0.039657373	-3966	000$\bar{1}$ 0000 1000 0010	2	1	3
24	0.0071369656	714	0000 010$\bar{1}$ 0$\bar{1}$00 1010	3	2	5
25	-0.019994041	-1999	0000 1000 0$\bar{1}$01 000$\bar{1}$	2	2	4
26	0.0022951644	230	0000 0001 00$\bar{1}$0 10$\bar{1}$0	2	2	4
27	0.013960509	1396	0000 10$\bar{1}$0 $\bar{1}$00$\bar{1}$ 0100	2	3	5
28	-0.0065141204	-651	0000 00$\bar{1}$0 $\bar{1}$00$\bar{1}$ 0101	2	3	5
29	0.0075733471	757	0000 010$\bar{1}$ 000$\bar{1}$ 0101	3	2	5
30	0.00099026691	99	0000 0001 0$\bar{1}$00 010$\bar{1}$	2	2	4
31	-0.0057534026	-575	0000 00$\bar{1}$0 0$\bar{1}$00 0001	1	2	3
			TOTAL ADDS/SUBS			**147**

governed by the CSD conversion process shown in Table 2 and Figure 1. The skip logic signals a shift to the register and bypasses the addition process when a zero CSD is encountered. The overall complexity of the CSD multiplier is lower than the radix-4 modified Booth multiplier since only a 2:1 multiplexer is required for each bit position of the adder as opposed to a 5:1 multiplexer for each bit position of the Booth multiplier. The delay of the CSD multiplier can be less than the radix-4 Booth multiplier if the maximum number of non-zero CSD digits for the filter coefficients is limited. For the case where n = 16 with three allowable non-zero CSD digits, the CSD multiplier only requires three cycles to complete the multiplication, whereas the radix-4 Booth multiplier requires eight cycles. This leads to better than a 60% reduction in multiplier delay if the shift register delay is ignored, i.e., when we skip the zero partial products in the CSD multiplier. See Table 3 for a comparison between hardware multipliers and the

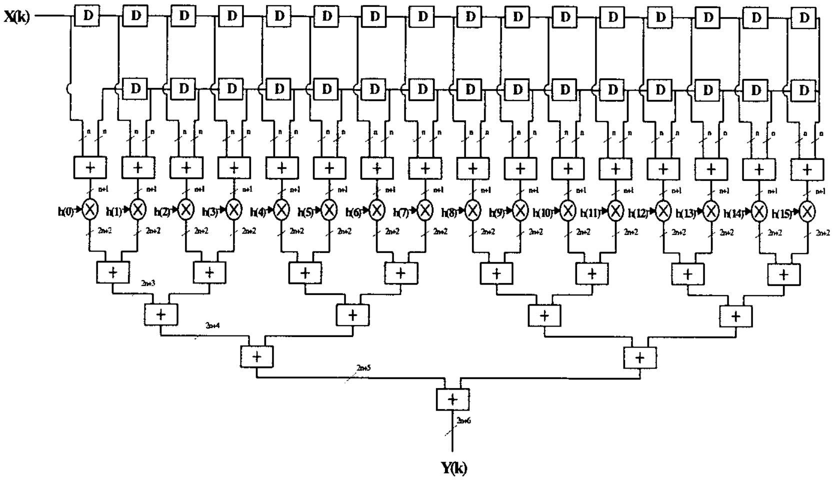
X(k)
D
h(0) h(1) h(2) h(3) h(4) h(5) h(6) h(7) h(8) h(9) h(10) h(11) h(12) h(13) h(14) h(15)
n
n+1
2n+2
2n+3
2n+4
2n+5
2n+6
Y(k)

Figure 3. Architecture of M=32 FIR Filter

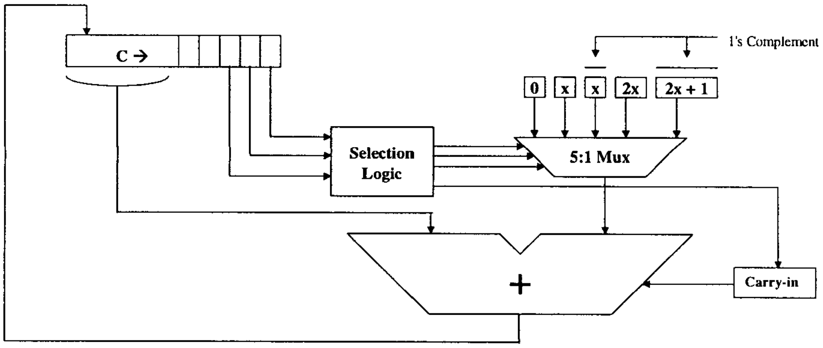

Figure 4. Radix-4 Modified Booth Multiplier

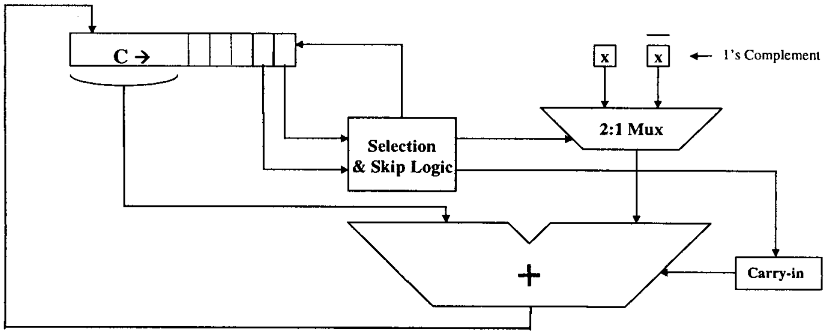

Figure 5. CSD Multiplier

Potential delay savings realized by a CSD digit-limited implementation. A spreadsheet was used to determine the frequency response for each CSD digit reduction for 16-bit numbers. For n = 16, there are maximum of eight possible non-zero digits for each filter coefficient, although, as shown in Table 4a and 4b, there are no more than seven non-zero digits required for this particular filter. Thus, the number of non-zero CSD digits will be limited from seven down to two in single digit increments (i.e. 7, 6, 5, 4, 3, & 2).

RESULTS

Table 5 shows the decimal values of the filter coefficients for each set of CSD digits. Note that there is very little effect on the coefficients until only three non-zero CSD digits are allowed. The filter frequency responses for 4, 3 and 2 non-zero CSD digits are shown in Figures 6, 7, and 8, respectively. The phase response is unaffected by coefficient quantization and thus exhibits no error. No appreciable difference is seen for allowable CSD digits above 4. The error in the magnitude response is plotted in Figure 9. Most of the error occurs in the stopbands at the extrema and may be discounted somewhat since these errors may not affect the filter performance. As can be seen in the error plots and frequency response, the passband error is relatively insignificant for all non-zero CSD digit reductions. If the stopband limitation of the filter is arbitrarily set to at least 50dB of attenuation, then the CSD implementation with no more than 3 non-zero digits is an optimal solution.

Table 5. CSD Digit Reductions

8 CSD Digits	7 CSD Digits	6 CSD Digits	5 CSD Digits	4 CSD Digits	3 CSD Digits	2 CSD Digits
-575	-575	-575	-575	-575	-575	-576
99	99	99	99	99	100	96
757	757	757	757	756	752	768
-651	-651	-651	-651	-652	-656	-640
1396	1396	1396	1396	1392	1408	1536
230	230	230	230	230	232	224
-1999	-1999	-1999	-1999	-1999	-2000	-2016
714	714	714	714	712	704	768
-3966	-3966	-3966	-3966	-3966	-3966	-3968
1126	1126	1126	1126	1128	1120	1152
6623	6623	6623	6623	6624	6656	6144
-1050	-1050	-1050	-1050	-1050	-1048	-1056
8514	8514	8514	8514	8514	8512	8448
-12025	-12025	-12025	-12025	-12024	-12032	-12288
-29679	-29679	-29679	-29679	-29680	-29696	-28672
30411	30411	30412	30416	30400	30464	30720

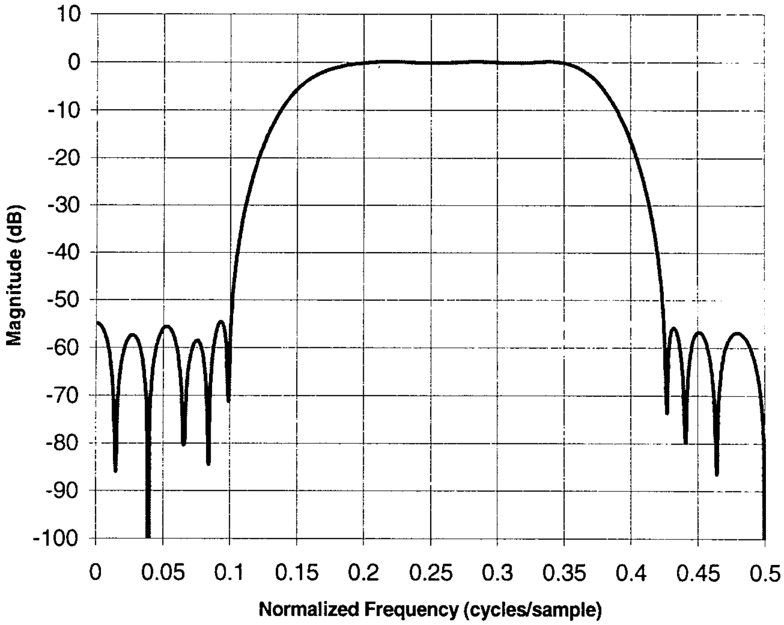

Figure 6. Frequency Response, 4 Non-Zero CSD Digits.

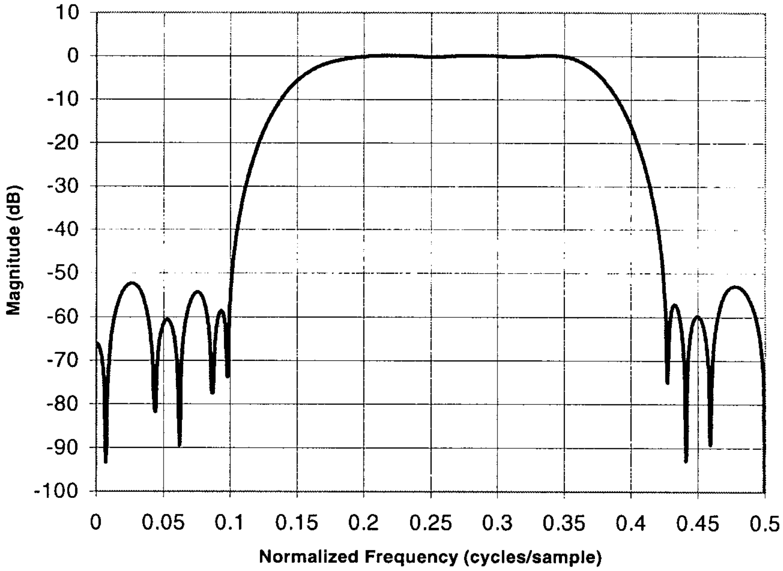

Figure 7. Frequency Response, 3 Non-Zero CSD Digits.

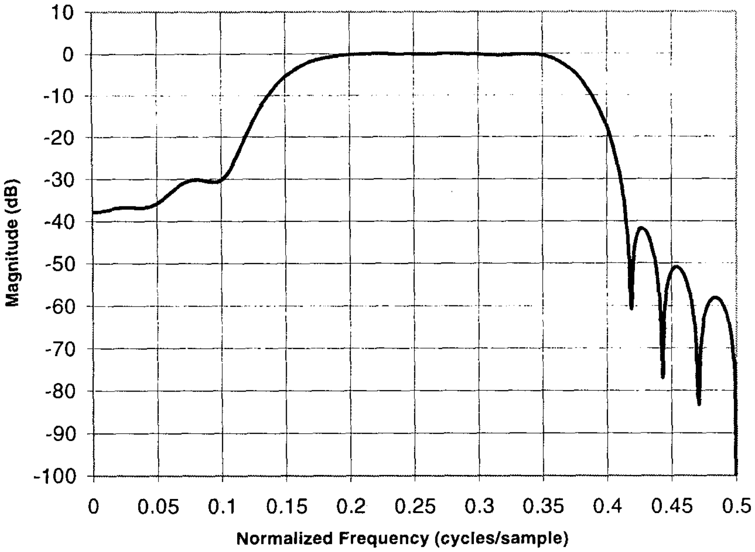

Figure 8. Frequency Response, 2 Non-Zero CSD Digits.

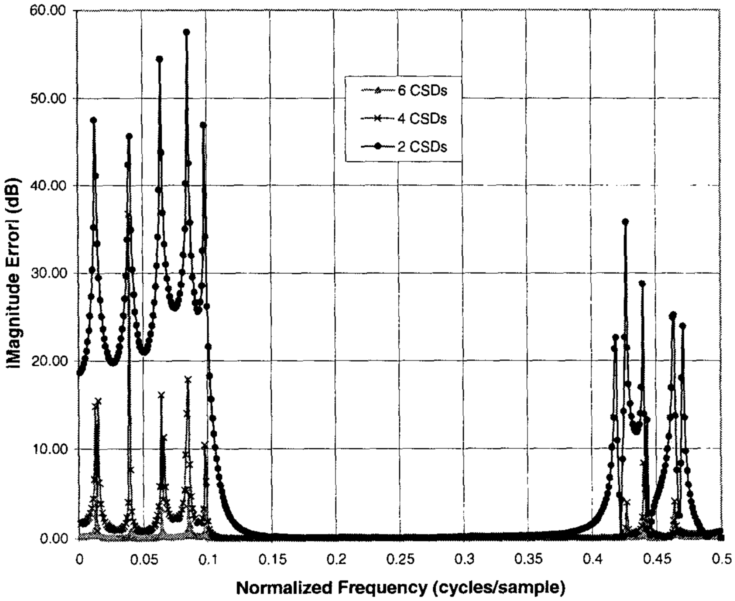

Figure 9. Error in Frequency Response.

CONCLUSIONS

CSD representation for FIR filters can reduce the delay and complexity of the hardware implementation by a significant amount. It is also shown that further reductions in the number of non-zero CSD digits of the filter coefficients reduces the multiplication process to a few additions or subtractions of partial products. For the bandpass filter discussed here, a restriction to 3 non-zero digits gives good frequency response with limited error. Thus, with 3 non-zero digits, the multiplier can be reduced to an adder that shifts and adds or subtracts at most three partial products. A full Booth-type multiplier is not required for each filter tap which reduces the overall hardware area requirements.

Filter design algorithms that restrict coefficients to powers-of-two already exist, but further work could be done in this area to formulate a design algorithm that would conform to only "easy" CSD numbers. This would reduce the expense of full multipliers, but maintain a satisfactory frequency response of the FIR filter.

References

[1] Fred J. Taylor, *Digital Filter Design Handbook*, New York: Marcel Dekker, 1983.

[2] John G. Proakis and Dimitris G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications,* 2nd Edition, New York: Macmillan, 1992.

[3] Hwan-Rei Lee, et al, "A new hardware-efficient architecture for programmable FIR filters," *IEEE Transactions on Circuits and Systems*, Vol. 43, pp. 637-644, 1996.

[4] Henry Samueli, "An improved search algorithm for the design of multiplierless FIR filters with powers-of-two coefficients," *IEEE Transactions on Circuits and Systems*, Vol. 36, pp. 1044-1047, 1989.

[5] Algirdas Avizienis, "Signed-digit number representations for fast parallel arithmetic," *IRE Transactions on Electronic Computers*, Vol. EC-10, pp. 389-400, 1961.

[6] Kei-Young Khoo, et al, "A programmable FIR digital filter using CSD Coefficients," *IEEE Journal of Solid-State Circuits*, Vol. 31, pp. 869-874, 1996.

[7] Behrooz Parhami, *Computer Arithmetic: Algorithms and Hardware Designs,* New York: Oxford University Press, 2000.