論文 / 著書情報
Article / Book Information

Title	Design of FIR Digital Filters with CSD Coefficients Having Power-of-Two DC Gain and Their FPGA Implementation for Minimum Critical Path
Authors	Mitsuru Yamada, Akinori Nishihara
Citation	IEICE Trans. Fundamentals., Vol. E84-A, No. 8, pp. 1997-2003
Pub. date	2001, 8
URL	http://search.ieice.org/
Copyright	(c) 2001 Institute of Electronics, Information and Communication Engineers

PAPER

Design of FIR Digital Filters with CSD Coefficients Having Power-of-Two DC Gain and Their FPGA Implementation for Minimum Critical Path

Mitsuru YAMADA[†a)] ***and*** **Akinori NISHIHARA**[†], ***Regular Members***

SUMMARY For low-complexity linear-phase FIR digital filters which have coefficients expressed as canonic signed digit (CSD) code, a design method to impose power-of-two DC gain is proposed. Output signal level can easily be compensated to that of input so that cascading many stages do not cause any gain errors, which are harmful in, for example, high precision measurement systems. The design is formulated as an optimization problem with magnitude response constraints. The integer linear programming modified for CSD codes is solved by the branch and bound method. The design example shows the effectiveness of the obtained filter in comparison with existing CSD filters. Also, an evaluation method for the area to implement the filter into field programmable gate array (FPGA) is proposed. The implementation example shows that the minimum critical path is obtained with only a little increase in the die area.

key words: *FIR digital filters, CSD, optimization, integer programming, FPGA*

1. Introduction

In digital signal processing for industrial measurement, for instance an analog-to-digital conversion for signals from a detector, the sampling rate is often changed from a range of several times to hundreds of times. The sampling rate can be converted by using a multistage digital filters and up or down samplers.

In image signals, most of signal power lies around DC and lower part of the frequency. Thus, it is desirable that no DC gain error occurs in the filter. In each filter of the multistage system, high speed, less computational complexity, small die area and linear phase are preferred. Filters with coefficients using canonic signed digit (CSD) code [1]–[4] are suitable for such purposes.

The design method [2] allows arbitrary DC gain to avoid deterioration in frequency response caused by non-uniform nature of coefficient values. Thus, a gain compensator is needed to compensate total gain so that the magnitude of the output signal of the filter becomes the same as that of the input signal. This can be realized by a divider with divisor being the filter's DC gain.

However, a divider has high computational complexity and low computational speed. If the divider is replaced by wired shifters and adders, the result of compensation becomes only an approximate value.

We propose a design method of FIR digital filters with CSD coefficients having power-of-two DC gain. In this filter gain can be compensated by using a wired shift operation without any errors. Design of these filters is formulated as the integer programming with the FIR filter's coefficient as variables. The tolerant range of desired frequency response of the filter is specified as constraints of the integer programming. Design examples show that good frequency responses are obtained in some cases even with tight constraint in DC gain. The resultant filters are superior to conventional ones [2] with gain compensation in terms of gain error.

We also show an implementation example by using field programmable gate array (FPGA). Recently, FPGAs become popular devices for not only trial production of digital circuits but also mass production [8]. Thus, study of implementation of high speed and high precision digital filters by using FPGAs will be interesting. FPGAs are different from semi-custom LSIs such as gate-arrays from certain points of view of configurable minimum logic elements. The method [3], [4] assume the use of semi-custom LSIs, and the filter complexity is evaluated in terms of the number of adders and die area. In the case of FPGA implementation, different evaluation method which considers internal structure of FPGAs is necessary. By taking that evaluation in account, we can modify the filter structure such that critical path length is minimized with a little increase in the die area (the number of logic blocks).

It is thus shown that high-precision and low-complexity filters are obtained and that those filters are modified to have very high speed with a little increase in die area when implemented on FPGAs.

2. Canonic Signed Digit Representation

In the case of binary representation, a set of digit {0,1} is used as each place of values. A numerical representation by a set of digit $\{\bar{1}, 0, 1\}$ is called SD (Signed-digit) representation, where $\bar{1}$ denotes -1. Among the SD numbers having minimum weight, the numbers which

Manuscript received August 17, 2000.
Manuscript revised February 9, 2001.

[†]The authors are with the Graduate School of Science and Engineering, Tokyo Institute of Technology, Tokyo, 152-8552 Japan.
a) E-mail: myamada@ec.ss.titech.ac.jp

do not have consecutive non zero digits is called CSD (Canonical signed-digit) representation. For example, $111_{(binary)}$ is represented as $100\bar{1}_{(CSD)}$. In CSD representation, the number of non-zero digits is always less compared to conventional binary representation. Multipliers with CSD coefficients can be realized using wired shifters, adders and subtracters. So the number of non-zero digits is critical in evaluation of computational complexity. The amount of calculation is less when CSD representation is used. The amount of calculation in a filter is further reduced by limiting the number of non-zero digits in its coefficients, in return for deterioration in frequency response.

3. Design Procedure

3.1 Frequency Response Constraints

Let the coefficients of the filter be integers so that the LSB is placed at 2^0. Let the DC gain of the filter, the sum of the coefficients of the filter, be 2^N, where N is an integer. This is the same as the case where the coefficients are represented by fractions with the LSB located at 2^{-N} and the DC gain of the filter is unity. In this paper, we choose the former definition.

Let the magnitude-frequency response of a linear-phase FIR filter be $H(\omega)$, and $\omega_0, \omega_1, \cdots, \omega_{K-1}$ are frequency points for evaluation. Specifications of the frequency response are given as shown in Fig. 1. Constraints of the frequency response are expressed as follows,

$$H(\omega_0) = 2^N$$

$$H(\omega_1) \leq H_a$$

$$H(\omega_1) \geq H_b$$

$$\vdots$$

$$H(\omega_{K-1}) \leq H_c$$

$$H(\omega_{K-1}) \geq H_d$$

where Ha and Hb are magnitude limits in the passband, and Hc and Hd are magnitude limits in the stop-band. Using the negative sign, we can reverse the direction of inequality. The left-hand side of the above constraints can be represented as

$$\mathbf{H} = [H(\omega_0), -H(\omega_0), \cdots, H(\omega_{K-1}), -H(\omega_{K-1})]^T. \quad (1)$$

Let the right-hand side of the above constraints be

$$\mathbf{d} = [2^N, -2^N, H_a, -H_b, \cdots, H_c, -H_d]^T. \quad (2)$$

Then, the constraints are expressed as

$$\mathbf{H} \leq \mathbf{d}. \quad (3)$$

On the other hand, let $\mathbf{x}$ be the coefficient vector of an FIR filter as

$$\mathbf{x} = [x_1, x_2, \cdots, x_M]^T \quad (4)$$

where odd number of taps is assumed here. The case for even number of taps can be treated similarly. Let $\mathbf{a}(\omega)$ be a trigonometric function vector of frequency as

$$\mathbf{a}(\omega) = [1, 2\cos\omega T, 2\cos 2\omega T, \cdots, 2\cos(M-1)\omega T]^T. \quad (5)$$

Then, frequency response of linear-phase FIR digital filters $H(\omega)$ is expressed as

$$H(\omega) = \mathbf{a}(\omega)^T \mathbf{x}. \quad (6)$$

Let $\mathbf{A}$ be

$$\mathbf{A} = [\mathbf{a}(\omega_0), -\mathbf{a}(\omega_0), \cdots, \mathbf{a}(\omega_{K-1}), -\mathbf{a}(\omega_{K-1})]^T. \quad (7)$$

Using Eqs. (3), (6) and (7), the constraints are written as

$$\mathbf{A}\mathbf{x} \leq \mathbf{d}. \quad (8)$$

3.2 Objective Function

The objective function z of this optimization problem works so that the frequency response nears H_a as shown in Fig. 1, that is, z is defined as

$$z = \sum_{i=0}^{p} H(\omega_i). \quad (9)$$

The objective function is a linear formula to be used in the linear programming. Then the optimization problem is solved to maximize z.

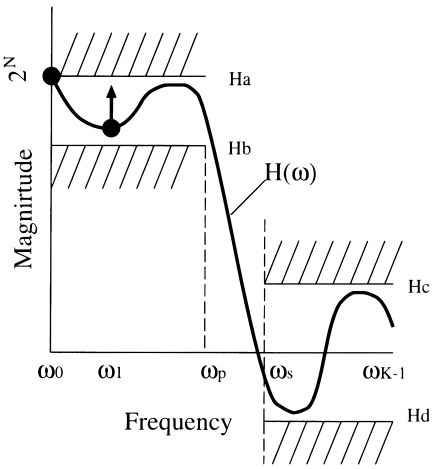


Fig. 1 Specifications of a frequency response.

3.3 Optimization Algorithm

Filter coefficients can be obtained by solving the optimization problem composed of the above constraints and the objective function, that is

Maximize

$$z = \sum_{i=0}^{p} H(\omega_i) \tag{10}$$

Subject to

$$\mathbf{Ax} \leq \mathbf{d}, \tag{11}$$

where the elements of optimal solution of $\mathbf{x}$ are integers represented by CSD numbers.

This problem is solved by the integer programming [6], [7]. The linear programming is performed by the simplex method [6]. The search of the solution is represented as follows.

At first, a real number solution is obtained under no CSD constraint on Eq. (11) as shown in node 1 of Fig. 2. Second, x_m, an element of the solution is constrained so that it can be represented by a CSD number. A new constraint $x_m \leq x_a$ is added to Eq. (11) in the left branch of the searching tree. This x_a is the maximum CSD number less than x_m. Then linear programming is performed again as shown in node 2 of Fig. 2. On the other hand, a new constraint $x_m \geq x_b$ is added to Eq. (11) in the right branch, where x_b is the minimum CSD number greater than x_m. Then, linear programming is performed in the same manner as the left branch as shown in node 3 of Fig. 2. Thus it is possible that that element of solution becomes CSD number in node 2 or node 3 in Fig. 2.

When there are feasible solutions at both nodes, a larger objective function node is selected for the next search. Then a new constraint is added to the previous constraint in the same way as above so that another element of the solution becomes CSD number. If there is no feasible solution in one side, that node is terminated. If there is no feasible solution in both sides, the search goes back to a node which has feasible solutions in both sides, and the other side is selected for the next search.

When all elements of a solution can be represented by CSD numbers, let this solution be a temporary solution. And let the value of the objective function be the temporary maximum value. Then search goes back to a node which has feasible solutions in both sides, and the other side is selected for the next search. At that moment, if the upper bound of the objective function is less than the temporary maximum value, this node is terminated, and go back to the node towards the top of the search tree which has feasible solution in both sides.

If all elements of the solution are obtained as integer numbers represented by CSD, and the value of the objective function is greater than the temporary maximum value, the temporary solution is replaced by the latest solution. The temporary maximum value is also replaced by the latest value of the objective function. Finally, if no feasible node remains, the temporary solution becomes the optimum solution.

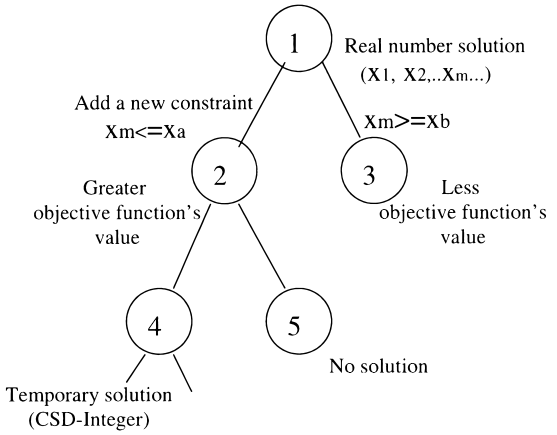


Fig. 2 Search tree of the branch and bound method.

3.4 Programming and Execution

The algorithm is programmed by using MATLAB, and Fig. 3 shows the flow chart. The amount of the program is approximately 800 lines. In the case of the example as described later which has 64 frequency response evaluation points, 5 variables and 56 nodes, the search was executed on Compaq Alpha Server 8400 type computer in 4 minutes. Similarly, a personal computer (Pentium II, 350 MHz) needs double the execution time. The execution time is much affected from the frequency response constraints. If the constraint tolerance is wide, there are many feasible solutions and the search needs much time.

4. Design Example

As an example a half-band filter is designed by using the proposed method. In the case of half-band filters, about a half of coefficients of the filter are forced to be zero. Then the Haar condition is not satisfied and the optimization cannot be performed. Therefore, we use the Ansari's method [5].

First, a type-2 one-band filter having gain 2^{N-1} is designed. The type-2 filters have a zero at normalized frequency π. Second, zeros are inserted to the coefficients, and the center of coefficients is given as a half of DC gain so that the magnitude response is double the type-2 filter's. From the above, a type-1 half-band filter can be designed.

Table 1 shows the specifications and the result of the design example. In this specification, the number

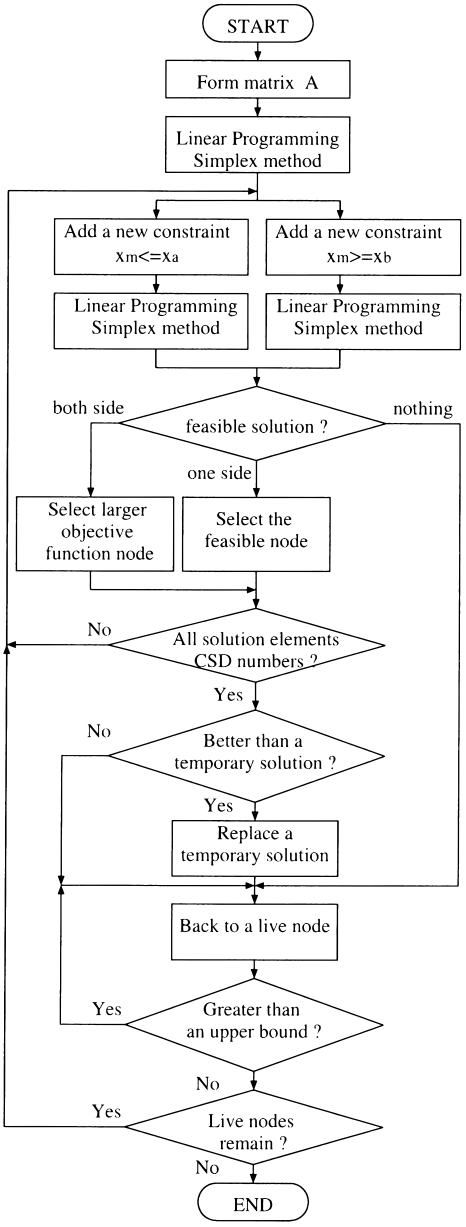


Fig. 3 Flow chart of the optimization.

Table 1 Specifications and results.

# of taps	19
Coefficient	8 bit
# of nonzero digits	3
DC gain	2^8
H_b	0.9×2^8
ω_p	0.155

$$
\begin{aligned}
h_8 &= 128 \\
x_1 = h_7 &= h_9 = 78 \\
h_6 &= h_{10} = 0 \\
x_2 = h_5 &= h_{11} = -19 \\
h_4 &= h_{12} = 0 \\
x_3 = h_3 &= h_{13} = 6 \\
h_2 &= h_{14} = 0 \\
x_4 = h_1 &= h_{15} = -1 \\
x_5 &= 0
\end{aligned}
$$

of tap of the half-band filter is 19, thus the number of tap of one-band filter which is to be obtained first is 10.

Shorter word length of coefficients reduces the possibility of existence of solutions which satisfy the specification. On the other hand, longer word length of coefficients improves the frequency response of the filter at the price of large die area due to larger word length for adders and delay elements. Therefore, the word length of coefficients is chosen as minimum under the condition that the optimization solution exists. The optimization is repeated with increasing the word length until the solution exists. In this design of one-band filter, solutions exist when the word length of the coefficients is more than 6-bit. Then the word length of the coefficients of the one-band filter is determined as 7-bit and the half-band filter's is 8-bit.

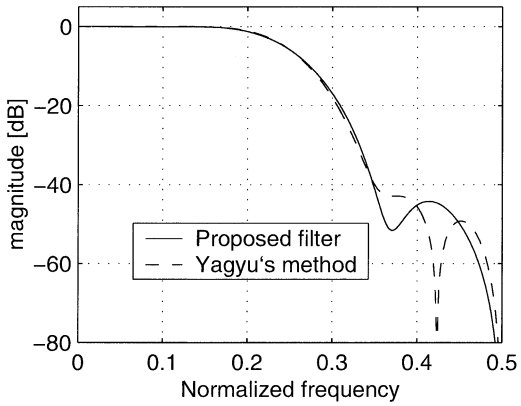


Fig. 4 Design example of a half-band filter.

The larger the number of non-zero digits of the coefficients, the better frequency response can also be obtained. However, the area increases because the number of adder increases. The number of non zero digit of the coefficients is given as 3-bit in this example. The DC gain is determined to 2^8 from the word length of coefficients.

When H_b is chosen near the DC gain, the number of feasible solutions is reduced and the optimizing calculation time becomes short. If H_b is too close to the DC gain, there is no feasible solution. In this example, H_b is given as 0.9×2^8. The normalized pass-band edge frequency, ω_p is specified as 0.155. The optimization described in section 3.3 results in filter coefficients $h_1, \cdots, h_{15}$ shown in Table 1. The number of taps of the filter becomes 15 instead of 19 because coefficients at both ends are solved as 0 in the optimization. Figure 4 shows the magnitude-frequency response of the obtained filter. Figure 4 also shows a response obtained by the conventional Yagyu's method [2]. Even under the stringent DC gain constraint, the similar response as conventional's can be obtained.

Figure 5 shows the block diagram of the designed filter. In this case, the gain compensator is a wired shift of 2^{-8} which corresponds to the DC gain of 2^8. The 20-bit internal signal is truncated to 12-bit output signal. When it is necessary to avoid a 0.5-LSB DC-offset due to the truncation, the gain compensator should use the

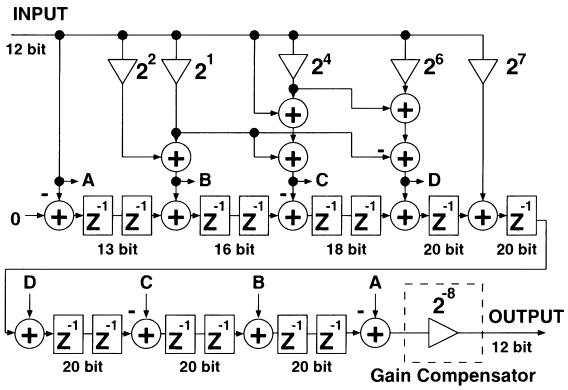


Fig. 5 Block diagram of the designed FIR filter.

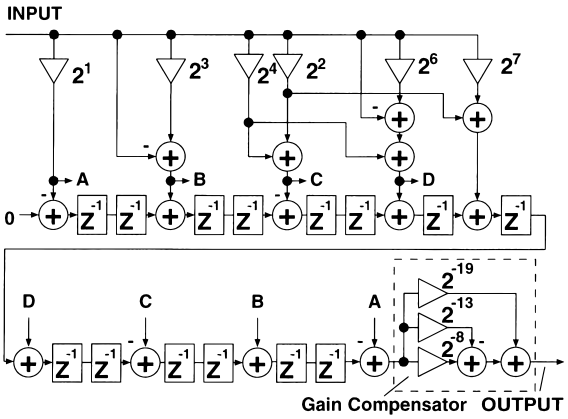


Fig. 6 Block diagram of the Yagyu's FIR filter with a gain compensator.

Table 2 Results of the proposed and Yagyu's methods.

	Proposed	*Yagyu*
DC gain	1.00000	0.99953
Stop-band gain	−44.3 dB	−42.9 dB
Number of adders	14	16

roundoff operator. The roundoff operator can be realized by an adder. Figure 6 shows the block diagram of the filter designed by Yagyu's method [2] with an additional gain compensator. In that case, the DC gain of the filter is 264. Then, the gain compensator is determined as $2^{-8} - 2^{-13} + 2^{-19}$ so that the DC gain is close to and does not exceed 1.0.

Table 2 shows the results of the proposed and the conventional methods. The DC gain is exactly 1.0 by using the proposed method. In contrast, there is an error of 4.7×10^{-4} by using the conventional method. This means that the error exceeds 1-LSB in 12-bit word length of 2's complement. Therefore, in the case where the word length exceed 12-bit, the proposed method is better than the conventional method in terms of a DC gain error.

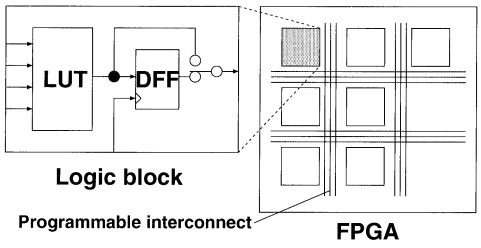


Fig. 7 Typical internal structure of FPGAs.

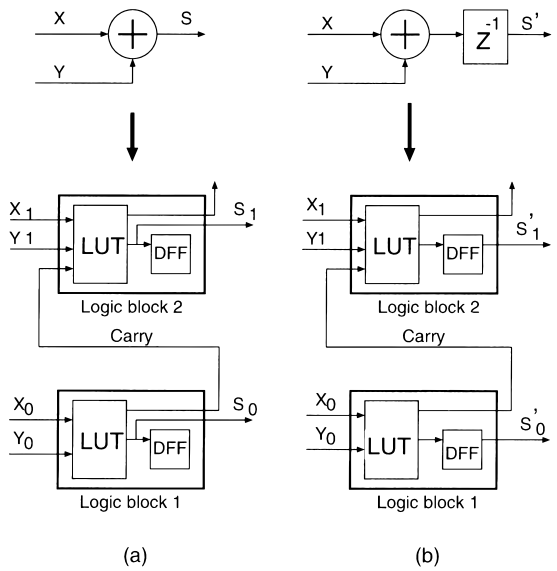


Fig. 8 Implementation of an adder and an adjacent delay element in FPGAs.

5. FPGA Implementation

5.1 Features of FPGAs

In semi-custom LSIs, such as gate arrays or standard cells, digital circuits are implemented by macro-cells which consist of transistors and interconnects to configure certain logic functions. In contrast, most of FPGAs have the internal structure shown in Fig. 7 [8]. FPGAs consist of larger logic blocks that are configured. The logic blocks consist of a D-type flip-flop (DFF) and a configurable look-up table (LUT) which has several inputs. Therefore, area evaluation for FPGAs should be different from that for semi-custom LSIs.

Suzuki et al. used the area ratio of an adder and a delay element as 1:0.9 in semi-custom LSIs [4]. FPGA implementation of an adder and the following delay element are shown in Fig. 8. Figure 8(a) shows the case where the only an adder is implemented in an FPGA, and Fig. 8(b) shows the case where an adder and a delay element are implemented. In the latter case, a delay element is implemented by enabling the DFF which is included in the adder's logic block. Thus putting a delay element just after an adder does not increase the

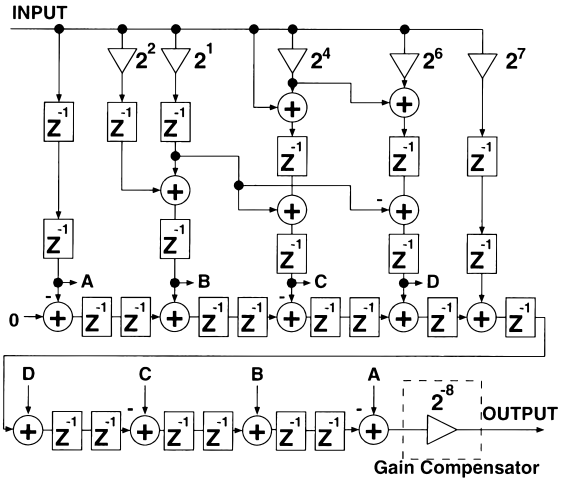


Fig. 9 Block diagram of the designed FIR filter having minimum critical path of one adder.

Table 3 Estimation of the area and the group-delay of the designed filters.

	Fig. 5	*Fig. 11*
# of adders	14	14 (100%)
# of delay elements	14	21 (150%)
Critical path length (Adder)	3	1 (33%)
Area (custom LSI)	26.6	32.9 (124%)
Area (FPGA)	20	22 (110%)
Group-delay	7	9 (129%)

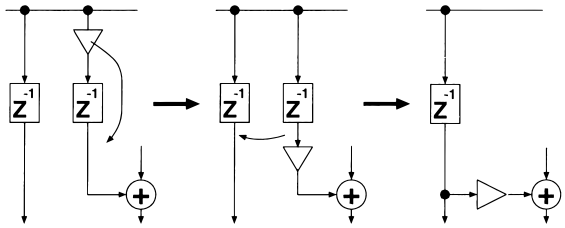


Fig. 10 Equivalent transformation of the signal flow.

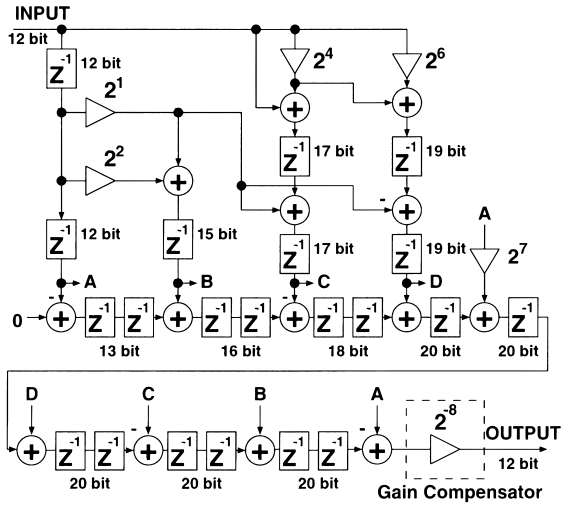


Fig. 11 Block diagram of the designed FIR filter after equivalent transformation from Fig. 9.

area at all.

The above observation shows that delay elements can be inserted between consecutive adders or subtracters as pipeline registers to minimize the critical path of the filter without increasing the area.

5.2 Minimization of the Critical Path

The input signal word length is specified as 12-bit in this design and the maximum internal word length is 20-bit. The differences of word length among adders and delay elements in the filter are ignored to make the estimation simple.

First, in order to reduce the critical path, delay elements are inserted between adders which act as parts of multiplication as shown in Fig. 9.

Suppose the propagation delay of an adder is T_a, and the number of nonzero digits of the filter coefficients is N_z. In the conventional signal flow shown in Fig. 5, the length of the critical path is $N_z T_a$. After the above pipeline registers are inserted, it becomes the minimum T_a, the propagation delay of a single adder. In this case, the length of the critical path decreases to 33% as shown in Table 3 by inserting delay elements.

Second, the number of delay elements is reduced by an equivalent transformation shown in Fig. 10. The order of a shifter and a delay element is reversed and the delays having the same input are unified. Figure 11 shows the resultant structure. The number of delay elements decreases from 25 to 21.

Table 3 shows the estimation of the area. In semi-custom LSI, the structure shown in Fig. 11 has 24% more area than that in Fig. 9. On the other hand, in the case of FPGAs, delay elements following to adders do not occupy any extra area. Then the increase in total area is estimated as 10% as shown in Table 3. This confirms that FPGA implementation has less overhead compared to custom LSI when the above speedup techniques are applied.

Insertion of delay elements increases the group delay of the filter. The number of taps of a half-band filter is $4M-1$, and so the group-delay is $((4M-1)-1)/2$ in Fig. 5. In Fig. 11, the group delay is $((4M-2)/2)+N_z-1$. If the number of taps is large and the number of nonzero digits is small, the group-delay increases only a little. Table 3 shows that in the case of Fig. 11, the increase in the group-delay is 29%.

5.3 Implementation Example

The circuit of the proposed FIR filter is described by VHDL for implementation. The word length of the delay elements and the adders are determined as a length sufficient to avoid a calculation overflow of the adders. In Figs. 5 and 11, the word length is shown near each

Table 4 Implementation result for the FPGA.

	Fig. 5	*Fig. 11*
Signal word length	12 bit	12 bit
# of Adders	14	14 (100%)
# of Delay elements	14	21 (150%)
Critical path length (Adder)	3	1 (33%)
Group-delay	7	9 (129%)
Area (Logic block)	204	215 (105%)
Actual critical path length	25.3 ns	12.7 ns (50%)
Sampling frequency	39.5 MHz	78.6 MHz

delay element. The circuit is actually implemented on XC4028XL-1 (Xilinx) using FPGA Compiler (Synopsys), and Alliance M1.4 device-fitter (Xilinx). Table 4 shows the results of the implementation.

Minimization of the critical path increases the area to 105%, where the area is measured by the number of logic blocks used. The increase in the area is less than the estimation of Table 3, because the FPGA has 2 DFFs in each logic block. Although the critical path length is estimated to decrease to 33%, the actual critical path decreases to 50%. This is because the CAD tools minimize the critical path very well in the case that 3 adders are consecutive and the carry propagation length of 3 adders is shorter than 3 times of the carry propagation length of a single adder.

The maximum sampling frequency is obtained as 78.6 MHz. It means that a high speed FIR filter is obtained by the proposed method.

The results can also be considered as trade-off among the group-delay, the area and the length of critical path. Increasing the group-delay to 129% and the area to 105% makes it possible to reduce the length of actual critical path to 50%.

This FPGA device is installed onto an evaluation board with an A-D converter and a D-A converter. The observed response is the same as Fig. 4 by using an analog gain-phase analyzer.

From the above, in implementation on FPGAs, the length of critical path of the FIR filter with CSD coefficients can be reduced without much increase in the area. Improvement of operating speed for FIR filters by using the proposed method is confirmed to be effective for FPGAs.

6. Conclusions

For low-complexity linear-phase FIR digital filters which have coefficients expressed as canonic signed digit (CSD) code, a design method to impose power-of-two DC gain is proposed. This design problem is formulated as integer programming. As a result, reasonable frequency responses are obtained even with the stringent condition on the DC gain.

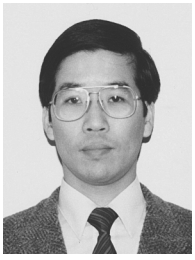
We show the area estimation method which takes internal structure of FPGAs into account. With that in mind FIR digital filter structures are modified so that the length of critical path becomes minimum of a single adder. It is proved that despite delay elements are inserted into the designed FIR filter, the area increase is only a little.

Therefore, high speed, compact and linear-phase FIR digital filters are obtained.

References

[1] Y.C. Lim and S.R. Parker, "FIR filter design over a discrete powers-of-two coefficient space," IEEE Trans. ASSP, vol.ASSP-31, no.3, pp.583–591, June 1983.
[2] M. Yagyu, A. Nishihara, and N. Fujii, "Design of FIR digital filters using estimates of error function over CSD coefficient space," IEICE Trans. Fundamentals, vol.E79-A, no.3, pp.283–290, March 1996.
[3] M. Yagyu, A. Nishihara, and N. Fujii, "Fast FIR digital filter structures using minimal number of adders and its application to filter design," IEICE Trans. Fundamentals, vol.E79-A, no.8, pp.1120–1129, Aug. 1996.
[4] K. Suzuki, H. Ochi, and S. Kinjo, "A VLSI design of FIR filter using CSD with minimum number of registers," IEICE Trans., vol.J81-A, no.2, pp.190–197, Feb. 1998.
[5] R. Ansari, "Satisfying the Haar condition in halfband FIR filter design," IEEE Trans. ASSP, vol.36, no.1, pp.123–124, Jan. 1988.
[6] H.A. Taha, Integer Programming Theory Applications and Computations, Academic Press, 1975.
[7] R.S. Garfinkel and G.L. Nemhauser, Integer Programming, Wiley-Interscience, 1972.
[8] A.K. Sharma, Programmable Logic Handbook, McGraw-Hill, 1998.

Mitsuru Yamada was born on May 22, 1960. He took B.E. and M.E. degrees from Kagoshima University, Kagoshima, Japan in 1983 and 1985, respectively. He joined JEOL Ltd., in 1985. He is now a Dr.E. student of graduate school of Science and Engineering, Tokyo Institute of Technology. His research interests are in high speed digital signal processing and its FPGA implementation.



Akinori Nishihara received the B.E., M.E. and Dr.Eng. degrees in electronics from Tokyo Institute of Technology in 1973, 1975 and 1978, respectively. Since 1978 he has been with Tokyo Institute of Technology, where he is now Professor of the Center for Research and Development of Educational Technology (CRADLE). His main research interests are in 1-D and multi-D signal processing and its application to educational technology. Dr. Nishihara served as an Associate Editor of the IEEE Transactions on Circuits and Systems II from 1995 to 1997. He was Student Activities Committee Chair of IEEE Region 10 (Asia Pacific Region) during 1995–1996, and is now IEEE Region 10 Treasurer. He also served as Editor-in-Chief of the Transactions of IEICE, Part A, from 1998 to 2000. He received the 1999 IEICE Best Paper Award and the IEEE Third Millennium Medal in 2000. He is a member of IEEE, EURASIP, ECS and JET.