AN EFFICIENT FIR FILTER ARCHITECTURE

Joseph B. Evans

Telecommunications & Information Sciences Laboratory
Department of Electrical & Computer Engineering
University of Kansas
Lawrence, KS 66045-2228

Abstract – This paper describes an efficient architecture for FIR filters. By exploiting the reduced complexity made possible by the use of sparse powers-of-two coefficients, an FIR filter tap can be implemented with only $2B$ full adders and $2B$ (or $4B$) latches, where B is the intermediate wordlength. Word and bit level parallelism allows high sampling rates, limited only by the full adder delay. This novel architecture allows the implementation of high sampling rate filters of significant length on a single field-programmable gate array (FPGA), as well implementation using more conventional VLSI techniques.

I. INTRODUCTION

In recent years, considerable attention has been placed on the implementation of signal processing algorithms in VLSI, ranging from full custom VLSI to general purpose digital signal processors. A variety of approaches to high speed implementation of FIR filters have been pursued [1, 2, 3, 4, 6, 8, 9, 10, 11, 14, 15, 17, 19]. In order to attain high performance, parallel implementation strategies such as the systolic and wavefront methods have been applied. Word-parallel, bit-parallel processing techniques appear to scale well with improvements in implementation technology and increasing demands for higher performance.

This paper presents a new parallel FIR filter building block suited for implementing filters where each of the coefficient values is a sum or difference of several power-of-two terms. It is particularly useful for the case where coefficients are a sum or difference of only two power-of-two terms. This architecture allows high sampling rate FIR filters of substantial length to be implemented on the current generation field-programmable gate arrays (FPGAs), as well as in more traditional CMOS custom and semi-custom circuitry. The high sampling rates obtained through this architecture are due to extensive pipelining; the implementation efficiency is a result of the use of the highly constrained coefficient values.

II. BACKGROUND

In binary arithmetic, multiplication by a power-of-two is simply a shift operation. Implementation of systems with multiplications may be simplified by using only a limited number of power-of-two terms, so that only a small number of shift and add operations are required.

The improvements in speed and saving in integrated circuit area are, however, achieved at the expense of a deterioration in the frequency response characteristics. The extent of the frequency response

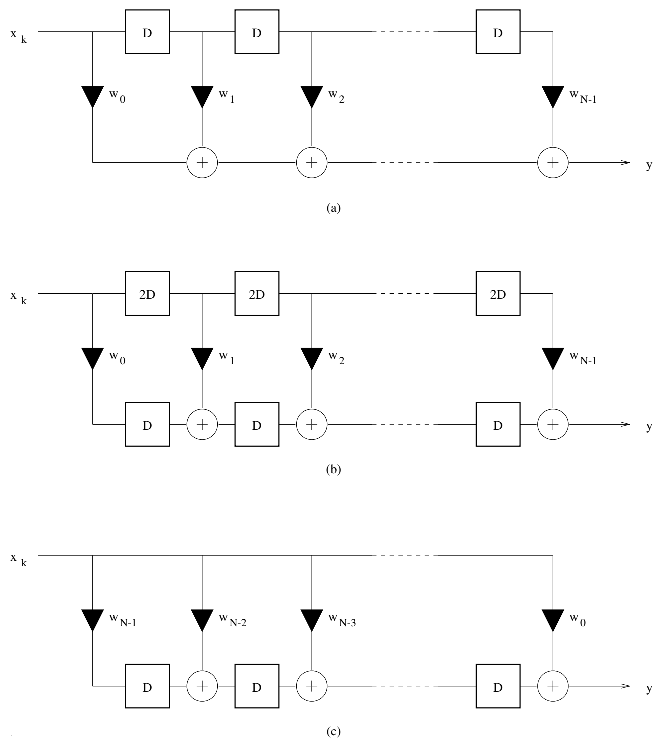

Figure 1. FIR Filter Architecture, (a) canonical form, (b) pipelined, (c) inverted form.

deterioration depends on the number of power-of-two terms used in approximating each coefficient value, the architecture of the filter, and the optimization technique used to derive the discrete space coefficient values. It was demonstrated in [13] that an FIR filter with -60dB of frequency response ripple magnitude can be realized using two power-of-two terms for each coefficient value, given that the filter is in cascade form and the coefficient values are derived using mixed integer linear programming.

The basic structure of an FIR filter is illustrated in Figure 1 (a). Using cut-set retiming, the pipelined version shown in Figure 1 (b) can be obtained. An inverted form FIR filter, which will be used in FPGA implementations is depicted in Figure 1 (c).

If the coefficient value is an integer power-of-two, or a sum of two powers-of-two, the multipliers can be replaced by shifters, as depicted in Figure 2. Since the coefficients will be fixed for this class of filter, the coefficient values can be realized by appropriately routing the inputs to the full adders in the filter structure. That is,

The author can be contacted via e-mail at evans@tisl.ukans.edu. This research is supported by the University of Kansas General Research allocation 3626-20-0038.

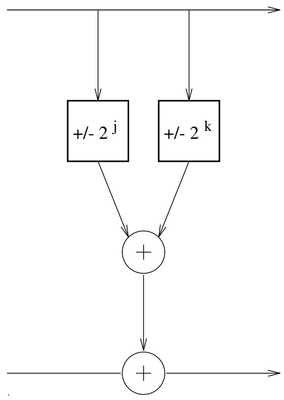

Figure 2. FIR Filter Tap Arithmetic Unit, Coefficients with Two Powers-of-Two

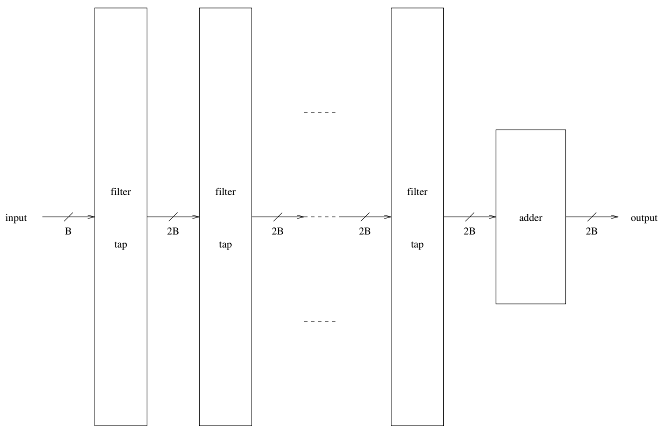

Figure 3. FIR Filter Architecture

moving the adder inputs k places to the left achieves the same effect as would a coefficient value of 2^k.

III. ARCHITECTURE

In order to attain high sampling rates using conventional FPGAs or low cost CMOS processes, bit-level parallelism is exploited. The overall filter architecture is shown in Figure 3, where the filter taps and final adder stage are shown. The adder is required to resolve the carries that are generated and propagated through the pipeline.

The structure of a filter tap is shown in Figure 4, where the internal pipelining is depicted. The two adders, which are necessary for coefficients that are a sum of two signed powers-of-two, are implemented as two rows of full adders, whose inputs are configured with the appropriate shift for the given coefficients. The sign of the coefficients is controlled by inverters. The sum and carry signals from the full adders are pipelined using a carry-save addition (CSA) techniques in order to increase the sampling rate and alleviate potential routing delays in the target implementation technologies. The input data bus passes through the bit-slice array to provide short interconnection distances to the first row of full adders. This bus may be optionally pipelined depending on the particular implementation technology, FPGA or full custom, among other factors. The hardware requirements for a tap with B_d input datapath bits and B_i intermediate accumulation path bits are then $2B_i$ full adders and a minimum of $2B_i$ flip-flops.

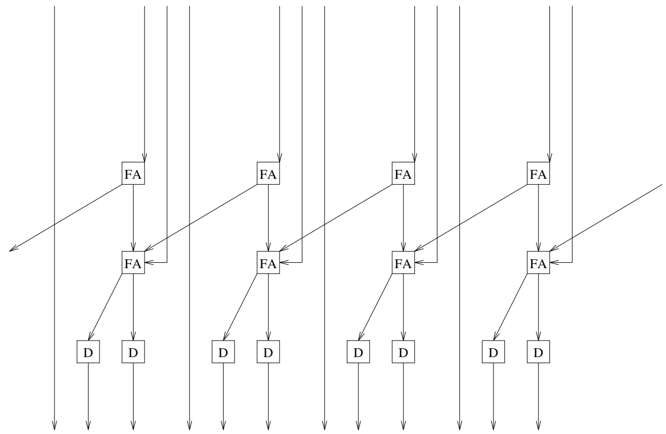

Figure 4. FIR Filter Tap Structure

IV. FPGA IMPLEMENTATION

Advances in field programmable gate array technology have enabled FPGAs to be applied to a variety of problems. In particular, FPGAs prove particularly useful in datapath designs, where the regular structure of the array can be utilized effectively. The programmability of FPGAs adds flexibility not available in custom approaches, while retaining relatively high system clock rates. The disadvantages of FPGAs are primarily related to the limited number of logic operations that can be implemented on a particular device, and the limited signal routing options that are available for connecting logical operators on the array.

The architecture proposed here is well suited to FPGA implementation; only minor modifications need to be considered to map it to the array. In particular, each of the taps of the FIR filter shown in Figure 3 can be implemented in two array columns of Xilinx XC3100-series FPGAs. Because of the high degree of spatial and temporal locality, signal routing delays are not a major concern, as they are with typical high performance FPGA designs.

The architecture of an FPGA-based FIR filter tap with two powers-of-two coefficients is identical to that shown in Figure 4, based on the inverter form FIR structure. Each of the bit slices for the tap require two combinational logic blocks (CLBs) in the array for implementation. The extensive local routing capability of typical FPGAs can be used for the majority of signals within and between taps. Figure 5 illustrates the local routing required between CLBs, where column "1" maps to the first set of full adders for a given tap, and column "2" maps to the second set. The globally routed input data signals are distributed using the horizontal and vertical nets running the length and width of the chip between the rows and columns of CLBs.

The primary concern is with routing of the shift lines. In most realizations, the accumulation path will have a wider word width than the input data from the shifter, in order to account for the overflow and round-off problems that are inherent in a design of this type. For example, if the input data is B_d bits wide, the accumulation path will most likely be $B_i \geq 2B_d$ bits wide. This implies that the input datapath will be using fewer routing lines in each FPGA column than will the accumulation path. By exploiting these extra, unallocated resources, the low delay vertical routing lines of the FPGA can be used more effectively. The extra resources allow the number of vertical routing lines to be minimized, as illustrated in Figure 6, where the additional datapath leads to lower congestion in the routing channel between the columns. A tap with B_d input

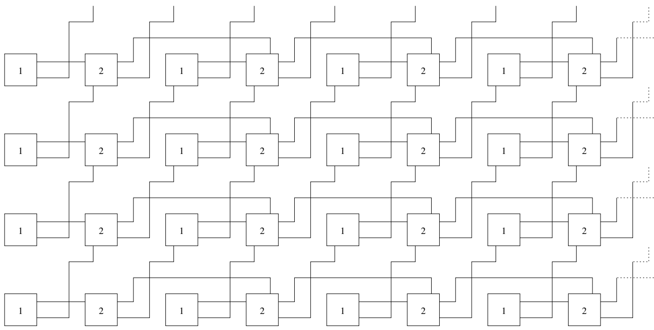

Figure 5. FPGA Filter Tap Local Interconnection

Table 1. Example FPGA Filter Impulse Response

filter taps	coefficient value
w_0, w_{10}	-30
w_1, w_9	6
w_2, w_8	24
w_3, w_7	48
w_4, w_6	65
w_5	72

datapath bits and B_i accumulation path bits can thus be implemented using $2B_i$ logic blocks. The final adder required by the filter can be implemented on an FPGA or using an additional chip.

Typical filter characteristics have been implemented on an Xilinx XC3195 FPGA using this architecture. The XC3195 has an array of 22 by 22 (484) CLBs. For example, an eleven tap lowpass FIR filter with the passband cut-off at $0.1f_s$, the stopband beginning $0.15f_s$, and -18dB stopband rejection has the discrete-space impulse response shown in Table 1. An input data word size of 10 bits was used; the 22 rows provide sufficient intermediate word width protection against overflow. All of the columns of the array were required for the eleven taps. The final accumulation stage was not performed on the array. The maximum sampling rate for this particular design was 30 MHz. The delay is highly dependent on the input data routing, and so higher sampling rates may be attainable for other filter responses (with careful routing).

V. CUSTOM IMPLEMENTATION

A prototype CMOS implementation has been developed to evaluate this architecture. This implementation was designed in a 2.0 micron, double level metal, single level polysilicon CMOS process, using the MSU/ITD standard cells [7]. The architecture of a tap in the full custom chip is shown in Figure 7. The primary difference between this structure and that used for the FPGA implementation is the application of input data pipelining to reduce global signal distribution delays.

Some details of the implementation are given in Table 2. The custom implementation is substantially smaller (by a factor of approximately two and a half) than the equivalent filter based on full two's complement multipliers and adders. The area/speed performance of this design compares favorably with those in [5, 12, 16, 18, 19] when normalized for the older technology used in the present case. In particular, the architecture in [19], based on the FIRGEN compiler [8], also uses canonic signed-digit (CSD) coefficients. In that architecture, however, four terms are used, as opposed to the two used in this work. The design described in [12] also uses four term CSD coefficients. The present custom design makes more extensive use of pipelining than either of these designs, in particular for the input datapath.

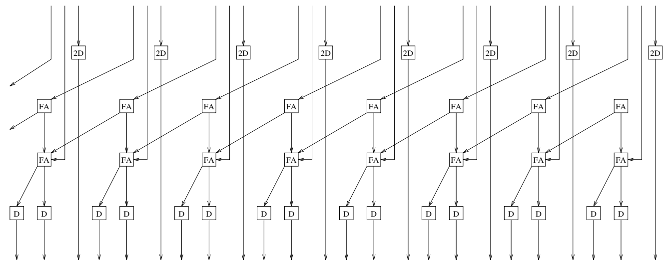

Figure 7. Filter Tap Architecture, Custom Implementation

The area and speed results for the architecture in this 2.0 μm process suggest that 60-70 tap chips with sampling rates exceeding 100 MHz should be feasible in more modern processes, even using the standard cell paradigm. Full custom implementation can most likely yield an additional factor of two in density, due the relatively poor utilization of area in the standard cell method.

Table 2. FIR Filter ASIC Core (2.0 μm DLM SCMOS)

input data	16 bits
coefficients	16 bits + 2 sign bits
intermediate data	16 bits
filter length	15 taps
maximum sampling & clock rate	25 MHz
active area	3.768 mm by 6.167 mm

VI. CONCLUSION

A new parallel FIR digital filter structure which is suitable for efficient implementation of filters whose coefficient values are sums or differences of power-of-two terms has been presented. By exploiting the constraints on the coefficient values, this architecture yields extremely efficient and high speed programmable and custom implementations. Example FPGA and standard cell CMOS implementations illustrating the application of this architecture were presented.

REFERENCES

[1] D. E. Borth, I. A. Gerson, J. R. Haug, and C. D. Thompson. A flexible adaptive FIR filter VLSI IC. *IEEE Journ. Select. Areas Commun.*, SAC-6(3):494–503, Apr 1988.

[2] P. R. Cappello, editor. *VLSI Signal Processing*. IEEE Press, 1984.

[3] J. B. Evans, Y. C. Lim, and B. Liu. A high speed programmable digital FIR filter. In *IEEE Int. Conf. Acoust., Speech, Signal Processing*, Apr 1990.

[4] J. Gallia et al. High-performance BiCMOS 100k-gate array. *IEEE J. Solid State Circuits*, SC-25(1):142–149, Feb 1990.

[5] R. Hartley, P. Corbett, P. Jacob, and S. Karr. A high speed FIR filter designed by compiler. In *IEEE Cust. IC Conf.*, pages 20.2.1–20.2.4, May 1989.

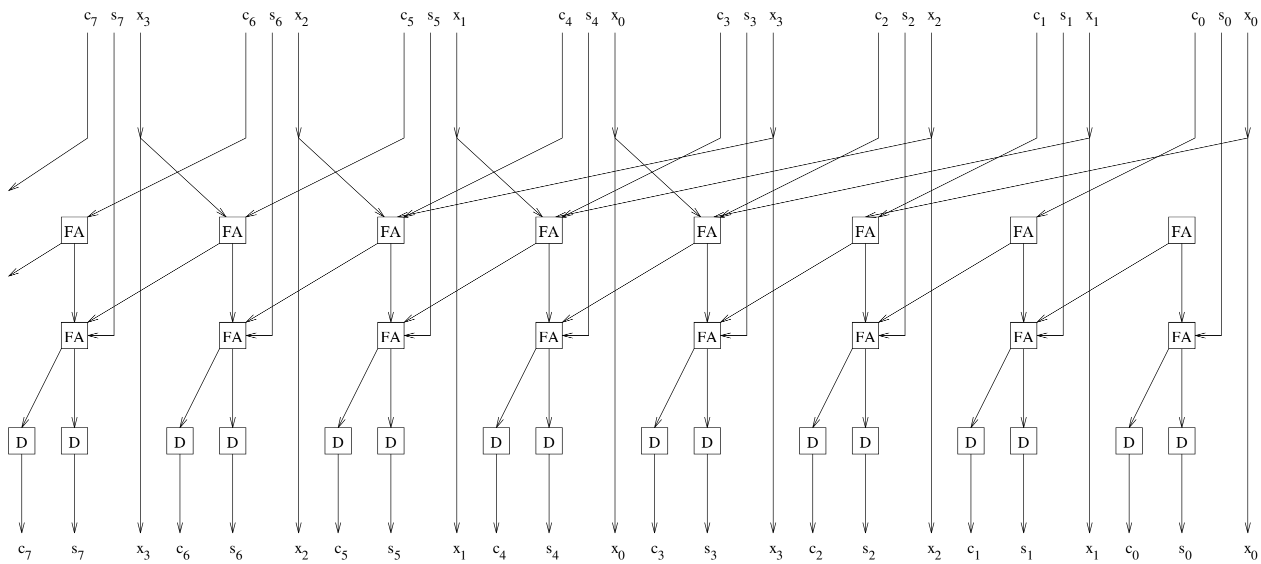

Figure 6. FPGA Filter Tap with Shifter Routing

[6] M. Hatamian and S. Rao. A 100 MHz 40-tap programmable FIR filter chip. In *IEEE Int. Symp. Circuits and Syst.*, pages 3053–3056, May 1990.

[7] Institute for Technology Development - Advanced Microelectronics Division, Mississippi State University. *Scalable CMOS (SCMOS) Standard Cell Library: dlm V2.2*, March 1990.

[8] R. Jain, P. Yang, and T. Yoshino. Firgen: A computer-aided design system for high performance FIR filter integrated circuits. *IEEE Trans. Signal Processing*, 39(7):1655–1668, Jul 1991.

[9] S. Y. Kung. *VLSI Array Processors*. Prentice-Hall, 1988.

[10] S. Y. Kung, R. E. Owen, and J. G. Nash, editors. *VLSI Signal Processing II*. IEEE Press, 1986.

[11] S. Y. Kung, H. J. Whitehouse, and T. Kailath, editors. *VLSI and Modern Signal Processing*. Prentice-Hall, Inc., 1985.

[12] J. Laskowski and H. Samueli. A 150-Mhz 43-tap half-band FIR digital filter in 1.2-μm CMOS generated by compiler. In *IEEE Cust. IC Conf.*, pages 11.4.1–11.4.4, May 1992.

[13] Y. C. Lim and B. Liu. Design of cascade form FIR filters with discrete valued coefficients. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-36:1735–1739, Nov 1988.

[14] H. Moscovitz, W. Bullman, J. Carelli, et al. Automatic full-custom design of high performance DSP chips. In *Proc. IEEE Globecom*, 1988.

[15] S. Powell and P. Chau. Reduced complexity programmable FIR filters. In *IEEE Int. Symp. Circuits and Syst.*, pages 561–564, May 1992.

[16] P. Ruetz. The architectures and design of a 20-MHz real-time DSP chip set. *IEEE J. Solid State Circuits*, 24(2):338–348, Apr 1989.

[17] P. Yang, T. Yoshino, R. Jain, and W. Gass. A functional silicon compiler for high speed FIR digital filters. In *IEEE Int. Conf. Acoust., Speech, Signal Processing*, pages 1329–1332, Apr 1990.

[18] F. Yassa, , J. Jasica, et al. A silicon compiler for digital signal processing: Methodology, implementation, and applications. *Proc. IEEE*, 75(9):1272–1282, Sep 1987.

[19] T. Yoshino, , R. Jain, et al. A 100-MHz 64-tap FIR digital filter in 0.8 μm BiCMOS gate array. *IEEE J. Solid State Circuits*, 25(6):1494–1501, Dec 1990.