Efficient FIR Filter Architectures Suitable for FPGA Implementation

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

Abstract

This paper describes efficient architectures for FIR filters. By exploiting the reduced complexity made possible by the use of two powers-of-two coefficients, these architectures allow the implementation of high sampling rate filters of significant length on a single field-programmable gate array (FPGA).

The author can be contacted via e-mail at evans@tisl.ukans.edu. Portions of this work will be presented at *ISCAS '93* in Chicago, Illinois. This research is supported by the University of Kansas General Research allocation 3626-20-0038.

1 Introduction

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 custom implementation of FIR filters have been pursued [1, 2, 3, 4, 5, 7, 8, 9]. In order to attain high performance, parallel implementation strategies such as systolic 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.

Advances in field programmable gate array (FPGA) technology have enabled FPGAs to be used in a variety of applications. 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, the constraints on the inputs and outputs to the atomic logic units, and the limited signal routing options that are available for connecting logical operators on the array. Many current FPGAs architectures are implemented using memory technologies, and hence the advances in that area will be reflected in improved FPGA density and speed.

This paper presents new parallel FIR filter building blocks suited for implementing filters where each of the coefficient values is a sum or difference of two power-of-two terms. These architectures allow high sampling rate FIR filters of substantial length to be implemented on current generation FPGAs.

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. These simplifications are, however, achieved at the expense of a deterioration in the frequency response characteristics, the extent of which 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 [6] 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.

If the coefficient value is an integer power-of-two, or a sum of two powers-of-two, the multipliers in a filter tap can be replaced by shifters, as depicted in Figure 1. 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, moving the adder inputs k places to the left achieves the same effect as would a coefficient value of 2^k.

2 Architectures

The signal flow graphs (SFG) of the FIR filter architectures discussed in this work are illustrated in Figure 2. The SFG shown in Figure 2 (a) can be applied to FPGA architectures since the use of global communication can be tolerated in such systems, although more pipelining can be used if needed. An SFG appropriate for linear phase filters is shown in Figure 2 (b). In order to attain high sampling rates using conventional FPGAs, 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 the filter tap of Figure 2 (a) is shown in Figure 4. 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) technique in order to increase the sampling rate and alleviate potential routing delays in the target implementation technology. 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 FPGA implementation technology, among other factors. The bits of the input are shifted before summation, as represented by the dotted lines.

The linear phase filter tap of Figure 2 (b) is depicted in Figure 4. This architecture is similar to the previous case, with the addition of the upper set of adders and registers which implement the delay and sum operations on the input data stream. In this case, the delayed and global data bits are summed prior to shifting, as represented by the dotted lines, due to logic unit input/output restrictions. While the ripple carry structure does limit performance, most recent FPGA architectures support high speed carry logic which minimizes the problem.

3 FPGA Implementation

An FIR filter tap as shown in Figure 4 can be implemented in two array columns of Xilinx XC3100-series FPGAs. Because of the high degree of spatial and temporal locality, most signal routing delays are not critical, as they are with typical high performance FPGA designs. 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 6 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 overflow and round-off problems. 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 use 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 4, where the additional datapath leads to lower congestion in the routing channel between the columns. A tap with B_d input 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 the 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 was designed. 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).

Linear phase FIR filter taps can be implemented in three array columns of Xilinx XC4000-series FPGAs, as depicted in Figure 5. Because the XC4000 series supports dedicated carry logic, the ripple carry chain can be used to implement the adder for the input and delayed data. A 19-tap linear phase filter can be supported on an XC4020 component, which has 900 CLBs. Based on the Xilinx timing analyzer, sampling rates on the order of 15-20 MHz can be obtained.

4 Conclusion

A new parallel FIR digital filter structure which allows efficient FPGA implementation of filters whose coefficient values are sums or differences of power-of-two terms was presented. Digital FIR filters with over one hundred taps based on this architecture should be possible by the end of the decade if current technological trends continue. Examples based on Xilinx XC3100 and XC4000 FPGAs were given, although other programmable logic devices such as the AT&T ORCA components will also support this architecture. Automatic programming, from filter specifications to FPGA program, is straightforward.

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] 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.

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

[4] 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.

[5] 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.

[6] 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.

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

[8] 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.

[9] 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.

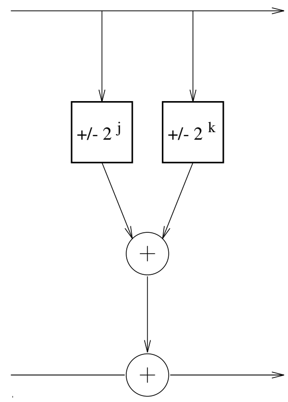

Figure 1: FIR Filter Tap Arithmetic Unit, Coefficients with Two Powers-of-Two

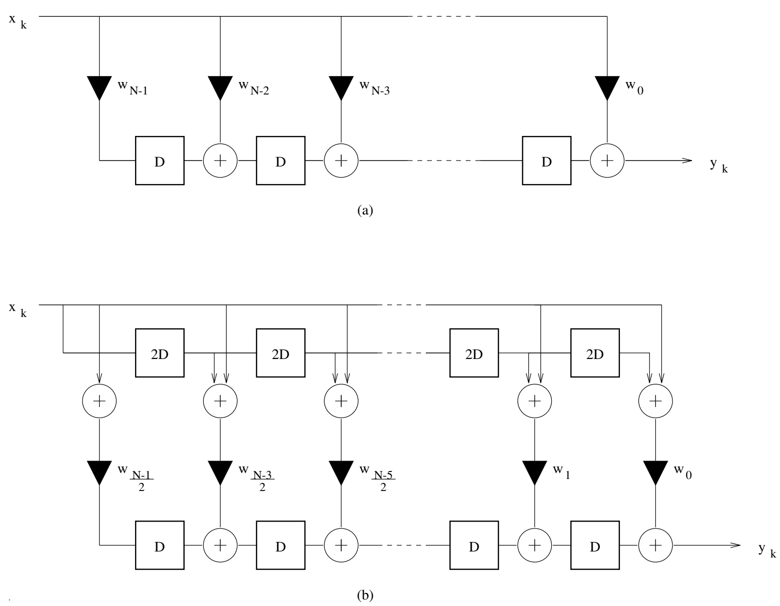

Figure 2: FIR Filter Architecture, (a) inverted form, (b) linear phase.

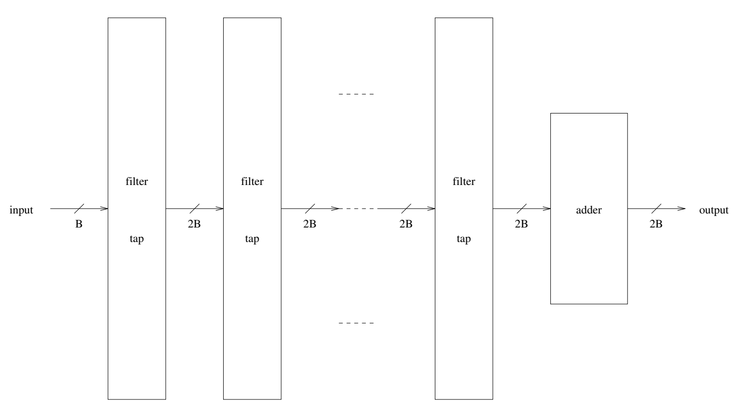

Figure 3: FIR Filter Architecture

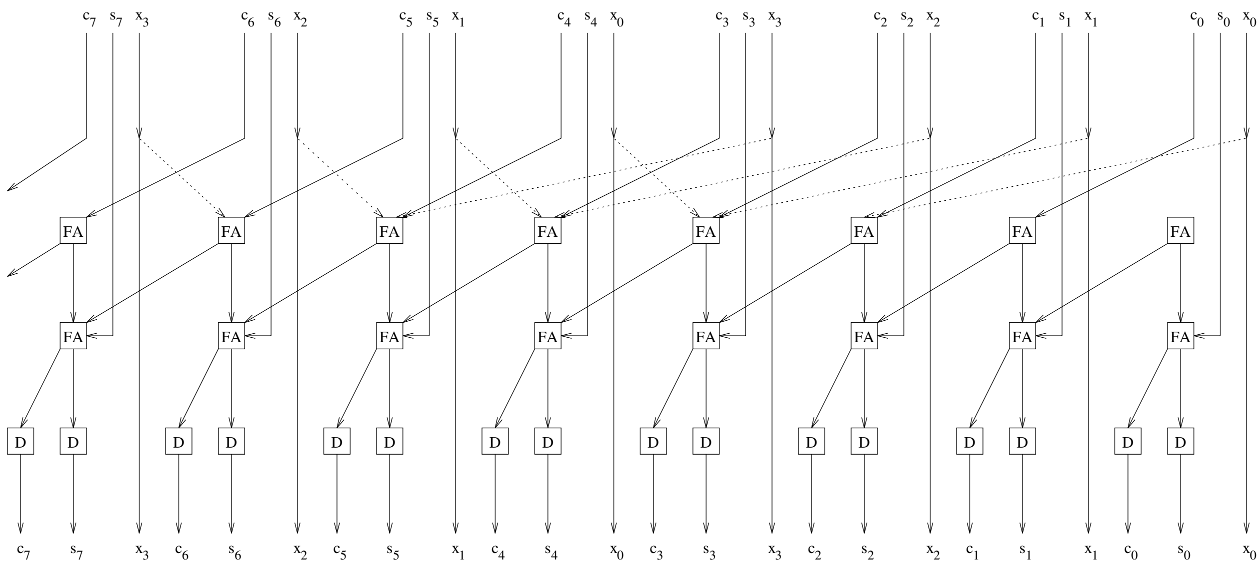

Figure 4: FPGA Filter Tap Structure, with Shifter Routing

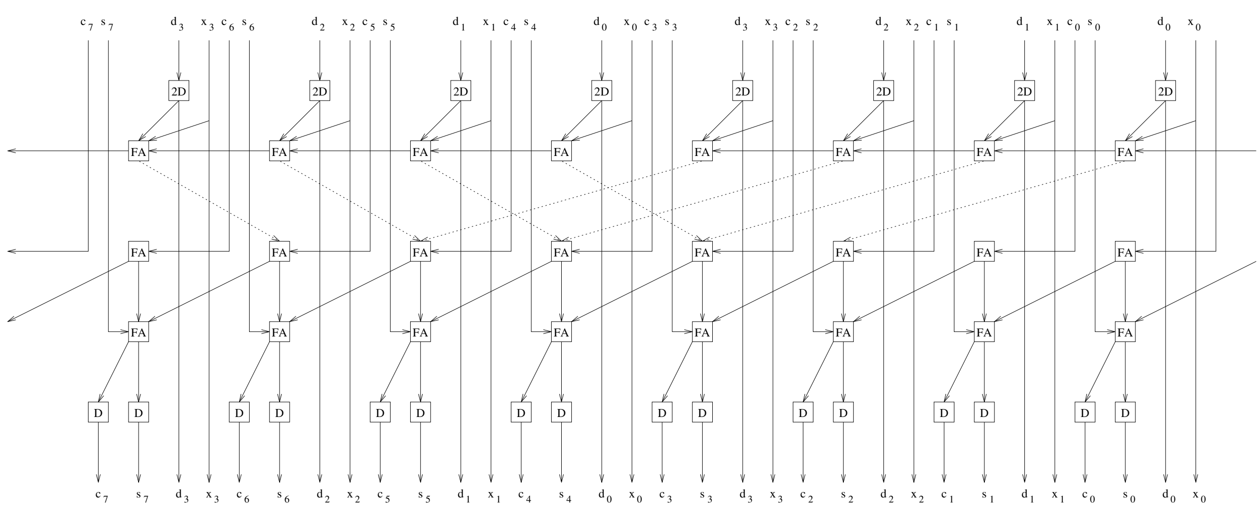

Figure 5: FPGA Linear Phase Filter Tap Structure

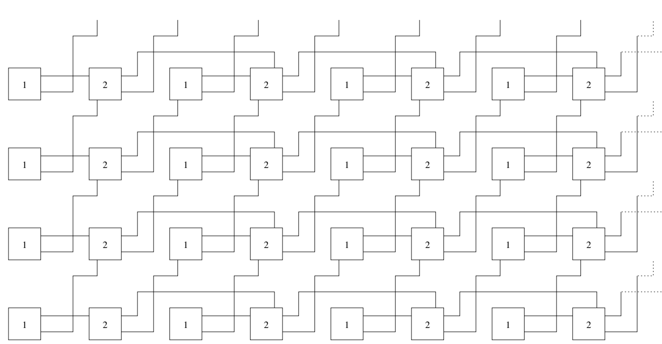

Figure 6: FPGA Filter Tap Local Interconnection