Journal of VLSI Signal Processing 21, 31–35 (1999)

Scalability of Programmable FIR Digital Filters

DING-MING KWAI
C/o 47 Ln. 80 Chang-Shin St., Taipei 106, Taiwan

BEHROOZ PARHAMI
Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106-9560, USA

Received September 17, 1996; Revised August 5, 1997

Abstract. Previous designs of programmable FIR digital filters have demonstrated that the use of broadcast data and control can lead to a high performance-to-cost ratio. As the technology advances to the deep sub-micrometer regime, such an approach should be re-examined by taking the effect of interconnections into account. In this paper, we show that the contribution of interconnect delay to the cycle time is no longer negligible and will hamper the scalability of such broadcast designs. Further speed and density improvements through scaling can be secured by the fully pipelined design in which both data and control signals are restricted to local connections.

1. Introduction

In its transposed form, a finite impulse response (FIR) digital filter can be implemented as a cascade architecture consisting of N fixed-size cells, one for each tap. Input data are broadcast to all filter taps and the responses computed as inner-product steps in pipelined fashion. As the required number of taps may vary, designs must be modular and expandable. Tap counts from 8 to 64 are found in different applications [1–7].

For adaptive applications, programmable coefficients or weights must be supported to allow modifying the response in real time. The coefficients can be loaded through the pipeline registers which form a shift-register chain when the broadcast input is set to zero. All coefficient registers are connected to a common *enable* line which, when asserted, simultaneously stores the contents of the pipeline registers as coefficients (Fig. 1 without the dotted register z). The fact that the data bus was already in place led previous designers to adopt such a simple control scheme, making the broadcast overhead relatively insignificant.

As the throughput rate is often the primary concern, it is advantageous to introduce an intermediate pipeline level into the multiplier-accumulator module. The operating frequency can be increased by a factor close to two, if the multiply-add latency is evenly divided in two stages. The expense of an extra register (z in Fig. 1) is quite modest compared to the highly complex multiplier and adder that would be needed to achieve comparable performance improvement with the single-stage module.

While the preceding assessment may be accurate for short filters, the reduced depth of pipeline stages heightens the significance of propagation delays on wires [8–10]. For a long filter, the broadcast overhead for input signals becomes a major portion of the cycle time. Realistic analysis, therefore, requires a technology- and architecture-dependent model that predicts the cycle time of an N-tap FIR digital filter as a function of N. The cycle time is affected by two factors: broadcasting data to all filter taps and performing a pipelined operation on data in each tap. The required cycle time increases with N simply because longer interconnections will be involved and a larger load must be driven.

In this paper, we show that the broadcast design may cease to be cost-effective as scaling of dimensions in integrated circuits is used to improve performance. We note that the broadcast overhead cannot be alleviated by

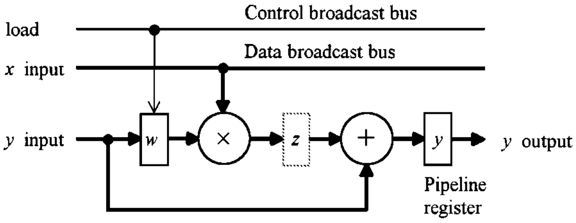


Figure 1. A single tap of the FIR digital filter with broadcast data/control and an optional intermediate pipeline register z. Although z is drawn between the multiplier and adder for clarity, it can in fact be placed at a suitable midpoint within the multiply-add computation.

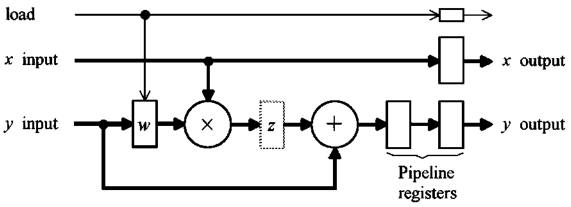


Figure 2. A single tap of the FIR digital filter with pipelined data and control. The intermediate pipeline register z is still optional, but its inclusion will yield greater benefits in view of the absence of broadcast overhead.

merely pipelining the data signals, since the overhead will then shift to the control signals. As an alternative, we show that it is possible to pipeline the control signals with the data signals through the filter, thus limiting the global connections to clock and power supply. The increased area to accommodate more pipeline registers can be compensated for by the higher operating frequency, thus maintaining the throughput rate with scaling.

Our presentation is organized as follows. Section 2 describes the fully pipelined design and its variants, followed by cost-performance tradeoffs among various designs. Section 3 discusses the effects of scaling on the above designs. Section 4 contains our conclusions.

2. Fully Pipelined Design

If the input x is to be pipelined through the FIR digital filter, a register must be inserted on the data bus to stop the direct propagation between cells. The output y will then lead x by one clock cycle. Regulating the data flow requires the insertion of an extra register on the y output.

The preceding design can be easily obtained by applying the retiming technique [11] to the signal flow graph of the transposed direct-form realization. Consequently, when the coefficients are loaded through the pipeline registers, they reach the taps on alternate clock cycles, implying that the load signal can also be pipelined. Figure 2 shows the resulting design.

More general designs can be derived by dividing the number N of taps into N/k segments. In each segment, $k-1$ cascaded cells of the type shown in Fig. 1 are followed by a single cell shown in Fig. 2. Thus, the data and control signals are allowed to propagate within a segment terminated by the pipeline registers, which act as repeaters between segments.

The segment size k has cost and speed implications. As k increases, the cost is reduced due to the need for fewer pipeline registers and throughput is degraded due to the longer interconnections within segments. It also affects the latency since data must propagate through $N + N/k$ pipeline registers. The extreme case of $k = N$, corresponding to the broadcast design, has the smallest area; the fully pipelined design with $k = 1$ has the highest throughput. Clearly, tradeoffs can be made by selecting k to achieve the most cost-effective design.

To facilitate our analyses, we assume that data are represented by fixed-point numbers with 12-bit inputs, 12-bit coefficients, and 26-bit outputs. The coefficients are encoded by applying Booth's algorithm (which replaces a sequence of ones representing the value $2^j + 2^{j-1} + \cdots + 2^i$ by one positive and one negative binary digit denoting $2^{j+1} - 2^i$). The partial products for the current tap, along with the result from the previous tap, are collected in the form of sum and carry bits by carry-save addition. To avoid carry propagation in each tap, the true result is not computed until the final tap. This technique is well known and commonly used in designing FIR digital filters [6].

Presently, designers tend to put the critical path on the carry-save adder, trying to prevent the broadcast overhead from being a dominating factor in the cycle time. In reality, the overall delay to be partitioned into two pipeline stages is taken from the broadcast input, through the partial-product generator, to the carry-save adder. The Booth encoder is not on the critical path, since the coefficients are held in static registers. The remaining registers can be dynamic and operate under a single-phase clock.

We estimate the areas and cycle times for different values of k and a fixed number $N = 64$ of taps using a 1.0 μm CMOS technology. The area and speed characteristics of each component are listed in Table 1.

Table 1. Area and speed characteristics of the implemented components.

Component	Delay (ns)	Area (μm^2)
Register	0.9	2.0×10^4
Adder	6.1	9.8×10^4
Multiplier	14.1	2.3×10^5

Figure 3 shows the decrease in area and increase in cycle time, relative to the fully pipelined design. The most cost-effective design, having the smallest area-time product, is obtained when the segment size is chosen to be $k = 4$; for such an optimal design, the speed degradation is more then offset by the area saving.

Because of segmentation, the resulting cycle time depends only on the segment size k, rather than on the filter length N. Figure 3 also indicates that for $N = 2, 4, \ldots, 32$, broadcast design with $k = N$ is more cost-effective that fully pipelined design with $k = 1$. In these cases, the propagation delays incurred on long wires do not significantly affect the cycle time. Thus, trading speed for area is worthwhile.

The broadcast design allows the expansion of the number of taps in a limited range (2–32), while still enjoying high cost-effectiveness. This explains why previous designs of short programmable FIR digital filters have been based on broadcasting, rather than pipelining, of the data and control signals. As the filter length grew to $N = 64$ taps, broadcasting overhead emerged as the primary limiting factor on performance.

3. Scalability Analysis

The preceding results, however, cannot be simply extended as we continue to exploit the downward scaling of feature sizes in switching devices and wires to improve performance. In this section, we further analyze the scalability of the fully pipelined design and show that as linear dimensions are scaled down to the deep submicrometer regime, the broadcast design in no longer cost-effective even for shorter filters.

Figure 4 shows the interconnect delay on wires for 1.0, 0.5, and 0.25 μm CMOS technologies, presented as ratios with respect to the device switching time for the corresponding technology [12–14]. For the broadcast design, we have conservatively estimated the wire length in the range of 2 mm (8 taps) to 6 mm (64 taps), using a 1.0 μm CMOS technology and distributing the broadcast data by a standard tree-like network [15]. These base points appear on the bottom curve in Fig. 4. As the linear dimensions are scaled down by a given factor, the wire lengths are expected to be reduced by the same factor.

The interconnect delay is a function of the wire resistance and capacitance. As the dimensions are scaled down, the wire resistance increases due to the reduction in cross-sectional area, while there is no corresponding decrease in the wire capacitance, leading to increased interconnect delay. Figure 4 indicates that the broadcast overhead will dramatically increase, even though the wire length is shortened with more advanced technologies. Hence, we need to reexamine the relative

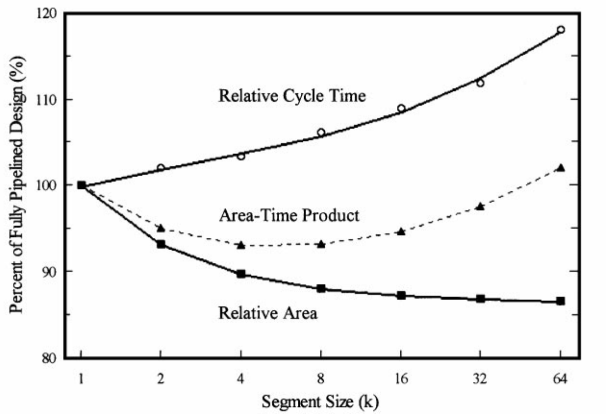


Figure 3. Cycle time and VLSI layout area, using 1.0 μm CMOS technology, for designs with different segment sizes k. Percent values are relative to the fully pipelined design with $k = 1$.

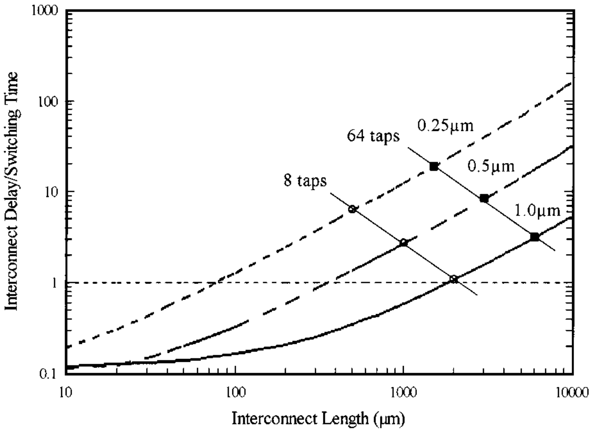


Figure 4. The ratio of interconnect delay to device switching time versus wire length for 1.0, 0.5, and 0.25 μm CMOS technologies using standard aluminum/silicon dioxide-based metallization.

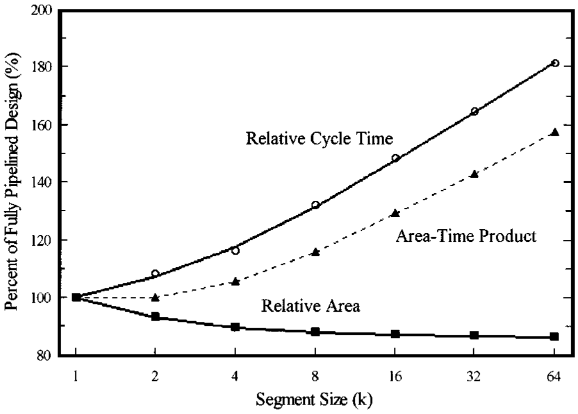


Figure 5. Cycle time and VLSI layout area for designs with different segment sizes k relative to the fully pipelined design with $k = 1$, using 0.25 μm CMOS technology.

measures of layout area and cycle time with each change in technology.

We repeat the analysis previously done for 1.0 μm CMOS in Section 2 (Fig. 3) using the more advanced 0.25 μm CMOS technology. The results are shown in Fig. 5. Because of our assumption that the scaling factor applies to each dimension, the relative area remains the same. The relative cycle time obviously increases in order to accommodate the additional delays introduced by the long data bus used to broadcast the data to k taps.

Compared to the case of the 1.0 μm CMOS technology, the knee point of the relative cycle time curve has now moved to a much smaller segment size ($k = 4$). Beyond this point, the broadcast overhead becomes a dominating influence on the cycle time. The speed degradation can no longer be compensated for by the area saving which never exceeds 15%. As a result, the expandability of the broadcast design is limited to a very small range. On the contrary, the fully pipelined design shows its superiority in preserving the speed and density benefits of scaling by virtue of its lower area-time product. We expect that with further advances in technology, the advantages will become apparent for even smaller values of k.

4. Conclusion

Scalability, as considered in this paper, has two aspects. One is the downward scaling of fabrication technology in terms of the feature size of components; the other is the upward scaling of the cascade architecture to implement longer programmable FIR digital filters. As the device switching times continue to improve, propagation delays on wires begin to curtail the benefits of scaling.

We have shown that by restricting the segment size to a small number of taps, the benefits of scaling can be secured. In view of the significant overhead involved in broadcasting, resorting to such a fully pipelined design is necessary if the continuous improvement of throughput rate with scaling is to be maintained. Our results serve to reiterate the importance of local and pipelined communication in modular designs, based on state-of-the-art VLSI technology.

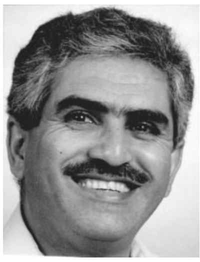
Modifications to reduce the wire resistance and capacitance through the use of new conductor (e.g., copper) and insulator (e.g., polymer) materials, as well as multi-level interconnections, do not significantly affect our conclusions. Optimistically, the interconnect RC time constant can be reduced from 3 to 1 ns/cm^2 [16]. Such a change is equivalent to dividing the interconnect delay/switching time ratio of Fig. 4 by a factor of 3, and thus shifting the optimal segment size of Fig. 5 by the same factor. Furthermore, such a change involves higher development and production costs. The resulting cost factors must be taken into account for a fair and complete comparison.

References

1. M. Hatamian and S.K. Rao, “A 100 MHz 40-tap programmable FIR filter chip,” *Proc. Int’l Symp. Circuits & Systems*, New Orleans, LA, pp. 3053–3056, May 1990.
2. T. Yoshino et al., “A 100-MHz 64-tap FIR digital filter in 0.8-μm BiCMOS gate array,” *IEEE J. Solid-State Circuits*, Vol. 25, pp. 1494–1501, Dec. 1990.
3. J. Laskowski and H. Samueli, “A 150-MHz 43-tap half-band FIR digital filter in 1.2-μm CMOS generated by silicon compiler,” *Proc. IEEE Custom Integrated Circuits Conf.*, Boston, MA, pp. 11.4.1–11.4.4, May 1992.
4. J.B. Evans, “Efficient FIR filter architectures suitable for FPGA implementation,” *IEEE Trans. Circuits & Systems-II: Analog and Digital Signal Processing*, Vol. 41, pp. 490–493, July 1994.
5. L.E. Thon, P. Sutardja, F.-S. Lai, and G. Coleman, “A 240 MHz 8-tap programmable FIR filter for disk-drive read channels,” *Digest IEEE Int’l Solid-State Circuits Conf.*, San Francisco, CA, pp. 82/83, Feb. 1995.
6. R.A. Hawley et al., “Design techniques for silicon compiler implementations of high-speed FIR digital filters,” *IEEE J. Solid-State Circuits*, Vol. 31, pp. 656–667, May 1996.
7. K.Y. Khoo, A. Kwentus, and A.N. Wilson, “A programmable FIR digital filter using CSD coefficients,” *IEEE J. Solid-State Circuits*, Vol. 31, pp. 869–874, June 1996.
8. M. Afghahi and C. Svensson, “Performance of synchronous and asynchronous schemes for VLSI systems,” *IEEE Trans. Computers*, Vol. 41, pp. 858–872, July 1992.
9. D. Audet, Y. Savaria, and N. Arel, “Pipelined communications in large VLSI/ULSI systems,” *IEEE Trans. VLSI Systems*, Vol. 2, pp. 1–10, March 1994.

10. D.-M. Kwai and B. Parhami, “Area-time tradeoffs in FIR digital filters with broadcast and pipelined designs,” *Proc. Midwest Symp. Circuits & Systems*, Sacramento, CA, Vol. 1, pp. 449–452, Aug. 1997.
11. C.E. Leiserson and J.B. Saxe, “Optimizing synchronous systems,” *J. VLSI Computer Systems*, Vol. 1, pp. 41–67, 1983.
12. S. Bothra, B. Rogers, M. Kellam, and C.M. Osburn, “Analysis of the effects of scaling on interconnect delay in ULSI circuits,” *IEEE Trans. Electron Devices*, Vol. 40, pp. 591–597, March 1993.
13. B. Davari, R.H. Dennard, and G.G. Shahidi, “CMOS scaling for high performance and low power—the next ten years,” *Proc. IEEE*, Vol. 83, pp. 595–606, April 1995.
14. M. Gilligan and S. Gupta, “A methodology for estimating interconnect capacitance for signal propagation delay in VLSIs,” *Microelectronics J.*, Vol. 26, pp. 327–336, May 1995.
15. C.Y. Wu and M.-C. Shiau, “Delay models and speed improvement techniques for RC tree interconnections among small geometry CMOS inverters,” *IEEE J. Solid-State Circuits*, Vol. 25, pp. 1247–1256, Oct. 1990.
16. D.C. Edelstein, G.A. Sai-Halasz, and Y.-J. Mii, “VLSI onchip interconnection performance simulations and measurements.” *IBM. J. Research & Development*, Vol. 39, pp. 383–401, July 1995.

Ding-Ming Kwai received the B.S. and M.S. degrees in Taiwan from the National Cheng Kung University, Tainan, and the National Chiao Tung University, Hsinchu, in 1987 and 1989, respectively, and the Ph.D. degree from the University of California, Santa Barbara, in 1997. He was with the Chung Cheng Institute of Technology, Taoyuan, Taiwan, as a reserve officer during 1989–1991 and with the Hualon Microelectronics Corporation, Hsinchu, Taiwan, as a design engineer during 1991–1993. His research interests include parallel processing, VLSI architectures, and fault-tolerant computing.



Behrooz Parhami received his Ph.D. in computer science from University of California, Los Angeles, in 1973. Presently, he is Professor in the Department of Electrical and Computer Engineering, University of California, Santa Barbara. His research deals with parallel architectures and algorithms, computer arithmetic, and reliable computing. In his previous position with Sharif University of Technology in Tehran, Iran (1974–1988), he was also involved in the areas of educational planning, curriculum development, standardization efforts, technology transfer, and various editorial responsibilities, including a 5-year term as Editor of *Computer Report*, a Farsi-language computing periodical. His technical publications include over 170 papers in journals and international conferences, a Farsi-language textbook, and an English/Farsi glossary of computing terms. Two textbooks on computer arithmetic (Oxford, 2000) and parallel processing (Plenum, 1999) have recently been completed.

Dr. Parhami is a Fellow of both the IEEE and the British Computer Society, a member of the Association for Computing Machinery, and a Distinguished Member of the Informatics Society of Iran for which he served as a founding member and President during 1979–1984. He also served as Chairman of IEEE Iran Section (1977–1986) and received the IEEE Centennial Medal in 1984.
parhami@ece.ucsb.edu