16th European Signal Processing Conference (EUSIPCO 2008), Lausanne, Switzerland, August 25-29, 2008, copyright by EURASIP

EFFICIENT SERIAL AND PARALLEL IMPLEMENTATION OF PROGRAMMABLE FIR FILTERS BASED ON THE MERGING TECHNIQUE

Dimitris Bekiaris, George Economakos and Kiamal Z. Pekmestzi

Department of Electrical and Computer Engineering, National Technical University of Athens
Iroon Polytechneiou 9, Zografou, 15780, Athens, Greece
phone: + 30 2107722500, fax: + 30 2107722428, email: {mpekiaris, geconom, pekmes}@microlab.ntua.gr
web: www.microlab.ntua.gr

ABSTRACT

This paper presents a novel architecture for the efficient implementation of parallel and serial programmable FIR filters. In the parallel merged architecture, both the input data and the coefficients operate in bit-parallel form. In the serial merged architecture, input data enters the circuit in Modified-Booth encoded digits, while the coefficients are kept in bit-parallel form. The proposed schemes are based on a low latency filter structure, where adjacent multiply-add units are merged to reduce the number of carry-save registers of the accumulation path to the half. The computation of intermediate terms is implemented using the carry-save arithmetic. Based on theoretical estimation models of hardware complexity and switching activity, it is shown that the presented schemes result in circuits with reduced area and power consumption, compared to other parallel and serial FIR filter architectures presented in the bibliography.

1. INTRODUCTION

The efficient implementation of Digital Signal Processing (DSP) algorithms is always critical for the design of embedded systems suitable for real-time applications. Apart from performance, nowadays power dissipation is also becoming a critical parameter, as long as battery is the only source of energy in portable devices like mobile communication systems. Therefore, a power-performance trade-off should be considered for the realization of DSP cores, so that the maximum performance is combined with the lowest possible power consumption.

Such a problem appears in the design of Finite Impulse Response (FIR) digital filters, which are the fundamental components of several DSP algorithms. Although their general architecture permits an easy implementation, it requires an excessive amount of hardware complexity, leading also to significant amount of power. Thus, numerous design techniques have been presented in the bibliography, targeting the minimization of the circuit's area and the reduction of total power dissipation.

For the present VLSI manufacturing technologies, dynamic power dissipation is the dominating factor of total power consumed, compared to the leakage current, which will be critical in the rising, sub-100nm, technologies. In a CMOS VLSI implementation, dynamic power dissipation is strongly dependent on the switching activity of inputs and outputs of the circuit. So, several design techniques have explored the design and implementation of high throughput FIR filters with low complexity and with reduced switching activity per output sample.

In [1] and [2], folded digit-serial architectures with reduced hardware complexity are presented, using a tree accumulation structure for the final addition of intermediate products. These schemes suffer from high latency, while their corresponding transposed forms require a large number of carry-save registers. Low power folded schemes are also implemented in a single multiply-add unit [3] and a single-multiplier DSP processor [4]. In these designs, the coefficients are reorganized based on the minimum Hamming distance, so that the switching activity in the multiplier is strongly reduced. A mapping and scheduling algorithm of the coefficients in programmable folded FIR filters, based on the idea of Hamming distance, is also presented in [5] and the transpose form is again selected. Architectures introducing the reuse of common sub-expressions are presented in [6], [7] and [8], but the proposed design techniques are not applicable to programmable filters. An efficient implementation of non-merged parallel FIR filters is also shown in [9], where, instead of a novel architectural concept, several custom-based design techniques are presented, targeting the reduction of the filter's power dissipation. Also, hardware efficient programmable filter schemes pipelined at the bit-level are shown in [10], where both the input data and the coefficients operate in bit-parallel form.

This paper introduces two novel low-latency programmable FIR filter architectures. In the first, input data is kept in bit-parallel form, while in the second it is transformed into Modified-Booth encoded digits. In both schemes, the coefficients operate in bit-parallel form. The structure of the proposed scheme is partially based on the one presented in [11], where adjacent multiply-add units are merged to reduce the registers of the accumulation path to the half. Theoretical estimation models of hardware complexity and switching activity demonstrate that the proposed merged schemes perform better that relative parallel and serial filter implementations presented in the literature, from the aspect of area and power consumption.

The organization of the paper is as follows: In the next section, the architecture of the proposed merged parallel FIR filter is given and it is compared to the transpose direct FIR scheme. The proposed digit-serial merged scheme is demon-

strated in Section 3, along with its corresponding circuits. In the next section, theoretical estimations of hardware complexity and switching activity per output sample is given for the proposed designs, which are compared to other parallel and serial architectures respectively. Finally, a conclusion and hints for future work are included in the last section.

2. THE PROPOSED MERGED PARALLEL FIR ARCHITECTURE

In general, an FIR filter output, y_n, is given by the following relation:

$$y_n = \sum_{i=0}^{k-1} a_i x_{n-i} \quad (1)$$

In (1), x_{n-i} imply the delayed input data samples and a_i the filter coefficients. For the hardware implementation of the sum-of-products shown in (1), several FIR architectures can be employed. The most prevalent is the transpose form, because is has low latency and reduced switching activity in the data path, as input data remains stable, until a new output sample is produced. The main drawback of the transposed architecture is the increased number of registers of the accumulation path, especially when the multiply-add (M-A) units are implemented based on the carry-save arithmetic to achieve high performance. The implementation of the transpose FIR filter form is shown in Fig. 1 for an 8-tap filter.

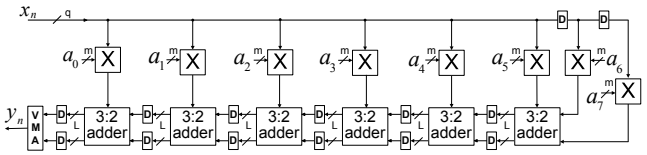

Figure 1. An 8-tap transpose FIR filter scheme.

In the above figure, both the input data and the coefficients operate as q-bit and m-bit parallel words respectively. So, for a k-tap filter, each M-A unit is composed by a 3:2 adder, which takes an input of $(q+m)$ bits, while the bits of the carry-save registers are equal to $L=(q+m+\log_2 k)$. The 3:2 adder produces the result in carry-save form to achieve high performance, so two registers of $(q+m+\log_2 k)$ bits are required in order to store the result. For the multiplication operations, a typical carry-save array multiplier is selected. The output sample is transformed from carry-save to binary form by a fast Vector-Merged Adder (VMA). The total number of delay elements in this architecture is equal to $\{2k(q+m+\log_2 k) + 2q\}$.

Such a large amount of delay elements consumes significant power, while they also require a large portion of silicon area. This drawback can be effectively reduced by the merged parallel FIR architecture, which is derived by applying merging and retiming on the transpose form of Fig. 1. The proposed merged parallel FIR scheme is shown in Fig. 2 for 8 taps. Thus, the produced filter structure has low latency, while the half number of registers is required in the accumulation path. The main difference of the proposed scheme is the use of 4:2 carry-save adders or compressors, instead of 3:2 adders. The structures of both the 3:2 and the 4:2 adder are shown in Fig. 3 and Fig. 4 respectively. Also, $qk/2$ delay elements are placed in the data path, while there are k delay elements of $(q+m+\log_2 k)$ bits in the accumulation path. Therefore, the total number of delay elements in the proposed parallel scheme is equal to $k(q+m+\log_2 k)+(qk)/2$.

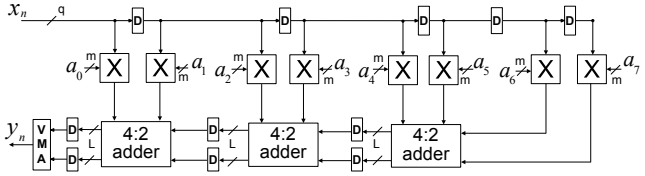

Figure 2. An 8-tap merged parallel FIR implementation.

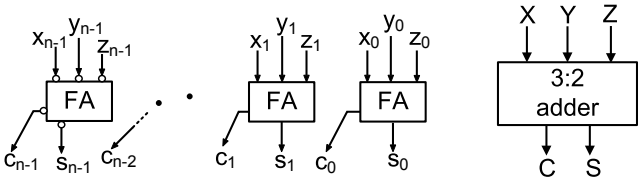

Figure 3. The 3:2 adder.

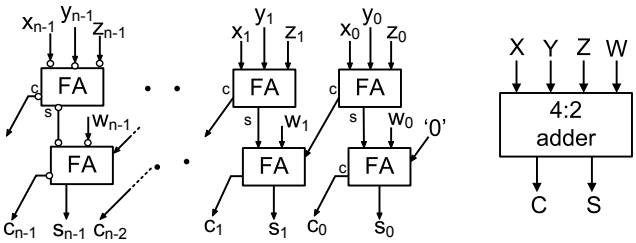

Figure 4. The 4:2 adder.

3. THE PROPOSED MERGED SERIAL FIR ARCHITECTURE

The proposed FIR scheme is strongly based on the merged parallel architecture shown in Fig. 2. The main difference of the novel scheme is that input data enters the circuit in digit-serial form. The proposed FIR architecture is shown in Fig. 5 for 8 taps. In this scheme, the q-bit input data, considered to be in 2's complement form, is transformed into $f=(q/2)$ Modified-Booth (MB) encoded digits, while entering into the circuit. Thus, input data can be written as follows:

$$x_n = \sum_{j=0}^{f-1} \mathbf{x}_n^j 2^{2j} \text{ , where } \mathbf{x}_n^j = \pm 2, \pm 1, 0 \quad (2)$$

In Eq. 2 and from now on, the superscripts declare the weight of the input data digit, while the subscripts imply the input data sample. Also, the bold-type terms represent the MB encoded digits of x_n .

By replacing Eq. 2 in Eq. 1, the filter output y_n^j of weight 2^{2j} is given by the following equation:

$$y_n^j = \sum_{i=0}^{k-1} \mathbf{x}_{n-i}^j a_i \quad (3)$$

Thus, the final output, y_n, is computed by accumulated the previous terms properly weighted, according to the following relation: $y_n = \sum_{j=0}^{f-1} y_n^j 2^{2j}$ (4)

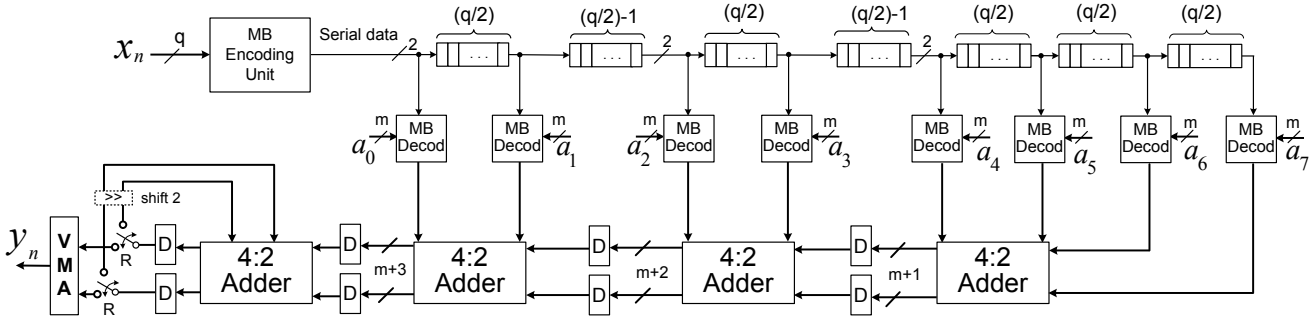

Figure 5. The architecture of the proposed 8-tap digit-serial FIR filter scheme.

The proposed digit-serial merged architecture is shown in Fig. 5 for 8 taps. In this structure, the MB encoding unit takes as input the q-bit data and generates the f MB digits with a skew of one clock cycle. Also, the components MB Decod of Fig. 5 imply the MB decoders, which perform the computation of the (q/2) partial products. The coefficients operate as word of m bits. The shift registers between successive taps are composed by f or (f-1) 2-bit registers of the output digits. The additions are implemented by 4:2 adders and a 4:2 carry-save accumulator. The accumulator is implemented using a 4:2 adder structure and the shift is hardwired with a 2-bit sign-extension for the correct addition of two's complement numbers. The result, in carry-save form, is driven to the VMA every f clock cycles and the output sample is transformed into binary form.

The signal R controls the loading of the accumulated result into the VMA and it is generated by a synchronous modulo-f global counter. The counter sets R to '1' every f clock cycles, as f steps are required for the computation of an output sample. For the implementation of the VMA, a typical carry-ripple adder should be used, as new output samples are generated every f clock cycles from the accumulator. The MB encoding unit and the MB decoder are presented in detail below.

3.1 The Modified-Booth encoding unit

According to the MB algorithm [12], the values of an encoded digit are mapped into the set {-2, -1, 0, +1, +2}. In the conventional implementations, three bits are required, in order to represent the MB digits. In this implementation, we present a novel encoding, which is strongly based on the one proposed in [1]. In this approach, the MB encoder generates two, instead of three bits, which are the sign and the left shift. The logic of these two bits is shown in the following equations:

$$s_j = b_{2j+1} \quad (5)$$

$$d_j = b_{2j} \, xnor \, b_{2j-1} \quad (6)$$

In (5) and (6), b_{2j+1}, b_{2j} and b_{2j-1} are the input data bits of the encoding unit, while the bits s_j and d_j compose the encoded digit, as it is also shown in Fig. 6. The signal *Sel* of Fig. 6 selects the proper three input data bits to be encoded at each clock cycle and it is produced by the global modulo-f counter of the filter's circuit.

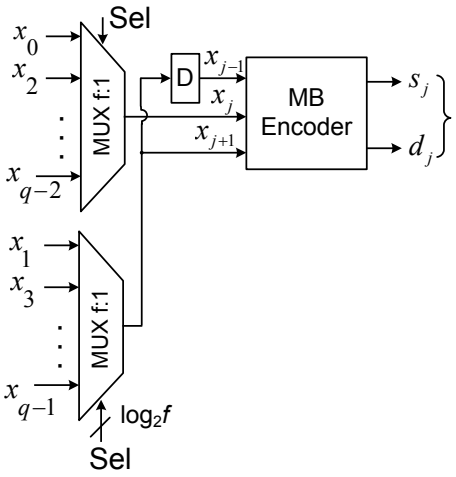

Figure 6. The MB encoding unit.

3.2 The Modified-Booth decoder

The cell of the Modified-Booth decoder is given in Fig. 7, where a^w and a^{w-1} are the bits of the coefficient a, of weight w and w-1 respectively. PP_j^w is the w-th bit of the partial product PP_j, whose weight is 2^{2j}, according to the MB digit $\{s_j, d_j\}$. The signal z_j is generated by the *xor* gate of the current sign bit, s_j, and of the sign bit of the previous digit, s_{j-1}. Every f cycles, s_{j-1} must be reset to zero, according to the MB algorithm, so R performs this control with an AND gate, as it is shown in Fig. 7.

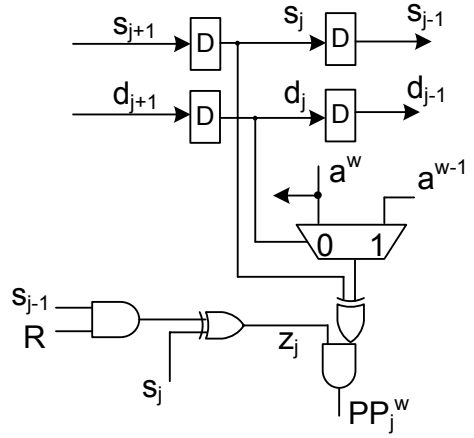

Figure 7. The cell of the Modified-Booth decoder (MB Decod).

4. ESTIMATION OF HARDWARE COMPLEXITY AND SWITCHING ACTIVITY

In the following tables, the proposed scheme is compared with other non-merged digit-serial architectures from the aspect of hardware complexity and switching activity. Hardware complexity is estimated as the area or the number of transistors required for the implementation of a circuit in a CMOS technology. In the next tables, hardware complexity is estimated as the number of delay elements per filter tap, as the remaining circuit is the same in both parallel and digit-serial compared schemes.

From Table 1, it is shown that the merged parallel FIR structure requires almost the half number of delay elements in the accumulation path and therefore significantly less amount of area, compared to the scheme of Fig. 1.

This advantage is also revealed in Table 2, where the digit-serial proposed architecture is compared to the unfolded serial scheme shown in [1], from the aspect of hardware complexity. The proposed serial scheme requires the half number of registers per tap in the accumulation path and finally requires less number of delay elements, while the combinatorial part of the compared circuits is the same.

Table 1. Hardware comparison of the parallel architectures.

Filter scheme (*k* taps)	Total number of Delay elements
Proposed Scheme	$k(q+m+\log_2 k)+(qk)/2$
Transpose Fig. 1	$2k(q+m+\log_2 k)+2q$

Table 2. Hardware comparison of the serial architectures.

Filter scheme (*k* taps)	Total number of Delay elements
Proposed Scheme	$k(m+\log_2 k)+(qk)/2$
Digit-Serial Transpose[1]	$2k(m+\log_2 k)+2q$

In Tables 3 and 4, a metric of the switching activity per output sample is given, for the parallel and the serial compared schemes respectively. Based on this metric, the schemes shown in Table 1 and Table 2 are now compared from the aspect of switching activity per output sample, in the forthcoming tables.

The columns SA_{ACCUM} and SA_{DATA} represent the switching activity of the accumulation path and the input data path, while SA_{TOTAL} is the sum of these two terms.

So, based on this approach and according to the SA_{ACC} column of Table 3, the proposed merged parallel FIR scheme has half switching activity per output sample in the accumulation path, compared to the transpose scheme of Fig. 1, while it has more switching activity per output sample in the input data path. It must be noted that in both parallel schemes, one clock cycle is needed for the generation of an output sample. Finally, from the SA_{TOTAL} column, it is proven that the switching activity of the proposed parallel scheme is less than the scheme of Fig. 1 for the production of an output sample. Therefore, the proposed merged parallel scheme performs better than the transpose architecture of Fig. 1 from the aspect of dynamic power consumption. Provided that it has also less hardware complexity, as it is shown in Table 1, it performs better than the structure of Fig. 1 from the aspect of total power consumption, considering a certain CMOS VLSI technology [14].

On the other hand, from the SA_{ACC} column of Table 4, it is shown that the proposed digit-serial merged scheme has half switching activity per output sample, compared to the design shown in [1], while the two schemes have equal switching activity in the input data path. It is also noticed that *f* clock cycles are required for the production of an output sample in both compared digit-serial schemes. Thus, as it is also shown in the SA_{TOTAL} column, the proposed digit-serial filter scheme has half total switching activity per output sample, compared to the scheme presented in [1]. Therefore, it performs significantly better than the circuit shown in [1], from the aspect of dynamic power dissipation. Furthermore, as that the proposed serial scheme results in a circuit with less hardware complexity, as shown in Table 2, it will result into a circuit with less total power dissipation, when implemented in a certain CMOS VLSI technology.

Table 3. Switching activity comparison of the parallel schemes.

Filter scheme (*k* taps)	SA_{ACC}	SA_{DATA}	SA_{TOTAL}
Proposed Scheme	$k(q+m+\log_2 k)$	$(k-1)q$	$2kq+k(m+\log_2 k)$
Transpose Fig. 1	$2k(q+m+\log_2 k)$	q	$2kq+2k(m+\log_2 k)$

Table 4. Switching activity comparison of the serial schemes.

Filter scheme (*k* taps)	SA_{ACC}	SA_{DATA}	SA_{TOTAL}
Proposed Scheme	$(k/2)(m+\log_2 k)$	$(k-1)q^2$	$(k-1)q^2+(k/2)(m+\log_2 k)$
Digit-Serial Transpose[1]	$k(m+\log_2 k)$	$(k-1)q^2$	$(k-1)q^2+(k/2)(m+\log_2 k)$

5. CONCLUSION AND FUTURE WORK

In this paper, two novel hardware efficient and reduced power programmable FIR architectures with low latency are presented, based on the merging technique. In the first, input data enters the circuit in bit-parallel form, while in the second it is transformed into Modified-Booth encoded digits. Both schemes perform better than relative parallel and serial FIR implementations presented in the bibliography. The proposed architectures can be further explored, targeting the efficient implementation of symmetrical linear phase FIR filters, while they can also be enriched with synthesis results of critical time delay, area and total power dissipation.

6. REFERENCES

[1] O. T.-C. Chen and L.H Chen, "A Hardware-Efficient FIR Architecture with Input-Data Folding and Tap-Folding", Proceedings on IEEE International Symposium on Circuits and Systems (ISCAS) (1) 2005, pp.544-547.

[2] O.T.-C. Chen and L.H. Chen, "A Hardware-Efficient Programmable FIR Processor using Input-Folding and Tap-Folding", EURASIP Journal on Advances on Signal Processing, Volume 2007, Article ID 95523, 14 pages.

[3] A. T. Erdogan and T. Arslan, "Low Power FIR Filter Implementations Based on a Coefficient Ordering Algorithm", Proceedings of the IEEE Computer Society on VLSI Emerging Trends in VLSI System Design (ISVLSI'04), 2004.

[4] A. T. Erdogan and T. Arslan, "On the Low Power Implementation of FIR Filtering Structures on Single Multiplier DSPs", IEEE Transactions on Circuits and Systems II- Analog and Digital Signal Processing, vol. 49, no 3, March 2002, pp. 223-229.

[5] Vijai Syndararajan and Keshab. K. Parhi, "Synthesis of Low Power Folded Programmable Coefficient FIR Digital Filters", ASP-DAC 2000, pp.153-156.

[6] Richard I. Hartley, "Subexpression sharing in Filters using Canonic Sign Digit Multipliers", IEEE Transactions on Circuits and Systems II-Analog and Digital Signal Processing, vol. 43, no. 10, October 1996, pp. 677-688.

[7] R.Pasko et al., "A New Algorithm for Elimination of Common Subexpressions", IEEE Transactions on Computer-Aided Design, vol. 18, no. 1, January 1999, pp. 58-68.

[8] Andrew. G. Dempster, Malcolm D. Macleod, "Use of Minimum-Adder Multiplier Blocks in FIR Digital Filters", IEEE Transactions on Circuits and Systems II- Analog and Digital Signal Processing, vol. 42, no. 9, September 1995, pp. 569-577.

[9] Jonksung Park et al., "High Performance and Low Power Filter Design Based on Sharing Multiplication", International Symposium of Low Power Electronics and Design (ISLPED) 2002, Monterey, California, USA, August 12-14, 2002.

[10] Paraskevas Kalivas, Paul Bougas, Andreas Tsirikos, George Oikonomakos and Kiamal Z. Pekmestzi, "New Systolic and Low Latency Parallel FIR Filter Schemes", International Journal of Signal Processing 1 (1), 2004, 1-12.

[11] Dimitris Bekiaris, Isidoros Sideris, George Economakos and Kiamal Z. Pekmestzi, "Power-Efficient and Low-Latency Implementation of Programmable FIR Filters using Carry-Save Arithmetic", 14th International Symposium on Electronics, Circuits and Systems (ICECS), Marakech, Morocco, December, 11-14, 2007.

[12] Mahesh Mendehale, Sunil D. Shrelekar and G. Venkatesh, "Low Power Realization of FIR Filters on Programmable DSPs", IEEE Transactions on VLSI Systems, vol. 6, no. 4, December 1998, pp. 546-553.

[13] I.Sideris, K.Anagnostopoulos, P. Kalivas and K.Z. Pekmestzi, "Novel Systolic Schemes for Serial-Parallel Multiplication", EUSIPCO 2005.

[14] N. Weste, D. Harris, "CMOS VLSI Design: A Circuits and Systems Perspective", Addison-Wesley Publishing Company (2005).