

Therefore, matching like powers in (68), we get

$$c_2(n) = \frac{g(n)}{n + 1/2} \quad (69)$$

or

$$c_2(n) = \frac{2}{2n + 1} \cdot \frac{1}{n!} \cdot \left(\frac{1}{2}\right)_n. \quad (70)$$

REFERENCES

- [1] J. Le Bihan, "Maximally linear FIR digital differentiators," *J. Circuits, Syst., Signal Processing*, vol. 14, pp. 633–637, May 1995.
- [2] —, "Coefficients of FIR digital differentiators and Hilbert transformers for midband frequencies," *IEEE Trans. Circuits Syst. II*, vol. 43, pp. 272–274, Mar. 1996.
- [3] B. Carlsson, "Maximum flat digital differentiator," *Electron. Lett.*, vol. 27, no. 8, pp. 675–677, Apr. 11, 1991.
- [4] DSP Committee, *Selected Papers in Digital Signal Processing, II*. New York: IEEE Press, 1976.
- [5] E. Hermanowicz and M. Rojewski, "Design of FIR first order digital differentiators of variable fractional sample delay using maximally flat error criterion," *Electron. Lett.*, vol. 30, no. 1, pp. 17–18, Jan. 6, 1994.
- [6] O. Herrmann, "On the approximation problem in nonrecursive digital filter design," *IEEE Trans. Circuit Theory*, vol. 18, pp. 411–413, May 1971.
- [7] I. R. Khan and R. Ohba, "New design of full band differentiators based on Taylor series," in *Proc. Inst. Elect. Eng.—Visual Image Signal Process*, vol. 146, Aug. 1999, pp. 185–189.
- [8] B. Kumar and S. C. D. Roy, "Coefficients of maximally linear, FIR digital differentiators for low frequencies," *Electron. Lett.*, vol. 24, no. 9, pp. 563–565, Apr. 28, 1988.
- [9] —, "Design of digital differentiators for low frequencies," *Proc. IEEE*, vol. 76, no. 3, pp. 287–289, Mar. 1988.
- [10] —, "Design of efficient FIR digital differentiators and Hilbert transformers for midband frequency ranges," *Int. J. Circuit Theory Appl.*, vol. 17, pp. 483–488, 1989.
- [11] —, "Maximally linear FIR digital differentiators for midband frequencies," *Int. J. Circuit Theory Appl.*, vol. 17, pp. 21–27, 1989.
- [12] B. Kumar, S. C. D. Roy, and H. Shah, "On the design of FIR digital differentiators which are maximally linear at the frequency π/p , $p \in \{\text{positive integers}\}$," *IEEE Trans. Signal Processing*, vol. 40, pp. 2334–2338, Sept. 1992.
- [13] T. W. Parks and C. S. Burrus, *Digital Filter Design*. New York: Wiley, 1987.
- [14] S.-C. Pei and P.-H. Wang, "Closed-form design of maximally flat FIR Hilbert transformers, differentiators, and fractional delayers by power series expansion," *IEEE Trans. Circuits Syst. I*, vol. 48, pp. 389–389, Apr. 2001.
- [15] M. Petkovšek, H. S. Wilf, and D. Zeilberger, *A = B*. Natick, MA: A.K. Peters, 1996. [Online]. Available: <http://www.cis.upenn.edu/~wilf/AeqB.html>.
- [16] W. Preuss, , NY: Symbol Technologies, 2000. Personal communication.
- [17] R. Rabenstein, "Design of FIR digital filters with flatness constraints for the error function," *Circuits, Syst., Signal Processing*, vol. 13, pp. 77–97, Jan. 1993.
- [18] L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975.
- [19] L. R. Rabiner, J. H. McClellan, and T. W. Parks, "FIR digital filter design techniques using weighted Chebyshev approximation," *Proc. IEEE*, vol. 63, pp. 595–610, Apr. 1975.
- [20] L. R. Rabiner and C. M. Rader, Eds., *Digital Signal Processing*. New York: IEEE Press, 1972.
- [21] M. R. R. Reddy, S. C. D. Roy, and B. Kumar, "Design of efficient second and higher degree FIR digital differentiators for midband frequencies," in *Proc. Inst. Elect. Eng.*, vol. 138, Feb. 1991, pp. 29–33.
- [22] S. Samadi, T. Cooklev, A. Nishihara, and N. Fujii, "Multiplierless structure for maximally flat linear phase FIR filters," *Electron. Lett.*, vol. 29, no. 2, pp. 184–185, Jan. 21, 1993.
- [23] S. Samadi, A. Nishihara, and H. Iwakura, "Universal maximally flat low-pass FIR systems," *IEEE Trans. Signal Processing*, vol. 48, pp. 1956–1964, July 2000.
- [24] H. W. Schüssler and P. Steffen, "Some advanced topics in filter design," in *Advanced Topics in Signal Processing*, J. S. Lim and A. V. Oppenheim, Eds. Englewood Cliffs, NJ: Prentice-Hall, 1988, ch. 8, pp. 416–491.
- [25] I. W. Selesnick and C. S. Burrus, "Exchange algorithms for the design of linear phase FIR filters and differentiators having flat monotonic passbands and equiripple stopbands," *IEEE Trans. Circuits Syst. II*, vol. 43, pp. 671–675, Sept. 1996.

On the Low-Power Implementation of FIR Filtering Structures on Single Multiplier DSPs

Ahmet Teyfik Erdogan and Tughrul Arslan

Abstract—The authors present three multiplication schemes for the low-power implementation of finite-impulse response (FIR) filters on single multiplier complementary metal-oxide-semiconductor (CMOS) digital signal processors (DSPs). The schemes achieve power reduction through the minimization of switching activity at one or both inputs of the multiplier. In addition, these schemes are characterized by their flexibility since they tradeoff implementation cost against power consumption. Results are provided for a number of example FIR filters demonstrating power savings ranging from 20% with schemes which can be implemented on existing common DSPs, and up to 51% with schemes using enhanced DSP architectures.

Index Terms—Digital signal processors (DSPs), finite-impulse response (FIR) filters, low power design, switching activity.

I. INTRODUCTION

The advent of portable computing has led to a significant increase in research work targeting the reduction of power consumption in high throughput digital signal processor (DSP) devices. It can be shown that the most significant factor affecting power consumption in a complementary metal-oxide-semiconductor (CMOS) very large-scale integration (VLSI) device is the switching power, which is expressed by the product $[1/2 \times (\text{supply voltage})^2 \times \text{switched capacitance} \times f]$ [1]. In this product, the switched capacitance is a combination of the physical capacitance, C , and the switching activity factor, k . This factor is defined as the average number of times that a gate makes a logic transition ($1 \rightarrow 0$ or $0 \rightarrow 1$) in each clock cycle. Therefore, for achieving low power in CMOS circuits one must target minimizing one or more of the parameters V_{dd} , C and k .

Recently a number of researchers have targeted the minimization of switching activity with the aim of reducing power consumption for generic digital circuits, e.g., [2] and [3]. For example, in [2] a synthesis system is developed which synthesises both finite state machines and combinational logic for low-power applications. Low power is achieved by minimizing the average number of transitions at the internal nodes of the combinational circuits and state assignment which minimizes the total number of transitions occurring at the

Manuscript received September 1, 1999; revised June 1, 2001. This paper was recommended by Associate Editor J. LeBlanc.

The authors are with the Department of Electronics and Electrical Engineering, University of Edinburgh, Edinburgh EH9 3JL, Scotland, U.K. (e-mail: ate@ee.ed.ac.uk, arslan@ee.ed.ac.uk).

Publisher Item Identifier S 1057-7130(02)06229-8.

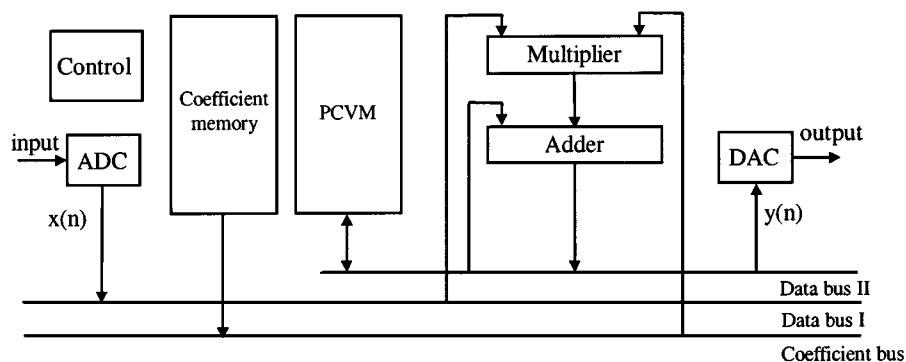


Fig. 1. DSP processor architecture for TDF FIR filter implementation.

present state inputs of the finite state machine. A low-power technology decomposition procedure is introduced in [3] which produces a decomposed network with minimum switching activities. This is achieved together with a low-power technology mapping algorithm that hides high switching nodes inside gates and lowers the fanout load driven by high switching activity nodes. Researchers have also targeted reducing the switching activity within DSP systems. These have included (1) the use of coding techniques [4], (2) the design and manipulation of arithmetic units such as adders and multipliers [5], [6], (3) the use of various techniques to exploit bit correlation in coefficients [7]–[9]. Other techniques include dynamic minimization of filter orders [10], following a differential approach to processing coefficients [11], and use of multirate architectures [12].

This paper presents a number of multiplication schemes for low-power implementation of FIR filters on DSPs. The schemes target reducing power by reducing switching activity at one or both multiplier inputs, depending on the filter realization structure used. Results are provided which show up to 51% saving in power. The schemes also provide the designer with flexibility in terms of power/cost tradeoff.

II. MULTIPLICATION SCHEMES

The authors developed a number of multiplication schemes for the low-power implementation of FIR filters on CMOS based single multiplier DSPs. The schemes vary in the technique employed in the manipulation of data and coefficient inputs of the multiplier and the use of filter realization structure, where either one of the direct form (DF) or the transpose direct form (TDF) realization is utilized.

A. Scheme I: TDF Structure

A DF filter implemented on a single multiplier DSP is executed such that at each clock cycle a new data sample $x(n)$ and the corresponding filter coefficient $h(k)$ are fetched from the memory simultaneously and applied to the multiplier. Therefore, for each multiplication both inputs of the multiplier receive new data. Due to this continuous change at both inputs, using the DF realization will cause a high level of switching activity at both multiplier inputs. This will in turn cause a correspondingly high-switching activity within the multiplier, hence leading to a higher overall power consumption. Using the TDF realization, however, the switching activity at data inputs of the multiplier is significantly reduced since the data input remains unchanged for a substantial number of multiplication operations, corresponding to the filter length. This results in considerably fewer switching activities within the multiplier circuit and consequently leads to less power consumption than the DF realization [13].

In a TDF realization a single input sample, $x(n)$, and a number of intermediate values, which we term, precalculated values (PCVs), cor-

responding to the outputs of the respective filter stages are required for the calculation of an output sample, $y(n)$. The PCV for a filter stage k at time instant n is given by

$$PCV_k(n) = h(k)x(n) + PCV_{k+1}(n-1). \quad (1)$$

Hence, PCVs at different stages of an N -tap filter are given by:

$$\begin{aligned} PCV_{N-1}(n) &= h(N-1)x(n) + PCV_N(n-1) \\ &\vdots \\ PCV_2(n) &= h(2)x(n) + PCV_3(n-1) \\ PCV_1(n) &= h(1)x(n) + PCV_2(n-1) \\ PCV_0(n) &= h(0)x(n) + PCV_1(n-1). \end{aligned}$$

The PCVs above could in turn be expressed as follows:

$$\begin{aligned} PCV_{N-1}(n) &= h(N-1)x(n) \\ &\vdots \\ PCV_2(n) &= h(2)x(n) + h(3)x(n-1) + \dots \\ &\quad + h(n-1)x(n-[N-1]) \\ PCV_1(n) &= h(1)x(n) + h(2)x(n-1) + \dots \\ &\quad + h(n-1)x(n-[N-1]) \\ PCV_0(n) &= h(0)x(n) + h(1)x(n-1) + \dots \\ &\quad + h(n-1)x(n-[N-1]). \end{aligned}$$

Note that the final value, $PCV_0(n)$, is equal to the filter output, $y(n)$. The above implies the need for the allocation of a special memory block, within the data memory unit, for the storage of PCVs. This we term precalculated value memory (PCVM). Secondly, the conventional multiply-accumulate (MAC) operation can not be used with the TDF realization, since the output of each filter stage has to be saved in memory in order to realize the delays between the adders in the TDF structure. Instead, the result of multiplying the current data sample with a given coefficient is added to the PCV read from the PCVM resulting in another PCV. This will be stored back in the PCVM for use with the next data sample.

A TDF filter can be implemented more efficiently by performing simple modifications to the DSP hardware architecture. For example, by allowing a separate data bus path for writing back the PCVs generated above, see Fig. 1, significant increase in throughput could be achieved. This is currently possible since a large number of common DSP cores are available as intellectual properties (IPs) in a single chip configurable DSP units which allow modification to the DSP archi-

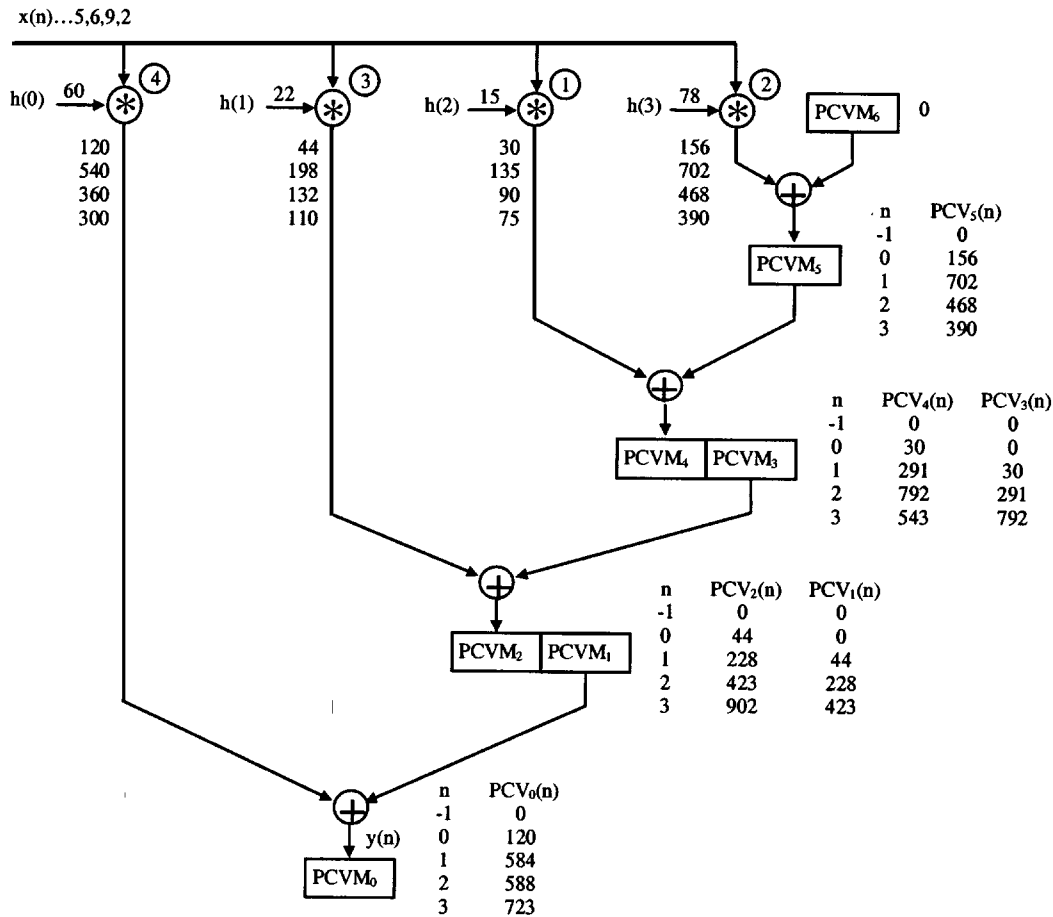


Fig. 2. Example of a 4-tap TDF filter structure with coefficient ordering.

ture for aspects such as added functionality and/or enhancement to performance [14].

The sequence of steps for the implementation of this scheme is as follows:

- 1) get the data sample $x(n)$;
- 2) get the filter coefficient $h(k)$;
- 3) multiply $x(n)$ and $h(k)$;
- 4) get the PCV corresponding to the output of the next filter stage for the previous data sample, $PCV_{k+1}(n-1)$;
- 5) sum the results of 3) and 4) above and store as $PCV_k(n)$ in PCVM;
- 6) repeat steps 2) to 5) for the remaining coefficients;
- 7) get the filter output, $y(n)$, from PCVM₀;
- 8) repeat steps 1) to 7) for the next filter output, $y(n+1)$.

B. Scheme II: TDF Structure With Coefficient Ordering

A further reduction in switching activity within the multiplier section of the TDF filter can be achieved by implementing the filter such that respective data samples are multiplied with filter coefficients in a nonsequential order of their respective filter stages. The ability to re-order multiplication operations is desirable since it provides the potential for the switching activity at the coefficient input of the multiplier to be reduced by minimizing the *Hamming distance* between those filter coefficients used in successive multiplication operations. The authors have verified this in [15]. This results in reducing the switching activity at both multiplier inputs. The choice of an ordered coefficient set in which successive coefficients are highly correlated is both computationally complex and NP-complete for practical size filters. The

identification of such an order will require an effective search algorithm. A number of probabilistic algorithms have proved successful in solving NP-complete search problems such as above. Examples are simulated annealing and genetic algorithms [16]. Due to both their inherent parallelism and their ability to cope with complex discontinuous search spaces, genetic algorithms have been selected for ordering the coefficients in this scheme. A combinatorial order-based genetic algorithm was used to produce coefficient orders with significant reduction in switching activity. Further details of this algorithm could be found in [17].

It must be noted that the ordering of coefficients is performed only once prior to the commencement of filtering. Subsequent use of the filter will utilize the same order of coefficients. For this reason, coefficient ordering has no implications on the speed of the filtering procedure.

When coefficients are processed in nonsequential order two possible situations may arise.

- a) The previous filter stage, *stage* $k-1$, has been evaluated in a previous cycle, in which case a single PCVM location is required to store the new $PCV_k(n)$, as in (1).
- b) *Stage* $k-1$ has not been evaluated yet, in which case two PCVM locations will be required, one to store the new PCV, i.e., $PCV_k(n)$, resulting from the current evaluation of *stage* k [as in (a)] and another to preserve the original PCV, i.e., $PCV_k(n-1)$, as $PCV_{k-1}(n)$ [see (2)] for the evaluation of *stage* $k-1$

$$PCV_{k-1}(n) = PCV_k(n-1). \quad (2)$$

The sequence of steps for the implementation of this scheme is as follows:

- 1) apply the coefficient ordering algorithm and produce the re-ordered coefficient set;
- 2) get the data sample $x(n)$;
- 3) get the filter coefficient $h(k)$;
- 4) multiply $x(n)$ and $h(k)$;
- 5) get the output of the next filter stage for the previous data sample, $PCV(n-1)$;
- 6) sum the results of 4) and 5) above;
- 7) if situation a), see above, arises then store the result of 6) into the reserved PCVM location. Otherwise, situation b), transfer the contents of the first PCVM location to the second PCVM location and then store the result of 6) into the first PCVM location;
- 8) repeat steps 2) to 7) for the remaining coefficients;
- 9) get the filter output, $y(n)$, from $PCVM_0$;
- 10) repeat steps 2) to 9) for the next filter output, $y(n+1)$.

Fig. 2 illustrates a signal flow graph of a 4-tap TDF filter structure where the multiplication operations have been reordered. Circled numbers in the graph indicate the execution order, i.e., input data samples are multiplied by coefficients in the order of $h(2)$, $h(3)$, $h(1)$, and $h(0)$. Initially, at $n = -1$ the seven memory locations $PCVM_0$ to $PCVM_6$, corresponding to PCV_0 to PCV_6 , are all initialized to zero. At $n = 0$, the data sample $x(0)$ ($= 2$) is processed by the filter. First, the output of filter stage 2 is calculated by multiplying $x(0)$ ($= 2$) with the coefficient $h(2)$ ($= 15$). The result ($2 \times 15 = 30$) is then summed with the output of the next filter stage for the previous data sample, $PCV_5(-1)$ ($= 0$), and the result ($30 + 0 = 30$) is stored as $PCV_4(0)$. However, because the current value of $PCVM_4$ ($= PCV_4(-1)$) will be needed for the evaluation of the previous filter stage, it is saved as $PCV_3(0)$ in $PCVM_3$ before $PCV_4(0)$ ($= 30$) is stored in $PCVM_4$. The output of the filter stage 3 is next calculated by multiplying $x(0)$ ($= 2$) with the filter coefficient $h(3)$ ($= 78$). The result ($2 \times 78 = 156$) is then stored as $PCV_5(0)$. The outputs of the remaining filter stages are then calculated in succession before the second data sample $x(1)$ ($= 9$) is processed at $n = 1$. The third and fourth samples $x(2)$ and $x(3)$ are subsequently processed in the same manner.

To demonstrate the flexibility of these schemes a pure software implementation of scheme II is illustrated in Fig. 3. The figure shows the assembler code compiled for TMS320C54x target as an example.

C. Scheme III: DF Structure With Coefficient Ordering

The previous two schemes are restricted to TDF realization of FIR filters which dictate the need for a modified DSP processor architecture. Hence, these schemes could not be implemented efficiently on traditional DSPs using the conventional MAC unit. However, most common DSPs in the market are based upon the conventional MAC unit. Examples are Texas Instruments' and Motorola's DSP series. For this reason, there is a need for low-power implementation schemes for the above processors with minimum (or no) modifications to their architectures. In this section we describe another multiplication scheme for low-power implementation of DF FIR structures. The scheme uses coefficient ordering (as in scheme II). However, since the scheme is implemented with the DF structure it does not require any PCVs. Therefore, it can easily be adapted to most common DSPs in the market without the need for any alterations to their architectures.

The sequence of steps for the implementation of this scheme is as follows:

- 1) apply the coefficient ordering algorithm and produce the re-ordered coefficient set;
- 2) clear the accumulator;

```

_main:
    PSHM    AR1
    FRAME  #-4
    STM    #_x, AR1
    STM    #_y, AR2
    STM    #99, BRC
    RPTB   L3-1
    ; loop starts

L2:
    LD     *AR1+, 16, T
    DLD   *(_pcv+8), A
    DST   A, *(_pcv+6)
    MPY   *(_h+2), A
    DADD  *(_pcv+10), A
    DST   A, *(_pcv+8)
    MPY   *(_h+3), A
    DST   A, *(_pcv+10)
    DLD   *(_pcv+4), A
    DST   A, *(_pcv+2)
    MPY   *(_h+1), A
    DADD  *(_pcv+6), A
    DST   A, *(_pcv+4)
    MPY   *(_h), A
    DADD  *(_pcv+2), A
    DST   A, *(_pcv)
    SSBX  SXM
    SFTA  A, 8
    SFTA  A, -8
    SFTA  A, #-15, B
    STL   B, *AR2+
    ; loop ends

L3:
    FRAME  #4
    POPM   AR1
    RET
    ; return occurs

```

Fig. 3. TMS320C54x assembler code for the filter example in Fig. 2.

- 3) get the filter coefficient $h(k)$;
- 4) get the corresponding data sample $x(n-k)$;
- 5) multiply $h(k)$ and $x(n-k)$;
- 6) add the product into the accumulator;
- 7) repeat steps 3) to 6) for the remaining coefficients;
- 8) get the filter output, $y(n)$, from the accumulator;
- 9) repeat steps 2) to 8) for the next filter output, $y(n+1)$.

Fig. 4 illustrates the signal flow graph for scheme III using the same filter example as in Section II-B with a different coefficient order.

III. EVALUATION OF THE SCHEMES

The efficiency of the above schemes is demonstrated using a number of practical FIR filters. For each filter specification a coefficient set, *norm*, is obtained using the MATLAB¹ software program. Next, a second coefficient set, *min*, is derived from the original set by ordering the coefficients such that Hamming distance between adjacent coefficients is minimized. This is followed by generating a uniformly distributed set of data samples. These data samples are associated with each of the coefficient sets obtained in the previous stage, *norm* and *min*, using the filter structures DF and TDF. This ensures that the appropriate data and coefficient inputs are applied to the multiplier. At this stage an input simulation file is generated for each of the structure/ordering combinations, i.e., DF/*norm*, DF/*min*, TDF/*norm*, and TDF/*min*. Each simulation file is run with a

¹MATLAB is a registered trademark of The MathWorks, Natick, MA.

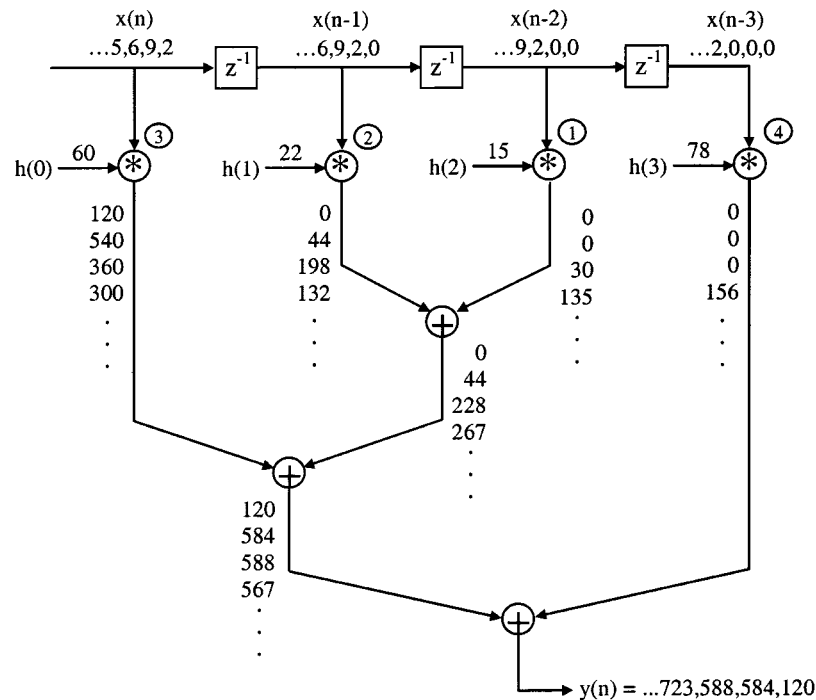


Fig. 4. Example of a 4-tap DF filter structure with coefficient ordering.

TABLE I
AVERAGE REDUCTIONS IN SWITCHED CAPACITANCE

Multiplier size	Structure/ Ordering	Transitions per sample at data inputs	Reduction (%)	Transitions per sample at coefficient inputs	Reduction (%)	Switched capacitance per sample (pF)	Reduction (%)
8-bit	DF/norm	137	-	90	-	522	-
	DF/min	138	-1	26	71	417	20
	TDF/norm	4	97	90	0	311	40
	TDF/min	4	97	26	71	100	81
16-bit	DF/norm	276	-	261	-	4093	-
	DF/min	279	-1	123	53	3499	15
	TDF/norm	8	97	261	0	2794	32
	TDF/min	8	97	123	53	1457	64
24-bit	DF/norm	412	-	402	-	14792	-
	DF/min	415	-1	238	41	13411	9
	TDF/norm	12	97	402	0	10860	27
	TDF/min	12	97	238	41	7656	48

separate version of the Cadence's Verilog-XL² digital logic simulator. Verilog-XL uses a gate-level netlist of the multiplier circuit for the simulation procedure. During a given simulation, the number of signal transitions for each gate within the multiplier circuit is monitored. The capacitance of each gate, which consists of the output capacitance of the gate itself, input capacitances of the loading gates and capacitance of the wires connected to the output, is extracted by performing a layout of the multiplier circuit using the Envisia² Silicon Ensemble² place-and-route software targeting a 0.7- μm standard cell CMOS technology. Both capacitive information and the switching activity figures are used to obtain the switched capacitance of each gate. This is then accumulated to obtain the overall switched capacitance of the entire

multiplier circuit. This is repeated for each of the structure/ordering combinations. The overall switched capacitance of each TDF/norm, TDF/min, and DF/min structure/ordering combinations is compared to the overall switched capacitance of DF/norm in order to calculate the amount of power saving.

IV. SIMULATIONS AND RESULTS

Simulations were performed on 8×8 , 16×16 and 24×24 -b Baugh-Wooley two's complement array multipliers [18] for three low-pass and two band-pass linear phase filter examples with filter lengths varying from 54 to 89. Each filter was simulated using 1000 randomly generated data samples. Table I shows the average results of simulating these filter examples. The results reflect the reduction in switching activity at data and coefficient inputs of the multiplier,

²Verilog-XL, Envisia, and Silicon Ensemble are registered trademarks of Cadence Design Systems, Inc., San Jose, CA.

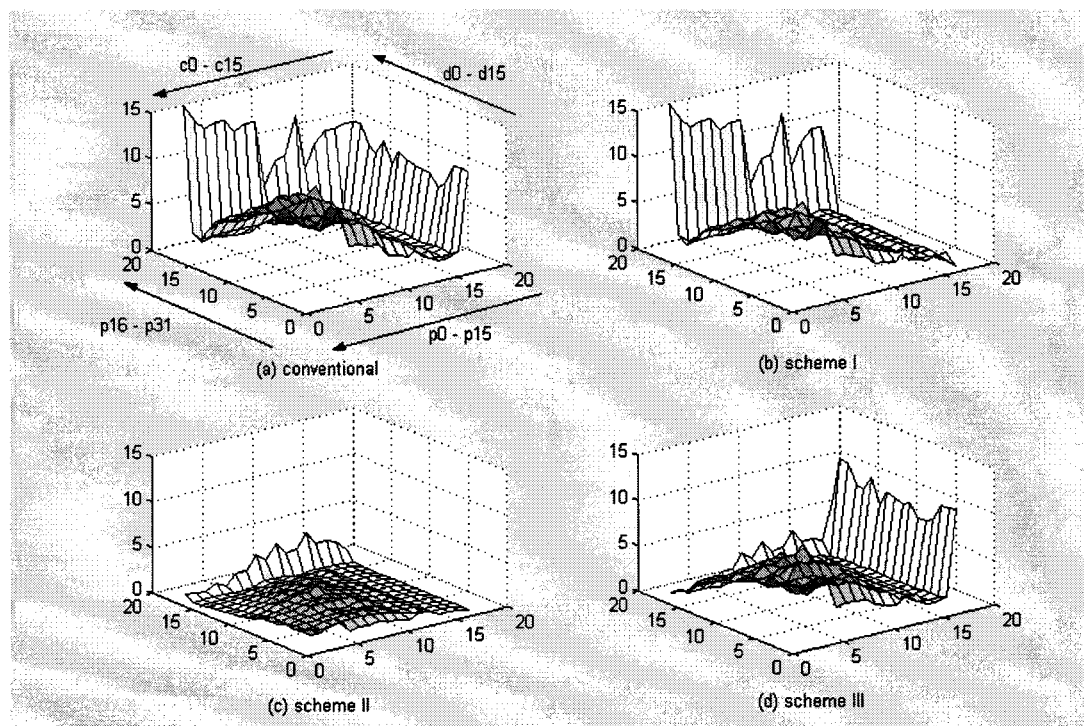


Fig. 5. Switched capacitance distribution for different multiplication schemes.

and in switched capacitance of the entire multiplier. The reduction in switching activity at the data input is 97% for TDF since the data input remains unchanged until all the coefficients are processed. Whereas the data samples for DF are uncorrelated and hence the change in their order may yield a slight positive or negative reduction in switching activity. It happens that in our case the switching activity at the data input increases slightly (by only 1%) although it may also decrease for other data sets. The reduction in switching activity at the coefficient input is up to 71% for DF/min and TDF/min due to coefficient ordering. There is no reduction for TDF/norm since in this case the coefficients are processed in their normal order. The switched capacitance for the multiplier is directly related to the switching activity of its inputs. In all cases the results demonstrate a reduction in switched capacitance. The best results are obtained with TDF/min achieving 81%, 64%, and 48% reductions using 8×8 , 16×16 and 24×24 -b multipliers, respectively. The second best (40%, 32%, and 27%) and the third best results (20%, 15%, and 9%) are obtained with TDF/norm and DF/min using 8×8 , 16×16 and 24×24 -b multipliers, respectively.

In order to examine the effect of the reductions achieved in switched capacitance on the internal circuitry of the multiplier, for the various schemes, 3D diagrams have been used. These reflect the switched capacitance at the output of each constituent gate within the multiplier circuit. Fig. 5(a) illustrates the switched capacitance distribution for a 16×16 -b array multiplier circuit when conventional filtering is employed. As expected, the switched capacitance is high at both inputs (data inputs: d0–d15 and coefficient inputs: c0–c15) of the multiplier. This is due to both high switching activity and high loading capacitance at the primary inputs. The high switching activity at the inputs is propagated toward the internal circuitry, having a relatively higher impact on switched capacitance in the gates toward the circuit outputs. Applying scheme I reduces the switching activity at data inputs of the multiplier, whereas the activity remains the same at the coefficient inputs, see Fig. 5(b). This decrease in switching

activity is reflected in a small region of high switched capacitance toward the outputs of the multiplier [compared with Fig. 5(a)]. With scheme II, however, the switching activity at both multiplier inputs is reduced, hence significantly reducing the switched capacitance of the internal gates of the multiplier circuit, see Fig. 5(c). The effect of applying scheme III is similar to that of scheme I. In this case, the switching activity at the coefficient inputs is reduced while the switching activity at the data inputs being the same. Therefore, a similar effect is propagated through the internal gates of the multiplier circuit, see Fig. 5(d).

V. DETERMINATION OF OVERHEADS

The use of TDF structure imposes some overheads, mainly due to PCVs, in comparison with the DF structure. In order to account for these overheads the PCVM unit was simulated and its switching activity monitored. In order to obtain an accurate estimate of the overheads, the switched capacitance of the bus and memory units is required, in addition to that of the multiplier. For this we utilize a high level switched capacitance model for the critical modules. The model is based on the principles used by the authors in [4] and assumes that the total switched capacitance, SC_{total} , is given by (3) as shown below:

$$SC_{total} = k_{mult}C_{mult} + k_{mem}C_{mem} + k_{bus}C_{bus}. \quad (3)$$

The parameters k_{mult} , k_{mem} , and k_{bus} represent average number of transitions for the multiplier, memory, and bus units, respectively. Verilog-XL simulator was used to obtain k_{mult} , whereas k_{mem} , and k_{bus} were obtained through functional simulations.

The choice of the “average” capacitance values (C_{mult} , C_{mem} , and C_{bus}) was performed after the analysis of a number of different FIR filter chips, developed by the group here. The values 1, 2, and 5 were chosen to be a true reflection of C_{mult} , C_{mem} , and C_{bus} , respectively.

TABLE II
AVERAGE OVERHEADS

Multiplier size	Structure/ordering	# transitions/sample			switched-capacitance/sample			SC _{total}	Reduction (%)
		Multiplier (k _{mult})	Memory (k _{mem})	Buses (k _{bus})	Multiplier (k _{mult} ·C _{mult})	Memory (k _{mem} ·C _{mem})	Buses (k _{bus} ·C _{bus})		
8-bit	DF/norm	6813	4	227	6813	8	1135	7956	-
	TDF/norm	4269	299	393	4269	598	1965	6832	14
	TDF/min	1385	395	429	1385	790	2145	4320	45
16-bit	DF/norm	57196	8	537	57196	16	2685	59897	-
	TDF/norm	39972	720	989	39972	1440	4945	46357	22
	TDF/min	21097	1074	1205	21097	2148	6025	29270	51
24-bit	DF/norm	198938	12	814	198938	24	4070	203032	-
	TDF/norm	149301	1067	1481	149301	2134	7405	158840	21
	TDF/min	105983	1578	1828	105983	3156	9140	118279	41

These values were chosen to reflect the amount of switched capacitance within the circuitry of these units. Table II illustrates the average overheads. Column 3 shows the average number of signal transitions per sample for the multiplier, memory, and bus units, respectively. The estimated switched capacitance values are shown in column 4. Column 5 displays the total switched capacitance of the multiplier, memory and bus units. The reductions in switched capacitance, after including the effect of overheads, achieved by TDF/norm and TDF/min in comparison with DF/norm are shown in column 6. As the table illustrates the maximum reduction (51%) is obtained in the case of a 16 × 16-b multiplier with TDF/min structure/ordering.

VI. CONCLUSION

The paper has compared a number of multiplication schemes for the low-power implementation of FIR filters on single multiplier DSPs. The schemes target reducing power consumption through a reduction in the amount of switched capacitance within the multiplier section of the DSPs. Our results indicate that with an added cost penalty, reflected in the modification of the DSP architecture, up to 51% power saving can be achieved. Whereas with least cost penalty, involving modification at software level, up to 20% power saving can be achieved, hence providing the DSP designer with implementation flexibility.

REFERENCES

- [1] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low power CMOS digital design," *IEEE J. Solid-State Circuits*, vol. 27, pp. 473–484, Apr. 1992.
- [2] K. Roy and S. C. Prasad, "Circuit activity based logic synthesis for low power reliable operations," *IEEE Trans. VLSI Syst.*, vol. 1, pp. 503–513, Dec. 1993.
- [3] C. Y. Tsui, M. Pedram, and A. M. Despain, "Power efficient technology decomposition and mapping under an extended power consumption model," *IEEE Trans. Computer-Aided Design*, vol. 13, pp. 1110–1122, Sept. 1994.
- [4] M. R. Stan and W. P. Burleson, "Bus-invert coding for low-power I/O," *IEEE Trans. VLSI Syst.*, vol. 3, no. 1, pp. 49–58, Mar. 1995.
- [5] T. K. Callaway and E. E. Swartzlander, "Optimizing arithmetic elements for signal processing," in *VLSI Signal Processing V*, K. Yao, R. Jain, W. Przytula, and J. Rabaey, Eds. New York: IEEE Press, 1992, pp. 91–100.
- [6] A. P. Chandrakasan and R. W. Brodersen, "Minimizing power consumption in digital CMOS circuits," *Proc. IEEE*, vol. 83, no. 4, pp. 498–523, Apr. 1995.
- [7] A. T. Erdogan and T. Arslan, "Low power coefficient segmentation algorithm for FIR filter implementation," *Inst. Elect. Eng. Electron. Lett.*, vol. 34, no. 19, pp. 1817–1819, Sept. 1998.
- [8] T. Arslan and A. T. Erdogan, "Data block processing for low power implementation of direct form FIR filters on single multiplier CMOS DSP's," in *Proc. IEEE ISCAS*, May–June 31–3, 1998, pp. D441–D444.
- [9] M. Mehendale, S. D. Sherlekar, and G. Verukatesh, "Coefficient optimization for low power realization of FIR filters," in *Proc. IEEE Workshop VLSI Signal Processing*, 1995, pp. 352–361.
- [10] T. Ludwig, S. H. Nawab, and A. P. Chandrakasan, "Low power digital filtering using approximate processing," *IEEE J. Solid-State Circuits*, vol. 31, pp. 395–399, Mar. 1996.
- [11] N. Sankaraya, K. Roy, and D. Bhattacharya, "Algorithms for low power and high speed FIR filter realization using differential coefficients," *IEEE Trans. Circuits Syst. II*, vol. 44, pp. 488–497, June 1997.
- [12] M. Mehendale, S. D. Sherlekar, and G. Verukatesh, "Low power realization of FIR filters using multirate architectures," in *Proc. IEEE 9th Int. Conf. VLSI Design*, Jan. 1996, pp. 370–375.
- [13] A. T. Erdogan, "Low power FIR filter implementations for VLSI based DSP systems," Ph.D. dissertation, Cardiff Univ., Wales, U.K., 1999.
- [14] M. Mehendale, S. D. Sherlekar, and G. Verukatesh, "Techniques for low power realization of FIR filters," in *Proc. Asia and South Pacific Design Automation Conf.*, Aug. 29–Sept. 1, 1995, pp. 447–450.
- [15] A. T. Erdogan, T. Arslan, and D. H. Horrocks, "Low power multiplication schemes for single multiplier CMOS based FIR digital filter implementations," in *Proc. IEEE ISCAS*, Hong Kong, June 9–12, 1997, pp. 1940–1943.
- [16] T. Arslan, D. H. Horrocks, and E. Ozdemir, "Structural synthesis of cell-based VLSI circuits using a multi-objective genetic algorithm," *Inst. Elect. Eng. Electron. Lett.*, vol. 32, no. 7, Mar. 1996.
- [17] M. F. Rohani, "A genetic algorithm for coefficients optimization of single multiplier CMOS DSP's," M.Sc. thesis, Cardiff Univ., Wales, U.K., 1998.
- [18] K. Hwang, *Computer Arithmetic: Principles, Architecture, and Design*. New York: Wiley, 1979.