

## LOW POWER IMPLEMENTATION OF LINEAR PHASE FIR FILTERS FOR SINGLE MULTIPLIER CMOS BASED DSPs

A.T. Erdogan and T. Arslan

Cardiff University of Wales  
Cardiff School of Engineering  
Cardiff CF2 3TF, United Kingdom.

ERDOGAN@Cardiff.ac.uk, ARSLAN@Cardiff.ac.uk

### ABSTRACT

Recently, a new scheme for the single multiplier implementation of low power digital filters on CMOS-based DSPs was presented [1,2]. In this paper the scheme is generalised to include linear phase FIR filters (LPFIRs) and its implementation is investigated with two common methods of LPFIR realisation structures. The paper also describes an effective framework which combines layout, timing, and capacitive information, for the evaluation of power consumption for FIR filters. New results are provided which demonstrate up to 85% reduction in overall power consumption. The paper discusses the generalisation of the scheme to LPFIRs and the effects of added overheads, describes the evaluation environment, and provides comprehensive results which show the effectiveness of the scheme as a generic power saving framework for the implementation of FIR filters exploiting coefficient symmetry, where exists, without the need for specialised realisation structures.

### 1. INTRODUCTION

Digital filters are now widely used throughout the electronics industry, with one of the most common types being the Finite Impulse Response (FIR) filter. A typical FIR filter may require a great many stages in order to provide a suitable response and therefore, rather than having a respective multiplier for each filter stage, a more economical means of realising such a device is to use a CMOS based digital signal processing device wherein all multiplication operations are multiplexed through a single multiplier [3]. Of increasing concern to engineers however, is the power consumption of such CMOS based devices, due in particular to their widespread use in a variety of portable, battery powered electronic devices such as mobile phones. It has been shown that the main source of power dissipation in a typical CMOS logic gate is due to switching power,  $P_{sw}$ , given by [4]:

$$P_{sw} = \frac{1}{2} k C_{load} V_{dd}^2 f \quad (1)$$

where  $V_{dd}$  is the supply voltage,  $f$  is the clock frequency,  $C_{load}$  is the load capacitance of the gate and  $k$  is the switching activity factor which is defined as the average number of times that the gate makes a logic transition ( $1 \rightarrow 0$  or  $0 \rightarrow 1$ ) in each clock cycle.

In [1,2] the authors presented a multiplication scheme for single multiplier implementation of FIR filters on CMOS based DSP processors. The scheme operates in conjunction with the transpose direct form FIR filter structure and a modified DSP architecture, which offers a high degree of flexibility since it is open to exploitation by different coefficient ordering algorithms. This architecture incorporates a multiply-add unit and memory units for storing the coefficients and intermediate multiply-add values (we term Pre-Calculated Values, PCVs). Since the transpose direct form structure is used by the scheme, only one data sample is required at a time. This avoids the use of a special data memory unit. Prior to the filtering operation, the coefficients are ordered according to the ordering algorithm used and are stored in the coefficient memory in a special form which incorporates the coefficient value, the address of the corresponding PCV, and a shift flag, SF, indicating a shift operation in the memory. More detailed description of the scheme is provided in [1,2]. An algorithmic outline of the multiplication scheme is given below:

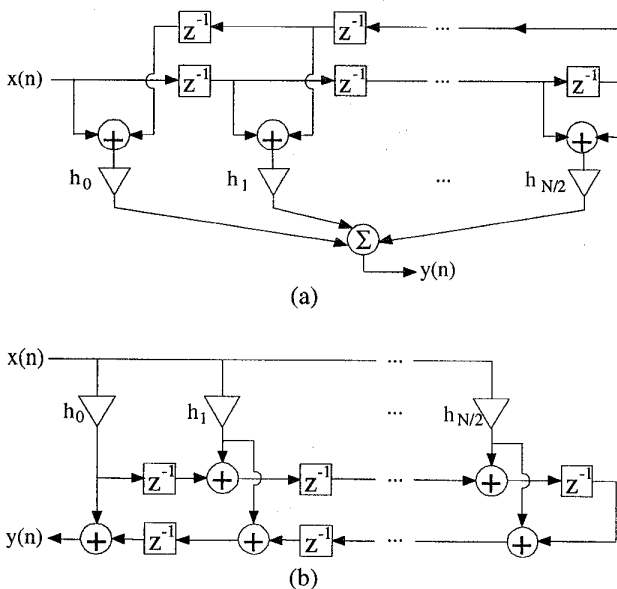
```

for each data sample  $x_n$  ( $n=0,1,2, \dots$ ) {
  for each coefficient-word  $cw_i$  ( $i=0,1,2, \dots,N-1$ ) {
    decode  $cw_i$  into  $h_k$ , PCVMA and SF;
    multiply  $x_n$  and  $h_k$ ;
     $PCV_k = x_n * h_k + [PCVMA]$ ;
    if (SF = 1) {
       $[PCVMA-2] = [PCVMA-1]$ ;
       $[PCVMA-1] = PCV_k$ ;
    }
    else  $[PCVMA-1] = PCV_k$ ;
  } endfor
  get filter output  $y_n$  from PCVM0;
} endfor

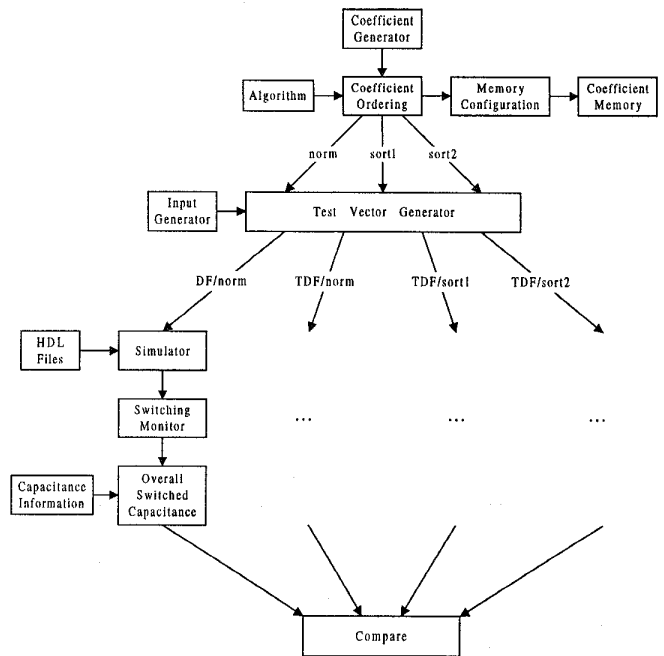
```

where PCVMA is the Pre-Calculated Value Memory Address and [PCVMA] indicates the contents of PCVMA.

Using this scheme the authors showed that up to 63% reduction in switching activity could be achieved. However, the investigations carried out in [1,2] assumed the following: (1) the switching power is proportional to the switching activity with disregard to loading and parasitic capacitances, and spurious timing information such as glitches, (2) a randomly distributed pattern of inputs on both data and coefficient inputs of the multiplier, (3) all filter coefficients are different. In many filtering applications a phase distortion can not be tolerated and hence filters are required to have a linear phase response. Such filters require a symmetric or anti-symmetric (about the central tap) set of coefficients [5]. In this paper the authors extend the scheme to linear phase FIR filters and show a more significant *overall* power reduction of up to 85% using a comprehensive simulation environment which combines layout, timing, and capacitive information. In addition to the transpose direct form realisation, in this work the scheme has been used with another common realisation structure for linear phase FIR filters [5], see Figure 1. This structure, commonly known as *folded* transpose direct form as opposed to the *unfolded* realisation of the transpose direct form, exploits the symmetry in filter coefficients in order to reduce the number of multiplications and coefficient memory locations to about half of that for the unfolded transpose direct form realisation.



**Figure 1:** (a) Folded Direct Form Structure.  
(b) Folded Transpose Direct Form Structure.



**Figure 2:** Scheme for Evaluating Power Consumption.

## 2. PROCEDURE

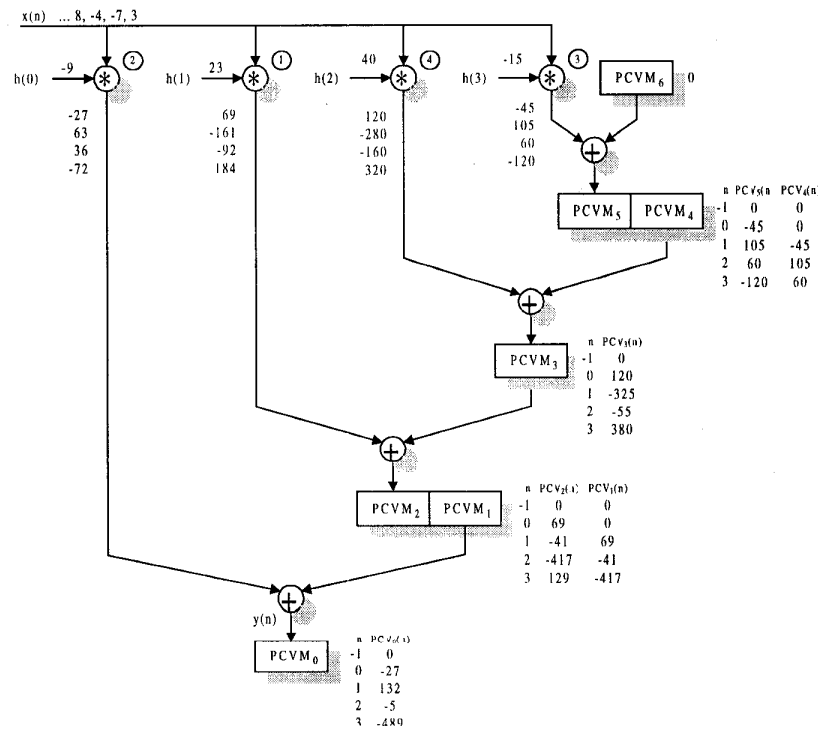
In this work we demonstrate the efficiency of the above scheme using two basic example ordering algorithms, we term  $sort_1$  and  $sort_2$ . In the first, the coefficients to the multiplier are sorted in an ascending order. In the second, the coefficients are ordered by reducing the hamming distance between adjacent coefficients. The coefficients are obtained by designing a number of practical FIR filters, including low-pass, band-pass etc., for different filter specifications. Ninety filters are designed (Ten for each of 32, 64, and 128 filter taps, repeated for wordlengths of 8, 16, and 24-bits). This is followed by randomly generating uniformly distributed sets of data samples, each set corresponding to one of the ninety FIR filters. Next, three coefficient sets are derived from the original coefficient set as shown in Figure 2. The first set is a replica of the coefficients in their original order, *norm*. In the second and third, however, the coefficients are ordered according to  $sort_1$  and  $sort_2$  respectively. The generated input patterns are associated with each of the coefficient sets obtained in the previous stage. This ensures that the appropriate data and coefficient inputs are applied to the multiplier, taking into account the filter structure, either direct form (DF) or transpose direct form (TDF), and the coefficient ordering algorithm. This stage will generate an input simulation file for each of the structure/ordering combinations, i.e. DF/norm, TDF/norm,

TDF/*sort*<sub>1</sub>, and TDF/*sort*<sub>2</sub>. Each simulation file is run with a separate version of the Verilog-XL simulator [6] on a 140 MHz Sun Ultra platform. Verilog-XL uses a hardware description language, HDL, form of the multiplier circuit for the simulation procedure. For each simulation the number of signal transitions for each gate is monitored. Capacitive information (wiring and loading capacitances) are extracted by performing a layout, using Cadence VLSI layout tools with the 0.7 μm CMOS technology, of the multiplier circuit. Both capacitive information and the switching activity figures are used to obtain the switched capacitance of each gate which is then accumulated to obtain the overall switched capacitance of the multiplier. This is performed for each of the structure/ordering combinations. The above procedure is performed for both folded and unfolded filter structures. The best structure/ordering is that with the smallest overall switched capacitance.

Figure 3 illustrates an example of a 4-tap filter for TDF structure with *sort*<sub>2</sub> ordering algorithm having coefficients,  $h(k)=\{-9,23,40,-15\}$  and input sample values,  $x(n)=\{3,-7,-4,8,\dots\}$ . The numbers inside the circles indicate the order in which the coefficients are multiplied by the input samples. The Pre-Calculated Value Memory (PCVM) cells are initialised to '0' and the filtering terminates with the PCVM having values  $\{-489,-417,129,380,60,-120,0\}$  with the final output  $y(3)$ , being -489.

### 3. RESULTS

Simulations were performed on 8 x 8, 16 x 16 and 24 x 24-bit two's complement array multipliers for filter orders of 32, 64 and 128 as shown in tables 1 and 2. The simulations were repeated for 32000, 64000 and 128000 multiplications respectively (i.e., 1000 input samples for each multiplier size/filter order cases). In each case the results obtained are the average of 10 simulations. Table 1 shows the results of simulating the linear phase FIR filters with unfolded TDF structure. The results reflect the reduction in switching power compared to that using the unfolded DF structure. In all cases the results demonstrate a reduction in switching power with a maximum power saving of up to 91.69% with an 8 x 8-bit multiplier, 128-tap filter, and using *sort*<sub>2</sub> ordering algorithm. Table 2 demonstrates a similar comparison between the folded TDF and DF structures with up to 83.68% saving. Figure 4 is an overall comparison of the four filter structures in terms of power consumption with filters of 8-bit wordlengths only. It could be deduced from the figure that without ordering the filter coefficients, i.e. using *norm*, the folded structures achieve approximately twice the power savings of the unfolded structures. This is expected since the latter structures utilise twice the number of multipliers. However, when *sort*<sub>1</sub> and *sort*<sub>2</sub> coefficient ordering algorithms are utilised then the power savings achieved by all structures are approximately the same. This implies that



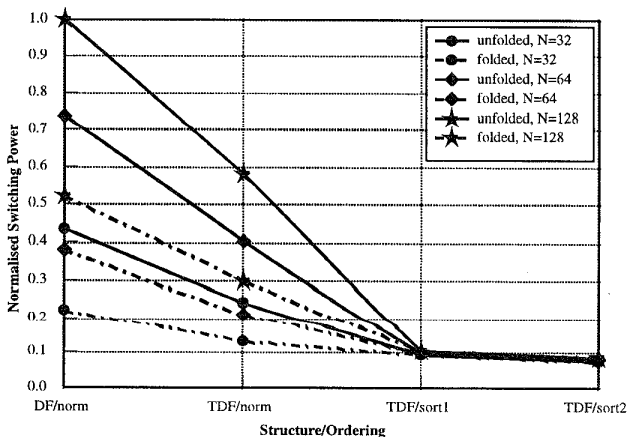
our multiplication scheme exploits the symmetry of the coefficients without the need to specialised implementation structures for linear phase filters. Similar conclusions could be made using 16 x 16 and 24 x 24-bit multipliers.

Multiplier size	Ordering	Filter order		
		32	64	128
		Reduction in power [%]		
8-bit	norm	44.35	45.53	42.10
	sort <sub>1</sub>	78.94	86.92	89.67
	sort <sub>2</sub>	83.08	89.12	91.69
16-bit	norm	35.78	35.41	34.08
	sort <sub>1</sub>	69.80	74.55	78.78
	sort <sub>2</sub>	75.57	80.65	84.54
24-bit	norm	29.97	28.52	22.81
	sort <sub>1</sub>	64.56	66.05	68.05
	sort <sub>2</sub>	72.08	73.32	76.12

**Table 1:** Average Power Reduction with Unfolded Structure.

Multiplier size	Ordering	Filter order		
		32	64	128
		Reduction in power [%]		
8-bit	norm	42.00	44.62	42.64
	sort <sub>1</sub>	59.13	74.51	80.13
	sort <sub>2</sub>	66.68	79.34	83.68
16-bit	norm	34.38	35.22	36.99
	sort <sub>1</sub>	41.95	50.73	60.18
	sort <sub>2</sub>	52.63	62.24	69.92
24-bit	norm	28.03	27.97	27.66
	sort <sub>1</sub>	32.08	35.38	40.16
	sort <sub>2</sub>	44.27	48.45	55.46

**Table 2:** Average Power Reduction with Folded Structure.



**Figure 4:** Switching Power Comparison for Unfolded and Folded Structures.

The results above do not include the effects of overheads due to the scheme. Such overheads are mainly due to the

The results for the unfolded structure are summarised in table 3. As could be seen from the table the maximum overhead is 6.85% in the case of 8-bit multiplier with a 128-tap filter, hence providing a net reduction in power of 84.84%.

Multiplier size	Ordering	Filter order		
		32	64	128
		Average overheads [%]		
8-bit	norm	2.63	3.07	4.59
	sort <sub>1</sub>	3.85	4.54	6.83
	sort <sub>2</sub>	3.92	4.64	6.85
16-bit	norm	0.52	0.54	0.61
	sort <sub>1</sub>	0.76	0.81	0.91
	sort <sub>2</sub>	0.76	0.81	0.91
24-bit	norm	0.21	0.22	0.25
	sort <sub>1</sub>	0.32	0.33	0.37
	sort <sub>2</sub>	0.31	0.32	0.37

**Table 3:** Average Overheads for Unfolded Structure.

#### 4. CONCLUSION

In this work, we have generalised a new scheme for the single multiplier implementation of low power digital filters, on CMOS-based DSPs, to linear phase FIR filters. New results are presented which provide up to 85% net power reduction and show that the scheme provides a generic power saving framework for FIR filters exploiting coefficient symmetry, where exists, without the need for specialised implementation structures.

#### 5. REFERENCES

- [1] A.T. Erdogan and T. Arslan, "Low power multiplication scheme for FIR filter implementation on single multiplier CMOS DSP processors", IEE Electronic Letters, vol. 32, no. 21, pp. 1959-60, 10th October 1996.
- [2] A.T. Erdogan, T. Arslan and D.H. Horrocks, "Low power multiplication schemes for single multiplier CMOS based FIR digital filter implementations", IEEE ISCAS'97, pp. 1940-43, Hong Kong, 9-12 June 1997.
- [3] S.K. Mitra and J.F. Kaiser, "Handbook for Digital Signal Processing", Wiley Interscience Publication, 1993, New York, USA.
- [4] T. Arslan, A.T. Erdogan and D.H. Horrocks, "Low power design for DSP: methodologies and techniques", Microelectronics Journal, vol. 27, no. 8, pp. 731-744, November 1996.
- [5] P. Duncan, K. Kindsfater, L. Liu, and R. Jain, "Strategies for design automation of high speed digital filters", Journal of VLSI Signal Processing, no. 9, pp. 105-119, 1995.
- [6] D.E. Thomas, and P.R. Morby, "The Verilog Hardware Description Language", Kluwer Academic Publishers, 1995.