

High Throughput and Low Power FIR Filtering IP Cores

C. H. Wang, A. T. Erdogan, T. Arslan

School of Engineering and Electronics
University of Edinburgh, Scotland

Abstract - This paper presents the implementation of high throughput and low power FIR filtering IP cores. Multiple datapaths are utilized for high throughput and low power is achieved through coefficient segmentation, block processing and combined segmentation and block processing algorithms. The paper presents the complete architectural implementation of these algorithms for high performance applications. The paper describes the design methodology, evaluation environment, and provides results which show up to 33% reduction in power consumption with less than 10% increase in area depending on the number of datapaths.

1. INTRODUCTION

Recently, more and more traditional applications and functionalities have been targeted to palm-sized devices, such as Pocket PCs and camera-enabled mobile phones with colorful screen. Consequently, not only is there a demand of provision of high data processing capability for multimedia and communication purposes, but also the requirement of power efficiency has been increased significantly.

FIR filters are widely used in most of the Digital Signal Processing applications and are characterized by the extensive sequence of multiplication operations. Intuitively, there are two approaches, sequential and parallel, to implement FIR filters. The sequential implementation can be cost and area-effective in hardware but its bottleneck is low throughput; hence it is not suitable for high performance applications. On the other hand, the parallel implementation can maximize the throughput with some additional hardware, such as multipliers and adders.

Furthermore, power dissipation is becoming a crucial factor in the realization of parallel mode FIR filters. There is increasing number of published techniques to reduce power consumption of FIR filters. The authors in [1] utilize differential coefficients method (DCM) which involves using various orders of differences between coefficients along with stored intermediate results rather than using the coefficients themselves directly for computing the partial products in the FIR equation. To minimize the overhead while retaining the benefit of DCM, differential coefficient and input method (DCIM) [2] and decorrelating (DECOR) [3] have been proposed. Another approach used in [4] is to optimise word-lengths of input/output data samples and coefficient values. This involves using a general search based methodology, which is based on statistical precision analysis and the incorporation of cost/performance/power measures into an objective function through word-length parameterisation. In [5], Mehendale et al. presents an algorithm for optimising the coefficients of an FIR filter, so as to reduce power consumption in its implementation on a programmable digital signal processor. The use of

coefficient segmentation, block processing and combined segmentation and block processing algorithms for low power FIR filter implementations has been shown in [6]. The authors in [7] and [8] have introduced high throughput FIR filter implementations.

This paper presents the implementation of high throughput and low power FIR filtering Intellectual Property (IP) cores. The IP cores are based on the low power algorithms presented in [6] for single datapath implementations. This paper shows their implementation for increased throughput as well as low power applications, through employing multiple datapaths. The paper studies the impact of parameterisation in terms of datapath parallelisation on the power/speed/area performance of these algorithms. The paper describes the design methodology, evaluation environment, and provides results which show up to 33% reduction in power consumption with less than 10% increase in area depending on the number of datapaths.

The paper is organized as follows. A brief introduction to a conventional multiple datapath FIR filtering IP core is provided in Section 2. Three different techniques for high throughput and low power FIR filtering IP cores are introduced in Section 3. Simulation results are discussed in Section 4, and finally the paper concludes with Section 5.

II. MULTIPLE DATAPATH ARCHITECTURE

A direct form (DF) realization of an N-tap FIR filter is shown in Figure 1. In this realization, there are delay units between multipliers. This means that the present and N-1 previous data samples of input $x(n)$ are multiplied by N-tap coefficients, and then are summed together to form the filter output $y(n)$. Furthermore, it implies that with a single datapath unit, N cycles will be needed in order to generate a filter output. The block diagram of a generic DF FIR filter implementation is illustrated in Figure 2. It consists of two memory blocks for storing coefficients (HROM) and input data samples (XRAM), two registers for holding the coefficient (HREG) and input data (XREG), an output register (OREG), and the controller along with the datapath unit. The XRAM is realised in the form of a latch-based circular buffer for reducing its power consumption. The controller is responsible for applying the appropriate coefficients and data samples to the datapath.

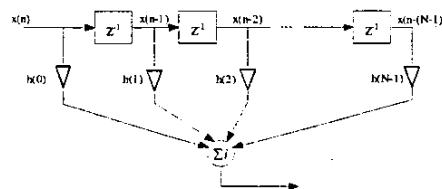


Figure 1. Direct form FIR filter architecture.

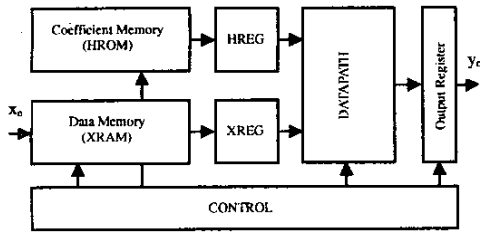


Figure 2. Block diagram of an FIR filter implementation.

In order to increase the throughput, the number of datapaths should be increased and data samples and coefficients should be allocated to these datapaths in each clock cycle. For example, for a 4-tap FIR filter with 2 datapaths, the coefficient data can be separated into 2 parts, (h_3, h_2) and (h_1, h_0) each allocated to a different datapath with corresponding set of input data samples, as shown in Figure 3. Therefore, an output will be obtained in $\lceil N/M \rceil$ clock cycles, where N is the number of taps and M is the number of datapaths. For example, for a 4-tap filter, an output can be obtained in 2 clock cycles with 2 datapaths.

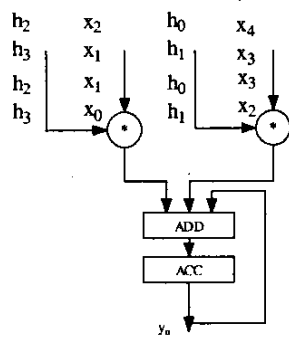


Figure 3. A 2 datapath architecture for a 4-tap filter

III. MULTIPLE DATAPATH LOW POWER ARCHITECTURES

A. Coefficient Segmentation Algorithm

Two's complement representation is most commonly used in DSP applications due to the ease of performing arithmetic operations. Nevertheless, sign extension is its major drawback and causes more switch activity when data toggles between positive and negative values. For this reason, the coefficient segmentation algorithm proposed in [6] segments a coefficient h_k into two parts; one part, m_k , for the multiplier and one part, s_k , for the shifter. Segmentation is performed such that m_k is the smallest positive value in order to minimize the switching activity at the multiplier input. On the other hand, s_k is a power of two number and could be both positive and negative depending on the original coefficient. The MSB bit of s_k acts as the sign bit and remainder are the measure of shift. For instance, if a coefficient is 11110001, the decomposed number and shift value will be 00000001 and 10100, respectively. An example of 2 datapath implementation architecture of this algorithm is shown in Figure 4.

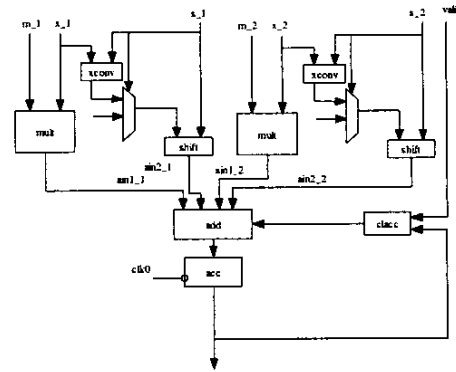


Figure 4. The segmentation algorithm with 2 datapaths.

B. Data Block-Processing

Due to the successive change of both coefficient and data samples at each clock cycle, there is a high switching activity within the multiplier unit of the datapath. This high switching activity can be reduced significantly, if the coefficient input of the multiplier is kept unchanged and multiplied with a block of data samples [6]. Once a block of data samples are processed, then a new coefficient is obtained and multiplied with a new block of data samples. However, this process requires a set of accumulator registers corresponding to the size of the data block size. The previous results have shown that a block size of 2 provides the best results in terms of power saving. An example datapath allocation for $N=6$ and $M=2$ and its corresponding architecture is shown in Figures 5 and 6 respectively.

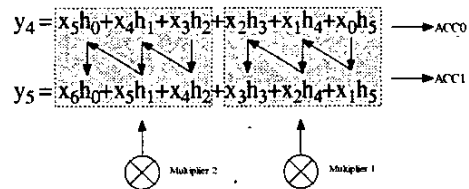


Figure 5. An example of a 6-tap filter with $M=2$.

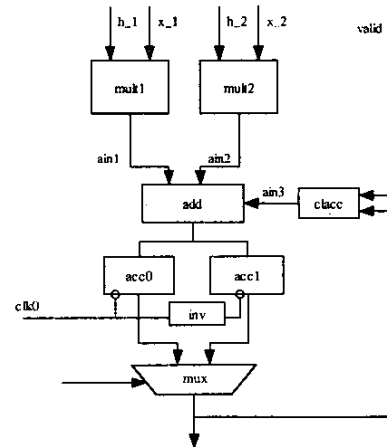


Figure 6. The block processing algorithm with 2 datapaths.

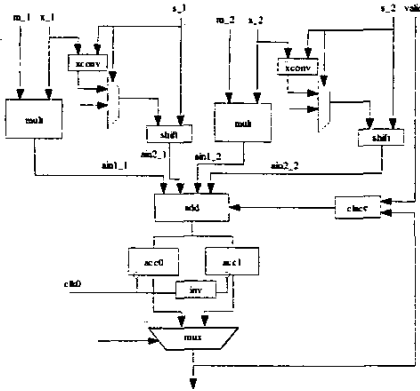


Figure 7. Combined Coefficient Segmentation and Block Processing algorithm with 2 datapaths.

C. Combination Coefficient Segmentation and Block Processing Algorithm

The architectures of coefficient segmentation and block processing algorithms can be merged together. This will reduce the switching activity at both coefficient and data inputs of the multiplier units within the datapaths with only slight overhead in area. An example of 2 datapath implementation of this algorithm is shown in Figure 7.

IV. SIMULATIONS AND RESULTS

We have analyzed throughput, area and power consumption of the different FIR IP cores for all algorithms, namely conventional (CON), coefficient segmentation (CSEG), block processing (BP), and combined CSEG and BP (COMB). In order to evaluate their performance for different number of datapaths, we have implemented these with 1, 2, 4 and 8 datapath cases. All IP cores were implemented for an example of 73-tap band-pass filter, using 16x16-bit Booth multipliers in their datapaths. The cores were designed using Verilog HDL and then synthesized using Synopsys Design Compiler, targeting a 0.18 μm standard cell CMOS library. Simulations were performed for 1000 randomly generated data samples using Verilog-XL simulator. This was followed by computing their power consumption with Synopsys Design Power. In all cases, a clock rate of 10MHz and 1.8V were assumed.

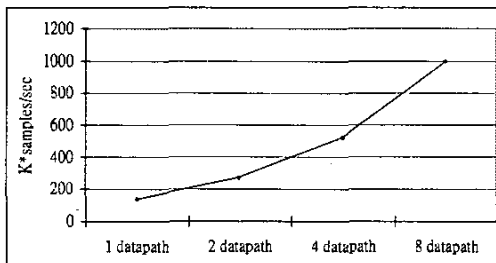


Figure 8. FIR filter throughput.

Table 1 Power and Area analysis

# of datapath	Algorithm	Power (mW)	saving (%)	Area (mm ²)	increase (%)
1	CON	1.759	-	0.161	-
	CSEG	1.585	10	0.166	3
	BP	1.207	31	0.165	2
	COMB	1.177	33	0.172	6
2	CON	3.099	-	0.234	-
	CSEG	2.943	5	0.246	5
	BP	2.227	28	0.244	4
	COMB	2.132	31	0.256	9
4	CON	6.127	-	0.384	-
	CSEG	5.963	3	0.409	7
	BP	4.273	30	0.394	3
	COMB	4.139	32	0.419	9
8	CON	10.52	-	0.688	-
	CSEG	10.69	-2	0.738	7
	BP	8.057	23	0.702	2
	COMB	7.955	24	0.754	10

The comparison of throughput for different number of datapaths is shown in Figure 8. Because the number of taps is not always a multiple of the number of datapaths, the throughput will be slightly lower than the expected throughput. For example, for a 73-tap filter, a filter output can be generated in 73, 37, 19, and 10 clock cycles using 1, 2, 4, and 8 datapaths respectively. Therefore, the throughput will be 137, 270, 526 and 1000 K samples/sec for 1, 2, 4, and 8 datapaths respectively.

The results for power consumption and area usage for all algorithms and different number of datapaths are given in Table 1. Clearly, in all cases the best power saving is achieved with COMB algorithm with a 6-10 % increase in area. The second best power saving is achieved with BP algorithm with only 2-4 % area increase. However, for 8 datapath case COMB and BP achieve similar power savings, 24% and 23% respectively. This can be explained in respect to CSEG's performance in power saving. As the number of datapaths increases the power saving of CSEG decreases significantly and eventually for 8 datapath case it results in some power increase. The main reason for this decrease in power efficiency is the increase in power consumption of the addition tree due to the increased number of shifters employed by this algorithm. For example, there are 17 inputs in the adder-tree in 8-datapath case for CSEG compared to only 9 inputs for CON. Although, the increased number of datapaths reduces the power of the multipliers, this reduction becomes less than the overheads due to the additional shifters and increased complexity of the adder-tree. Therefore, this degradation in power saving of CSEG directly contributes to COMB's power efficiency as well, since it also employs the same number of additional shifters. As a result, for small number of datapaths COMB achieves the best power savings, but as the number of datapaths increases BP offers better power and area tradeoff compared to COMB.

Figure 9 illustrates the power contribution of main blocks to the overall power consumption for each case. Clearly, as

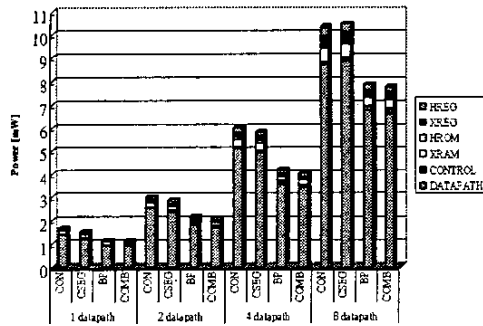


Figure 9. Overall power consumption

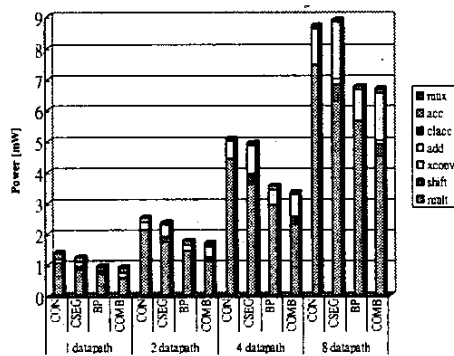


Figure 10. Datapath power consumption

expected most of the power (77.4%-78.2% for 1-datapath, 80.3%-81.6% for 2-datapath, 82.9%-83.6% for 4-datapath, and 82.7%-84.1% for 8-datapath cases) is consumed by the datapaths. The power contributions of the main blocks within the datapath units are shown in Figure 10. Clearly, in all cases the multiplier block is responsible for most power consumption (47.9%-60.6% for 1-datapath, 53.4%-69.3% for 2-datapath, 49.0%-72.1% for 4-datapath, and 56.7%-70.9% for 8-datapath) and for all cases CSEG reduces multiplier power by 16%-18.9%, BP by 24.3%-33.5% and COMB by 39.5%-54.1%. The Figure also clearly shows the overheads due to shifter and more complex adder power consumptions for CSEG and COMB algorithms.

The area usage analysis for all algorithms and different number of datapaths is given in Table 1 and Figure 11. Clearly, BP results in least area increase (2-4%), followed by CSEG (3-7%), and COMB (6-10%) compared to CON area. However, in overall the area increase is not linear with the number of datapaths and only increases by 1.5, 2.4, and 4.4 fold for 2, 4, and 8-datapath cases compared to 1-datapath case.

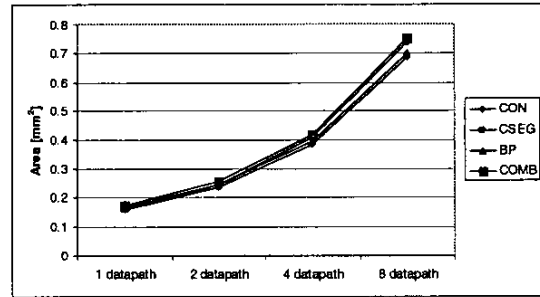


Figure 11. Area comparisons.

V. CONCLUSION

In this paper, we have presented the complete architectural implementations of a number of low power algorithms for high performance applications. The paper showed the impact of parameterisation in terms of datapath parallelisation on the power, area and speed metrics for FIR IP cores. Results have been provided which show up to 33% reduction in power consumption with less than 10% increase in area depending on the number of datapaths employed. It has been shown that for small number of datapaths COMB achieves the best power savings, but as the number of datapaths increases BP offers better power and area tradeoff compared to COMB.

REFERENCES

- [1]. N. Sankarayya, K. Roy, and D. Bhattacharya: "Algorithms for Low Power and High Speed FIR Filter Realisation Using Differential Coefficients", IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing, vol. 44, pp. 488-497, June, 1997.
- [2]. T-S Chang, Y-H Chu, and C-W Jen: "Low Power FIR Filter Realization with Differential Coefficients and Inputs", IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing, vol. 47, no. 2, pp. 137-145, Feb., 2000.
- [3]. S. Ramprasad, N.R. Shanbhag, and I.H. Hajj: "Decorrelating (DECOR) Transformations for Low Power Digital Filters", IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing, vol. 46, no. 6, pp. 776-788, June, 1999.
- [4]. H. Choi and W.P. Burleson: "Search-Based Wordlength Optimisation for VLSI/DSP Synthesis", VLSI Signal Processing, vol. 7, pp. 198-207, 1994.
- [5]. M. Mehendale, S. D. Sherlekar, G. Venkatesh: "Low Power Realization of FIR filters on Programmable DSP's", IEEE Trans. on VLSI Systems, Vol. 6, No. 4, pp. 546-553, Dec. 1998.
- [6]. A. T. Erdogan, M. Hasan, T. Arslan, "Algorithm Low Power FIR Cores." Circuits, Devices and Systems, IEE Proceedings, 2003, Page(s): 155-160
- [7]. A.T. Erdogan, T. Arslan: "Low power implementation of high throughput FIR filters", IEEE Int. Conf. on Circuits and Systems, pp. 373-376, May 2002.
- [8]. D.A. Parker, K.K. Parhi: "Area-efficient parallel FIR digital filter implementations", Application Specific Systems, Architectures and Processors, pp.93-111, Aug. 1996.