Optimal Combination of Number of Taps and Coefficient Bit-Width for Low Power FIR Filter Realization

João Portela[*] Eduardo Costa[†] José Monteiro[‡]

Abstract — **This paper addresses the optimization of FIR filters for low power. We propose a search algorithm to find the combination of the number of taps and coefficient bit-width that leads to the minimum number of total partial sums, and hence to the least power consumption. We show that the minimum number of taps does not necessarily lead to the least power consumption in fully parallel FIR filter architectures. This is particularly true if the reduction of the bit-width of the coefficients is taken into account. We show that power is directly related to the total number of partial sums in the FIR filter, which in turn is determined by the number of bits set to 1 in the coefficients. We have developed a search algorithm that achieves up to 36% less power consumption when compared to an implementation using the minimum number of taps.**

1 INTRODUCTION

Power consumption in VLSI digital signal processing systems (DSP) has gained special attention mainly due to the proliferation of high-performance portable battery-powered electronic devices, such as cellular phones, laptop computers, etc. In DSP applications, one of the most basic operations are Finite Impulse Response (FIR) filter computations. In this paper, we propose a new optimization technique to reduce power consumption in parallel FIR filter implementations.

Power consumption is directly related to the amount of computation. At first, this may indicate that for a low power implementation, the least number of taps that allows for the desired filter precision should be used. One of the contributions of this work is to show that this is not necessarily so. In an implementation that uses shift-adders for the multiplications, the total number of these elements will define the power consumption of the filter. In turn, the number of shift-adders is determined by the number of bits set to 1 in the coefficients of the filter. Hence, it may happen that a larger number of taps leads to a reduction in the total number of bits set to 1 in the coefficients.

To compound this problem, an additional variable that has great impact in the power consumption of the filter is the bit-width of the coefficients.

[*]IST/INESC-ID, Lisbon, Portugal, jpop@algos.inesc.pt
[†]UCPel, Pelotas, Brazil, ecosta@ucpel.tche.br
[‡]IST/INESC-ID, Lisbon, Portugal, jcm@algos.inesc.pt

We propose in this work an algorithm to explore the best combination of coefficient bit-width and number of taps. This search can not be performed efficiently while estimating power for each combination. A second major contribution of this work is to demonstrate empirically that we need only compute the total number of 1s in the coefficients of each combination and use this value has the cost function for our search. Hence, given that the range for the number of taps and coefficient bit-width is limited, it is possible to use efficiently a simple exhaustive search method. We show that this search takes negligible time, even for large examples, and that implementations that consume close to 36% less power when compared to the least number of taps can be obtained.

2 FIR FILTER DESIGN

In this section we discuss the main aspects related to the implementation of a low-pass band FIR filter, namely its frequency response, architecture and coefficient calculation.

2.1 Frequency Response

FIR filtering is achieved by convolving the input data samples with the desired unit impulse response of the filter. The output $Y[t]$ of a M-tap FIR filter is given by the weighted sum of latest M+1 input data samples $X[t]$, as shown in Equation 1.

$$Y[t] = \sum_{i=0}^{M} c_i X[t+M-i] \tag{1}$$

The characteristics of digital filters are often specified in the frequency domain. For frequency selective filters, such as low-pass and band-pass filters, the specifications are often in the form of tolerance schemes. A typical specification of a low-pass filter is depicted in Figure 1. In the Figure 1, the dashed horizontal lines indicate the tolerance limits. In the pass-band ($< W_p$) and the stop-band ($> W_s$), the magnitude response has a limit deviation of s. The width of the transition band determines how sharp the filter is. The magnitude response decreases monotonically from the pass-band to the

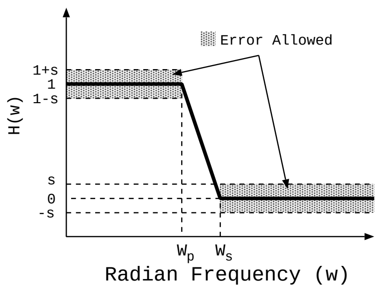

Figure 1: Specification of a low-pass FIR filter.

stop-band in this region.

2.2 Architecture

In the direct form FIR filter implementation, each clock cycle a new data sample and the corresponding filter coefficient are simultaneously applied to each multiplier. The result of all multipliers is added simultaneously, producing a significant amount of glitching [1].

In our work, we address this problem by implementing an alternative fully-parallel architecture, called the transposed form, depicted in Figure 2 [2]. This architecture presents the same complexity as the direct form, but it involves multiplying all the coefficients by the same input data. Because the registers are now between the adders, most of the glitching is filtered, resulting in a significant power reduction compared to the direct form architecture. As can be observed in Figure 2, the transposed

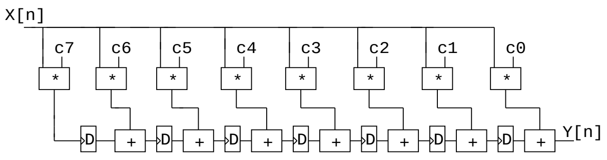

Figure 2: Transposed form FIR filter architecture.

implementation presents a large number of multiplier modules (as many as the direct form, and equal to the number of taps), which contribute to large area, delay and power consumption in the architecture. In order to optimize these parameters, shift-add circuits are used as multipliers. With this approach, for each bit 1 in the coefficient an adder circuit must be used in order to provide the partial product terms. Therefore, the higher the number of bits 1 used in the coefficients, the higher the complexity presented by the shift-add operators.

2.3 Coefficient Calculation

The window method is the simplest method of FIR filter design [3]. The choice of window is governed

# Taps	36	37	38	39	40	41
Power	0.91	0.78	0.89	0.86	0.93	0.82

Table 1: Filter power consumption in mW as a function of the number of taps.

by the desire to have the window response as short as possible, so as to minimize computation in the implementation of the filter. In this work, we use the Kaiser window approach [4].

The Kaiser window is defined according to Equation 2 [3], where $\alpha = \frac{M}{2}$ and I_o represent the zeroth-order modified Bessel function of the first kind.

$$c_i = \begin{cases} \frac{I_0[\beta(1-[(i-\alpha)/\alpha]^2)^{(1/2)}]}{I_0(\beta)}, & 0 \le i \le M, \\ 0, & otherwise. \end{cases} \quad (2)$$

3 OPTIMIZATION ALGORITHM

3.1 Number of Taps vs. Power

The simplification and tailoring of the digital systems plays an important role in architectural exploration for low power. This rule of thumb may erroneous lead a designer to search for a fully parallel FIR filter implementation that uses the least number of taps that allow the desired filter accuracy. This is not necessarily true, as exemplified in Table 1. For this filter example, designed using the Kaiser window method [4] and implemented with the fully-parallel architecture of Section 2.2, the power consumption with 36 taps is significantly larger than with fewer taps. For this filter, the best solution in terms of power would be to use 37 taps.

This behavior can be readily explained when we consider the use of shift-adders in place of the multipliers. As we vary the number of taps, different values for the coefficients are obtained. The number of bits set to 1 in each of these coefficients determines the total number of shift-adders used in the parallel implementation of the filter. In turn, the number of shift-adders will have a large contribution to the total power dissipation. Thus, although a larger number of coefficients has higher probability of having a larger overall number of bits set to 1, in many cases this is not the case.

For a low power implementation, it may be worthwhile to investigate the power consumption of a range of taps. As in the example of Table 1, significant optimizations can be found.

3.2 Coefficient Bit-Width vs. Power

To make this problem more interesting and give us a larger exploration space for this optimization problem, we take into account the number of bits

Bit Width	Number of Taps					
	36	37	38	39	40	41
10			0.61	0.57	0.64	0.58
11		0.60	0.76	0.61	0.77	0.62
12		0.71	0.85	0.70	0.85	0.74
13	0.91	0.78	0.89	0.86	0.93	0.82

Table 2: Filter power consumption in mW as a function of the number of taps and the coefficient bit-width.

Bit Width	Number of Taps					
	36	37	38	39	40	41
10			79	74	81	73
11		79	105	81	105	80
12		99	112	95	119	99
13	132	107	128	119	132	111

Table 3: Total number of bits set to 1 in the coefficients.

(or bit-width) used for the coefficients. The behavior of power consumption with the coefficient bit-width is monotonic, the smaller the bit-width the less power consumption. Naturally, as the coefficient bit-width is reduced, the precision of the filter's frequency response also reduces.

We present in Table 2 the power estimates for the filter example discussed above, this time varying both the number of taps (horizontally) and the coefficient bit-width (vertically). The empty slots correspond to combinations that lead to a solution that violates the specified error, and therefore are not considered. To obtain the coefficients with different bit-widths, we are simply using the Kaiser window method as before to obtain the coefficients, which we then truncate and use the most significant bits (these are values between -1 and 1 in fixed-point representation).

We can observe from the table that for a given number of taps power decreases with the bit-width. For this example, the best solution in terms of power would be to use 39 taps and coefficients with 10 bits.

3.3 Power per Bit Set to 1

From the discussion above we conclude that the optimization procedure should test each combination of number of taps and coefficient bit-width for validity and power consumption, and select the valid solution with lowest power. The problem is that performing power estimation on each of these combinations is extremely inefficient.

We have computed for a set of FIR filter examples with different configurations the relation between power dissipation and the total number of bits set to 1 in the coefficients, *i.e.*, the power per bit set to 1. The first graph of Figure 3 shows these values for the filter example being presented. We have observed that for a given filter this relation is practically constant. This indicates that for our optimization procedure, a good cost function is the total number of bits set to 1 in the coefficients, a value easy to obtain.

We present in Table 3 the total number of bits set to 1 for the filter under consideration. We can observe that the lower numbers correspond in fact to the filter solutions with lower power consumption. Note that this relation is not completely monotonic and small errors may occur. Using just the total number of bits set to 1, the solution we obtain uses 41 taps and 10 bits for the coefficients (73 bits set to 1). However, we had observed that the best solution in terms of power would be to use 39 taps and 10 bits, which presents 74 bits set to 1. This is a negligible error as these two solutions present about the same power consumption, 0.58mW vs. 0.57mW, especially when compared to the original value of 0.91mW.

3.4 Optimization Algorithm

Using the total number of bits set to 1 as the cost function to minimize, an efficient search can be performed for the combination of number of taps and coefficient bit-width corresponding to the least power consumption implementation of the FIR filter.

For each number of taps M, the coefficients are calculated using the Kaiser window method [4], according to the user-defined filter parameters and allowed error. We then enter a loop that starts with the minimum bit-width for the coefficients ($\mathrm{BW}_{\mathrm{min}}$) and successively increase it, truncating the least significant bits. Using the truncated coefficients $trunc\_coefs$, the resulting transfer function is tested for validity, by verifying that it does not violate the specified error at any point. If a valid transfer function is found, then we exit this loop since we know that any other solution with larger bit-width will necessarily increase (or maintain) the total number of bits set to 1. In our experiments we have used $\mathrm{BW}_{\mathrm{min}}$=8 since we have not found a case where coefficients with this bit-width would yield a valid transfer function. $\mathrm{BW}_{\mathrm{max}}$ corresponds simply to the untruncated coefficients.

When exiting this inner loop, if the solution is invalid for all bit-width then we simply increase the number of coefficients. Otherwise, the total number of bits set to 1 is computed and if it is the lowest found so far, the current solution is saved. Note that if two solutions with the same best cost function are found, the one with least number of

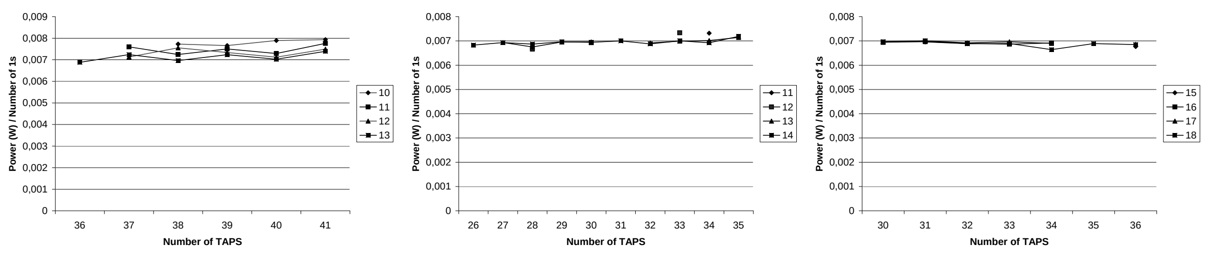

Figure 3: Relationship between power dissipation and the number of bits set to 1 in the coefficients for filters A, B and C, respectively.

	W_p	W_s	s	P Orig	P Opt	%
A	1.8	2.1	0.025	0.91	0.58	36.3
B	1.1	1.75	0.005	0.62	0.59	4.8
C	0.5	1.5	0.0005	0.94	0.79	16.0
D	1.0	2.6	0.001	0.46	0.36	22.0
E	2.2	2.6	0.01	0.96	0.76	21.0
F	0.5	0.6	0.05	1.84	1.32	28.0

Table 4: Parameters and power (in mW) savings for the FIR filters used as benchmarks.

taps is kept as this will typically lead to a best solution in terms of power.

One open issue is the range to use for the number of taps, M_{min} and M_{max}. The value for M determined according to the Kaiser window method [4] gives a good hint on M_{min}. However, since valid solutions can be found with less number of coefficients, we start with M_{min} much lower than that. Given that our cost function is trivial to compute, we can evaluate very efficiently hundreds of solutions. Therefore, we are able to test large values for M_{max}.

4 RESULTS

In this section, we present results on a set of FIR filters with different parameters and allowed error. The characteristics of these filters, pass-band (W_p), stop-band (W_s) and allowed error (s), are given in the first columns of Table 4.

The last three columns of Table 4 presents the results, comparing the power consumption of the filter selected from the search algorithm proposed in this paper (Section 3.4) with the filter that uses the least possible number of taps as given by the Kaiser window method [4]. The power results presented were obtained using the switch-level simulator SLS [5].

We should stress that for the largest filter, filter F, each combination that is tested in the inner loop of the algorithm described in Section 3.4 takes 12ms to run on a AMD-Duron processor running at 900MHz, thus justifying our claim that this search can be performed efficiently.

5 CONCLUSION

In this paper we have proposed a search algorithm to find the combination of number of taps and coefficient bit-width that leads to the least power consumption in a parallel implementation of FIR filters. We have showed that this search can be performed efficiently using as cost function the number of bits set to 1 in the coefficients. The results show that significant power savings are possible when compared to a solution that uses the least number of taps.

6 Acknowledgments

This research was supported in part by CAPES-Brazil and in part by FCT-Portugal under project POCTI.

References

[1] A. Erdogan and T. Arslan. High Throughput FIR Filter Design for Low Power SOC Applications. In 13^{th} *Annual IEEE International ASIC/SOC Conference*, pages 21–24, 2000.

[2] A. Nannarelli, M. Re, and G. Cardarilli. Tradeoffs between residue number system and traditional FIR Filters. In *IEEE International Symposium on Circuits and Systems*, May 2001.

[3] A. Oppenheim and R. Shafer. *Discrete-Time Signal Processing*. Prentice Hall Signal Processing Series, 1989.

[4] J. Kaiser. Nonrecursive Digital Filter Design Using the I_O-Sinh Window Function. In *IEEE International Symposium on Circuits and Systems*, pages 20–23, 1974.

[5] A. Genderen. SLS: An Efficient Switch-Level Timing Simulator Using Min-Max Voltage Waveforms. In *Proceedings of the International Conference on Very Large Scale Integration*, pages 79–88, 1989.