Power Macro-Models For DSP Blocks With Application to High-Level Synthesis†

Subodh Gupta and Farid N. Najm

ECE Dept. and Coordinated Science Lab.
University of Illinois at Urbana-Champaign

Abstract – In this paper, we propose a modeling approach for the average power consumption of macro-blocks that are typically used in *digital signal processing* (DSP) systems, such as adders, multipliers and delay elements, in terms of their input/output signal switching statistics. The resulting *power macro-model*, consisting of a quadratic or cubic equation in four variables, can be used to estimate the average power consumed in the macro-block for any given input/output signal statistics. This enables high-level power estimation and allows one to compare the power performance of different competing DSP systems during high-level synthesis. This approach has been implemented and models have been built and tested for many macro-blocks.

1. Introduction

Power consumption of very large scale integrated (VLSI) circuits has become a critical design concern due to high-density and high-speed micro-electronic devices. Due to limited battery life, reliability issues, and packaging/cooling costs, power reduction has become a crucial design concern, which motivates the development of low-power synthesis tools.

A number of low-power high-level synthesis techniques have appeared in the literature (see [1] for a survey), which target DSP circuits. These techniques require power cost functions to be evaluated for the different alternative architectures. Thus, it is typically necessary to estimate the power for a given hardware data flow graph (DFG) or for the register transfer level (RTL) description of the fixed point implementation of an algorithm. Each node in a DFG correspond to a macro-block. The total power dissipated by a hardware DFG is the sum of the power dissipated by each of the macro-block. Hence, there is a need for the power macro-models for these macro-blocks.

Recently, a number of techniques for high-level power modeling/estimation of DSP systems has been proposed. The method in [2], treats all the circuit input bits as digital "white noise" and due to this assumption can give errors of up to 80% in comparison to gate-level tools. In [3], the authors presented activity sensitive capacitance models for various library components. In [4], the authors presented macro-models for estimating the power of DSP macro-blocks in terms of word-level statistics. However, these and other techniques either assume some input distribution or require a priori selection of a specific analytical equation form.

†This research was supported in part by the National Science Foundation (NSF MIP 97-10235) and by the Semiconductor Research Corporation (SRC 97-DJ-484, customization funds from Texas Instruments Inc.).

Since filtering is a common operation in DSP systems, we will focus on finite impulse response (FIR) digital filters. We will present power macro-models for adders, both *general* (both word values are changing) and *fixed-coefficient* (only one word value is changing), *fixed-coefficient* multipliers, and delay elements (latches). Our models are equation-based. Specifically, we construct a quadratic or cubic equation on four variables, which are functions of primary inputs and outputs statistics. Moreover, we do not assume any specific probability distribution for the input signals. We use an automatic characterization procedure based on the method of Recursive Least Squares (RLS) to build the macro-models.

This paper is organized as follows. In section 2, we will present power macro-modeling for the macro-blocks. In section 3 we will give an overview of macromodel construction. In section 4, we give empirical results that show the effectiveness of this model. In section 5 we present an application of the power macro-models to high-level synthesis, and we summarize and conclude in section 6.

2. Power Macro-modeling

We have previously presented a general 4-dimensional (4D) table-based macro-model in [5] for combinational circuits. The 4D macro-model considers the average power of a circuit to be a function of four variables:

$$P_{avg} = f(P_{in}, D_{in}, SC_{in}, D_{out}) \tag{1}$$

These four variables are defined as:

$$P_{in} = \frac{1}{n}\sum_{i=1}^{n} p_i \qquad D_{in} = \frac{1}{n}\sum_{i=1}^{n} d_i \tag{2}$$

$$SC_{in} = \frac{2}{n\,(n-1)}\sum_{i=1}^{n}\sum_{j=i+1}^{n} SC_{ij} \qquad D_{out} = \frac{1}{m}\sum_{i=1}^{m} d_i \tag{3}$$

where n and m are number of primary inputs and outputs respectively, p_i and d_i are individual signal probability and zero-delay transition activity respectively, at the respective input or output nodes. SC_{ij} is the probability of two signals x_i and x_j being high simultaneously. It was also observed [5] that P_{in}, D_{in} and SC_{in} should satisfy the following constraints:

$$\frac{D_{in}}{2} \le P_{in} \le 1 - \frac{D_{in}}{2} \quad \text{and} \quad \frac{nP_{in}^2 - P_{in}}{(n-1)} \le SC_{in} \le P_{in} \tag{4}$$

While the above macro-model gives very good results in the general case, it does not give good accuracy for the case when one of the two numbers applied to the input of a multiplier or an adder is a constant. This, however, is a very common case in DSP systems, where adders and multipliers used to implement digital filters have one constant input; we will refer to these as being *fixed-coefficient* multipliers and adders. Table 1 shows the average error in estimating the total power of *fixed-coefficient* multipliers and adders using the macro-model (1). For finding the average error, 1000 blocks of input vectors having different values of P_{in}, D_{in}, and SC_{in} satisfying (4) were generated. One of the two binary words (i.e half of the primary inputs) was kept fixed to a randomly chosen value for every block of input vectors.

Total power was estimated using [7] which also provides estimate of D_{out}. Total power was also estimated using (1) and the relative absolute value of the error was calculated between the simulated and predicted power and the average of this error was calculated for all blocks of input vectors. The macro-blocks that we tested are signed multipliers (Baugh-Wooley, bit sizes of 8×8, 10×10, and 16×16), unsigned multipliers (Array, bit sizes of 8×8, 12×12, and 16×16), and ripple carry adder (RCA, bit sizes of 8, 16, and 20). It is clear from the table that macro-model (1) needs to be improved for this case of *fixed-coefficient* multipliers and adders.

Table 1. Average error when total power of *fixed-coefficient* multipliers and adders is estimated

Circuit	Avg. Error	Circuit	Avg. Error	Circuit	Avg. Error
Mult8(BW)	39.2%	Mult8(Array)	43.3%	Adder8(RCA)	27.3%
Mult10(BW)	65.2%	Mult12(Array)	69.8%	Adder16(RCA)	35.3%
Mult16(BW)	59.1%	Mult16(Array)	55.2%	Adder20(RCA)	30.7%

Let us assume A and B are the two input words of a multiplier or adder and A is fixed coefficient (or constant). We start by expressing (1) in-terms of separate statistics at A and B, as follows:

$$P_{avg}^{FC} = f(P_{in}^{A}, D_{in}^{A}, SC_{in}^{A}, P_{in}^{B}, D_{in}^{B}, SC_{in}^{B}, D_{out}) \tag{5}$$

where superscript FC denotes *fixed-coefficient*.

Let us consider that we have to generate a block of N consecutive input vectors and let us denote the kth vector by V_k. SC_{in} and P_{in} can be written in terms of the input vectors as [5]:

$$\begin{aligned} SC_{in} &= \lim_{N\to\infty} \frac{2}{Nn\,(n-1)} \sum_{k=1}^{N} \frac{n_1\,(k)\,(n_1\,(k)-1)}{2} \\ P_{in} &= \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \frac{n_1\,(k)}{n} \end{aligned} \tag{6}$$

where $n_1\,(k)$ is number of 1s in V_k. For the constant word A, the number of 1s in each vector V_k is the same. Let us denote this by n_1, then SC_{in} becomes:

$$SC_{in}^{A} = \lim_{N\to\infty} \frac{2}{Nn\,(n-1)} \frac{Nn_1\,(n_1-1)}{2} = \frac{n_1\,(n_1-1)}{n\,(n-1)} \tag{7}$$

Similarly, for word A, P_{in}^{A} can be written as:

$$P_{in}^{A} = \lim_{N\to\infty} \frac{1}{N} \frac{Nn_1}{n} = \frac{n_1}{n} \Rightarrow n_1 = nP_{in}^{A} \tag{8}$$

Substituting (8) into (7) leads to:

$$SC_{in}^{A} = \frac{P_{in}^{A}\left(nP_{in}^{A}-1\right)}{(n-1)} \tag{9}$$

It is clear from (9) that SC_{in}^{A} is completely determined from the value of P_{in}^{A}. Therefore, we do not consider SC_{in}^{A} as one of the variable in our macro-model. Moreover, D_{in}^{A} is 0 in (5) as A is a constant word. Furthermore, we found experimentally that dropping P_{in}^{B} also, does not reduce the accuracy significantly. Therefore, by dropping these three variables, (5) reduces to:

$$P_{avg}^{FC} = f(P_{in}^{A}, D_{in}^{B}, SC_{in}^{B}, D_{out}) \tag{10}$$

For the delay element, the macro-model is constructed for one-bit, as there is no interaction between different bit slices, and the capacitance associated with each bit is approximately same for all the bits. Furthermore, we assume that all the internal capacitances are lumped at the output node, given by C_{out}. Since, this is the only capacitance which is going to switch, the simplified macro-model for one bit delay element becomes:

$$P_{avg}^{L} = 0.5 f V_{dd}^{2} C_{out} D_{out} \tag{11}$$

where superscript L represents delay (or latch) power. For a 0.5μ CMOS technology we choose $C_{out} = 100fF$ as being a typical output capacitance value. Normally, of course, this value would be imported from the library for different latch types. Note that the clock power is not considered, while making the power macro-model for one-bit delay element, as it is a constant number.

3. Macro-model Construction

We have demonstrated in [6] that it is possible to fit a quadratic or cubic polynomial to the function $f(\cdot)$ in (1). We have found that the same methodology works as well for *fixed coefficient* multipliers and adders. By following the procedure of [6], we found that a good choice of the function $f(\cdot)$ in (10), was cubic and quadratic for *fixed coefficient* multipliers and adders respectively. Furthermore, we also used automatic characterization procedure based on RLS [6], to construct these analytical macromodels. More details on macromodel construction and characterization, can be found in [6].

4. Model Accuracy Evaluation

In this section we show the results of our power macro-modeling approach for different adder and multiplier circuits. We have considered both sign (Baugh-Wooley, bit width of 8, 10, and 16) and unsigned (Array, bit width of 8, 12 and 16) *fixed-coefficient* multipliers. We also considered both *general* and *fixed-coefficient* ripple carry (bit width of 8, 16, and 20) adders.

For *general* adders, we start by randomly generating blocks of input vectors for various values of P_{in}, D_{in}, and SC_{in} that satisfy (4), using the approach described in [5]. Similarly, for *fixed-coefficient* multipliers and adders, we generated blocks of input vectors for various values of P_{in}^{A}, D_{in}^{B}, and SC_{in}^{B}. A total of 1000 such blocks were generated for every macro-block, for which power was estimated using Monte Carlo simulation [7], based on a gate-level simulation with a scalable-delay gate timing model. The Monte Carlo simulation also provides accurate estimation of D_{out}. The power values predicted by the analytical function were compared to those from the gate-level Monte Carlo simulation, and the average relative errors are shown in Table. 2. In table 2 also shown, is the order of the model finally used by RLS (where C and Q implies cubic and quadratic respectively), and the number of RLS iterations (#RLS) required to build the equation. Moreover, under the column marked "Circuit", the superscript FC signifies *fixed-coefficient*. It can be seen from the table that the average error for all the macro-blocks is less than 20%. Also, shown in Figs. 1 and 2 are the scatter plots for *fixed-coefficient* 10×10-bit Baugh-Wooley multiplier and 20-bit ripple carry adder respectively. The fit is very good and shows that it is indeed possible to do high-level power modeling across the whole range of input switching statistics.

5. Application to High-level Synthesis of DSP Systems

In this section, we present an application of power macro-models in the context of high-level synthesis of DSP systems. The application is to find the power savings obtained after applying an algorithm transformation technique (ATT) under which we have considered decorrelating transformation [8] (DECOR).

We constructed macro-models for *general* adder, *fixed-coefficient* multiplier and delay-element. The macro-model for *fixed-coefficient* adder was not used, as in the application that we considered, there were no *fixed-coefficient* adder. Therefore, in the remainder of the section whenever we say adder and multiplier we imply *general* adder and *fixed-coefficient* multiplier respectively. Also note that the FIR filters considered below are the FIR sections of an adaptive filter.

Table 2. Average error when total power is estimated

Circuit	Model	Avg. Error	#RLS	Circuit	Model	Avg. Error	#RLS
Mult8^{FC}(BW)	C	12.1%	145	Adder8^{FC}(RCA)	Q	5.8%	177
Mult10^{FC}(BW)	C	15.3%	201	Adder16^{FC}(RCA)	Q	4.6%	149
Mult16^{FC}(BW)	C	16.5%	151	Adder20^{FC}(RCA)	Q	4.3%	241
Mult8^{FC}(Array)	C	10.2%	259	Adder8(RCA)	Q	8.1%	180
Mult12^{FC}(Array)	C	10.5%	222	Adder16(RCA)	Q	7.0%	141
Mult16^{FC}(Array)	C	11.1%	215	Adder20(RCA)	Q	7.4%	194

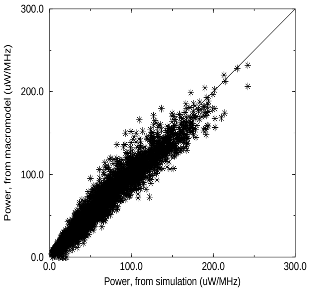

Figure 1. Scatter plot for 10×10-bit Baugh-Wooley *fixed-coefficient* multiplier.

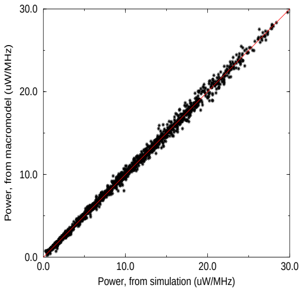

Figure 2. Scatter plot for 20-bit ripple carry *fixed-coefficient* adder.

5.1 ATT-DECOR

In DECOR [8], correlations between adjacent coefficients are exploited to decrease the coefficient precision, which leads to less complex multiplier with reduced power dissipation. Fig. 3 shows a direct-form [9] (DF) implementation of an N-tap FIR filter with coefficients $c_i, i = 0, \cdots, N-1$. The coefficients c_i are chosen to satisfy the desired frequency spectrum. For this study, we assume $N = 41$, and a low-pass frequency spectrum with cut-off frequency 0.2π. The floating point filter coefficients are designed via the *fir1.m* program in MATLAB, and then quantized to 10 bits to get their integer values. The precision of input signal $x(n)$ and output signal $y(n)$ was chosen to be 10 and 20 bits respectively. The sizes of the different hardware blocks in Fig. 3 were chosen as follows:

1. multipliers: 10×10-bit Baugh-Wooley multiplier
2. adders: 20-bit Ripple Carry Adder
3. latches: 10-bit static latches

The filter obtained after applying DECOR [8] to the direct-form filter in Fig. 3 is shown in Fig. 4. The filter in Fig. 4 has two extra multipliers, adders and latches, but the precision of the coefficient in the multiplier has been reduced to 8 bits, which leads to reduction in power dissipation.

We applied different input vectors, having different mean, variance, and correlation values, and estimated the power of the filters by using our macro-models and by using gate-level power estimation [7]. To use our macro-models and power estimator, input/output statistics were obtained for each of the adders, multipliers and latches, by propagating word values from input to the output. Fig. 5 shows the total power dissipation of the filters. It can be seen from the figure that power estimates from the macromodel and those from the gate-level simulation of the filters are very close. Moreover, power savings from macromodel and actual simulations are approximately the same, which shows the feasibility of our approach.

6. Conclusion

In this paper, we proposed a modeling approach for the average power consumption of DSP macro-blocks, such as adders, multipliers and delay elements, in the special case when one of the inputs is a constant. These macro-models can be used as part of high-level synthesis or design exploration to estimate the power consumed in the macro-block for any given input/output signal statistics. We presented experimental validation of the proposed models. The average error was shown to be less than 20% for all the macro-blocks. We also presented an application of these macro-models in the context of high-level synthesis, to obtain power-optimal DSP circuits.

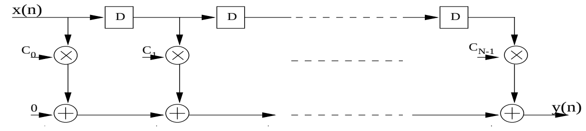

Figure 3. A N-tap DF digital filter.

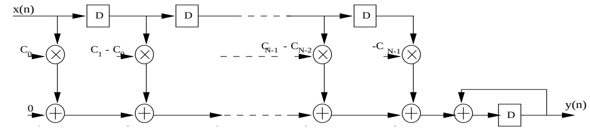

Figure 4. A N-tap DF digital filter after applying DECOR.

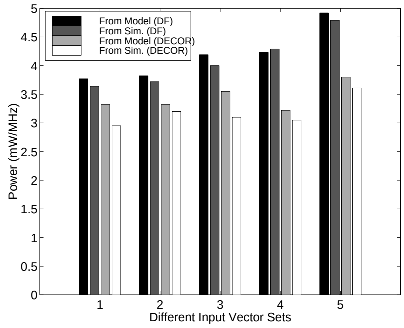

Figure 5. Power comparison between DF and DECOR.

References

[1] A. Chandrakasan, M. Potkonjak, R. Mehra, J. Rabaey, and R. W. Brodersen, "Minimizing Power Using Transformations," *IEEE Transactions on Computer-Aided Design*, vol. 14, no. 1, pp. 12-31, January 1995.

[2] S. R. Powell and P. M. Chau, " Estimating Power Dissipation of VLSI signal Processing Chips: The PFA technique," *VLSI Signal Processing IV*, pp. 250-259, 1990.

[3] P. E. Landman and J. M. Rabaey, "Architectural Power Analysis: The Dual Bit Type Method," *IEEE Transactions on VLSI*, vol. 3 pp. 173-187 June 1995.

[4] S. Bobba, I. N. Hajj, and N. R. Shanbhag, "Analytical Expressions for Power Dissipation of Macro-blocks in DSP Architectures," *VLSI Design Conference*, January 1999.

[5] S. Gupta and F. N. Najm, "Power Macromodeling for High Level Power Estimation," accepted for publication in *IEEE Transactions on VLSI*, 1999.

[6] S. Gupta and F. N. Najm, "Analytical Model for High Level Power Modeling of Combinational and Sequential Circuits," *IEEE Alessandro Volta Memorial International Workshop on Low Power Design*, March 4-5, 1999.

[7] M. Xakellis and F. N. Najm, "Statistical Estimation of the Switching Activity in Digital Circuits," *31st ACM/IEEE Design Automation Conference*, pp. 728-733, June 1994.

[8] S. Ramprasad, N. R. Shanbhag, and I. N. Hajj, "Decorrelating (DECOR) Transformations for Low-Power Adaptive Filters," *Proc. International Symposium on Low Power Electronics and Design*, pp. 250–255, 1998.

[9] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, 2nd edition. Englewood Cliffs, NJ: Prentice-Hall