VLSI DESIGN
2001, Vol. 12, No. 2, pp. 233–243
Reprints available directly from the publisher
Photocopying permitted by license only

Signal Strength Based Switching Activity Modeling and Estimation for DSP Applications*

LIH-YIH CHIOU[a,†], KHURRAM MUHAMMAND[b,‡] and KAUSHIK ROY[a,¶]

[a]*School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47906, USA;*
[b]*Texas Instruments, Dallas, TX 75243, USA*

(Received 20 June 2000; In final form 3 August 2000)

We present an effective switching activity modeling and estimation technique for components under resource sharing. The model uses word-level signal statistics to generate a single parameter, called signal strength. By using the signal strength, we can construct power models for the both cases of sharing and non-sharing of computing resources. The model enables us to effectively estimate switching activity at higher level of design abstraction. We have conducted several experiments using both synthetic and real data to evaluate our method. We have compared competing architectures for their relative power consumption for different components. The results show that average difference between the proposed method and very accurate power simulation (as opposed to switching estimation) using PowerMill is up to 12%.

Keywords: High-level power modeling; High-level power estimation; Switching activity estimation; DSP synthesis; Power estimation; VLSI

I. INTRODUCTION

Power efficient applications such as portable computing and wireless communication devices have driven the VLSI industry to take power consumption as one of the major implementation constraints. High-level power estimation at architectural or register-transfer level enables designers to evaluate competing architectures in the early phase of the design process. The designers can explore design space with larger flexibility and perform better trade-offs at higher levels. Hence, costly redesign steps can be avoided.

Several bottom-up approaches have been proposed to address this issue [1–5]. The power factor approximation technique in [1] uses a constant

* This research was funded in part by DARPA and by Purdue Research Foundation.
† Corresponding author. e-mail: lihyih@ecn.purdue.edu
‡ e-mail: k_muhammadl@ti.com
¶ e-mail: kaushik@ecn.purdue.edu

type model to determine weighting factor to model the average power consumed by a given module. The model assumes switching activity is constant regardless of the difference in inputs. In contrast, activity-sensitive power estimation techniques account for the variation of power dissipation due to different signal statistics. A bitwise data model is used in [2], while others use word-level signal statistics to construct macromodels. In [5], a method to characterize switching activity based on average signal power is proposed. The systematic way of characterizing switching activity provides an effective basis to hierarchically estimate power consumption and to evaluate different designs efficiently. One drawback of the method is that it is not applicable to designs when sharing of computing resources is required. The purpose of resource sharing is to reuse computing components and to reduce the number of resources needed in a design such that the area is minimized. This is common practice in DSP processor based implementation. Though several studies mention the possible increment in switching energy when resources are shared [6, 7], high-level power modeling has not really indicated how sharing resources impact power consumption.

In the current day CMOS technology, power consumption is dominated by dynamic components [8]. Hence, we will use switching activity as the metric for comparing power consumption. We count the number of switches at each output of basic cell in a functional module and sum them up to obtain the switching activity. For example, the basic cells for an array multipliers are AND gates, half adders and fuller adders. Relative weighting factors are applied to different kinds of internal cells to reflect output load capacitance driven by the gate. We also count the switches at the input pins and multiply them by the bit width, N, to account for input buffers/drivers. Delay-free model is assumed in our current work. As we ignore the internal node switching activity, this method cannot be used for accurate power estimation. However, it is our objective to provide high-level models such that we can compare competing architectures with fast and reasonable accuracy for low switching activity (and hence, low power).

We focus our work on data-dominated applications, such as DSP. By constructing analytical or table-lookup models for basic building blocks, such as arithmetic modules, multiplexers and registers, power (switching activity) estimation in a variety of DSP architectures can be obtained in a hierarchical way.

We will present a power macromodeling approach to estimate switching activity of designs. The approach takes into account the effect of sharing resources and does not treat the building blocks as independent black boxes.

The rest of the paper is organized as follows. Section II provides background for our work. Section III describes our techniques employed to characterize modules in resource sharing designs. Experimental results are reported in Section IV. Finally, Section V draws the conclusions.

II. PRELIMINARIES

The power model we presented uses word-level statistics, while the switching activity occurs at bit level. A bit-level model based on word-level statistics is first introduced. Following that we will define a signal parameter called signal strength, η, and state procedures to characterize switching activity. Throughout this paper the sign-magnitude number representation is assumed.

II.1. Word-level Statistics and Sign-magnitude Bit-level Model

A bit-level switching activity model based on word-level signal statistics is proposed in [2]. In the bit-level model, the effects of input word-level statistics on the least significant bits (LSBs) and the most significant bits (MSB) are divided into three regions. The first region (MSBs) has low switching activity. The second region (LSBs) has high switching activity. The last region in between is considered to be a linear transition connecting

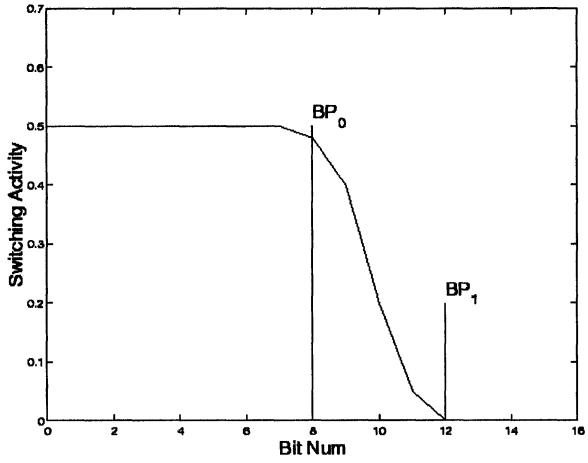

FIGURE 1 Switching activity of bits of a zero-mean white Gaussian process expressed in 16-bit SM form. Sign bit is not shown.

the other two. This is shown in Figure 1 for sign-magnitude (SM) representation. There are two break points, BP_0 and BP_1, that separate the three regions. Switching activity at the MSBs, the region above the BP_1, is considered to be zero. Switching activity in the lower-bit region, *i.e.*, below BP_0, is 0.5.

II.2. Signal-strength Based Power Model

The signal-strength based power model uses a parameter called signal strength to characterize power. Let the $X[n]$ be the input sequence to a DSP system. Usually $X[n]$ can be assumed to be a stationary Gaussian process. The signal strength, η, is defined as the number of bits needed to represent the average signal power. η is given as

$$\eta = \log_2\left((2^{N-1}-1)\times\frac{\sqrt{E(X^2[n])}}{d}+1\right) \quad (1)$$

where X[n] represents an input sequence, $E(X^2[n])$ is the average signal power of X[n] and N is number of bits used to represent the signal value. $\sqrt{E(X^2[n])}$ equals the standard deviation of zero-mean signals. All signals are assumed to be uniformly quantized in a dynamic range of $\pm d$ and are represented in sign magnitude form using N bits. With the given statistics of an input sequence, we can compute η of the sequence by using Eq. (1). The breakpoints BP_0 and BP_1 that define three regions of a multi-bit signal can be calculated from the word-level signal statistics (η and temporal correlation ρ) using Eq. (2) [5].

$$\begin{aligned} BP_0 &= \eta - 2.1(1-\rho)^{-0.1293} + 1.1 \\ BP_1 &= \eta - 2.1(1-\rho)^{-0.1293} + 4.1 \end{aligned} \quad (2)$$

For uncorrelated signals, $BP_0 \approx \eta - 1$ and $BP_1 \approx \eta + 2$.

In [5], switching activities of building blocks are formulated based on empirical equations of the statistics of the signals applied at their input. An example that uses η to characterize components is illustrated as follows. Consider an adder whose two primary input ports are A and B. Let L be the length of primary input vectors. The sequences to two inputs are $\langle a^{(1)}, a^{(2)}, \ldots, a^{(L)}\rangle$ and $\langle b^{(1)}, b^{(2)}, \ldots, b^{(L)}\rangle$, respectively. The η_A (or η_B) of the sequence, $\langle a^{(i)}\rangle$ (or $\langle b^{(i)}\rangle$), to input A (or B) is obtained by gathering signal statistics and applying Eq. (1). After η's of input signals are obtained, the switching activity of a functional block can be formulated as a function of η's of input sequences by extensive simulation and proper approximation. One can perform linear approximation by fitting simulated results to the first order of the polynomial. As an example in [5], the switching activity of an adder can be expressed as $SW = 1.59 + 0.49\eta_A + 0.49\eta_B$.

This kind of modeling method works well for switching activity estimation of fully parallel implementations, since no operations (multiplications) share the same components (multipliers). In real implementations, sharing of computing resources must be considered for some resource-constraint applications.

II.3. Definitions for Array Multipliers

Before we consider generalized models, we need to define some terms for array multipliers which are

used in our experiments. Two attributes are used to classify the array multiplier [9]. One is the order of the operand's bits: most significant bit (MSB) first or least significant bit (LSB) first. The other attribute is what kind of adders are used to perform summation of partial products: a ripple (RP) carry type or a carry-save (CS) type. An MSB-first CS multiplier will be referred to as most significant bit first array multiplier with carry-save structure adders.

III. POWER MODELING FOR RESOURCE SHARING

III.1. Observations

The idea of sharing resources implies that a shared functional unit has to multiplex its input from different sources. Hence the assumption about stationary signals for original signal strength based switching modeling is not valid any more. Therefore, it is necessary to have a model that can be used when sharing resources are needed. Consider a module that is shared by two operations. For the input A of a module, the new sequence is formed by alternating between two source sequences, $\langle a_1^{(i)} \rangle$ and $\langle a_2^{(i)} \rangle$ and is equal to $\langle a_1^{(1)}, a_2^{(1)}, a_1^{(2)}, a_2^{(2)}, \ldots, a_1^{(M)}, a_2^{(M)} \rangle = \textit{interleave}(\langle a_1^{(i)} \rangle, \langle a_2^{(i)} \rangle)$. *Interleave*( ) is a function that mixes two alternating sequences as shown above. Parameters that can relate switching activity differences of $\langle a_1^{(i)} \rangle$ and $\langle a_2^{(i)} \rangle$ must be added for more accurate modeling.

Difference in switching activity of a resource, which will be referred to as Δsw, is defined as the difference in switching without sharing and with sharing. The difference, Δsw, can be positive or negative. Experimentally, we have observed an increment in switching activity in most cases of resource sharing. We observed that the Δsw of a shared functional unit is affected by the difference of η's of input sequences, which are $\langle a_1^{(i)} \rangle$ and $\langle a_2^{(i)} \rangle$. Δ_η will be used as the notation for difference of η's for any two signals. Qualitatively speaking, the larger the Δ_η of two sequences, the larger is the Δsw observed. Δ_η can be used for estimating switching activity in case of resource sharing for the following reasons: First, η is related to average signal power, so the Δ_η of two signals will indicate Δsw. Second, it is shown in [5] that $\eta+2$ marks the beginning of the most significant bits (MSBs) region in word-level statistical model. The switching activity in the MSB region is very low and is generally assumed zero when modeling. However, the width of the MSB region will affect the switching activity. Therefore the difference in the width of two MSB regions can be an indication of Δsw due to resources sharing.

III.2. Generalized Signal-strength Based Switching Activity Model

By including $\Delta\eta$, the signal strength difference of sharing sequences, we can generalize the switching activity model. The generalized switching activity model of a functional unit consists of a base switching table and a set of difference switching tables. The base table is constructed under the condition of sharing no computing resources. The switching activity in the base table can be derived based on the original signal strength based power model. The difference tables are constructed under different sharing conditions.

The sharing condition is based on the difference of η's at every primary input of a functional unit. Consider a two-input functional unit, FU, that is shared by two operations, OP_1 and OP_2. Each operation originally has two input sequences, $\langle a_1^{(i)} \rangle$ and $\langle b_1^{(i)} \rangle$ for OP_1 and $\langle a_2^{(i)} \rangle$ and $\langle b_2^{(i)} \rangle$ for OP_2. By applying Eq. (1), the corresponding η's can be computed and they are denoted as η_{a1}, η_{b1}, η_{a2} and η_{b2}. The shared FU will then have two new input sequences $\langle v_a^{(j)} \rangle$ and $\langle v_b^{(j)} \rangle$, where $\langle v_a^{(j)} \rangle = \textit{interleave}$ $(\langle a_1^{(i)} \rangle, \langle a_2^{(i)} \rangle)$ and $\langle v_b^{(j)} \rangle = \textit{interleave}$ $(\langle b_1^{(i)} \rangle, \langle b_2^{(i)} \rangle)$. The Δ_η of two sequences into input A is computed as $D_a = |\eta_{a1} - \eta_{a2}|$. $D_b = |\eta_{b1} - \eta_{b2}|$ is for input B. The switching activity of the shared FU will be constructed using D_a and D_b as well as η_{a1} and η_{b1}.

Consider the construction of a difference table for the FU under the sharing condition $D_a = 3$ and $D_b = 0$. For every entry in the table, two sequences whose η's are η_{a1} and η_{a2} will be first generated by a

sequence generator where $\eta_{a2}=\eta_{a1}+D_a=\eta_{a1}+3$. Then they are alternately fed to input A of FU. The same procedures are applied to the other two sequences at input B of FU, whose η's are η_{b1} and η_{b2} $(=\eta_{b1}+D_b=\eta_{b1}+0=\eta_{b1})$. To form a difference table, η_{a1} and η_{b1} range from 1 to $16-D_a$ and from 1 to $16-D_b$, respectively for 16-bit datawidth. Figure 2 plots a difference table when $D_a=3$ and $D_b=0$ for a 16-bit MSB-first CS array multiplier.

In our method, the switching activity of a shared FU consists of two parts: first part is switching activity without resource sharing, and the second part is switching activity due to the sharing of resources. The switching activity can be expressed as:

$$SW(OP_1, OP_2) \tag{3}$$

$$= SW_{\text{no\_sharing}} + \Delta sw \tag{3(a)}$$

$$= SW_{\text{base}}(OP_1) + SW_{\text{base}}(OP_2) + SW_{\text{diff}}(OP_1, OP_2) + SW_{\text{diff}}(OP_2, OP_1) \tag{3(b)}$$

$$= SW_{\text{base}}(\eta_{a1}, \eta_{b1}) + SW_{\text{base}}(\eta_{a2}, \eta_{b2}) + SW_{\text{diff}}(\eta_{a1}, \eta_{b1}, D_{a_1}, D_{b_1}) + SW_{\text{diff}}(\eta_{a2}, \eta_{b2}, D_{a_2}, D_{b_2}) \tag{3(c)}$$

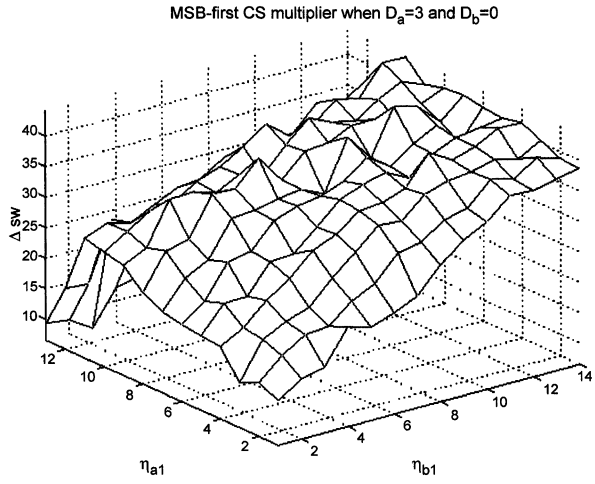

FIGURE 2 Switching differences (Δsw) for a 16-bit MSB-first CS array multiplier under resources sharing when mixing condition is $D_a=3$ and $D_b=0$.

where $SW_{\text{base}}(\cdot)$ corresponds to the first part of switching activity and $SW_{\text{diff}}(\cdot)$ is the second part. $SW_{\text{base}}(\cdot)$ can be looked up from the base table, while $SW_{\text{diff}}(\cdot)$ can be found from the difference tables.

Though we used a two-port functional module to explain our idea, our method can be applied to other common DSP modules such as registers and multiplexers. A register can be modeled as a one-port module. The Eqs. (3(a)) and (3(b)) remain the same, while the last part of Eq. (3) is modified to

$$SW(OP_1, OP_2) = SW_{\text{base}}(\eta_{a1}) + SW_{\text{base}}(\eta_{a2}) + SW_{\text{diff}}(\eta_{a1}, D_{a_1}) + SW_{\text{diff}}(\eta_{a2}, D_{a_2}) \tag{4}$$

Let us now turn our attention to multiplexers (MUXes). Since our modeling for resource sharing actually considers a module under multiplexing situations, the modeling of the MUX is quite unique in our model and requires special treatment. We first focus on a 2×1 MUX and expand our model to a $n \times 1$ MUX later. Since we focus on data-dominated applications, we assume that a $n \times 1$ MUX is multiplexing among its input sources with equal opportunity. So each input sequence of a 2×1 MUX will have 50% chance to appear at the output of the MUX.

Since the MUX is always in multiplexing mode, there is no need for base table. We can consider a 2×1 MUX as an one-input module that shares two sequences from its two inputs. Therefore, we can formulate the switching counts as follows:

$$SW(\eta_{a1}, \eta_{a2}) = SW_{\text{diff}}(\eta_{a1}, D_{a1}) + SW_{\text{diff}}(\eta_{a2}, D_{a2}) \tag{5}$$

III.3. Sharing of *n* Operations

Our work can be generalized to characterize switching activity (power) for sharing a functional unit among n operations. It is observed that switching transitions occur when two consecutive vectors are different. If the order of interleaving sequences is known, the variations of switching

activity is the summation of n individual mixings. Suppose $n=3$ and the order of interleaving sequences is $\langle a_1^{(i)}\rangle$, $\langle a_2^{(i)}\rangle$ and $\langle a_3^{(i)}\rangle$. Three operations, OP_1, OP_2 and OP_3, share one functional unit. The first part of switching activity ($SW_{\text{no\_sharing}}$) is the summation of individual switching activity without sharing a resource. The second part of switching activity can be expressed as $\Delta sw = SW_{\text{diff}}(OP_1,OP_2,OP_3) = SW_{\text{diff}}(OP_1, OP_2) + SW_{\text{diff}}(OP_2,OP_3) + SW_{\text{diff}}(OP_3,OP_1)$. Δsw is the summation of every consecutive pair in the order of mixing. For sharing of n, we need to look up n pairs. This makes our technique very effective and fast in estimating switching activity (or power).

$$\begin{aligned} SW(\{OP_i\})_{i=1\ldots n} &= \sum_{i=1}^{n} SW_{\text{base}}^{(i)}(OP_i) \\ &\quad + \sum_{j=1}^{n-1} SW_{\text{diff}}^{(j)}(OP_j, OP_{j+1}) \\ &\quad + SW_{\text{diff}}(OP_n, OP_1) \\ &= \sum_{i=1}^{n} SW_{\text{base}}^{(i)}(\eta_a, \eta_b) \\ &\quad + \sum_{j=1}^{n} SW_{\text{diff}}^{(j)}(\eta_a, \eta_b, D_a, D_b) \end{aligned} \quad (6)$$

where n is number of sharing of one resource.

The generalization for a $n \times 1$ MUX is the expansion of Eq. (5) for a 2×1 MUX. The switching counts can be formulated using the following equation.

$$\begin{aligned} &SW_{\text{MUX}}(\{\eta_{a_i}\})_{i=1\ldots n} \\ &\quad = SW_{\text{diff}}(\eta_{a_1}, D_{a_1}) + SW_{\text{diff}}(\eta_{a_2}, D_{a_2}) \\ &\qquad + \cdots + SW_{\text{diff}}(\eta_{a_n}, D_{a_n}) \\ &\quad = \sum_{i=1}^{n} SW_{\text{diff}}(\eta_{a_i}, D_{a_i}) \end{aligned} \quad (7)$$

III.4. Storage Requirements

For a two-input functional unit, all tables for switching activity characterization require $O(N^4)$ storage, where N is the bit width. We need to construct one base table and $(N-2)\times(N-2)$ difference tables. Each difference table has $(N-D_a)\times(N-D_b)$ entries. The storage requirement is $\sum_{D_a=1}^{N-2}\sum_{D_b=1}^{N-2}(N-D_a)\times(N-D_b)+(N-2)\times(N-2)$. This storage can be reduced to $O(N^2)$ if polynomial surface fitting can be applied to each table. Since we have experimentally observed as high as 20% of mismatch between data and quadratic approximation, we do not use polynomial surface fitting to determine the switching activity. Instead, we use tabular forms to store the results.

III.5. Analysis of Characterized Results of Functional Units

We have constructed a set of switching activity models for different types of multipliers and adders using the method described above. The relative switching increment using resource sharing is called *percentage switching increment* (γ) and is defined as

$$\gamma = \frac{SW_{\text{diff}}(\eta_a, \eta_b, D_a, D_b)}{SW_{\text{base}}(\eta_{a1}, \eta_{b1}) + SW_{\text{base}}(\eta_{a2}, \eta_{b2})} \times 100. \quad (8)$$

The quantity indicates the relative switching behaviors under sharing of resources. Figure 3 shows the percentage switching increment for an MSB-first CS array multiplier under resource sharing. There are five bars at every position of η_{b1}. Each represents $D_a = 1,2,3,4,9$. The mixing conditions are $D_b = 0$ while D_a is varying. The varying D_a is based on fixing η_{a1} to 6. As $D_a = |\eta_{a1} - \eta_{a2}|$, η_{a2} can take the following values: 7,8,9,10,15. We collect data from 5 different mixing conditions, $(D_a=1,\ D_b=0)$, $(D_a=2,\ D_b=0),\ldots,(D_a=9,\ D_b=0)$. We take the data at $\eta_{a1}=6$ and $\eta_{b1}=1,2,\ldots,14$ from each mixing condition and place them side by side. The percentage switching activity increment at $D_a=9$ averages 22% and reaches as high as 28%. It is observed that γ increases as D_a increases. Similar behaviors can be observed for LSB-first array

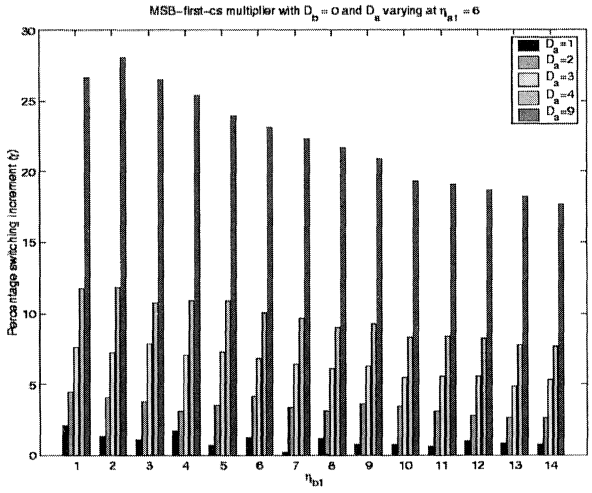

FIGURE 3 Percentage switching increment (γ) for a 16-bit MSB-first CS array multiplier under resources sharing when $D_b = 0$ and D_a is varying and η_{a_1} is fixed at 6.

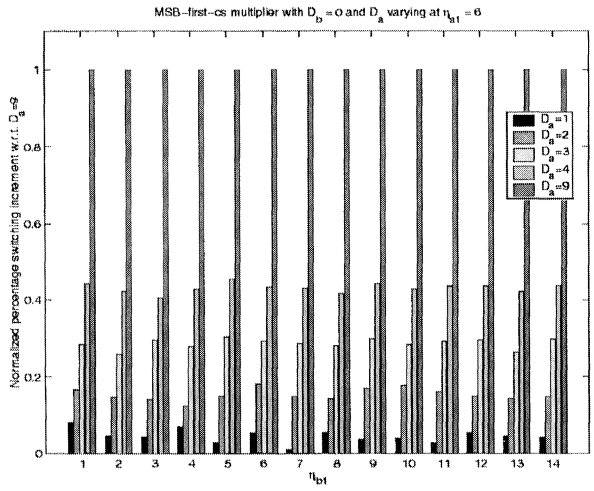

FIGURE 4 Normalized percentage switching increment for a 16-bit MSB-first CS array multiplier under resources sharing when $D_b = 0$ and D_a is varying and η_{a_1} is fixed at 6.

multipliers, tree multipliers and adders. We also normalize bars at every position of η_{b1} by the bar whose value is the largest among the five bars. For example, bars that represent γ at $D_a = 1,2,3,4,9$ are normalized by the γ at $D_a = 9$. Figure 4 shows the normalized increment in switching. We observe that no matter how large η_{b1} is, there is always about 90% difference between $D_a = 1$ and $D_a = 9$. These results suggest that Δ_η of sharing sequences have significant impact on switching activity.

IV. EXPERIMENTAL RESULTS

We compare our signal-strength based switching activity estimation (SSSAE) with a power simulator, PowerMill, for the relative power consumption of DSP functional units. SSSAE works on higher level of design abstraction, *i.e.*, architectural or register-transfer level (RTL), while PowerMill needs detailed implementation at the circuit level. The prototype SSSAE software has been implemented in the C++ programming language. Our evaluation shows that there is about 6–17% difference in switching activity estimation and very accurate power simulation on the relative power consumption of different components of an implementation.

IV.1. Setup and Procedures

We have two implementations for a 6-tap FIR filter at the RTL and at the circuit level. The two implementations for the filter use the same number of resources, but have different resource binding for functional units. Figures 5 and 6 show two implementations at RTL.

Information needed for switching (power) estimation for SSSAE and PowerMill are different. SSSAE requires a scheduled and allocated data flow graph (DFG) and input signal statistics, while PowerMill needs circuit-level description and data sequences. We use the following procedures to obtain the circuit-level implementation. The scheduled and allocated DFGs are first implemented in VHDL. Then we use Synopsys Design Compiler to synthesize the VHDL code using low-power standard cell library developed by Carnegie Mellon University [10]. The cells in the library are implemented using 0.35 μm CMOS technology with 3.3 V power supply.

We use variety of data sequences to evaluate SSSAE. These data sequences are obtained from

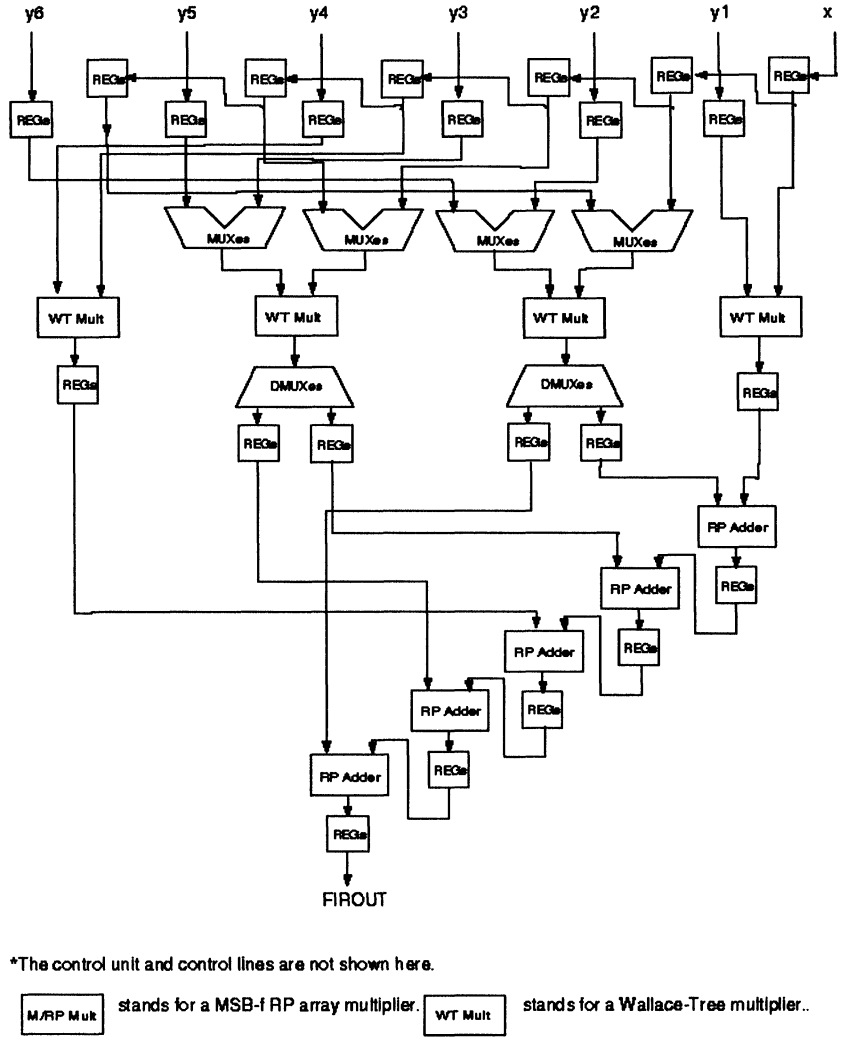

FIGURE 5 Implementation I of a 6-tap FIR Filter.

various sources as shown in Table I. AUD1, AUD2 and IMG1 are real audio and image data signals. SIG1 and SIG2 are stationary Gaussian input signals generated by computers. Input signal statistics for SSSAE are obtained by extracting test data sequences. The procedure to obtain the signal strength of internal nodes required by the SSSAE are as follows. Signal statistics are propagated from the primary inputs using analytical method given in [11]. This method enables to propagate the statistics with fast and reasonable accuracy. With the signal statistics such obtained, signal strength can be computed as described in Section II.

Our switching activity estimation is not designed for accurate estimation of real power consumption, but focuses on the relative power dissipation in components and rapid evaluation time. The results obtained from PowerMill have confirmed that our estimation are in good agreement with the circuit-level simulation. The execution time for each design using PowerMill took about

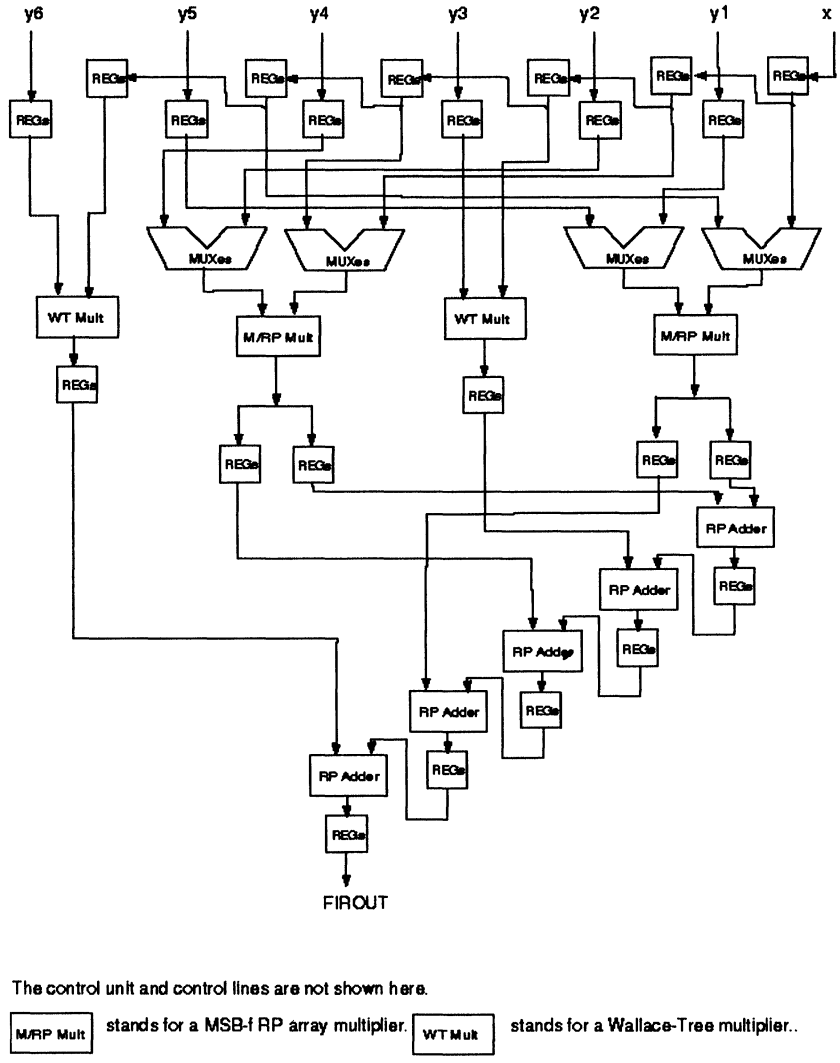

FIGURE 6 Implementation II of a 6-tap FIR Filter.

TABLE I Description and statistics of the data sets

Data set	Description	μ	σ
AUD1	speech	6.27	63.14
AUD2	speech	3.37	20.40
IMG1	picture	124.52	4473.64
SIG1	computer-generated	2.17	3667.90
SIG2	computer-generated	14.39	1397.41

20–30 hours on a Sun Ultra 30 with 128 M RAM, while our switching activity estimation at RTL took less than 30 secs on the same platform.

IV.2. Analysis of Results

We compare the relative power dissipation of different components of an implementation using two methods: one obtained using power simulation (*i.e.*, PowerMill) and the other by switching activity estimation (*i.e.*, SSSAE). Here FUs represent functional units, MUXes are for multiplexers and REGs are for registers.

Tables II and III show the relative power consumption for different components of the

TABLE II Simulated and estimated relative power dissipation for implementation I of Figure 5

Signal	Est. md.	FUs (%)	MUXes (%)	REGs (%)
AUD1	Sw. Act.	34.61	4.43	60.95
	Power	22.53	2.70	74.08
AUD2	Sw. Act.	42.10	4.05	53.84
	Power	25.66	2.61	71.10
IMG1	Sw. Act.	34.61	4.44	60.95
	Power	21.57	2.74	74.99
SIG1	Sw. Act.	32.36	4.54	63.11
	Power	19.34	2.82	77.13
SIG2	Sw. Act.	29.54	4.67	65.79
	Power	18.02	2.88	78.38

TABLE III Simulated and estimated relative power dissipation for implementation II of Figure 6

Signal	Est. Md.	FUs (%)	MUXes (%)	REGs (%)
AUD1	Sw. Act.	29.49	4.44	66.07
	Power	20.00	3.04	76.21
AUD2	Sw. Act.	36.83	4.13	59.04
	Power	24.15	2.98	72.19
IMG1	Sw. Act.	29.49	4.44	66.07
	Power	19.40	3.10	76.76
SIG1	Sw. Act.	27.33	4.51	68.16
	Power	19.65	3.07	76.55
SIG2	Sw. Act.	24.96	4.59	70.44
	Power	18.25	3.11	77.90

implementation. The average difference between the estimation and the simulation is 11.27% for FUs, 1.52% for MUXes and 12.09% for REGs. The power consumption due to the registers ranges from 70% to 80% of the total power consumption. FUs consumed about 18% to 26% and MUXes dissipated less than 4%.

It is observed from Table IV that the ratio between switching activity and accurate power simulation is similar for different rows of the table. The ratio is related to real capacitance in the circuit. By extracting the ratio and applying to our models, the accuracy can be increased. For example, we can obtain the empirical factor for FUs of Implementation I as 0.618 from Table IV.

TABLE IV The ratio for FUs between switching estimation and power simulation extracted from Table II

Signal	Sw. act. (%)	Power sim. (%)	Ratio (col3/col2)
AUD1	34.61	22.53	0.65
AUD2	42.10	25.66	0.61
IMG1	34.61	21.57	0.62
SIG1	32.36	19.34	0.60
SIG2	29.54	18.02	0.61
Average	–	–	0.618

TABLE V Simulated and estimated relative power dissipation for implementation I of Figure 5. The estimated switching activity for different components have been multiplied by the extracted empirical factor

Signal	Est. Md.	FUs (%)	MUXes (%)	REGs (%)
AUD1	Sw. Act.	21.58	2.78	75.64
	Power	22.53	2.70	74.08
AUD2	Sw. Act.	27.46	2.65	69.89
	Power	25.66	2.61	71.10
IMG1	Sw. Act.	21.58	2.78	75.64
	Power	21.57	2.74	74.99
SIG1	Sw. Act.	19.91	2.80	77.29
	Power	19.34	2.82	77.13
SIG2	Sw. Act.	17.89	2.84	79.28
	Power	18.02	2.88	78.38

Empirical factors for MUXes and REGs can be obtained similarly. They are 0.622 and 1.23 respectively. Use of such factors reduces the error between the power simulation and the switching activity estimation to within 5% as shown in Table V. Even though our modeling method does not include capacitance information of the design and glitches that occur during the operations of real circuits, the model still provides reasonable accuracy for the relative power consumption.

V. CONCLUSIONS

We have proposed an effective switching activity modeling and estimation technique that uses signal

strength to characterize and evaluate components under resource sharing. This scheme considers not only switching activities due to varying input signal statistics, but also transitions stemming from hardware sharing. The effectiveness of the proposed model is verified by two implementations of an FIR filter. A variety of synthetic and real data are applied to the FIR filters. The results show that our models track the circuit-level simulator reasonably well.

The transformation of parallel algorithms to a resource-limited design is very common in DSP applications. The proposed generalized signal-strength based model provides a valuable tool to estimate and compare switching activites in the early phase of design when the input signal statistics are known.

References

[1] Powell, S. and Chau, P. (1990). "Estimating Power Dissipation of VLSI Signal Processing Chips: the FA Techniques", *Proc. IEEE Workshop on VLSI Signal Processing*.

[2] Landman, P. E. and Rabaey, J. M., "Activity-sensitive Architectural Power Analysis", *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Jun., 1996.

[3] Raghunathan, A., Dey, S. and Jha, N. K. (1996). "Register-Transfer Level Techniques for Switching Activity and Power Consumption", *Proc. of the International Conference on Computer Aided Design*.

[4] Gupta, S. and Najm, F. (1997). "Power Macromodeling for High Level Power Estimation", *Proc. of 34th Design Automation Conf.*

[5] Lundberg, M., Muhammad, K., Roy, K. and Wilson, S. K. (1999). "High-level Modeling of Switching Activity with Application to Low-Power DSP System Synthesis", *IEEE International Conference on Acoustics, Speech, and Signal Processing*.

[6] Mehra, R., Guerra, L. M. and Rabey, J. M. (1996). "Low-power Architectural Synthesis and the Impact of Exploiting Locality", *Journal of VLSI Signal Processing*.

[7] Rabaey, J. (1996). *Digital Integrated Circuits: A Design Perspective*, Prentice Hall.

[8] Chandrakasan, A. P., Sheng, S. and Brodersen, R. W., "Low-Power CMOS Digital Design", *Journal of Solid-State Circuits*, April, 1992.

[9] Muhammand, K., Somasekhar, D. and Roy, K. (1999). "Switching Characteristics of Generalized Array Multiplier Architectures and their Applications to Low Power Design", *Proc. of International Conference on Computer Design: VLSI in Computers and Processors*.

[10] Chris Inacio, "*The Carnegie Mellon Synthesizable Digital Signal Processor Core*", CMU DSP Team Report, June, 1999.

[11] Ramprasad, S., Shanbhag, N. R. and Hajj, I. N., "Analytical Estimation of Signal Transition from Word-Level Signal Statistics", *IEEE Trans. Computer-Aided Design*, July, 1997.

Authors' Biographies

Lih-Yih Chiou is currently a Ph.D. degree candidate in the Department of Electrical and Computer Engineering at the Purdue University. He worked with Lucent Technology, Holmdel, NJ, in the summer of 1998. He is a member of the IEEE. His research interests include high-level power modeling, high-level synthesis for low power design and design automation for integrated circuits.

Khurram Muhammad is working in the mixed signal circuits lab at Texas Instruments, Dallas, TX 75243 as a member of technical staff. His research interests include simulation, low-power and high performance design, CAD techniques and methodologies for digital systems and mixed signal circuits. He is a member of IEEE and served in the technical committee of the 10th Great Lakes Symposium on VLSI in Chicago, IL, in March, 2000.

Kaushik Roy received his B.Tech. from Indian Institute of Technology, Kharagpur and his Ph.D. from University of Illinois at Urbana-Champaign. He is currently an Associate Professor of Electrical and Computer Engineering, Purdue University, West Lafayette. Prior to joining Purdue, he worked with Texas Instruments in Dallas. He has held visiting faculty positions at Intel and at the University of California, Berkeley. He is/was in the editorial board of IEEE Transactions on VLSI Systems, IEEE Design and Test magazine, and IEEE Transactions on Circuits and Systems I. Kaushik received NSF Career Award in 1995, best paper awards at IEEE International Test Conference and IEEE International Symposium on Quality of IC Design, and is currently a Purdue University Faculty Scholar Professor.