ACTIVITY MODELS FOR USE IN LOW POWER, HIGH-LEVEL SYNTHESIS

R. Henning and C. Chakrabarti
Department of Electrical Engineering
Arizona State University
Tempe, Arizona 85287, USA

ABSTRACT

Characteristics of the data being processed can be used to reduce the power consumption in the data path of a VLSI circuit by exploiting their relationship with transition activity during high-level synthesis. Important relationships between fixed-point, two's complement data characteristics and 0→1 transition activity in static CMOS circuits are presented in this paper. Models for computing transition activity in terms of a new set of transition parameters are developed. Propagation of data characteristics through multiplication and addition functional units is discussed. The use of the relationships and models to analyze and significantly reduce 0→1 transition activity with little computational effort is illustrated with examples.

1. INTRODUCTION

Though the study of power reduction techniques for VLSI implementations has received a great deal of attention in recent years, the majority of study has neglected the relationship between power dissipation and the data being processed. As a result, VLSI designers typically devote little attention to exploitation of data characteristics to reduce power. This paper addresses this issue by developing and demonstrating the importance of general relationships between fixed-point, two's-complement data characteristics and power dissipation in static CMOS circuits.

Power reduction can be quite significant if data characteristics are fully exploited. The equation for switching power in a static CMOS gate is

$$P_{switching} = \alpha_{0\to1} C_L V_{dd}^2 f_{clk}.$$

Data characteristics can be used to reduce 0→1 transition activity $\alpha_{0\to1}$. Reducing load capacitance, C_L, or supply voltage, V_{dd}, can also significantly reduce switching power and have been the focus of much study. However, significant reduction of these terms can be difficult from a technology standpoint, and reduction of V_{dd} is typically accompanied by a significant decrease in clock frequency, f_{clk}. In contrast, the power reduction obtained by reducing $\alpha_{0\to1}$ is fairly independent of integration technology and can be achieved without reducing processing speed.

This research was supported by the Motorola SABA program and by the Center for Low Power Electronics.

In the past, relationships between data characteristics and $\alpha_{0\to1}$ in the data path have been illustrated for special cases [1][2][3]. Few general relationships have been defined and modeled to aid in finding low $\alpha_{0\to1}$, high-level designs. Models that do exist for relating data characteristics and $\alpha_{0\to1}$ are targeted for high-level power estimation accuracy under certain assumptions, not analyzing the causes of $\alpha_{0\to1}$ [4][5]. Suggested approaches for finding low $\alpha_{0\to1}$ implementations effectively estimate $\alpha_{0\to1}$ for a large number of possible designs, rather than narrowing down the choices by inspection of data characteristics first [6]. Such methods require significant computation.

The general relationships presented in this paper can be used to identify and exploit data characteristics to reduce $\alpha_{0\to1}$ in the data path during high-level synthesis. Models are developed for $\alpha_{0\to1}$ in terms a new set of transition parameters meant for analyzing the causes of $\alpha_{0\to1}$. Use of these relationships and models to analyze and significantly reduce $\alpha_{0\to1}$ with relatively little computation is illustrated with examples.

This paper is organized as follows. In Section 2, a new set of important transition parameters that affect $\alpha_{0\to1}$ are identified. In Section 3, models that relate these parameters to $\alpha_{0\to1}$ are presented. Propagation of data characteristics through multiplication and addition functional units is shown in Sections 4 and 5. Examples of how to apply this knowledge to reduce $\alpha_{0\to1}$ during high-level synthesis are given in Section 6.

2. TRANSITION PARAMETERS

Two transition examples are used to illustrate the transition parameters defined in this section (see Figures 1 and 2). Each figure shows consecutive binary values occurring at a 16-bit node in a static CMOS circuit. The binary numbers are fixed-point, two's-complement representations of decimal values between -1 and 1 (also shown in figures).

	Binary Value	Decimal Value
Previous Value:	0000011010010000	0.05126953125
Current Value:	0000011011001000	0.052978515625
Node Bit Type:	IS CMD|SS IT	

Figure 1. Transition between two positive 16-bit values.

	Binary Value	Decimal Value
Previous Value:	0000011001010000	0.04931640625
Current Value:	1111001100110000	-0.10009765625
Node Bit Type:	IS IT	

Figure 2. Transition between positive and negative 16-bit values.

2.1 Sign Bits

Sign bits are consecutive bits following the most significant bit of a value that are the same as the most significant bit. In the transition examples of Figures 1 and 2, all of the values have five sign bits except for the current value in Figure 2 which has four. In a single node transition, there are two distinct parameters related to sign bits that affect $\alpha_{0\to1}$ of node bits. The first and most obvious parameter is the type of sign transition. In Figure 1, the MSB makes a 0→0 transition as a result of the +→+ sign transition, while in Figure 2, the MSB makes a 0→1 transition as a result of the +→-. The second parameter is the number of intersecting sign (IS) bits in a single transition, n_{IS}, which is defined as the minimum of the number of sign bits in the two values involved in the transition. In Figure 1, n_{IS} is five. It is four in Figure 2. The property of IS bits that makes n_{IS} useful is they only make 0→1 transitions if the sign change is +→-.

2.2 Truncation Bits

Truncation bits in a value are the consecutive bits beginning with the least significant bit (LSB) that are zero. All of the values in the Figures 1 and 2 have four truncation bits except the current value of Figure 1 which has three truncation bits. The parameter associated with truncation bits that affects 0→1 transition activity is the number of intersecting truncation (IT) bits between consecutive node values, n_{IT}. n_{IT} is defined similarly to n_{IS} as the minimum of the number of truncation bits in the two values involved in the transition. In Figure 1, n_{IT} is three. In Figure 2, it is four. The property of IT bits that makes n_{IT} useful is they all make 0→0 transitions. They never make 0→1 transitions.

2.3 Data Bits

Data bits includes all other bits of a value that are not sign or truncation bits. In Figures 1 and 2, the previous values both have seven data bits while the current values both have eight data bits. An important parameter associated with data bits that affects 0→1 transition activity is the number of consecutive matching data (CMD) bits between consecutive node values during a same sign transition, $n_{CMD|SS}$. $n_{CMD|SS}$ is defined as the number of consecutive data bits that match in value and position between the two values involved in a same sign transition (+→+ or -→-) beginning with the most significant data bit of either value. Note that this definition implies $n_{CMD|SS}$ is zero whenever the number of sign bits in the two transition values are not equal. In Figure 1, $n_{CMD|SS}$ is four, whereas in Figure 2, $n_{CMD|SS}$ is undefined. The property of CMD bits that makes $n_{CMD|SS}$ useful is they all make 0→0 or 1→1 transitions. They never make 0→1 transitions.

3. ACTIVITY MODELS

Using the transition parameters, sign transition type, n_{IS}, $n_{CMD|SS}$, and n_{IT}, useful models for 0→1 transition activity on an N-bit node have been found. In these models, it is assumed that bits that are independent of these parameters, have equal likelihood of being 0 or 1.

3.1 Sign Bit Activity

Assume that $\overline{n}_{CMD|SS} < 1$ and $\overline{n}_{IT} < 0.5$ for a practical stream of data encountered by an N-bit node. $\overline{n}_{CMD|SS}$ is the mean number of CMD bits for same sign transitions only. $\overline{n}_{IT}$ is the mean number of IT bits. A stream of this type may contain significant amounts of down-scaled data or data that does not require the full N-bit range of the representation. The probability of a 0→1 transition per bit of an N-bit node in this case can be modeled as

$$\begin{aligned}\alpha_N = \tfrac{1}{4N}\{&N + (1-P_{OS})(\tfrac{1}{3} - \overline{n}_{IS|SS}) \\ &+ P_{OS}(\tfrac{3}{2}\overline{n}_{IS|(+\to-)} - \tfrac{1}{2}\overline{n}_{IS|(-\to+)} - \tfrac{1}{3})\}.\end{aligned} \tag{1}$$

P_{OS} is the probability of an opposite sign transition. $\overline{n}_{IS|SS}$ is the mean number of intersecting sign bits for same sign transitions only. $\overline{n}_{IS|(+\to-)}$ is the mean number of intersecting sign bits for positive to negative value transitions only. $\overline{n}_{IS|(-\to+)}$ is the mean number of intersecting sign bits for negative to positive transitions only.

From (1), it is clear that arranging data such that $\overline{n}_{IS|SS}$ and $\overline{n}_{IS|(-\to+)}$ are increased, while $\overline{n}_{IS|(+\to-)}$ and P_{OS} are decreased will reduce α_N. Finding the arrangement that minimizes α_N, however, requires joint optimization of these parameters. In practice, focusing on reducing P_{OS} may produce the greatest reduction in α_N for the least amount of work due to the scaling effect P_{OS} has on the contributions of the other terms.

As a minor note, the $\frac{1}{3}$ terms in the equation could be modeled more accurately to provide more accurate estimates of α_N. There is a strong dependence of these terms on the number of CMD bits for both same sign and opposite sign of transitions. However, as will be demonstrated in Section 6, the effect of these terms does not appear to be critical for the purposes of high-level synthesis. So, detailed modeling of these terms is neglected here in order to simplify the model.

3.2 Data Bit Activity

Assume that the data stream being considered has $\overline{n}_{CMD|SS} \geq 1$ and $\overline{n}_{IT} < 0.5$. Such a stream of data can result, for example, at the accumulation register of an accumulator. In this case, α_N can be modeled as

$$\begin{aligned}\alpha_N = \tfrac{1}{4N}\{&N + (1-P_{OS})(\tfrac{3}{2} - \overline{n}_{CMD|SS} - \overline{n}_{IS|SS}) \\ &+ P_{OS}(\tfrac{3}{2}\overline{n}_{IS|(+\to-)} - \tfrac{1}{2}\overline{n}_{IS|(-\to+)} - \tfrac{1}{3})\}.\end{aligned} \tag{2}$$

Equation (2) is very similar to (1), except that the $\frac{3}{2} - \overline{n}_{CMD|SS}$ term has replaced the $\frac{1}{3}$ term. This term allows the effect of $\overline{n}_{CMD|SS}$ being significantly large to be modeled relative to the other parameters. It accounts for the significant reduction in α_N not attributable to IS bits that occurs when data is arranged in an increasing or decreasing order. This term, like the $\frac{1}{3}$ term it replaces, could be modeled as a more accurate function of $\overline{n}_{CMD|SS}$. However, the complexity of such a model does not appear to be necessary for high-level synthesis applications.

3.3 Truncation Bit Activity

Next, assume that the data stream being considered has $\overline{n}_{CMD|SS} < 1$ and $\overline{n}_{IT} > 0.5$. A stream of this type can result, for example, if data like that modeled in (1) is significantly truncated or up-scaled. In this case, α_N can be modeled as

$$\begin{aligned}\alpha_N = \tfrac{1}{4N}\{N + \tfrac{1}{3} - \overline{n}_{IT} + (1 - P_{OS})(\tfrac{1}{3} - \overline{n}_{IS|SS}) \\ + P_{OS}(\tfrac{3}{2}\overline{n}_{IS|(+\to-)} - \tfrac{1}{2}\overline{n}_{IS|(-\to+)} - \tfrac{1}{3})\}.\end{aligned} \quad (3)$$

Equation (3) is very similar to (1), except that the $\frac{1}{3} - \overline{n}_{IT}$ term is added. This term allows the reduction in α_N resulting from $\overline{n}_{IT}$ being significantly large to be modeled relative to the other parameters. This term, could also be modeled more accurately. However, the complexity of such a model does not appear to be necessary for high-level synthesis applications.

Finally, if $\overline{n}_{CMD|SS} \geq 1$ and $\overline{n}_{IT} > 0.5$, α_N can be modeled as

$$\begin{aligned}\alpha_N = \tfrac{1}{4N}\{N + \tfrac{1}{3} - \overline{n}_{IT} + (1 - P_{OS})(\tfrac{3}{2} - \overline{n}_{CMD|SS} - \overline{n}_{IS|SS}) \\ + P_{OS}(\tfrac{3}{2}\overline{n}_{IS|(+\to-)} - \tfrac{1}{2}\overline{n}_{IS|(-\to+)} - \tfrac{1}{3})\}.\end{aligned} \quad (4)$$

This model accounts for all of the previously mentioned effects when they all exist at once. One inconsistency of the model that has been noted occurs when $\overline{n}_{IT}$ is very large. In this case, some bits may be modeled as CMD bits that are really a result of truncation. So, even though the estimate may be close, the assumption that it is due in part to close valued consecutive data may be incorrect. However, such a case is unlikely in practice.

4. MULTIPLICATION RELATIONSHIPS

Consider an N by N bit, fixed-point, two's-complement multiplier where input values are constrained between -1 and 1. It produces a 2N-1 bit result at the output. Important relationships between input and output data characteristics are summarized below.

1. Sign transitions: If both inputs make a same sign transition or an opposite sign transition, the output makes a same sign transition. However, if one of the inputs makes a same sign transition while the other makes an opposite sign transition, the output will make an opposite sign transition.

2. Sign bits: The number of output sign bits from a single multiplication is the sum of the number of sign bits of the two inputs or one less than that sum, except when one of the inputs is zero.

3. Data bits: Significant numbers of CMD bits tend not to appear for consecutive outputs of a multiplier.

4. Truncation bits: The number of truncation bits at the output is the sum of the number of truncation bits at the two inputs, except when one of the inputs is zero.

5. ADDITION RELATIONSHIPS

Consider an N bit, fixed-point, two's-complement adder where input values are constrained between -1 and 1. It produces an N+1 bit result at the output. Important relationships between input and output data characteristics are summarized below.

1. Sign transitions: The relationship between sign transitions at the inputs and the sign transition at the output for two consecutive addition operations requires knowledge of the relative magnitude of the two inputs in certain cases. For example, if both inputs make a same sign transition, but the input 1 values have the opposite sign of the input 2 values, the output transition will depend on which input is larger during each addition operation.

2. Sign bits: The number of output sign bits from a single addition of two values with the same sign is the minimum of the number of sign bits of the two inputs or one less than that minimum. If the two inputs have opposite sign, then the output always has at least as many sign bits as the minimum of the number of sign bits of the two inputs. However, as the difference between absolute values of the inputs decreases, the number of sign bits at the output tends to increase.

3. Data bits: If a small magnitude number is added to a large magnitude number, consecutive data bits beginning with the most significant data bit of the output tend to match data bits of the large magnitude input in value and position.

4. Truncation bits: The number of truncation bits at the output is the minimum of the number of truncation bits at the two inputs when the number of truncation bits at each input is different. If both inputs have the same number of truncation bits, more truncation bits may result, but on average the number of truncation bits at the output will be near the minimum of truncation bits at the two inputs.

6. APPLICATION EXAMPLES

6.1 Data Characterization

The models of Section 3 can be used to characterize a stream of data. This information can then be used during high-level synthesis to help reduce $\alpha_{0\to1}$. Take, for instance, a speech signal generated by a male speaker uttering "Card games are fun to play." Estimated values of the parameters introduced in Section 3 for this signal are shown in Table 1. When these values are plugged into (2), an estimate of 0.173 for α_N is found. The actual $\alpha_N \approx 0.176$. Since the estimates are close, we can speculate that a large $\overline{n}_{IS|SS}$ is the dominant characteristic of the data stream. A large $\overline{n}_{IS|SS}$ can be a significant asset when this signal is filtered, as will be shown in Subsection 6.3. $\overline{n}_{CMD|SS}$ is fairly low, because even though speech is a slowly varying signal, the case where consecutive, large-magnitude values are close does not occur often.

Table 1: Parameter values for 16-bit speech data.

Param	P_{OS}	$n_{IS\|SS}$	$n_{CMD\|SS}$	$n_{IS\|(+\to-}$	$n_{IS\|(-\to+}$	$\overline{n}_{IT}$	Model α_N	Actual α_N
Value	0.143	6.661	1.301	7.487	7.627	0.313	0.173	0.176

6.2 Multiplexing

The relationships considered thus far can be used to estimate the switching activity that will result from multiplexing two or more streams of data onto a particular node. Such information can be used to make high-level synthesis decisions to reduce $\alpha_{0\to 1}$. For example, consider the two 16-bit streams of data that produce the $\alpha_{0\to 1}$ values shown in Tables 2 and 3. It appears that we may be able to multiplex these two streams by interleaving their values and still achieve low $\alpha_{0\to 1}$ in many of the most significant node bits. However, as shown in Table 4, this is not the case. This result is more easily predicted by determining that $\bar{n}_{IS|SS}$ is only 1.988 for the accumulation register data, while it is 8.334 for the stream of down-scaled data. On the other hand, if we had interleaved two streams of the type shown in Table 3, the resulting $\alpha_{0\to 1}$ would remain like that of Table 3. As will be shown in the next example, it is sometimes possible to demultiplex a stream of data to reduce $\alpha_{0\to 1}$ as well.

Table 2: $\alpha_{0\to 1}$ for bits of an accumulation register stream.

MSBs	0.000	0.004	0.008	0.016	0.032	0.063	0.127	0.254
LSBs	0.262	0.246	0.238	0.246	0.246	0.262	0.278	0.262

Table 3: $\alpha_{0\to 1}$ for bits of a down-scaled stream.

MSBs	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
LSBs	0.250	0.251	0.250	0.250	0.249	0.250	0.250	0.250

Table 4: $\alpha_{0\to 1}$ for bits of interleaved stream.

MSBs	0.000	0.262	0.241	0.247	0.247	0.258	0.218	0.239
LSBs	0.247	0.266	0.270	0.239	0.245	0.282	0.235	0.266

6.3 Filter Computation

The activity models, multiplication relationships, and addition relationships described in previous sections can be combined to guide high-level synthesis. Consider a typical implementation of a 10th order FIR filter shown in Figure 3(a), where A[] represents the stream of filter coefficients {a(1), a(2), ..., a(10),a(1), a(2),..., a(10),...} and X[] represents the stream of inputs {x(9), x(8), ..., x(0),x(10), x(9),..., x(1),...}. The multiplier inputs are 16-bit and the accumulation is 31-bit. It is desired to speed the execution of this filter by about a factor of two and at the same time keep the increase in power required low. The first objective could be met by simply doubling the clock speed at the expense of doubling the power dissipation in the data path.

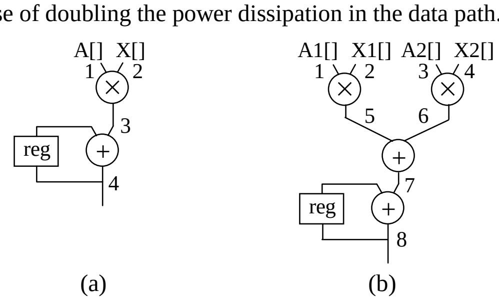

Figure 3. (a) Typical implementation of a 10th order FIR filter. (b) Improved implementation of a 10th order FIR filter.

If the filter coefficients and input data are characterized, it may be possible to find a significantly better implementation that achieves both objectives with little effort. For instance, in this case consecutive filter coefficients alternate in sign, consecutive filter inputs tend to have the same sign, and in both cases $\bar{n}_{IS}$ is large. This information implies an implementation that can isolate (demultiplex) the odd and even coefficients in separate streams while preserving the order of the input stream as much as possible could significantly reduce α_N at most of the nodes in the data path.

An implementation that does just this is shown in Figure 3(b) and results from using two multipliers in parallel rather than just one to isolate the odd and even coefficients from one another. The input streams are A1[]={a(1), a(3),..., a(9), a(1), a(3),..., a(9),...}, A2[]={a(2), a(4),..., a(10), a(2), a(4),..., a(10),...}, X1[]={x(9), x(7),..., x(1), x(10), x(8),..., x(2),...}, and X2[]={x(8), x(6),..., x(0), x(9), a(7),..., x(1),...}.

To verify the design choices, 40 samples of actual data from the application were run through the two implementations. Table 5 shows α_N for the nodes in Figure 3(a), and Table 6 shows α_N for the nodes in Figure 3(b). Estimates of α_N from (2) and the relationships of Sections 4 and 5 are very close to these values. By averaging activity over all nodes in each case, we find that average $\alpha_{0\to 1}$ is reduced from 0.280 to 0.206, a 26.4% reduction.

Table 5: Actual activities for implementation in Figure 3(a).

Node	1	2	3	4
α_N	0.299	0.158	0.308	0.306

Table 6: Actual activities for implementation in Figure 3(b).

Node	1	2	3	4	5	6	7	8
α_N	0.163	0.195	0.199	0.196	0.208	0.208	0.240	0.206

7. SUMMARY

The relationships considered here are for general use in exploiting data characteristics to reduce $\alpha_{0\to 1}$ in the data path during high-level synthesis. Models have been presented for $\alpha_{0\to 1}$ in terms of a new set of transition parameters, namely, sign transition type, n_{IS}, $n_{CMD|SS}$, and n_{IT}. Use of these relationships and models to reduce $\alpha_{0\to 1}$ by as much as 26% with little computational effort has been illustrated. Directions for future work include modeling other effects, improving accuracy, and integration into a high-level synthesis and power estimation tool.

REFERENCES

[1] Chatterjee A. and Roy R. "Synthesis of Low Power Linear DSP Circuits using Activity Metrics". *Proc of the IEEE Int Conf on VLSI Design*, 1994, pages 265-270.

[2] Chandrakasan A. and Brodersen R. "Minimizing Power Consumption in Digital CMOS Circuits". *Proc of the IEEE*, 83(4):498-523, 1995.

[3] Musoll E. and Cortadella J. "High-level Synthesis Techniques for Reducing the Activity of Functional Units". *Proc of the Int Symp on Low Power Design*, 1995, p 99-104.

[4] Landman P. and Rabaey J. "Architectural Power Analysis: The Dual Bit Type Method". *IEEE Trans on VLSI Systems*, 3(2):173-187, 1995.

[5] Raghunathan A., Dey S., and Jha N. "Register-Transfer Level Estimation Techniques for Switching Activity and Power Consumption". *Proc of the IEEE/ACM Int Conf on Computer-Aided Design*, 1996, pages 158-165.

[6] Raghunathan A. and Jha N. "Behavioral Synthesis for Low Power". *Proc of the Int Conf on Computer Design*, 1994, pages 318-322.