PARAMETERIZED LOGIC POWER CONSUMPTION MODELS FOR FPGA-BASED ARITHMETIC

Jonathan A. Clarke, Altaf Abdul Gaffar and George A. Constantinides

Department of EEE, Imperial College London
email: {jonathan.a.clarke, altaf.gaffar, g.constantinides}@imperial.ac.uk

ABSTRACT

The need for fast power estimation methods is a growing requirement in tools which perform power consumption optimization. This paper addresses the requirement by presenting a technique which is capable of providing a power estimate using only the word-level statistics of signals within an arithmetic hardware design. By abstracting away from the low-level details of a design it is possible to reduce the time required to calculate the power consumption dramatically. Power models for multiplication and addition have been constructed using an experimental method, and the operation of these models is illustrated by estimating the power consumed in logic for two example circuits: a sum of products and a parameterised polynomial evaluation. The proposed method is capable of providing an estimate within 10% of low-level power estimates given by XPower.

1. INTRODUCTION

The power dissipation in FPGAs has become an important design consideration in recent years due to the increasing costs manufacturers face for packaging and heat dissipation solutions, and the requirement for extended battery life in portable applications. Though FPGAs have higher power consumption than equivalent custom VLSI solutions due to the logic and routing overhead of FPGA circuits, their short time to market, small low-volume cost and steadily increasing performance make them an attractive alternative for many applications, and therefore optimizing the power consumption of FPGA designs is still an important task.

Power consumption in digital circuits can be divided into static and dynamic power, where static power consumption is due to leakage currents in the transistors of the circuit, and dynamic power is due to the switching of the circuit capacitances. The task of optimising the static power of an FPGA falls entirely on the manufacturer, who must develop new technologies to reduce static power in new devices. The dynamic power consumption of an FPGA can change significantly depending on the design which the device is configured to implement, leaving the task of optimising designs to reduce power consumption in the hands of the hardware designers.

In this paper we propose high-level techniques for estimating the dynamic power consumed in the arithmetic components of an FPGA. High-level power estimation tools such as this can be used before or during synthesis to allow high-level design changes which optimize power consumption. Previous work [1] presents an approach for word-level modelling of signal activities, which corresponds to the activities of wires connecting arithmetic components in a system, *i.e.* routing power. The work in this paper focuses on using the word-level statistics of the inputs to an arithmetic component to estimate logic power consumption within that component. A data-flow graph is used to describe all the components in the system and the connections between them. This technique gives very fast logic power consumption estimates compared with low-level techniques which need to either simulate or model the internal signals within an arithmetic component, which is computationally expensive due to the number of internal signals involved.

The main contributions of the work contained in this paper can be summarised as follows:

- an experimental analysis of which statistical parameters of signals have significant effect on the power consumption of the arithmetic components they drive, and
- a fast dynamic power consumption estimation technique based on this analysis which relies solely on high-level characteristics of the signals in the design.

The rest of this paper is organised as follows. Section 2 provides a review of current research work being carried out in hardware power modelling. In Section 3 a method for generating test signals with chosen characteristics which are used as inputs when doing power analysis is presented. The signal statistics which most strongly affect power consumption are identified in Section 4, and are used to develop the power models described in this section. Finally in Section 5 results from an evaluation of the developed models are presented.

The authors would like to acknowledge the support of Xilinx, Celoxica and the EPSRC under grant number EP/C512596/1.

2. BACKGROUND

In [2] the transition density technique was used to estimate dynamic power consumption in FPGAs. This technique requires designs to be fully placed and routed so that the capacitances of each signal in the design are known, and can be used to scale the activity rates of each signal appropriately to estimate the total power consumption of the system. In [3] a transition density based method is presented for estimating the dynamic power consumption of a design which has been mapped to a particular device; this means that the number of Look-Up Tables (LUTs) and their configuration is known, but their placement and the routing between them is still undetermined.

Using these techniques to automatically optimize the power consumption of a system during high-level synthesis would mean large computation times would be needed as each optimization made requires the system to be synthesized, then placed and routed before a new power consumption estimate can be made. Instead it would be preferable to use simpler models to approximate the power consumed in a design described using a high-level description.

The work in [4] presents a technique for modelling the dynamic power consumed in a 'block' within an FPGA, where a block might be a simple circuit such as an adder, or a more complex component such as an ALU or an FIR filter. The technique accounts for different bit-level statistics within the inputs to a block by partitioning these into several subsets according to the spatial correlation between each pair of signals, where within each subset all the signals have a similar level of spatial correlation. Each subset is then represented as a variable in an equation which estimates the average power consumed by the block, where each variable in the equation is multiplied by a coefficient whose value is determined through extensive simulation. This approach requires a different model for identical operations with different word-lengths however, whereas in the proposed approach the word-length of the arithmetic component is used as a variable in a single power model for that component.

The approach described in this paper uses word-level statistics to model the signals at the inputs to each arithmetic component in a system. Word-level statistics have previously been used by several groups as a means of estimating bit-level transition rates in a signal, and were first studied in the Dual Bit Type (DBT) method [1]. The DBT method and its variants recognised that signals in the data-paths of data-intensive systems, such as DSP circuits, are not well represented by temporally-uncorrelated noise signals, which have been traditionally used in power estimation techniques, but instead are well approximated by arbitrarily-correlated Gaussian signals. By studying the bit-level activities of typical signals, the authors identified that the LSBs in a signal are uncorrelated with each other and display activity rates similar to white noise, but that the MSBs are spatially correlated and have activity rates which can be related to the autocorrelation of the signal. The authors provide equations which estimate how many of the LSB bits exhibit white noise behaviour and how many MSB bits are correlated, and what their activities are; the activities in the region between these two can be approximated by interpolating between the LSB and MSB regions.

Though these techniques would work well for estimating the activity rates of the signals in the routing between components in an FPGA (assuming these are glitch-free), using bit-level statistics to estimate the activities of signals such as carries within arithmetic components, *i.e.* logic power, is a more complex problem due to fact that the MSBs of the input signals have spatial and temporal correlation. In the proposed approach the power consumed within FPGA arithmetic components is estimated directly from word-level statistics, hence avoiding the complexities of working with bit-level statistics.

3. SIGNAL MODELLING

3.1. Signal Representation

The switching activity in a synchronous digital circuit is entirely defined by the statistics of present and immediately-past signal values. Thus for a two-input arithmetic component with glitch-free inputs $x(n)$ and $y(n)$ at cycle n, the power consumption is entirely defined by the joint probability density function (PDF) $p(x(n), x(n-1), y(n), y(n-1))$. Moreover, if the signals are statistically stationary, the dependence of the joint-pdf on the absolute time index n may be dropped, resulting in $p(x_0, x_1, y_0, y_1)$.

In the DBT work [1] the authors demonstrate that simplifying the PDF of real-world signals to a zero-mean Gaussian distributions has minimal effect on the power-consumption observed. The joint-PDF of a zero-mean multi-variate Gaussian distribution is given by (1), where $\mathbf{x} = [x_1, \ldots, x_n]$ denotes the signal vector, and $\mathbf{C}$ is an $n \times n$ symmetric matrix with $[\mathbf{C}]_{ij} = E\{x_i x_j\}$.

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{n/2} \det^{1/2}(\mathbf{C})} \exp\left[-\frac{1}{2}\mathbf{x}^T \mathbf{C}^{-1} \mathbf{x}\right]. \qquad (1)$$

Let us define the cross-correlation function of two statistically stationary signals $p(n)$ and $q(n)$ by $r_{pq\tau} = E\{p(n)q(n-\tau)\}$. Then, for the particular case of a two-real-input arithmetic component with statistically stationary inputs $x(n)$ and $y(n)$, we obtain (2), where $\mathbf{C}$ is given by (3).

$$\mathbf{C} = \begin{bmatrix} r_{xx0} & r_{xx1} & r_{xy0} & r_{xy1} \\ r_{xx1} & r_{xx0} & r_{yx1} & r_{xy0} \\ r_{xy0} & r_{yx1} & r_{yy0} & r_{yy1} \\ r_{xy1} & r_{xy0} & r_{yy1} & r_{yy0} \end{bmatrix}. \qquad (3)$$

$$p(x_0, x_1, y_0, y_1) = \frac{1}{4\pi^2 \det^{1/2}(\mathbf{C})} \exp\left[-\frac{1}{2}[x_0\ x_1\ y_0\ y_1]\mathbf{C}^{-1}[x_0\ x_1\ y_0\ y_1]^T\right]. \quad (2)$$

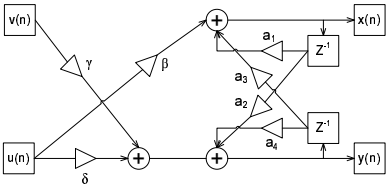

Fig. 1. Signal flow graph representation of signal generator

The important point to note is that all the information required for a complete characterisation of the power consumption of this component is contained in just six statistical parameters: r_{xx0}, the variance of signal x; r_{xx1}, the auto-correlation of signal x with unit time-lag; r_{xy0}, the cross-correlation of the two signals; r_{xy1}, the cross-correlation of the signals with unit time-lag in y; r_{yx1}, the cross-correlation of the signals with unit time-lag in x; and r_{yy0}, the variance of signal y, together with the word-length and scaling of each signal. Moreover, if each signal has been scaled (*i.e.* its binary point has been selected) appropriately, in proportion to the standard deviation of the signal, then we lose no generality by setting $r_{xx0} = 1$, resulting in a total of five statistical parameters and two word-length parameters.

3.2. Signal Generation

In order to investigate the effects of different input signal word-level statistics on the dynamic power consumption of arithmetic components it is necessary to develop a system for generating signals with chosen values for each of the signal characteristics of interest. In this section the motivation for selecting the statistics that were investigated will be given, followed by a description of the system used to generate the two input signals with the required statistics for the arithmetic component under test.

The system shown in Figure 1 was used to generate the signals $x(n)$ and $y(n)$ from two spatially and temporally uncorrelated zero-mean Gaussian signals $u(n)$ and $v(n)$, each having a unit variance, as produced by a standard software random number generator. An analysis of this results in (4-9), which show how to relate the variables: r_{xx0}, r_{yy0}, r_{xx1}, r_{yy1}, r_{xy0}, r_{xy1} and r_{yx1}, to the scaling coefficients in the system, allowing these to be selected appropriately to generate $x(n)$ and $y(n)$ with the required characteristics.

$$r_{xx0} = r_{pp0} + a_1 r_{xx1} + a_3 r_{xy1} \quad (4)$$
$$r_{yy0} = r_{qq0} + a_2 r_{yx1} + a_4 r_{yy1} \quad (5)$$
$$r_{xy0} = r_{pq0} + a_1 r_{yx1} + a_3 r_{yy1} \quad (6)$$

$$\begin{bmatrix} r_{xx1} \\ r_{xy1} \end{bmatrix} = \begin{bmatrix} r_{xx0} & r_{xy0} \\ r_{xy0} & r_{yy0} \end{bmatrix} \begin{bmatrix} a_1 \\ a_3 \end{bmatrix} \quad (7)$$
$$\begin{bmatrix} r_{yx1} \\ r_{yy1} \end{bmatrix} = \begin{bmatrix} r_{xx0} & r_{xy0} \\ r_{xy0} & r_{yy0} \end{bmatrix} \begin{bmatrix} a_2 \\ a_4 \end{bmatrix} \quad (8)$$
$$r_{pp0} = \beta^2, r_{qq0} = \gamma^2 + \delta^2, r_{pq0} = \beta\delta \quad (9)$$

The inputs to the signal generation system are the variances of the two signals r_{xx0} and r_{yy0}, together with correlation coefficients ρ_{xx1}, ρ_{yy1}, ρ_{xy0}, ρ_{yx1} and ρ_{yx1} defined as (10-14) and the output is a pair of Gaussian signals with the desired properties.

$$\rho_{xx1} = \frac{r_{xx1}}{r_{xx0}} \quad (10)$$
$$\rho_{yy1} = \frac{r_{yy1}}{r_{yy0}} \quad (11)$$
$$\rho_{xy0} = \frac{r_{xy0}}{\sqrt{r_{xx0} r_{yy0}}} \quad (12)$$
$$\rho_{xy1} = \frac{r_{xy1}}{\sqrt{r_{xx0} r_{yy0}}} \quad (13)$$
$$\rho_{yx1} = \frac{r_{yx1}}{\sqrt{r_{xx0} r_{yy0}}} \quad (14)$$

4. IMPORTANT FACTORS

In the preceding section, five correlation parameters were identified when considering two input arithmetic operators, each of which could affect the dynamic power consumption. In this section the effect of variations in each of these parameters on the dynamic power consumption is analyzed empirically.

From the graph in Figure 2 it can be seen that when the cross-correlation coefficient ρ_{xy0} is varied between -0.8 and $+0.8$ the variation in dynamic power consumption for a 16-bit multiplier is always less than 10%. On the other hand the variation in the value of the auto-correlation parameter ρ_{yy0} between -0.9 and $+0.9$ can cause a variation of up to 25% in the dynamic power consumption.

Similar results to those in Figure 2 were obtained for the variation in dynamic power consumption for a 16-bit adder. In this case when ρ_{yy1} is varied between -0.9 and $+0.9$, the maximum variation in the dynamic power consumption in the adder is about 6%, whilst when ρ_{xy0} is varied between -0.8 and $+0.8$, the variation is less than 1%.

The significance of these results is that dynamic power consumption is affected to a greater extent by auto-correlation than cross-correlation in these arithmetic components. Hence it is possible to ignore the cross-correlation values when deriving power models.

Other results which measured the effect of varying the word-length of a component on its logic power consumption indicated that a linear relationship exists between the two for adders, and that a quadratic relationship exists between the two for multipliers. These findings suggested the use

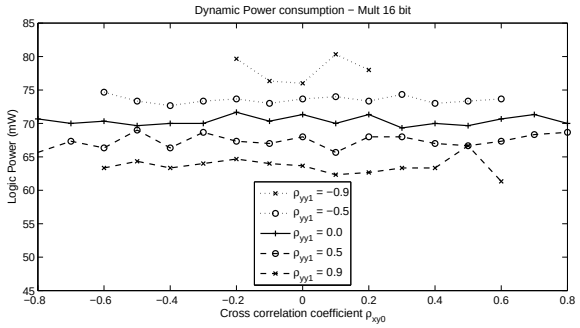

Fig. 2. Variation in dynamic power consumption obtained from Xilinx XPower, when the cross-correlation ρ_{xy0} and auto-correlation ρ_{yy0} are varied, other signal statistics are held constant, $r_{xx0} = 0.5, r_{yy0} = 0.01, \rho_{xx1} = 0.5$.

Table 1. The relationship between word-length (W) and dynamic power consumption (P) for adders and multipliers.

Component	Dynamic Power
Adder	$P = C_0W + C_1$
Multiplier	$P = C_0W^2 + C_1$

of simple equations such as those in Table 1 to estimate the power consumed within these components. In these equations C_0 and C_1 are determined by using the statistics of the signals driving the component to select their values from pre-made tables of coefficients. These tables are built by taking a series of measurements of the power consumed in the arithmetic components as the input signal statistics to each component are varied. The resulting tables use the variances and auto-correlations of each input to an arithmetic component to select the appropriate coefficients C_0 and C_1. When the measured signal statistics fall between available values in the tables linear interpolation is used to approximate C_0 and C_1. These tables are available for reference at: `http://infoeng.ee.ic.ac.uk/~gac1/Power`.

5. RESULTS AND CONCLUSION

To demonstrate the accuracy of the proposed method we consider two system implementations, a sum of products and a polynomial evaluation circuit. For both these examples, the power consumption is estimated with the proposed method, and the result is compared to the low level power estimation done by Xilinx XPower.

In Figure 3 the estimated and measured values for the logic power consumption for the sum of products example are considered. Each graph point represents a fully placed and routed design obtained by varying signal parameters. For the majority of cases the estimate provided by the proposed technique is accurate to within 10% of the measured value.

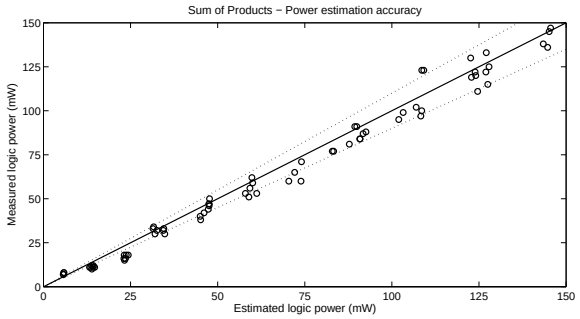

Fig. 3. Measured power consumption versus estimated power consumption. The solid line represents the case when both values are equal, the dotted lines on either side of it represent $\pm 10\%$ of the ideal value.

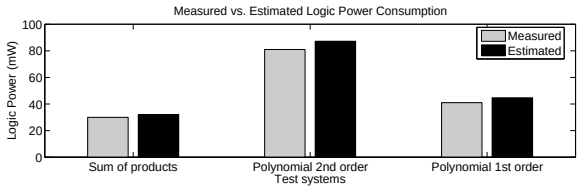

Fig. 4. Comparison of measured versus the estimated power consumption values for the median estimation error.

For the second example a 1st and 2nd order polynomial evaluation circuit were implemented. Figure 4 compares the estimated and measured logic power consumption values, showing the median difference between the two values. For the cases shown the maximum median difference between the estimated and measured values is less than 10%.

In conclusion this paper presents a high-level power estimation technique which uses empirically derived power models to estimate logic power consumption to within 10% of the low-level power estimate. Future work which has been identified is the development of techniques for further power model order reduction and the integration of the method within an arithmetic optimization system to perform power based optimisation.

6. REFERENCES

[1] P. Landman and J. Rabaey, "Architectural power analysis: The dual bit type method," *IEEE Trans. on VLSI Systems*, vol. 3, no. 2, pp. 173–187, 1995.

[2] K. K. W. Poon, A. Yan, and S. J. E. Wilton, "A flexible power model for FPGAs," in *FPL*, M. Glesner, P. Zipf, and M. Renovell, Eds. Springer, 2002, pp. 312–321.

[3] J. Anderson and F. Najm, "Power estimation techniques for FPGAs," *IEEE Trans. on VLSI Systems*, vol. 12, no. 10, pp. 1015–1027, 2004.

[4] L. Shang and N. K. Jha, "High-level power modeling of CPLDs and FPGAs," in *Proc. of the Int. Conf. on Comp. Design*. IEEE Computer Society, 2001, pp. 46–53.