Fast word-level power models for synthesis of FPGA-based arithmetic

Jonathan A. Clarke, Altaf Abdul Gaffar, George A. Constantinides, Peter Y. K. Cheung

Department of EEE, Imperial College London

Email: {jonathan.a.clarke, altaf.gaffar, g.constantinides, p.cheung}@imperial.ac.uk

Abstract— This paper presents power models for multiplication and addition components on FPGAs which can be used at a high-level design description stage to estimate their logic and intra-component routing power consumption. The models presented are parameterized by the word-length of the component and the word-level statistics of its input signals. A key feature of these power models is the ability to handle both zero mean and non-zero mean signals. A method for measuring intra-component routing power consumption is presented, enabling the power models to account for both logic and routing power in components. The resulting models are equations which can be used to estimate the power consumed in an arithmetic component in a fraction of a second at the pre-placement stage of the design flow. The models have a mean relative error of 7.2% compared to bit-level power simulation of the placed-and-routed design.

I. INTRODUCTION

Power consumption of Field-Programmable Gate Arrays (FPGAs) is rapidly increasing due to more tightly packed logic and higher clock speeds. This is causing costs to rise for larger devices as they now require more expensive packaging, heatsinks, or fans, and prevents smaller devices being used in portable applications where short battery life and little opportunity for heat dissipation are dominant factors. Digital circuits consume power when transistors within the circuit switch as computation is performed (dynamic power), and also when no switching is taking place, due to leakage currents in the device (static power). It is FPGA vendors who choose the number and layout of the transistors on a device, and hence they shoulder the task of optimizing the static power of future architectures. Hardware designers, however, can influence the dynamic power consumed on an FPGA through decisions in the design process, hence it is essential to provide tools which allow the comparison of the effects of different design changes on dynamic power consumption at various stages of the design flow.

Previous work by the authors presented a technique for estimating the logic power consumption in arithmetic components on an FPGA where the inputs to a component were modelled as zero mean Gaussian signals [1]. This work extends our previous work by relaxing the zero-mean constraint, and by introducing a technique for estimating the power consumed within the routing of multipliers, whilst using only a small number of parameters in the models developed. The resulting models are very fast compared to low-level power estimation techniques, but still account for glitching within components through characterization of the equations used.

The contributions of the work in this paper can be summarized as follows:

- an improved power model which allows for non-zero mean Gaussian input signals, allowing a greater range of signals to be modelled accurately, and,
- establishing a link between the logic power consumed by a component and the routing power consumed within the component, allowing intra-component routing power to be estimated.

II. BACKGROUND

The average power consumed in a particular capacitive element within a device can be calculated using (1), where: α is the average number of logic transitions that the signal on the element makes during one clock cycle, known as the switching activity of the signal, C is the capacitance of the element, V_{dd} is the power supply voltage, and f_{clk} is the clock frequency.

$$P = \frac{1}{2} \cdot \alpha \cdot C \cdot V_{dd}^2 \cdot f_{clk} \tag{1}$$

By using this equation on every capacitive component in an FPGA the dynamic power consumed on the device can be estimated. However this requires knowledge of the average activity of each signal in the circuit and the capacitance which that signal switches. These are only available once a design has been synthesized, placed and routed (to obtain the capacitance values), and then simulated at a low level (to obtain the activities of each signal) with test vectors which are expected to be typical inputs to the design. This process is very computationally intensive and so is unsuitable for use in power-conscious high-level synthesis tools where many iterations of power estimates followed by design modifications are made.

In [2] the transition density technique is used to avoid low-level simulation of an FPGA design. Switching density estimates are instead propagated through the circuit, reducing the computational cost of estimating switching activity rates for each node in the circuit, though it is still necessary to place and route the circuit. In [3] the transition density technique is also used, but the authors describe a technique for estimating the capacitance of routing wires between components without needing to perform place and route. Even so the synthesis and technology mapping stages which must still be performed

are too computationally intensive for use within a high-level synthesis loop.

Work in [4] and also in [5] uses characterized equations of functional components and mostly closely resembles the work presented here. In [4] a power model is described which is characterized using bit-level activity characteristics of the inputs and outputs to a component. The model is not parameterized by word-length however, meaning that separate models are created for identical operations with different word-lengths. In [5] a model is presented which uses bit-level characteristics of the inputs to a component, as well as its word-length, to calculate the power consumed in that component, however the work is unable to distinguish between signals with similar bit-level activities averaged across the signal, but with very different statistical distributions, which may cause the component driven by the signal to consume different amounts of power than predicted.

III. NON-ZERO MEAN GAUSSIAN SIGNALS

In [6] the authors demonstrate that it is possible to abstract away from the bit-level activities of a signal by grouping adjacent bits into regions which exhibit similar behaviour. Three different groups of bit-level activity were identified in a zero-mean Gaussian signal which are, in order from LSB to MSB: spatially and temporally uncorrelated LSB bits, a region with increasing spatial correlation towards the MSB, and the sign bit(s) at the MSB end of the signal. The authors demonstrate that it is possible to estimate the breakpoints between these groups of activity using only the lag-1 autocorrelation and standard deviation for zero-mean Gaussian signals.

For a statistically stationary Gaussian signal with non-zero mean, the signal varies with a Gaussian distribution around some constant mean value. This is reflected in the bit-level activity of the signal by a number of constant-valued bits at the MSB end, referred to here as the *mean bits* in a Gaussian signal, which are followed by the two lower groups of activity from [6]. The number of mean bits N_μ in a Gaussian signal with mean μ and standard deviation σ can be approximated using (2). Here $\log_2(|\mu| + 3\sigma)$ represents the bit-position relative to the binary point of the largest bit needed to represent the signal, and $\log_2(3\sigma)$, taken from [6], represents the bit-position relative to the binary point of the largest bit needed to represent the Gaussian variation about the mean of the signal (an uncorrelated Gaussian signal lies in the $\pm 3\sigma$ range 99.7% of the time). This simplifies as shown in (2), where it can be seen that the larger the ratio $\frac{|\mu|}{3\sigma}$, the more mean bits there are in the signal. Mean bits in a signal do not make any transitions, and so strongly affect the power consumed in arithmetic components driven by such signals. As such the power models described in Section IV use the number of mean bits in the input signals to help determine the power consumed in arithmetic components.

$$N_\mu = \log_2(|\mu| + 3\sigma) - \log_2(3\sigma) = \log_2\left(\frac{|\mu|}{3\sigma} + 1\right) \quad (2)$$

IV. POWER MODEL CONSTRUCTION

This section describes the power models developed which can be used at the design description stage to estimate the power consumed in an arithmetic component, given information about the component's word-length and some word-level signal statistics regarding the signals which pass through it. Currently the tools developed work with Xilinx System Generator [7], which allows arithmetic intensive FPGA designs such as DSP algorithms to be entered using a block diagram interface. Our tool monitors the signals at the inputs of each arithmetic component in the design during a high-level floating-point simulation, and measures the word-level signal statistics which are used in the power model for each signal. These are then used along with the word-length information in the design to estimate the power consumed in each component. This process can be done entirely without synthesizing or mapping the design to a target FPGA. The models described are intended to estimate the power consumed in adder and multiplier components implemented on Xilinx Virtex2 [7] devices, though the parameters selected for use in our power models are general enough to be used on other devices where ripple carry adders tree-based multipliers are used. Table I shows the equations for the power models developed, which are explained in detail in the following subsections.

A. Adders

Our previous work [1] indicated that the logic power consumed in an adder increases linearly with the word-length of the component, and can be estimated using $C_0(\sigma_y, \rho_{xx1}, \rho_{yy1})W_s + C_1(\sigma_y, \rho_{xx1}, \rho_{yy1})$, where σ_y is the standard deviation of the smaller of the two inputs, scaled according to the standard deviation of the larger signal, ρ_{xx1}, ρ_{yy1} are the lag-1 autocorrelation coefficients of the signals, C_0, C_1 are functions mapping these parameters into coefficient values, and W_s is the word-length of the component which captures the linear dependance of power on word-length. If both input signals to an adder contain a number of mean bits then a corresponding decrease in the power consumed in the component can be expected. An extension is proposed here to account for mean bits in the signal x with larger standard deviation, by modelling the power consumed in an adder of word-length W, and a number of mean bits N_{μ_x} in the input signal x, as equivalent to the power consumed in a smaller adder with word-length $W_s = W - N_{\mu_x}$, as shown in Table I, where N_{μ_x} is given by (2).

B. Multipliers

Multipliers implemented in Look-Up Tables (LUTs) on the Xilinx Virtex2 FPGA are constructed as a set of partial product generators at the top level followed by a summation tree to sum the partial products and generate the multiplication result. The work in [1] accounts for the effects of the autocorrelations ρ_{xx1}, ρ_{yy1} of the input signals using $C_0(\rho_{xx1}, \rho_{yy1})W^2 + C_1(\rho_{xx1}, \rho_{yy1})$, where C_0, C_1 are functions mapping the autocorrelations to coefficient values, and W^2 accounts for the quadratic dependence of power on word-length in multipliers.

TABLE I

THE RELATIONSHIP BETWEEN WORD-LENGTH (W) AND DYNAMIC POWER CONSUMPTION (P) FOR ADDERS AND MULTIPLIERS.

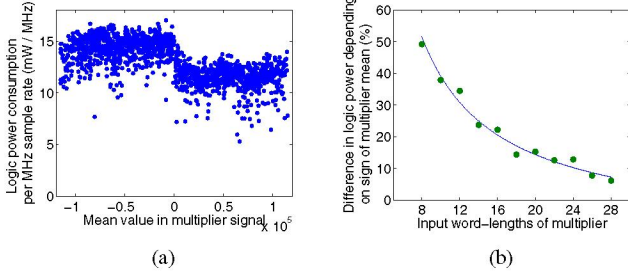| Component	Dynamic Power
Adder Logic	$P = C_0(\sigma_y, \rho_{xx1}, \rho_{yy1})W_s + C_1(\sigma_y, \rho_{xx1}, \rho_{yy1})$, where $W_s = W - N_{\mu_x}$
Multiplier Logic and Internal Routing	$P = \dfrac{[C_0(\rho_{xx1}, \rho_{yy1})W^2 + C_1(\rho_{xx1}, \rho_{yy1})] \cdot [Sign_x(\gamma_1 W^{-1} + \gamma_0) + 1]}{[N_{\mu_y}(\beta_1 W^{-1} + \beta_0) + 1]}$

(a)   (b)

Fig. 1. (a) The effects of the sign of the mean of the multiplier signal on the logic power consumption in an 18×18-bit multiplier component. 1500 tests are shown with independently chosen, uniformly distributed random input statistics. (b) How word-length affects the percentage difference between the power consumed when the mean value in the multiplier signal is positive or negative.
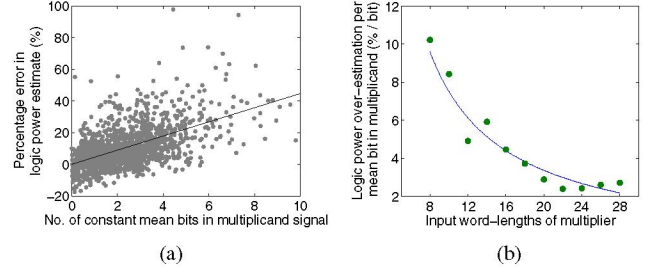

(a)   (b)

Fig. 2. (a) The error in the logic power estimate made using the autocorrelations of the inputs and sign of the multiplier signal mean, and its relationship to the number of mean bits in the multiplicand, in a 16×16-bit multiplier. 1500 tests are shown with independently chosen, uniformly distributed random input statistics. (b) The over-estimation factor (*i.e.* the gradient of the graph in Figure 2(a)) per mean bit in the multiplicand, across different word-length multipliers.

When considering non-zero mean signals, it was noted that the MSB of the multiplier signal has an approximately 50% higher fanout than all other input bits to a multiplier component, due to sign extension of this bit in each partial product pair which is summed at the top level of the component. Approximately two thirds of the logic driven by this signal within the multiplier uses it as a gating signal, thus when the multiplier signal is positive and its MSB bit is 0, transitions in the other signals driving this logic are not propagated through the rest of multiplier, causing lower power consumption than when the multiplier signal is negative and transitions are propagated through the component. As a result, the sign of the multiplier signal can have significant effect on power consumption, as shown in Figure 1(a). However this effect decreases as the word-length of the multiplier component increases, as its power consumption grows quadratically with word-length, whilst the amount of logic driven by the MSB bit in the multiplier signal (and hence the power consumed in this logic) only increases linearly with word-length. As such an inversely proportional relationship between the word-length of a multiplier and the change in power consumption due to the sign of the multiplier signal exists, as can be seen in Figure 1(b). This is reflected by the introduction of the term $[Sign_x(\gamma_1 W^{-1} + \gamma_0) + 1]$ into the multiplier logic power model in Table I, where $Sign_x$ is a binary variable equal to 0 iff the mean of the multiplier signal x is positive, while γ_1, γ_0 are architecture specific coefficients.

The other term introduced to the power model in Table I accounts for the number of mean bits in the multiplicand. Figure 2(a) shows that once the autocorrelation in the input signals and the sign of the multiplier signal have been accounted for, the logic power consumed by a multiplier decreases

approximately linearly with the number of mean bits in the multiplicand (hence a linear increase in over-estimation is seen in the data). Figure 2(b) shows that the rate of this decrease across multipliers of word-lengths 8 to 28 bits is inversely related to word-length. Again, this is due to the amount of logic affected by one mean bit in the multiplicand increasing linearly with word-length, while the overall power consumption of the component increases quadratically. The term $[N_{\mu_y}(\beta_1 W^{-1} + \beta_0) + 1]$ in Table I accounts for these effects.

C. Routing power within multipliers

The dynamic power consumed within the routing of a multiplier implemented in slices is directly related to the activity of the input signals of the multiplier and the power consumed in the logic of the multiplier. To establish the nature of this relationship a test system was developed which would allow for the separate measurement of intra-routing power consumption (*i.e.* within multipliers) and inter-routing power consumption (between components). The diagram in Figure 3(a) illustrates the test circuit used, where the input and output pins of the FPGA on which the multiplier is to be implemented have been constrained to lie in bit order at particular positions on the device. This forces the multiplier to be consistently placed in the same location independently of its word-length, as a result each multiplier will have approximately the same length of inter-component routing between its terminals and the pins of the chip.

The largest multiplier used was 28×28 bits; when smaller multipliers were placed on the chip those signals not used by the multiplier were still routed across the device, from the
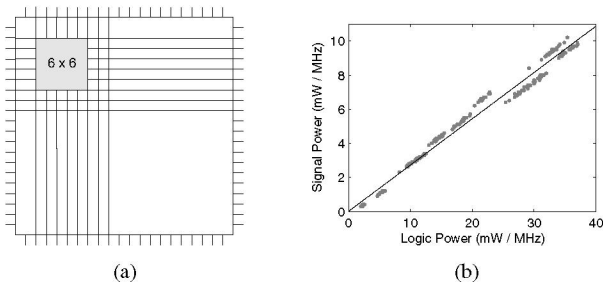
Fig. 3. (a) The test setup used to measure intra-component routing power. The inputs to the chip are constrained to lie in the same position for different word-length multipliers, and are routed across the chip even if not required by smaller multiplications. (b) The relationship between logic and internal routing power for multipliers. Note that the results have been normalized so that the base case has no routing power consumption.

TABLE II
ACCURACY RESULTS FOR MULTIPLIER MODEL W.R.T. XPOWER.

Estimation technique	Mean Relative Error
Work in [1]	32.7%
[1] + sign in multiplier	11.5%
[1] + sign in multiplier + mean bits in multiplicand	7.2%

unused input pins to the unused output pins. This allowed the routing power of a base case to be measured where only routing, and no multiplier, was placed on the chip. By subtracting the routing power consumed in the base case, measured using XPower [7], from the routing power consumed by the other circuits it was possible to obtain an estimate of the power consumed within the internal routing of the multipliers tested. Note that the outputs of each multiplier were registered to avoid glitch propagation through the inter-component routing to the chip pins.

Figure 3(b) shows a scatter plot of the logic and routing power consumed by multipliers for varying input word-lengths from 8 to 28 bits and varying input signal statistics. The routing power consumption in the base case was observed to be negligibly small (< 1 mW/MHz).

From these results it can be observed that when the outputs of components in a design are registered, preventing the pro-pagation of signal glitches, the inter-component routing power is small compared to the intra-component routing power. The results can be fitted to a linear curve with a mean relative error of 8.4%, as shown in Figure 3(b). Based on this linear curve a scaling factor of $(1 + \alpha)$, where α is the gradient of the curve, can be applied to the values of C_0, C_1 from [1], shown in the multiplier equation in Table I, to allow for both the logic and internal routing power consumed in a multiplier to be estimated.

V. RESULTS

The Mean Relative Error (MRE), used in this section to measure the accuracy of the models developed, is calculated using (3), where P_{e_i}, P_{m_i} are the ith power estimates and measurements, respectively.

$$MRE = \frac{1}{N} \sum_i^N \frac{|P_{e_i} - P_{m_i}|}{P_{m_i}} \quad (3)$$

The tests described in this section were performed using pairs of input signals with independently chosen, uniformly distributed random values for the mean, autocorrelation and standard deviation statistics for each signal. For the adder 1000 tests were performed on a 16-bit component. The MRE error

of the estimates made by the model in Table I was 4.98% compared to the estimates made by XPower.

The multiplier logic power model in Table I was charac-terized for each of the word-lengths 8, 10, 12,... 28, with 1500 tests for each word-length. Table II gives a breakdown of the accuracy of the technique in [1] and the modifications described in Section IV-B, when applied to the total of 16500 tests made across 11 different word-lengths. The MRE for the multiplier logic power model developed in this work is 7.2%.

VI. CONCLUSION

This paper has presented models to estimate the logic power consumption in adders and multipliers which are parameteri-zed using word-level statistics of the inputs to the compo-nents. The work presented has investigated the use of input signals with non-zero mean values and their effects on power consumption. The resulting power models have a MRE of at worst 7.2%, and can be used at the design description level to optimize a system before or during synthesis. A method for investigating the relationship between the logic power and internal routing power consumed within a multiplier was also presented which showed that an approximately constant relationship between the two exists, with a MRE of 8.27%. Future work will integrate these tools into a synthesis package allowing designs to be optimized for power at a high level.

REFERENCES

[1] J. A. Clarke, A. A. Gaffar, and G. A. Constantinides, "Parameterized logic power consumption models for FPGA-based arithmetic," in *FPL*, T. Rissa, S. Wilton, and P. Leong, Eds. IEEE, 2005, pp. 626–629.
[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. H. Anderson and F. N. 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.
[5] T. Jiang, X. Tang, and P. Banerjee, "Macro-models for high level area and power estimation on FPGAs," in *GLSVLSI*. ACM Press, 2004, pp. 162–165.
[6] P. E. Landman and J. M. Rabaey, "Architectural power analysis: The dual bit type method," *IEEE Trans. on VLSI Systems*, vol. 3, no. 2, pp. 173–187, 1995.
[7] System Generator, XPower, Virtex2. Xilinx. [Online]. Available: http://www.xilinx.com/