Modeling of glitch effects in FPGA based arithmetic circuits

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

Department of Electrical and Electronic Engineering, Imperial College London
London SW7 2AZ, United Kingdom
{altaf.gaffar,jonathan.a.clarke,g.constantinides}@imperial.ac.uk

Abstract— **One of the requirements when using high-level power optimization techniques is the ability to estimate circuit power consumption quickly. Bit-level estimation techniques which take into account the glitch activity in a circuit take too long to provide power estimates. In this paper we present a novel method which can provide fast estimates for the logic and intra-routing power consumption in digital circuits whilst taking into account the glitch activity but relying purely on the word-level statistics of the signals. The proposed method models the propagation of glitch activity in signals through the arithmetic components in circuits, and using this information estimates the logic and intra-routing power consumption. For arithmetic circuits implemented on FPGAs we demonstrate that previous macro-model based power estimation techniques consistently under-estimate the power consumption by up to 20 times, whilst this work can provide estimates to within a mean relative error of 30% compared to low-level power estimation.**

I. Introduction

In recent years due to increasing manufacturing costs and lack of flexibility associated with Application Specific Circuits (ASICs), Field Programmable Gate Arrays (FPGAs) have become increasingly popular for implementing custom hardware. One of the concerns of FPGA application designers is the minimization of the power consumption on the FPGA.

The power consumption in FPGA circuits can be categorized as being either dynamic power or static power. The static power consumption refers to the power consumed in the circuit when the circuit is powered on but the signals in the circuit have zero activity. The static power consumption is dependent upon the fabrication technology, circuit topology and transistor count.

The dynamic power consumption on the other hand refers to the power consumed due to the transition activity in the signals inside the circuit. The average dynamic power P consumed in a particular capacitive element within a device can be calculated using (1), where: $n(K)$ is the number of transitions in the element during K clock cycles, C is the capacitance of the element and V_{dd} is the power supply voltage.

$$P = \frac{1}{2} C V_{dd}^2 \left[\lim_{K \to \infty} \frac{n(K)}{K} \right] \quad (1)$$

P is a constant value, and the above limit holds when the assumption is made that the number of transitions within a given period of time is an ergodic stochastic process. By using this equation on every capacitive component in a design the dynamic power consumed by 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 is the approach taken by the XPower power estimation tool available from Xilinx [1] for estimating power consumption on FPGAs.

Simulating the operation of a device at a low level gives the best possible estimate of the activities of each signal, and coupled with accurate knowledge of the capacitance of internal wires and transistors of the device allows for the most accurate dynamic power consumption estimates to be made. This accuracy comes at the cost of very high computational complexity however, as low-level simulation attempts to mimic the switching activity of every transistor in a device to best approximate the effects of superfluous switching activity, *i.e.* glitching. Glitches are transitions that appear on signals driven by combinational logic, which occur in the period of time between when a new set of inputs arrive at the logic, and when the logic settles on a final output value for the new set of inputs. These glitches are (1) due to the logic and routing delay of the signals through the circuit causing them to appear at slightly different times at the input of the logic, (2) due to some of the input signals carrying glitches propagated from previous logic, or (3) due to both these reasons.

This paper concerns itself with the modeling of glitches in arithmetic components used in signal processing applications. These components include adders, subtracters and multipliers. Since these arithmetic components are high-level components comprising of many low-level logic elements connected in a fixed, regular topology, we propose the use of a high-level glitch modeling technique to estimate the power consumption.

These high-level techniques, although less accurate, do not require the compilation of designs to estimate power unlike low-level techniques, and therefore provide quick estimates for power. This makes these techniques attractive for use with optimization schemes which operate at the word-level. Although the results in this paper are specific to FPGA based circuits, this work can also extended to ASICs.

The key contributions presented by this work include.

- a novel technique to model and propagate glitch activity in arithmetic circuits using the word-level statistics of the signals.
- use of glitch activity based power models to reduce the over 20 times underestimation seen in previous macro-models [2]–[4], by providing power estimates which within a mean relative error of 30% compared to low-level power estimation.

The paper is organized as follows. Section II provides references to related work. Section III introduces our proposed model for glitch modeling. Section IV uses this model to consider the propagation effects of glitches through a circuit, and Section V considers the effect of glitches on circuit power consumption. Section VI evaluates the proposed power estimation models by comparing these with other previous models.

II. Background

When high-level optimization techniques targeting power consumption are applied to circuits, it is essential to be able to estimate the power consumption in a fast manner during the optimization process. This necessity has given rise to the development of *macro-models* to provide fast estimates for the power consumption in the functional units of the circuit. In this section we compare this work

0-7803-9729-0/06/$20.00 © 2006 IEEE

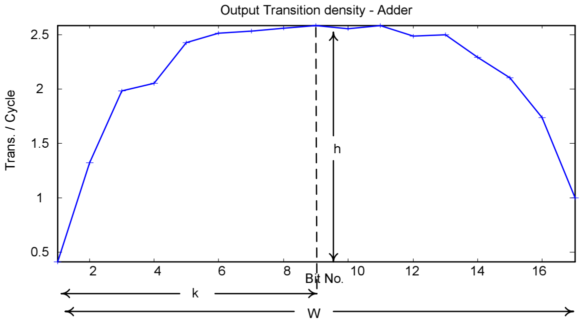

Fig. 1. Transition density variation for the output bus-signal from an adder with glitch free inputs.

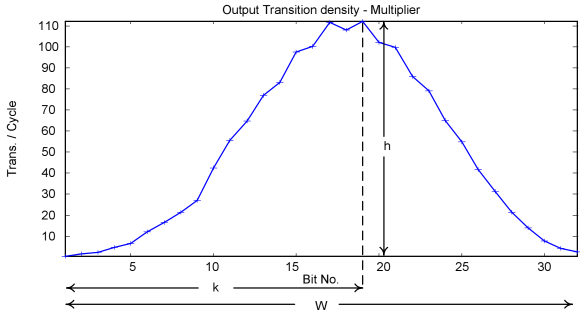

Fig. 2. Transition density variation for the output bus-signal from a multiplier with glitch free inputs.

with previous macro-models [2]–[4] and other power estimation techniques [5], [6] for FPGAs.

In [7] a technique know as the Dual Bit Type (DBT) is presented which attempts to model the bit-level activity in signals in terms of the word-level statistics. Later work in [2]–[4] focuses on the use of macro-models to model the power in circuits implemented on FPGAs. The macro-models presented in [2]–[4] assume that the inputs to the components are glitch-free, *i.e.* each input bit makes at most one transition per clock cycle. To ensure that the signals are glitch-free the outputs of components need to be fed into registers before they drive other components.

For many circuits it is necessary to feed the output of one component directly to another without the use of registers, due to latency constraints. In these cases glitches generated by one component would propagate to other components in the circuit, hence the assumption of that the signals are glitch-free made in [2]–[4] would cause these techniques to grossly underestimate the power consumed in components whose inputs are not glitch-free. In [8] the authors estimate that up to 70% of the activity inside arithmetic components is caused by glitches, and this would increase when glitches are allowed to propagate from one component to another, therefore it is important to consider the glitch activity when estimating power consumption.

In [9] the transition density technique is proposed, where the logical function of combinational blocks of logic is used to determine the probability of glitches at the inputs to a block being propagated through to the outputs. The low-level activity estimation techniques in [5], [6] propose the use of the *Boolean difference* function to find the output transition densities of a logic component based on its input transition density, but these methods fail to take into account the logic and inter component routing delays which give rise to most of the glitch activity.

In summary the techniques in [5], [6], [9] allow for activities in individual circuit elements to be estimated without using simulation by using transition density propagation, but still require some compilation of high-level design descriptions, which can be computationally expensive. Also, the transition density technique can still be computationally expensive for large combinational circuits.

The work presented in this paper, is to the best of our knowledge the first attempt to model the glitch activity and the related power consumption without the need for the compilation of the high-level design description or timing accurate simulations.

III. Modeling Glitch activity

The attention of this paper is focused primarily on modeling the glitches produced by arithmetic components such as adders and multipliers. Hence we deal with signals at the word-level and not at the bit level as in [5], [9] where the bit level signal inputs are considered to be independent. This allows the propagation of activity information through large combinatorial circuits such as multipliers without the use of boolean difference functions which would be computationally expensive.

The assumption of independence for the bit-signals in a bus is incorrect. The bits in a word are correlated with each other [2]. Our proposed method, being an empirical approach to modeling the glitches, considers the effect of logic and intra-component routing delays as well. The proposed method however does not consider the inter-component routing delays. Our experimental observations show that the greatest impact on routing delays is intra-component as opposed to inter-component.

Figure 1 presents the transition density [9] variation graph for the output of a 16-bit adder with glitch-free inputs. From the graph we derive the following three parameters, the peak height h, peak bit k and the signal word-length W.

A similar set of parameters can also be derived for the output transition density graph of the multiplier output shown in Figure 2. Using these transition density graphs it is possible to express a new metric to capture the transition density of a signal bus, we call this new metric the *Glitch Profile*, $\mathcal{G}$, and define it as the following tuple (2).

$$\mathcal{G} \;:=\; (W, h, k) \tag{2}$$

where W is the word-length of the signal, h is the peak height or the maximum value transition density and k is the peak bit number or the bit in the signal where this maximum value occurs.

The method use to obtained the transition density variation graphs in figures 1 and 2 is described in Section VI of this paper.

Although the total transition density given by the area under the curve is sufficient to model the intra-routing power, to model the logic and intra-routing power, the bit number in the signal at which the peak occurs as well is also required.

IV. Glitch Propagation

We now consider the estimation of the glitch profiles at the output of an arithmetic component based on the glitch profiles at the input of the component and the type of the component.

The amount of glitches generated depends upon the combinatorial logic depth from the input signals to the output signals of the component and the intra-routing delays associated with the signals inside the component. At a macro level we can model the glitches generated as being a function of the component type.

In addition to the glitches that are generated inside the component, the input signals to the component can themselves contain glitches as a result of being the outputs of other arithmetic component. These

TABLE I

FUNCTIONAL DESCRIPTION OF THE TEST CIRCUITS CONSIDERED, WHERE inA, inB, inC AND inD ARE THE INPUTS AND f IS THE OUTPUT OF THE CIRCUIT RESPECTIVELY.

Circuit Name	Description
c1	$f = (inA \times inB) + (inC \times inD)$
c2	$t1 = inA \times inB$ $f = (t1 + inC) \times t1$
c3	$t1 = inA + inB$ $f = (t1 \times inC) \times t1$
c4	$t1 = (inA \times inB) + (inC + inD)$ $f = t1 \times (inC + inD)$
c5	$f = (inA \times inB) \times (inC \times inD)$
c6	$t1 = inA \times inB \quad t2 = inC \times inD$ $t3 = inB \times inC \quad t4 = inA \times inD$ $f = (t1 \times t2) \times (t3 \times t4)$
c7	$f = (inA \times inB) + inC$
c8	$f = (inA \times inB) + (inC + inD)$

TABLE II

MEAN RELATIVE ERROR (MRE) FOR GLITCH PROFILE ESTIMATION

OP Type	Input Signal Type	MRE - h (%)	MRE - k (%)
Add	Glitch	9.0	14.0
Mul	Glitch	7.0	5.0
Add	Glitch Free	4.0	28.0
Mul	Glitch Free	12.0	4.0

input glitches can in turn trigger further glitch activity inside the component. This leads to another functional relationship between the output glitch profile and the input glitch profile.

We express both these functional relationships as a function of the input glitch profiles, the arithmetic component type and the output glitch profiles, as follows:

$$h_{out} = f_h(OP, \mathcal{G}_{in1}, \mathcal{G}_{in2}) \tag{3}$$

$$k_{out} = f_k(OP, \mathcal{G}_{in1}, \mathcal{G}_{in2}) \tag{4}$$

where $\mathcal{G}_{out} := \{W_{out}, h_{out}, k_{out}\}$, $\mathcal{G}_{in1}, \mathcal{G}_{in2}$ are the glitch profiles of the output and the two inputs respectively. $OP \in \{mul, add, sub\}$ represents the arithmetic operation being performed by the component and W_{out} is the word-length of the output signal.

Table I provides a functional description of the circuits used to test our proposed technique. The circuits feature several different combinations of the basic arithmetic operators, adder and multiplier. The circuits were designed to test as many different combinations of these components as possible.

In the proposed empirical approach described in more detail in Section VI, the sample circuits shown in Table I were built and the glitch profiles at the inputs and output of the arithmetic components were observed. Using these observations the functions f_h, f_k were approximated using a least-squares fitting method.

Table II shows the relative mean error for the estimated values of the Glitch profile components h, k. The mean relative error is calculated by the formula in equation (5).

$$MRE = \frac{1}{N}\sum_{i=1}^{N}\frac{|Est_i - Act_i|}{Act_i} \tag{5}$$

where Act_i is the actual value that is observed via net-list level simulation and Est_i is the estimated value obtained via our proposed method for a given component with given input glitch profiles. N is the number of different input glitch profiles considered.

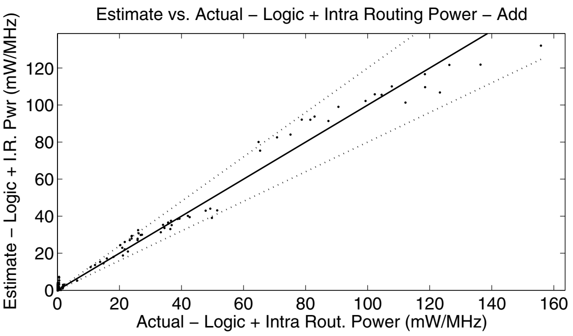

Fig. 3. Adder - Actual vs. Estimated logic and intra routing power consumption. The straight line represents the case when the actual and estimate values are the same, the dotted line represents ±20% variation bounds between the two values.

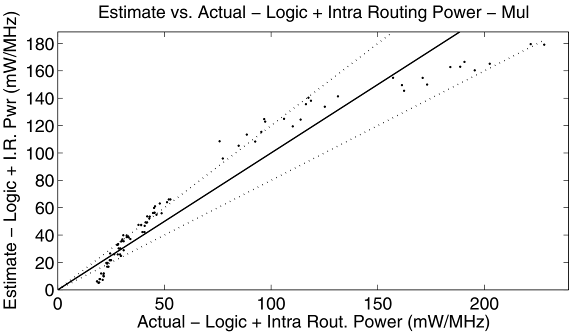

Fig. 4. Multiplier - Actual vs. Estimated logic and intra-routing power consumption. The straight line represents the case when the actual and estimate values are the same, the dotted line represents ±20% variation bounds between the two values.

V. EFFECT ON COMPONENT POWER

In previous work [2], we have shown that it is possible to model the logic power consumption in arithmetic components using signal word-level statistics when the inputs were glitch free.

When considering arithmetic components, which are composed of many logic elements connected together, it is possible to split the routing power as intra (within component) and inter (between components) routing power. In this work we consider the logic and intra-routing power of arithmetic components on FPGAs.

The logic and intra-routing power consumption inside the component is a function of the component's type and the transition activity of its inputs [2].

Similar to equation (3), the logic and intra-routing power consumption P_{LIR} can be expressed as a function of the input glitch profiles as:

$$P_{LIR} = f(OP, \mathcal{G}_{in1}, \mathcal{G}_{in2}) \tag{6}$$

The functional relationship between the inputs and outputs in equation (6) is derived again using the least-squares fitting approach.

Figures 3 and 4 show the estimated logic and intra-routing power consumption values derived using our proposed method shown in equation (6) and the actual logic and intra-routing power consumption values obtained using the Xilinx XPower [1] power estimation tool for adders and multipliers.

Although the estimates provided by Xilinx XPower are known to be inaccurate compared to the actual power consumption values obtained through bench measurements in FPGAs [10], XPower has the advantage in that it provides us the ability to estimate the power

TABLE III

Mean Relative Error (MRE) in the estimation of the total logic and intra-routing power, a comparison between the previous model described in [2] and the proposed model presented in this paper.

Circuit Name	Previous Model MRE (%)	Proposed Model MRE (%)
c1	71	9
c2	97	30
c3	72	30
c4	97	28
c5	94	40
c6	96	30
c7	41	31
c8	40	28

for individual components in the circuit based on activity at its inputs, which is what we require when building the parameterized power models. This information is difficult to obtain through device level measurement techniques.

VI. Method and Results

The circuits described in Table I were used to build and evaluate the power model proposed in this paper. The input signals to each of the circuits was registered in order to ensure that they are glitch free. The test circuits where synthesized and place and routed in to device level net-lists, these net-lists where then simulated with the test vectors produced using the signal generator described in [2].

The bit transition activity in signals during the simulation was observed and the signal glitch profiles were derived from it. Xilinx XPower was used to estimate the circuit power consumption, also based on the simulation of the net-list, from which the logic and intra-routing power consumption for the individual arithmetic components was obtained.

The power model was built by applying the above method to the circuits described in Table I and observing the glitch profiles at the inputs and output and power consumption in the arithmetic components.

In order to evaluate the logic and intra-routing power consumption estimates provided by our proposed method, we perform a comparison against the estimates obtained using the previous model we proposed in [2], where signal glitch activity is not considered when estimating power. Similarly other previous macro-models for FPGA arithmetic components [3], [4] also do not consider the case when the inputs to a component contain glitches, and are unable to predict how glitches propagate through a circuit, in contrast to this work. As a result we expect [3], [4] to perform as poorly as [2] for circuits shown here.

Table III presents the results of this comparison between the two different methods. The logic and intra-routing power consumption value calculated using Xilinx XPower is considered as the reference value or the Act_i value in equation (5) when calculating the MRE for estimates provided the two methods.

Each of the circuits described in Table I was parameterized with respect to its input and internal signal bit-widths. The MRE value for each circuit was found by estimating the total power consumption for the circuit for different internal signal bit-widths ranging from 12-bits to 28-bits.

From Table III it can be seen that the power consumption values obtained for circuits c1–c6 using the method described in [2] are an underestimation by up to 20 times. The proposed method on the otherhand can provide estimates that are within a mean relative error of 30% of the low-level estimate. For circuits c2–c6, which contain multipliers with inputs with glitches, the proposed method provides much better estimations than [2].

In circuit c1 both inputs to the adder are produced by multipliers and therefore cause a larger logic and intra-routing power consumption in the final adder than if it was fed with glitch free inputs.

However in circuits c7–c8 one of the inputs to the final adder is either glitch free(inC in c7) or produced by an adder($(inC + inD)$ in c8) therefore the increase in logic and intra-routing power consumption in the final adder due to glitch activity is not as large as the previous case. In this case the improvement in the MRE value is only 10% when the proposed method is used.

VII. Conclusion

In this paper we have described a novel method for modeling glitch activity inside arithmetic circuits using word-level statistics of signals. The proposed modeling technique is intended for use with optimization techniques and tools targeted at circuit design descriptions which are high-level or word-level based.

We have shown a method to capture the glitch activity by means of a metric known as the *Glitch Profile*. Using this metric we have presented a method to propagate the glitch activity through the arithmetic components of a circuit. We have also estimated the logic and intra-routing power consumed inside arithmetic components as a result of glitch activity and have shown that the estimates provided by the proposed method are better than those provided by estimation methods that do not take into account glitch activity.

Acknowledgments

The authors would like to acknowledge the support of Synplicity, Inc., Xilinx, Celoxica and the EPSRC under grant numbers EP/C512596/1 and EP/C549481/1.

References

[1] "Xilinx. http://www.xilinx.com/. Accessed: 01/06/2006."

[2] J. A. Clarke, A. A. Gaffar, G. A. Constantinides, and P. Y. K. Cheung, "Fast word-level power models for synthesis of FPGA-based arithmetic," in *IEEE International Symposium on Circuits and Systems*, Kos, Greece, 2006.

[3] L. Shang and N. Jha, "High-level power modeling of CPLDs and FPGAs," in *Proc. International Conference on Computer Design*, 2001, pp. 46–51.

[4] T. Jiang, X. Tang, and P. Banerjee, "Macro-models for high level area and power estimation on FPGAs," in *Proc. Great Lakes Symposium on VLSI*. ACM Press, 2004, pp. 162–165.

[5] K. K. W. Poon, A. Yan, and S. J. E. Wilton, "A flexible power model for FPGAs," in *Field-Programmable Logic and applications*. Montpellier, France: Berlin, 2002, pp. 312–321.

[6] J. H. Anderson and F. N. Najm, "Power estimation techniques for FPGAs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 10, pp. 1015–1027, 2004.

[7] P. E. Landman and J. M. Rabaey, "Architectural power analysis: the dual bit type method," *IEEE Trans. on Very Large Scale Integr. Syst.*, vol. 3, no. 2, pp. 173–187, 1995.

[8] A. Shen, A. Ghosh, S. Devadas, and K. Keutzer, "On average power dissipation and random pattern testability of cmos combinational logic networks," in *ICCAD '92: Proceedings of the 1992 IEEE/ACM international conference on Computer-aided design*, 1992, pp. 402–407.

[9] F. Najm, "Transition density: A new measure of activity in digital circuits," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, pp. 310–23, 1993.

[10] S. J. Wilton, S.-S. Ang, and W. Luk, "The Impact of Pipelining on Energy per Operation in Field-Programmable Gate Arrays," in *Proc. Field-Programmable Logic and Applications*, ser. LNCS, vol. 3203, 2004, pp. 719–728.