

ESTIMATING SPURIOUS POWER WHILE MAPPING K-LUT-BASED FPGA CIRCUITS

Ion Bucur, Nicolae Cupcea, Adrian Surpateanu, and Cornel Popescu
Computer Science and Engineering Department, University Politehnica of Bucharest
E-mail: (ion.bucur, nicolae.cupcea, adrian.surpateanu, cornel.popescu)@cs.pub.ro

Abstract

In this paper is presented a new approach for decreasing the spurious power consumption in K-LUT based FPGA implemented circuits. The approach is based on selective collapsing nodes in a direct acyclic graph (DAG) representing combinational or synchronous sequential circuits. It was used the simulation-based approach that estimates, using Monte Carlo experiment, the spurious switching activity of each net in the circuit. Traversing circuits in topological order, step by step best K-feasible cone are computed at the output of each node. Preserving the best depth of the circuits the mapping stage is done searching to minimize spurious switching power.

Keywords: *spurious switching power, K-feasible cones, optimum depth, optimal area and power.*

1. Introduction

Power consumption is becoming one of the most important considerations in VLSI design. High power often run hot and high temperature tends to exacerbate several silicon failure mechanisms. Spurious signals have large contribution in the dynamic power consumption. Therefore, in addition to performance and area optimization a great deal of research has been directed towards issues related to the low power FPGA circuit mapping.

Our main focus in this paper is on searching optimized solutions primarily for depth, and spurious power at the logic optimizing level. Our secondary focus is on minimal depth, area and spurious power at the same level. It is introduced a new power-driven mapping K-LUT based FPGA circuits based on previous published studies. The paper is organized as follows:

Section 2 describes power consumption issues in FPGAs. In section 3 there are briefly presented relevant previously published results concerning technology mapping that minimize spurious switching activity in FPGAs. Main definitions and used model are presented in section 4. Method of used spurious switching activity evaluation, at the gate level, is outlined in section 5. Section 6 presents procedures and introduces main features of the *PwDrvMap* algorithm. Experimental results and studies are presented in section 7. Conclusion and future work are exposed in section 8.

2. Power consumption in FPGAs

FPGAs circuits are developed in CMOS technology. Power dissipation in CMOS circuits comprises both static (leakage) power and dynamic power. Dynamic power is caused by signal transitions of a circuit and is governed by the equation:

$$P_d = 0.5 \sum_i C_i \cdot V_{dd}^2 d(i) \quad (1)$$

And in relation (1) introduced symbols represents:

- C_i is the capacitance of an arbitrary signal, i ,
- $d(i)$ is referred to as “transition activity” of logic signal i and represents the rate of transitions of signal i (i.e. the number of times that signal i changes its value in unit time), and
- V_{dd} is the supply voltage.

There are two types of signal transitions. One is the signal transition which is necessary to perform the required logic functions, and it is named functional transition. The other is the needless signal transition due to the uneven path delays to the inputs of a gate (hazards), and it is named spurious transition or glitch.

While spurious transition is not exclusive to FPGAs, the relatively high capacitive loading of programmable interconnect places a much important power cost to spurious signal. Most dynamic power in an FPGA is consumed in the programmable routing fabric. It is estimated [10] that about 50% - 70% of total power is dissipated in the interconnection network.

3. Related work

There have been done several works for decreasing the spurious power consumption in circuits mapped with FPGAs.

Anderson and Najm, in [1], are proposing a new programmable FPGA routing switch intended to function in three power appropriate modes: *sleep*, *low-power*, or *high-speed*. The conventional routing switch is mostly equivalent to their high speed mode, offering comparable performance and dissipated power. In their low-power mode, dissipated power is reduced by 36-40% but the performance is lower by about 28%. In their sleep mode the leakage power is 61% lower than in traditional switch. The applicability of the new switch is motivated through an analysis of timing slack in industrial FPGA designs.

Cheng *et al.* introduced a dynamic power evaluation model and an original technology mapping algorithm (named *GlitchMap*) addressing glitches [6]. Their experiments did show that *GlitchMap* was able to reduce dynamic power by 18.7% compared to a previous state-of-the-art power-aware algorithm, EMap published in a paper of Lamoureux *et al.* [8].

Lamoureux *et al.* presented in [9] *GlitchLess*, a circuit-level approach able to reduce spurious power in FPGAs by adding programmable delay elements to the logic blocks of the FPGA. Once performed routing the circuit and then performed the static timing analysis such delay elements are programmed in order to align arrival times of input signals of each look-up table. This method avoids new spurious power from being produced. Programmable delay elements are playing an important role in filtering and eliminating other parasitic transitions produced by upstream logic or off-chip circuits. This method did show, on average, 87% elimination of the spurious signals and a consumed power, in FPGA, by 17%. Supplementary introduced circuitry is increasing the chip area by 6% and delays on critical paths by less than 1%.

In [7] Hsieh, Cong, Zang, and Chang are discussing the optimization of the interconnect power for implementations using FPGA technologies. The authors are, in particular, interested to reduce the glitch power on lines associated with the output of functional blocks in projects. Their main idea was to introduce in circuit unused flip-flops in order to stop glitches' dissemination through the circuits. Putting at work vacant flip-flops is made with very little costs because there are plentiful flip-flops in actual FPGA circuits. Flip-flop usage method could cause data hazards issues. In order to prevent such situations Hsieh *et al.* developed a number of behavioral design techniques able to avoid such consequences. Their approach was evaluated for optimality and experimental results did show that, on average, their methods reduced by about 28% the dynamic power in the Xilinx Virtex-II technology.

4. Definitions and main model

Combinational part of a general logic network N can be represented as a direct acyclic graph (DAG) noted as $N(V, E)$ where V is the set of nodes and E is the set of directed edges. Each node in $N(V, E)$ represents a logical gate (possible complex), and a direct edge (u, v) exists if the output of gate u is an input of the gate v . The set of direct predecessors of gate v is expressed as $input(v)$ and the set of direct predecessors of a graph $H \subseteq N(V, E)$, is similar expressed as $input(H)$. Let u be an arbitrary gate in N and we are interested to compute generic K -feasible cone of node u (denoted C_u) in direct acyclic graph $N(V, E)$. Given an arbitrary node $u \in N(V, E)$ the level of u is computed, in general, using the expression:

$$level(u) = 1 + maximum_level\{ input(u) \}.$$

The level of a PI node is zero and the level (depth) of a network is the largest node level in the network.

5. Net switching activity estimation

Various approaches to computing dynamic switching activity have been proposed in the literature, and they can be, generally, considered as either simulation-based approaches, or as probability approaches. In [4] it was used the probability method. However this approach, in networks where there are re-convergent fanouts, is computing an upper bound of the stationary probability. It introduces an error of this important parameter of dynamic power dissipation in FPGAs circuits.

In this paper was used a simulation-based approach to determine dynamic switching activity for 15 combinational circuits of the MCNC benchmark. These benchmark circuits are first optimized using, traditionally, *rugged script* in SIS-1.2 application [11]. It was developed and built-up from the scratch new powerful simulator using structures and routines of SIS-1.2.

It was used, at gate-level, for accurate simulation-based logic activity analysis. This gate-level simulator has added capabilities for capturing the number of logic transitions on each net during simulation, as well as the proportion of time each net spends in the high and low logic states. One of the most important features of this simulator is the symbolic simulation. This facility is intended to leverage combinational and sequential circuits' optimization.

Simulation with zero or logic delays can be done for combinational or synchronous sequential circuits. Gate-level simulations provide both the functional (0 delay gate) and total activity. Spurious transition activity is computed as the difference between them. Circuits were simulated using almost 10 000 randomly generated input vectors. Simulation convergence is dynamically controlled. Each time an internal line is incrementing the logic activity (the number of changes from 1 to 0, or from 0 to 1), it is evaluated the expression:

$$\left| \frac{n}{N} - \frac{n+1}{N+m} \right| \leq \varepsilon \quad (2)$$

In (2) N is the number of simulated vectors after that the number of logic changes, on an arbitrary line w , become n . Let $N + m$ be the number of simulated vectors when the number of logic changes, on line w , is incremented ($n + 1$). Parameter ε is introduced at beginning of the simulation and it represents the precision of simulated-based logic activity determination. Simulator is checking when all internal lines and primary output lines of the circuit fulfill condition (2).

The power results that were computed are based on identical switching activities (0.5) for each primary input. In FPGAs, spurious transitions are generated at the output of LUT when the input signals change at dissimilar times. The pulse width of these spurious transitions depends on how uneven the input coming times are.

6. Power-driven FPGA mapping

Given a general network it is first optimized, independently of the target implementation technology. Our application is built-up using structures and routines of SIS-1.2 [11] and appropriate cost functions for power-driven minimal depth mapping. Basically, the approach used *K*-feasible-cones-enumeration method that mainly consists of *K*-feasible cones exhaustive generation.

K-feasible cones generation is made during network traverse from primary inputs to primary outputs and establish edge-delay of each feasible cone [2]. It is obvious that each *K*-feasible cone of the arbitrary node *u*, noted *C(u)*, could be identified by the set:

$$input(C(u)) = \{v_1, v_2, \dots, v_m\}, m \leq K \quad (3)$$

The set of all feasible cones of node *u*, noted *cones(u)*, can be represented as the Boolean sum (reunion) of each of the product (cube) representing the respective cone:

$$cones(u) = \bigcup_i input(C_i(u)) \quad (4)$$

Representing each *K*-feasible cone of the node *u* as a conjunction, in above relation, it becomes:

$$cones(u) = \bigcup_i v_{i_1}, v_{i_2}, \dots, v_{i_m} \quad (5)$$

Then it holds this Lemma:

Lemma1. Given a node *u* having as immediate predecessors: $input(u) = \{v, w, \dots, z\}$, each predecessor having already computed the set of all *K*-feasible cones, respective *cones(v)*, *cones(w)*, $\dots$, *cones(z)*. Than, the set *cones(u)* satisfy this relationship:

$$cones(u) \subseteq \bigcap_{v_i \in input(u)} (v_i \cup cones(v_i)) \quad (6)$$

Applications of the *Lemma1* are presented in [2]. Results of the generation of all *K*-feasible cones rooted in every node of DAG representing one of 10 large circuits from the *MCNC 91 ATPG* benchmark are shown in [3]. Actually implementation of *PwDrvMap* algorithm was successively developed with facilities for minimal depth, optimal area and dissipated power estimation. Once *K-feasible cones* generation completed, same network is traversed in reverse topological order, starting with the primary outputs lines that have largest delay mapping (that was evaluated during *K-feasible cones* generation [2]).

Making the right selection among the *K-feasible cones* of each node is guided using critical path in circuits and several appropriate cost functions (metrics). For each node is chosen the *K*-feasible cone having the lowest metric in the set of all *K*-feasible cones of the irrespective node. For nodes having no slack only *K*-feasible cones involving minimum network's depth will be considered. But, for nodes having slack it will be considered all *K*-feasible cones.

The problem of mapping for depth, of an arbitrary network, can be optimal computed in polynomial time using dynamic programming procedure [3]. Our heuristics are using several metrics. These heuristics are using data related to the number of internal nodes of each cone, as an efficiency measure of it, the count of internal nodes having fan-outs spreading in other feasible cones (possible duplicated nodes) etc.

Logic replication or duplication is performed implicitly when a LUT is used to implement a *K*-feasible cone. Replicating a node, in general, it has two consequences. First consequence is about connections, having high logic activity, from the replicated node to its successors.

Second consequence is about the increased fanout of nodes that are direct predecessors of duplicated node.

Depth Metric for node u is computed over one of the best depth K -feasible cone of u :

$$\text{DepthMetric}(\text{cones}(u)) = 1 + \min_v (\text{DepthMetric}(v \mid v \in \text{cones}(u))) \quad (7)$$

This metric is used mainly to quantify the depth criterion. In order to quantify the best suitable cone rooted in u is introduced the cost function *bestCone*. It is locally applied for the entire set of K -feasible cones of the node u and it is computed using the following relation:

$$\text{bestCone}(u) = \min_v \{ \text{multiCrit}(v \mid v \in \text{cones}(u)) \} \quad (8)$$

Multi-criteria function *multiCrit* is implemented as follows [8]:

$$\text{multiCrit}(v) = \frac{(1 + |LUT(v)|) \cdot S_{w \in \text{input}(v)}}{(1 + |\text{cone}(v)| - LUT(v))} \quad (9)$$

and

$$S_{w \in \text{input}(v)} = \sum_{w \in \text{input}(v)} \frac{q_1(w)(1 + q_2 \cdot \text{activity}(w))}{\text{fanout}(w)} \quad (10)$$

In (9) and (10) were used several entities that are essential in the defined multi-criteria cost:

- $|LUT(v)|$ is the number of internal nodes of cone v previously (already) chosen as LUTs,
- $|\text{cone}(v)|$ represents the number of internal nodes of cone v .
- $\text{activity}(w)$ is the spurious logic activity estimation of the node w ,
- $|\text{fanout}(w)|$ represent the number of nodes connected to the output of node w ,
- $q_1(w) = 1$ iff $|\text{fanout}(w)| < 3$,
- $q_1(w) = 0$ iff $|\text{fanout}(w)| \geq 3$, and
- q_2 is the parameter controlling the influence of the spurious activity estimation in multi-criteria cost.

Both parameters, q_1 and q_2 , were experimentally determined. Best results were obtained when $q_2 \gg q_1$ reflecting the preference for optimizing power over depth.

7. Experimental results

The basis of our approach is the exhaustive generation of all feasible K -feasible rooted in every node of the network. The high speed of this generation is offering enough time margins in order to search among all possible solutions the most appropriate one. It was assumed that all primary inputs have 0.5 switching activities, and all involved capacities have same value.

Our implemented algorithm did run for mapping into 5-LUT FPGAs several benchmark circuits, listed in Table 1 (it is possible, also to run the algorithm for $4 \leq K \leq 12$, with very small time penalties). To estimate power consumption using (1) it is required the capacitance of each net. The fanout of each mapped node was considered as an estimate of the capacitance of it.

Table 1. Estimated Spurious Power

Circuit	Depth	Optimum depth	Optimal depth	Optimal depth & area
5xp1	3	2,23	1,33	1,81
9symml	5	2,87	2,34	2,70
C499	5	9,89	7,88	8,04
C880	8	15,43	13,84	14,37
alu2	8	13,72	12,11	12,57
apex6	4	22,34	20,02	20,78
apex7	4	10,11	8,22	8,75
count	3	3,22	2,33	3,01
des	5	98,68	95,74	96,02
duke2	4	9,11	8,82	8,34
misex1	2	2,15	2,02	2,01
rd84	4	4,39	4,22	4,10
rot	6	35,46	34,03	35,23
vg2	4	4,22	2,55	2,91
z4ml	3	1,45	1,29	1,37

Our implemented algorithm did run for mapping into 5-LUT FPGAs several benchmark circuits, listed in Table 1 (it is possible, also to run the algorithm for $4 \leq K \leq 12$, with very small time penalties). To estimate power consumption using (1) it is required the capacitance of each net. The fanout of each mapped node was considered as an estimate of the capacitance of it.

Because our attempt was to build-up a tool able to evaluate medium-grain different network choices during logic optimization, the estimated dynamic power for each node u was simply computed mainly as the product of the node's spurious transition density $activity(u)$ and the fan-out of it:

$$EstimGlitchPower(u) = activity(u) \cdot fanout(u) \quad (11)$$

PwDrvMap is an efficient algorithm being able to compute several low-power optimal options, as can be seen in Table 1.

The first option keeps optimum depth and search among power-aware equivalent solutions. The second option is searching, on the base of the user's explicit option, one of the solutions with optimal depth but performing with improved spurious power consumption. For nodes situated on the critical paths of irrespective networks the optimal depth was computed using this relation:

$$optimalDepth(u)_{u \in criticalPath} = optimumDepth + \delta \quad (12)$$

Values listed in the second column of Table 1 were computed using $\delta=1$ for nodes belonging to the critical path, while for other nodes, not situated on critical paths, the optimal depth values were less or equal to the optimal depth of the circuit. Area minimization is of dominant importance for FPGA synthesis.

While maintaining an optimum depth of the network it is searched, among power-aware solutions, those having the minimal area (number of used LUTs). The third solution targets an optimal area and depth while keeping in low margin the dissipated power (illustrated in the third column of Table 1).

On average, the detailed experimental results are showing that power-aware mapping for optimal depth are producing 8.88% less estimated dissipated spurious power than mapping for optimum depth.

8. Conclusions and future work

Relaxing mapping conditions for circuits' depth it is leading to less dissipated power. But, introducing area minimal constraint it makes mapping, for both optimal depth and area, to be only 5.64% more efficient (concerning the spurious activity), on average, than mapping for optimum depth. Power-driven mapping both for depth and area optimal it appears to be more complex than mapping only for optimal depth.

Actual working heuristics have to be upgraded because it was searched only a limited part of mapping solutions' space. It is intended in the future approach to use dynamic programming together with refined heuristics to further develop *PwDrvMap* algorithm.

9. References

- [1] J.H. Anderson, and F.N. Najm, "A Novel Low-Power FPGA Routing Switch", *Proceedings of the IEEE 2004 Custom Integrated Circuits Conference*, ISBN: 0-7803-8495-43-6, pp. 719-722, October 2004.
- [2] I. Bucur, "Performance mapping of k -LUT based FPGAs", *University. Politehnica of Bucharest, Scientific Bulletin, Series: C*, Vol. 69, No. 2, Romania, pp.49-60, 2007.
- [3] I. Bucur, I. Fagarasan, C. Popescu, C.-A. Boiangiu, and G. Culea, "On K-LUT Based FPGA Optimum Delay and Optimal Area Mapping", *Proceedings of WSEAS International Conference on Mathematical and Computing Methods in Science and Engineering*, pp.137-142, Bucharest, 2008.
- [4] I. Bucur, C. Stefanescu, A. Surpateanu, and N. Cupcea, "Power-Aware and Optimal Depth Mapping of LUT Based FPGA Circuits", *Proceedings of the 17th International Conference on Control Science and Computer Science*, pp. 117-124, Bucharest, May 2009.
- [5] I. Bucur, C. Stefanescu, A. Surpateanu, and N. Cupcea, "Power Consideration in Mapping LUT based FPGA Circuits", *IEEE Intl. Conference on Intelligent Computer Communication and Processing*, August 27-29, Cluj-Napoca, Romania, Editor Ioan Alfred Letia, pp.369-372, ISBN: 978-1-4244-5007-7, Library of Congress Number: 200906791, IEEE Catalog Number: CFP0909D-PRT, 2009.
- [6] L. Cheng, D. Chen, M.D.F. Wong, "GlitchMap: An FPGA Technology Mapper for Low Power Considering Glitches", *The 44th ACM/IEEE, Design Automation Conference*, San Diego, CA, pp.318-323, 4-8 June 2007.
- [7] C.-T. Hsieh, J. Cong, Z. Zhang, and S.-C. Chang, "Behavioral Synthesis with Activating Unused Flip-Flops for reducing Glitch Power in FPGA", *ASP DAC'08*, Seoul, Korea, pp. 10-15, 21-24 March 2008.

- [8] J. Lamoureux, and S.J.E. Wilton, "On the interaction between power-aware FPGA CAD algorithms", *Proceedings of the International Conference on Computer-Aided Design*, ICCAD'03, IEEE Computer Society / ACM, San Jose, CA, USA, pp.701 – 708, 9-13 November 2003.
- [9] J. Lamoureux, G.G. Lemieux, and S.J.E. Wilton, "GlitchLess: Dynamic Power Minimization in FPGAs through Edge Alignment and Glitch Filtering", *IEEE Transactions on VLSI Systems*, Vol.16, No.11, pp.1521-1534, November 2008.
- [10] F. Li, Y. Lian, L. He, D. Chen, and Jason Cong, "Power Modelling and Characteristics of Field Programmable Gate Arrays", *IEEE TCAD*, Vol.24, No.11, pp.1712-1724, November 2005.
- [11] SIS-1.2, available on-line, <http://embedded.eecs.berkeley.edu/pubs/downloads/>.