Thermal-Aware Floorplanning for Partially-Reconfigurable FPGA-based Systems

Davide Pagano, Mikel Vuka, Marco Rabozzi, Riccardo Cattaneo, Donatella Sciuto, Marco D. Santambrogio
Politecnico di Milano, Milan, Italy
{davide1.pagano, mikel.vuka, marco.rabozzi}@mail.polimi.it,
{riccardo.cattaneo, donatella.sciuto, marco.santambrogio}@polimi.it

***Abstract*—Field Programmable Gate Arrays (FPGAs) systems are being more and more frequent in high performance applications. Temperature affects both reliability and performance, therefore its optimization has become challenging for system designers. In this work we present a novel thermal aware floorplanner based on both Simulated Annealing (SA) and Mixed-Integer Linear Programming (MILP). The proposed method takes into account an accurate description of heterogeneous resources and partially reconfigurable constraints of recent FPGAs. Our major contribution is to provide a high level formulation for the problem, without resorting to low level consideration about FPGAs resources. Within our approach we combine the benefits of SA and MILP to handle both linear and non-linear optimization metrics while providing an effective exploration of the solution space. Experimental results show that, for several designs, it is possible to reduce the peak temperature by taking into account power consumption during the floorplanning stage.**

I. Introduction

The use of FPGAs is steadily increasing in commercial applications that require high performance. This has lead to a higher attention in floorplan [1] design regarding performance, power consumption, area and wire length. In addition, in order to exploit modern FPGA capabilities like Partial Reconfiguration (PR) additional constraints must be taken into account: each region should cover the required amount of heterogeneous resources [2], while having the right shape in order to allow PR [3]. Different solutions have been developed ([4]–[9]), but only few of them ([4], [5], [9]) deal with the previous constraints.

Another important factor to consider in the floorplan process is the amount of power consumption, especially now that transistors density is increasing. Having a uniform temperature distribution has several advantages: lowers the probability of failure [10], contributes against the exponential increase of both static and dynamic power dissipation [11] and can lead to higher performance as well [12]. Several solutions have been proposed, but none of them was able to consider the floorplanning problem at a high level of abstraction: most of the works like [13] take in consideration sub-circuit partitioning or routing tracks [14]; others [15], [16] are more focused on power models. Therefore, following one of the few approaches existing in literature, we use a Node-Arc model (better described in Section II) similar to the one used in [17] where each node is a Reconfigurable Region (RR), thus allowing high level analysis.

Our contribution is to provide a high level floorplanning methodology for the most recent FPGAs, capable of dealing with heterogeneous resources and PR constraints, with a customizable objective function that takes into consideration wire length, area occupancy and wasted resources, while reducing the overall peak temperature attained, without considering lower level details about the FPGA.

The novel Thermal-Aware Floorplanner TAF that we present in this paper is based on SA and explores the solution space by means of the sequence pair representation [18]. To our knowledge, there is no other work that considers the temperature effects in floorplanning, so for comparison purposes we considered a second approach, Thermal Optimal TO, that extends the state-of-the-art floorplanner presented in [9]. In [9] the authors introduced two floorplan algorithms based on MILP: one called [HO] (used in TAF), able to locally improve the solution represented by the sequence pairs and another called [O], able to solve the problem to optimality. The latter has been extended into TO in this paper with new linearized constraints to enable thermal optimization of the floorplan as described in Section III. Both [O] and [HO], deal with the constraints of the newest FPGAs like PR and heterogeneous resources.

TAF stems from the MILP models presented in [9], it exploits the [HO] formulation in order to obtain a placement of the RRs starting from the sequence pair description. The use of [HO] allows to obtain the optimal solution with respect to the linear components of the objective function and the sequence pair considered at the current iteration of the SA, while the non-linear metrics are optimized by the annealer.

The remainder of this paper is organised as follows: Section II presents a description of our Node-Arc thermal resistive model, Section III shows how we extended the [O] MILP formulation to take into account thermal optimization, in Section IV we discuss our TAF algorithm, Section V provides details about the results obtained and Section VI concludes the paper.

II. Problem description and thermal model

In this section we present the thermal model used within TAF and TO approaches. In order to have a formal representation of the layout of the RRs and the thermal interaction between them, we reformulated the Node-Arc model presented in [17]. Assuming steady state conditions by considering an

978-3-9815370-4-8/DATE15/©2015 EDAA

average power consumption for the tasks assigned to the same RR, the heat flow can be modeled using the following equation:

$$K\nabla^2 T = p(x,y,z) \tag{1}$$

where T is the temperature, p is the power dissipation and K denotes the thermal conductivity.

In order to implement the Node-Arc model we apply the Finite Difference Method (FDM)[19] to solve equation 1. The resulting set of equations is analogous to that of electrical circuits [19] and for each node i it holds:

$$\sum_j \frac{t_i - t_j}{R_{i,j}} + p_i = 0 \tag{2}$$

where p_i is the power dissipated at node i and t_i, t_j are the steady state nodal temperatures at node i and j. $R_{i,j} = l_{i,j}/(K \cdot A_{i,j})$ is the thermal resistance between i and j, where A_{ij} is the sectional area normal to l and $l_{i,j}$ is the Manhattan distance between i and j.

The model described by equation 2 is considered at a higher level of abstraction: instead of representing the FPGA reconfigurable fabric, our nodes represent the RRs directly and the arcs denote routing channels separating adjacent RRs. The RR is represented as a point source as we do not seek to model the thermal distribution within the RR, while the value of the thermal resistance between nodes depends on the width of the wire channel.

III. Thermal extension of the MILP model

In this section we present the new TO MILP formulation that stems from the one presented for the [O] algorithm. In order to take into account the Node-Arc thermal model we need to introduce several new parameters, variables and constraints that are discussed in the following subsections.

A. Parameters and variables

From the [O] model, we recall that a tile represents the minimal area unit considered within the FPGA grid and we denote by $tileW$ and $tileH$ its width and height respectively. Within the FPGA grid there are $maxW$ tiles on the horizontal direction and $maxH$ tiles on the vertical direction. We also recall that N is the set of the reconfigurable regions that need to be floorplanned, while cx_n and cy_n are variables representing the x and y coordinates of region n centroid. What follows are the new thermal parameters added to the MILP formulation:

- $p_n :=$ average power dissipated by region n;
- $t_{ext} :=$ external temperature;
- $R_{ext} :=$ external thermal resistance;
- $rol_{i,j} :=$ thermal resistance for each unit of distance between regions i and j computed as $1/(K \cdot A_{i,j})$.

These are the new variables needed to characterize the thermal model:

- $t_n :=$ real variable ($\geq t_{ext}$) representing the temperature of region n;
- $dp_{i,j} :=$ real variable denoting the thermal power flowing from region i to region j.

The Node-Arc thermal equation 2 is hard to be considered within the MILP model since it involves divisions among variables. Indeed both the regions temperatures and the thermal resistance between regions can vary across different floorplans. For our purpose it is convenient to rewrite equation 2 as:

$$\forall i \in N : \sum_j dp_{i,j} + \frac{t_i - t_{ext}}{R_{ext}} + p_i = 0 \tag{3}$$

where:

$$dp_{i,j} = \frac{t_i - t_j}{rol_{i,j} \cdot l_{i,j}} \tag{4}$$

Equation 4 can be further rewritten as:

$$t_i = rol_{i,j} \cdot dp_{i,j} \cdot (|cx_i - cx_j| + |cy_i - cy_j|) + t_j \tag{5}$$

In order to include equation 5 within the MILP model we linearized it by introducing binary variables to solve the absolute values and by exploiting the binary expansion of variables cx_n and cy_n to compute the bilinear products. cx_n and cy_n are integer multiples of $tileW/2$ and $tileH/2$ respectively. We denote by xB and yB the positions of the most significant bits required for the binary expansion of variables cx_n and cy_n. What follows are the new variables needed for the model linearization:

- $rx_{i,j}$ $(ry_{i,j})$:= binary variable set to 0 if and only if region i centroid is at the left (at the bottom) of region j centroid;
- bx_n (by_n) := binary variable representing the b-th bit value of $cx_n \cdot 2/tileW \quad (cy_n \cdot 2/tileH)$;
- $qx_{i,j}$ $(qy_{i,j})$:= real variable representing the product: $dp_{i,j} \cdot cx_i \quad (dp_{i,j} \cdot cy_i)$;
- $vx_{i,j,b}$ $(vy_{i,j,b})$:= real variable representing the product: $dp_{i,j} \cdot bx_i \cdot 2^b \cdot tileW/2 \quad (dp_{i,j} \cdot by_i \cdot 2^b \cdot tileH/2)$;
- $dpx_{i,j}$ $(dpy_{i,j})$:= real variable representing the value: $dp_{i,j} \cdot |cx_i - cx_j| \quad (dp_{i,j} \cdot |cy_i - cy_j|)$.

B. Model constraints

In this subsection we consider a total ordering of the regions within set N. This ordering is used to exploit a symmetry of the Node-Arc thermal model to reduce the number of variables and constraints that are needed. Since the thermal resistance between two regions is the same regardless of the direction of the thermal flow ($rol_{i,j} = rol_{j,i}$), for each couple of regions we can enforce the following constraint:

$$\forall i \in N, j \in N \mid i < j : dp_{i,j} = -dp_{j,i} \tag{6}$$

To compute the temperatures of the regions, we need to ensure the constraints deriving from equation 3 and 5. Equation 3 can be included directly into the MILP model as a constraint, while the constraint related to equation 5 is linearized and rewritten as:

$$\begin{aligned} &\forall i \in N, j \in N \mid i < j : \\ &t_i = rol_{i,j} \cdot (dpx_{i,j} + dpy_{i,j}) + t_j \end{aligned} \tag{7}$$

Constraints 3 and 7 are enough to include the thermal model within the MILP formulation, however further consistency constraints are needed to guarantee the semantics of all the variables involved in the linearization process. These extra constraints are not reported here due to limited space, however they can be directly derived from the definition of the variables exploiting ”big M” constants where needed.

C. Objective function

Having computed the temperatures of the regions, we can now define an extra real variable T_{cost} and enforce the following constraint:

$$\forall i \in N : T_{cost} \geq t_i \tag{8}$$

In this fashion T_{cost} is bounded to be not less than the maximum temperature reached by a region and we can include it within the objective function to minimize the peak temperature of the design. T_{cost} can be easily normalized and included in the objective function proposed for [O] as follows:

$$\min \left\{ q_1 \cdot \frac{WL_{cost}}{WL_{max}} + ... + q_4 \cdot \frac{T_{cost} - t_{ext}}{DT_{max}} \right\} \tag{9}$$

where DT_{max} is defined as: $max_{n \in N} t_n - t_{ext}$ and it is computed considering the worst case maximum temperature for a region. Whereas q_1 and q_4 represent the weight assigned to wire length and temperature cost component respectively.

IV. Implementation of TAF algorithm

In this section we make reference to the [HO] MILP formulation and elaborate on the implementation details of the TAF algorithm. The general idea of TAF is to explore the solution space by means of a sequence pair representation that is optimized by SA. At each iteration of the annealer [HO] is invoked to locally improve the linear metrics of the user defined objective function and to obtain a placement for the RRs. If the [HO] MILP model is feasible, we are able to calculate the temperature of each RR in that particular floorplan according to our Node-Arc thermal model. Using the thermal map obtained, speculations can be made to obtain a better one by swapping RR slots in the sequence pair.

For each iteration of the annealer, a new sequence pair is generated swapping two slots in the former one. Having specified the new sequence pair, we write the [HO] MILP model and give it as input to a MILP solver (such as Gurobi [20]). If the solver is unable to find a solution, it means that the model is unsatisfiable: we discard that particular sequence pair and start over in another loop iteration. Otherwise, the solver returns an actual floorplan from which we can compute the distances and the thermal resistances between regions. The latter information, together with the power consumption of the regions that is given as input, allows us to compute the thermal map of the floorplan.

Based on the thermal map and the floorplan we associate a cost to the solution by means of equation 9. If the solution cost improves with respect to the best one found so far, we update the best solution and the related sequence pair is accepted or rejected depending on the current acceptance probability of the SA. Either when the SA timer runs out or when the uniformity of the thermal map has been reached, the loop exits and the best solution found is returned.

This algorithm performs rather well for small amount of RRs (at most 20 regions). In order to have good performance with problems having more regions it is necessary to warm start the algorithm with a sequence pair that leads to a feasible solution. For this purpose we use an incremental floorplanner that plans only a subset of RRs at each step, while considering fixed the position of the previous RRs.

V. Results Evaluation

We performed tests using designs with 5, 10, 15 and 20 RRs and within the objective function we considered both peak temperature and wire length for minimization. Specifically we performed 3 different tests for each set of reconfigurable regions as shown in table I and II: first giving equal weight to the thermal objective function and wire length, then focusing on the thermal map optimization by giving it a 0.95 weight, and finally focusing on the wire length by giving the thermal map optimization a weight of 0.05. The simulation experiments were held using a configuration specific to the Virtex-5 XC5VLX110T FPGA model. The models considered for testing required to occupy around 70% of the Configurable Logic Block (CLB) resources of the FPGA and the external temperature was set to 0 Celsius degrees.

Both TAF and TO were warm started using a solution achieved by the incremental floorplanner. The quality of the initial floorplan and the ones produced by TAF and TO are shown in table II, while within table I comparison are made considering the value of the objective function.

TABLE I
TAF objective improvement over initial solution and TO

# RRs	WL weight	T_{cost} weight	TAF Improvement over		Execution time [s]
			Initial solution	TO	
5	5%	95%	+8.53%	-0.41%	708
	50%	50%	+26.63%	-4.25%	745
	95%	5%	+64.52%	-8.23%	764
10	5%	95%	+3.40%	-1.57%	1633
	50%	50%	+11.36%	+4.16%	1653
	95%	5%	+44.29%	+35.93%	1693
15	5%	95%	+2.30%	-1.01%	3218
	50%	50%	+4.62%	-8.33%	3280
	95%	5%	+25.72%	-4.84%	3516
20	5%	95%	+4.21%	+4.21%	9220
	50%	50%	+12.82%	+12.82%	9445
	95%	5%	+32.25%	+32.25%	9138

As shown in the tables, TO gives better objective function values than TAF in cases where the number of reconfigurable regions is low (5 regions). When this number increases, the TO MILP model becomes too complex and the solver has serious problems to solve the continuous relaxation of the instances. As for the temperature, the maximum variation that can be obtained is about 1-2 degrees in the instance with 10 regions, going from an optimization that concentrates on wire length to one that gives priority to the maximum temperature.

TABLE II
APPROACH COMPARISON WITH DIFFERENT NUMBERS OF RRS AND OBJECTIVE FUNCTION WEIGHTS

# RRs	WL weight	T_{cost} weight	Initial solution			TO solution			TAF solution		
			Cost	T_{cost} [°C]	WL	Cost	T_{cost} [°C]	WL	Cost	T_{cost} [°C]	WL
5	5%	95%	0.392	4.047	4692	0.357	3.739	1978	0.359	3.731	3047
	50%	50%	0.303	4.047	4692	0.213	3.743	1246	0.222	3.878	1355
	95%	5%	0.213	4.047	4692	0.070	3.743	1246	0.076	3.901	1367
10	5%	95%	0.334	7.000	6587	0.318	6.662	6116	0.323	6.747	6871
	50%	50%	0.260	7.000	6587	0.240	6.753	5585	0.230	7.163	4092
	95%	5%	0.185	7.000	6587	0.161	7.198	5621	0.103	7.523	3322
15	5%	95%	0.310	6.477	19788	0.300	6.314	14194	0.303	6.275	24481
	50%	50%	0.251	6.477	19788	0.221	6.324	14178	0.240	6.673	16288
	95%	5%	0.192	6.477	19788	0.136	6.302	13550	0.143	6.493	14237
20	5%	95%	0.334	8.102	26728	0.334	8.102	26728	0.320	7.712	30424
	50%	50%	0.279	8.102	26728	0.279	8.102	26728	0.243	7.741	19781
	95%	5%	0.223	8.102	26728	0.223	8.102	26728	0.151	7.867	17459

VI. CONCLUSIONS

Our work presented an algorithm to generate efficient thermal aware floorplans at the RRs granularity for heterogeneous and partially reconfigurable FPGAs, by also taking into account the total wire length and giving the designer the possibility to fully customize the objective function and to optimize different metrics. The temperature reduction results in our paper are worse than the results presented in [17]. This had to be expected, since [17] considers a representation at the CLB level, while in this work we address single RRs. Although the maximum temperature reduction is only of few degrees, the temperature peak has been lowered resulting in a more uniform thermal map. Future works may include a better model to take into consideration the temperature of every point of the grid (and not only the one covered by a region) and a better policy to decide how to perform swaps in the sequence pair.

VII. ACKNOWLEDGMENTS

This work was partially funded by the European Commission in the context of the FP7 FASTER project (#287804).

REFERENCES

[1] L. Cheng and M. D. F. Wong, “Floorplan Design for Multimillion Gate FPGAs,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 25, no. 12, pp. 2795–2805, 2006.

[2] Xilinx Inc, “Floorplanning Methodology Guide ,” 2010.

[3] ——, “Partial Reconfiguration User Guide,” 2010.

[4] C. Bolchini, A. Miele, and C. Sandionigi, “Automated Resource-Aware Floorplanning of Reconfigurable Areas in Partially-Reconfigurable FPGA Systems,” in *FPL*, 2011, pp. 532–538.

[5] K. Vipin and S. A. Fahmy, “Architecture-aware reconfiguration-centric floorplanning for partial reconfiguration,” in *ARC*, 2012, pp. 13–25.

[6] A. Montone, M. D. Santambrogio, and D. Sciuto, “Wirelength driven floorplacement for FPGA-based partial reconfigurable systems,” in *IPDPS Workshops*, 2010, pp. 1–8.

[7] A. Montone, M. D. Santambrogio, D. Sciuto, and S. O. Memik, “Placement and Floorplanning in Dynamically Reconfigurable FPGAs,” *TRETS*, vol. 3, no. 4, p. 24, 2010.

[8] Y. Feng and D. P. Mehta, “Heterogeneous Floorplanning for FPGAs,” in *VLSI Design*, 2006, pp. 257–262.

[9] M. Rabozzi, J. Lillis, and M. D. Santambrogio, “Floorplanning for partially-reconfigurable fpga systems via mixed-integer linear programming,” in *FCCM*. IEEE, 2014, pp. 186–193.

[10] L. He, W. Liao, and M. R. Stan, “System level leakage reduction considering the interdependence of temperature and leakage,” in *DAC*, 2004, pp. 12–17.

[11] T. Tuan and B. Lai, “Leakage power analysis of a 90nm fpga,” in *Custom Integrated Circuits Conference, 2003. Proceedings of the IEEE 2003*. IEEE, 2003, pp. 57–60.

[12] K. Banerjee, A. Mehrotra, A. L. Sangiovanni-Vincentelli, and C. Hu, “On thermal effects in deep sub-micron vlsi interconnects,” in *DAC*, 1999, pp. 885–891.

[13] T.-C. Tai, “Power optimal partitioning for dynamically reconfigurable fpga,” in *ISIC*, 2012, pp. 37–40.

[14] S. Nishida, J. Eto, M. Amagasaki, M. Iida, M. Kuga, and T. Sueyoshi, “Power-aware fpga routing fabrics and design tools,” in *VLSI-SoC*, 2010, pp. 67–72.

[15] A. Vamos and M. Rencz, “Fpga power model for minimizing the thermal dissipation,” in *Thermal Inveatigation of ICs and Systems, 2008. THERMINIC 2008. 14th International Workshop on*. IEEE, 2008, pp. 148–151.

[16] V. Degalahal and T. Tuan, “Methodology for high level estimation of fpga power consumption,” in *ASP-DAC*, 2005, pp. 657–660.

[17] S. Bhoj and D. Bhatia, “Thermal modeling and temperature driven placement for fpgas,” in *ISCAS*, 2007, pp. 1053–1056.

[18] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, “VLSI module placement based on rectangle-packing by the sequence-pair,” *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 15, no. 12, pp. 1518–1524, 1996.

[19] C.-H. Tsai and S.-M. Kang, “Cell-level placement for improving substrate thermal distribution,” *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 19, no. 2, pp. 253–266, 2000.

[20] Gurobi Optimization Inc, “Gurobi optimizer.” [Online]. Available: http://www.gurobi.com/download/gurobi-optimizer