Congestion driven placement for mesh-based FPGA architecture with local interconnect

Zied Marrakchi
FlexRAS Technologies
Paris, france
zied.marrakchi@flexras.com

Mariem Turki
Habib Mehrez
LIP6, Paris 6
turki.mariem@lip6.fr

Jean-Luc Rebourg
CEA, DAM
Paris, France
rebourg.jean-luc@cea.fr

Mohamed Abid
CESlab
Sfax, Tunisia
mohamed.abid@ceslab.org

***Abstract*— In this paper we present an adaptation of a congestion driven placement technique to a Mesh based FPGA architecture containing a local interconnect connections. This techniques aims at spreading out congestion by considering white spaces and avoiding signals bounding boxes overlap. As shown in the experimentation section this technique reduces required routing channel width efficiently and consequently the device total area by 10% in average. In terms of circuit performance, we notice that congestion alleviation allows timing-driven router to reduce critical path delays despite wirelength increasing. This is due essentially to timing closure improvement.**

I. Introduction

Field programmable gate arrays (FPGAs) have become the key design and manufacturing vehicle for implementing a majority of digital circuits due to its advantages over application-specific integrated circuits (ASICs) such as lower depreciated mask costs and fast time to market. However, some features of FPGAs implemented with bulk-CMOS technology, especially radiation hardened performance, can still not meet the increased demands for nuclear, aerospace and military applications. In this work we propose a radiation hardened mesh based FPGA architecture fabricated in $65nm$ SOI ST technology. As shown in [1], the SEU (Single Event Upset) probability is proportional to the chip area exposed to radiation. Smaller the area is, less probability the SEU happens. In addition, hardening design has an area overhead. For example in [2] authors proposed a hardening technique with an area overhead of $38,3\%$. Thus it is very important to optimize device area to deal with this overhead and reduce the probability of SEU to happen.

Three factors combine to determine the characteristics of an FPGA: quality of its architecture, quality of the CAD tools used to map circuits into the FPGA, and its electrical design. The subject of this work is to propose a congestion aware placement technique to reduce device area. In fact interconnect is the dominant factor in term of area and occupies almost 80% to 90% [3]. Thus reducing routing channel tracks in an FPGA will effectively reduce the whole area.

To consider congestion in the placement objective function, we adapt the technique presented in [4]. In fact our architecture presents some particularities in terms of interconnect and logic blocks structures which induces some specific placement constraints.

II. FPGA architecture and configuration tools

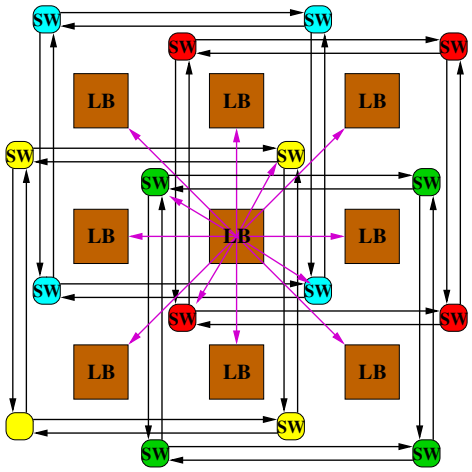

Fig. 1. Interconnet description

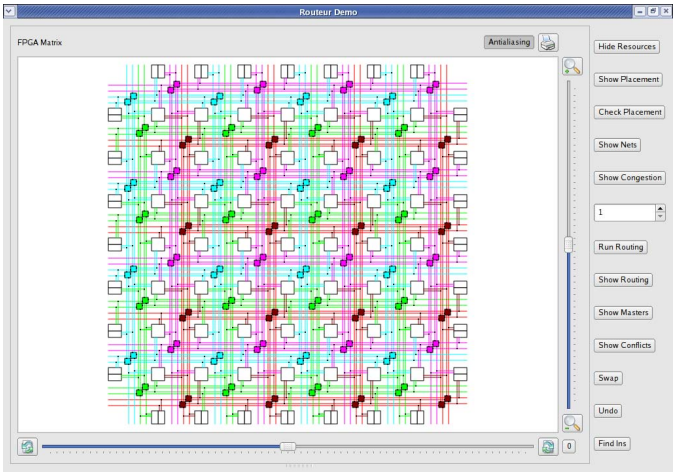

Fig. 2. FPGA architecture routing resources

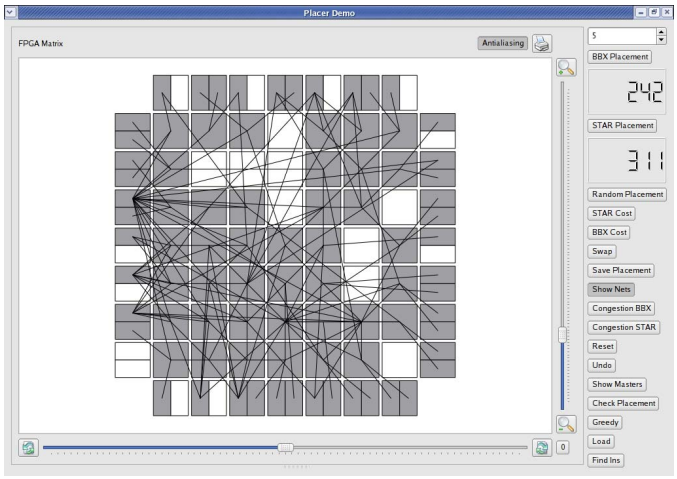

Fig. 3. Optimized netlist placement on FPGA architecture

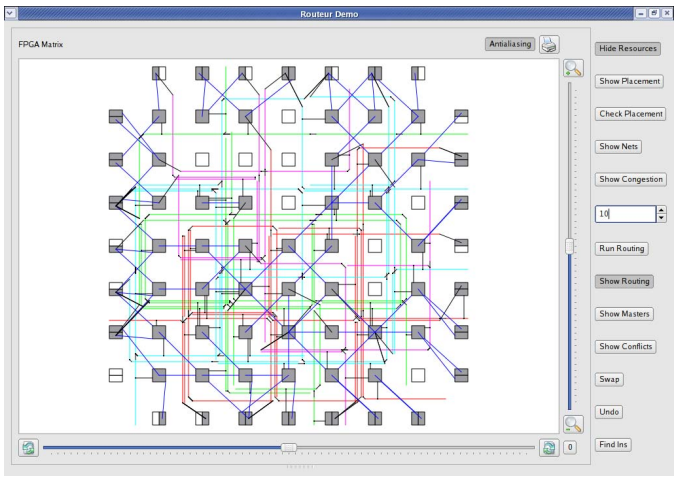

Fig. 4. Routed netlist on FPGA architecture

In this work, we propose a Mesh based FPGA architecture. Every CLB contains one LUT and a flip-flop for sequential functions. As shown in figure 1, there are two types of interconnection networks: global network and local network. The global network is composed of $4 * W$ disjoint sub-networks. W is the channel width. The length of the interconnection segments is equal to 2 which means that every segment spans 2 CLBs. All segments are unidirectionnal and driven by multiplexers. With the local network, each CLB can reach directly its 8 neighbors; this network provides fast connection and reduces congestion over the global network. The CLB inputs can be connected to the channel segment and its neighbors via a 50% populated connection block.

III. Placement objective

To implement a netlist application on the architecture, we use ABC [5] and TV-pack [6] to run mapping and packing respectively. We developped a VPR-like placer and router adaped to architecture particularities. Figures 2, 3 and 4 show screen shots of the developped placement and routing tools. The placer uses the simulated annealing schedule which is an adaptation of the Huang algorithm [7].

To estimate congestion in the cost function, we used an adaptation of the technique proposed in [8]. We compute the congestion coefficient and we multiply it by the sum of the half perimeters of the bounding boxes. The new cost function is:

$$Cost = CC^K * \sum_{i=1}^{Nnets} q(i) * [bbx(i) + bby(i)] \quad (1)$$

Unlike the coefficient expression used in [9], we propose the following expression:

$$CC = \left(\frac{\sum_{(i,j)} U_{i,j}^N}{nx * ny} \right) / \left(\frac{\sum_{i,j} U_{i,j}}{nx * ny} \right)^N \quad (2)$$

where $U_{i,j} = \sum_{CLB_{i,j} \in BB_{Net}} W_{net}$

$U_{i,j}$ is the congestion of each logic block and which corresponds to the sum of the congestion values of all bounding boxes which include this block. The parameter K allows to controle the weight of the congestion coefficient. According to equation (1), the cost function has two terms: the congestion coefficient and the sum of the half perimeters of the bounding boxes of all nets (wire length). In order to give more importance to congestion, we increase the value of K.

In equation (2), the parameter N is the weight that impacts the imbalance between the congestion related to each logic block. The congestion parameter of every net is noted W_{net}. In [4] authors extended the expression proposed in [8] and related it to bounding boxes area. They assigned to every signal a value which represents the amount of wires used by this signal to connect all its terminals. This value is added to all logic blocks included in the bounding box of the considered signal.

The expression of the wirelength per area used in [4] is:

$$W_{net} = L/A \quad (3)$$

where A is the area of the bounding box, and L is the amount of wire used by the signal to connect all its terminals. In [8], L is defined as follows:

$$L = \frac{1}{2} * BB + \beta * q \quad (4)$$

where BB is the perimeter of the bounding box, β is a tuning parameter and q is a fanout correction factor. As we notice here, L is only an estimation of the amount of wire.

As described previously, the architecture that we propose contains two networks: local and global. Local resources consist of direct connections and are reserved to neighboring blocks communication. Thus, signals between a driver and neighboring receivers are routed using local resources and do not create congestion for the global network.

Thus, to take this architecture specificity into account, we propose, another way to compute the amount of wire used by every signal. This is described by the following algorithm:

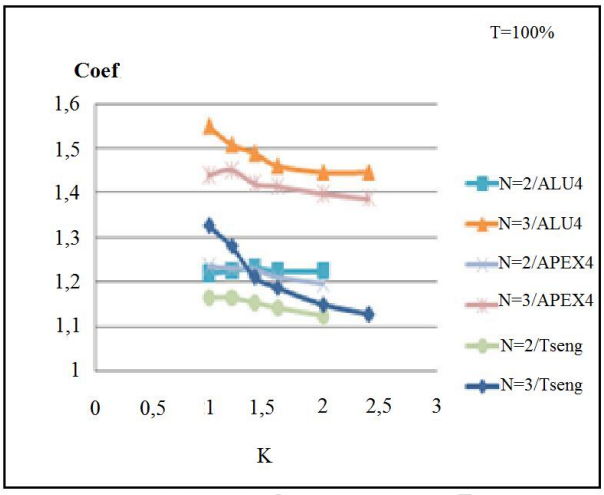

Fig. 5. The effect of CC weight (K) on congestion reduction

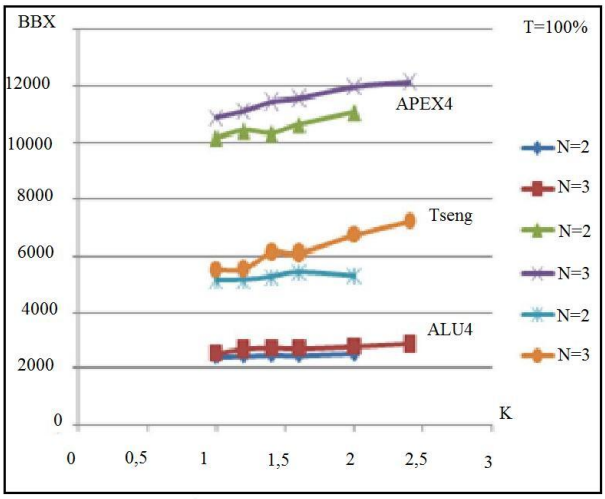

Fig. 6. The effect of CC weight (K) on wire length reduction

```
For every Net "N"{
    get the driver of "N"
    x_d=driver->x()
    y_d=driver->y()
    For each receiver rec {
        x_r=rec->x()
        y_r=rec->y()
        If(rec is not a neighbor of driver){
            L+=abs(x_d - x_r)+abs(y_d - y_r)
        }
    }
}
```

The idea is to compute the distance between the driver of the net and all its receivers except the receivers which are in the neighborhood. Indeed, the local network allows a direct connection between the neighbor blocks. Thus, we do not consider them when we estimate congestion.

IV. Experimental Results

To evaluate the efficiency of the placement technique, we place and route the largest MCNC benchmark circuits [10] on the proposed architecture.
We also evaluate the effect of each parameter of the cost function on congestion estimation.

A. Weightening parameter "K"

First, we analyze the behavior of the congestion coefficient CC when we vary the parameter K (equation (1)).
As shown in figure 5, we notice that CC decreases when we increase the parameter K. In fact, increasing K gives more priority to the congestion coefficient in the objective function.
Figure 6 shows that when we increase the K value the bounding box is increasing since the congestion is more and more taken into consideration and the bocks are more and more spread out over the whole chip. Thus, if we increase indefinitely the parameter K, the wire length increases, and the performance of the circuit may be impaired.

Thus, it is important to find a tradeoff between the way K is increased and the sum of bounding box decreasing. For this purpose, we propose the following expression for the parameter K:

$$K = 1 + SuccRatio \tag{5}$$

where SuccRatio is the rate of the accepted moves in one simulated Annealing loop (in the inner loop). At the beginning of the placement, almost all moves are accepted since the temperature has a big value. So the SuccRatio is almost equal to 1. At this stage, the weight has the highest value which is equal to 2. As the placement is proceeding, the temperature is decreased, and the rate of the accepted moves decreases, so the coefficient is also decreasing as the placement is more and more refined. At the last iteration of the placement the weight is almost equal to 1.
This new expression of the weight parameter K keeps the same gain in terms of congestion and reduces the wire length by 7,3% for ALU4 bench and 12,3% for TSENG bench.
Another conclusion is that the congestion coefficient depends on the occupancy rate variation which is the ratio between netlist instances and architecture logic blocks. In figure 7 we notice that congestion coefficient decreases when the occupancy ratio (T) decreases. This means that when we increase the number of blocks, the empty space also increases and consequently netlist instances are more spread out over the entire chip which reduces congested zones. To take advantage of empty regions in low occupancy cases, we integrate the occupancy ratio in the expression of the parameter K as shown in equation 6. The introduction of this new parameter allows

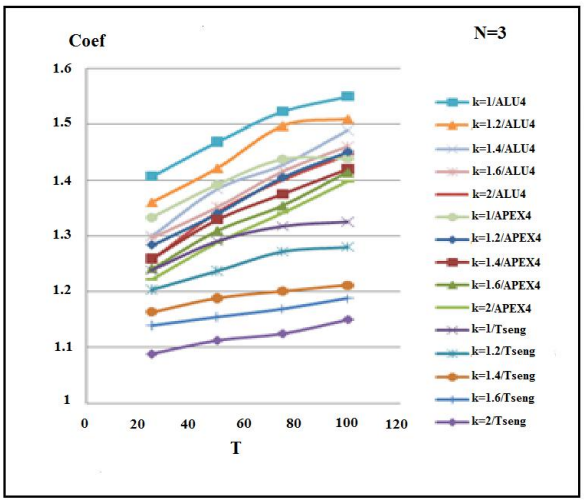

Fig. 7. The effect of occupancy on congestion reduction

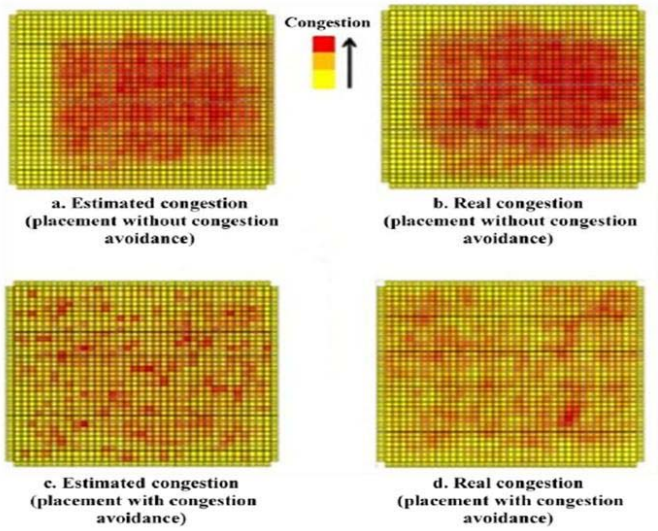

Fig. 8. Congestion maps

to take empty logic blocks into account.

$$k = \frac{1 + SuccRatio}{T} \tag{6}$$

with $T = \frac{NB_{instances}}{NB_{blocs}}$. When we increase the number of the blocks in the FPGA architecture, the congestion weight K is increasing, and we make more effort to reduce congestion. Thus, the circuit is more and more spread out over the entire area.

B. Weightening logic blocks parameter "N"

The parameter N impacts the imbalance between the congestion coefficients of every logic block. It increases the impact of highly congested blocks. We made several experiments in order to get the most adapted value of N to the congestion expression. Thus, we tried with n=1, 2, 3, 4 and 5. Experimentally we found that N=3 gives the best results. So the final coefficient expression is:

$$CC = \left(\frac{\sum_{(i,j)} U_{i,j}^3}{nx * ny}\right) / \left(\frac{\sum_{i,j} U_{i,j}}{nx * ny}\right)^3 \tag{7}$$

C. Congestion Estimation Quality

To generate visual congestion map comparisons, we presented the congestion maps of the same design before and after the routing (real congestion) using a temperature scale (yellow=low, red=high congestion). The estimated congestion maps appear in figure 8(a) and figure 8(c) which represent respectively the congestion map of the placed netlist using the default objective cost function (only signals bounding boxes optimization), and the congestion map of the placed netlist using the congestion objective function. The real congestion maps appear in figure 8(b) and figure 8(d) which represent respectively the congestion map of the placed netlist using the default objective cost function, and the congestion map of the placed netlist with the congestion objective function. The real congestion is evaluated after routing based on the number of used tracks in each routing channel.

According to figure 8 we notice that the estimation maps figure 8(a) and figure 8(c) are correlated to the real congestion maps respectively in figure 8(b) and figure 8(d). This means that the new congestion parameter included in the objective function gives a good estimation of the distribution of the congestion in the entire chip even before routing. In figure 8(a) and figure 8(b), the congestion is concentrated at the center of the FPGA chip. This is obvious since the objective cost function aims at bringing closer the blocks which communicate together in order to decrease the sum of the half perimeter of all signals bounding boxes. In the other side, in figure 8(c) and figure 8(d), the distribution of congestion is balanced over the entire chip and there is less congested regions compared to the figure 8(a) and figure 8(b). This balance induces a decrease of the required channel width and consequently a reduction of the required routing resources.

D. Area Optimization

We tested the impact of the congestion aware placement on the area. In our experimentation, we use the largest MCNC Benchmarks[10]. We place all benches with the proposed congestion aware technique and the common wire length based technique. Then we route all benches using a timing driven router with the smallest channel width. Finally, we evalute the resulting device area and the critical path delays. Area and timing charaterization are made on a real layout. The FPGA device is fabricated in $65nm$ SOI ST technologies. In table 1, area is computed in a symbolic unit λ. To get real values we replace it with the value $65nm$.

For all benches we use architectures with 75% logic occupancy. We notice that in most cases we obtain a reduction of area reaching in average 10%. In addition we notice that with the new objective function we obtain smaller delays, this is

MCNC		BBX Placement		Congestion Placement		Gain	
Circuits	Luts	Area (λ^2) $\times 10^6$	Delay (ns)	Area (λ^2) $\times 10^6$	Delay (ns)	Area %	Delay %
alu4	606	319	8,3	290	8,4	9	-1
apex2	1919	1541	9,2	1356	8,5	12	7
apex4	1290	1092	9,6	950	9,3	13	3
bigkey	1707	1065	6,6	947	7	11	-6
clma	8383	7672	19,2	6521	17,5	15	8
des	2092	2047	8	1739	8,1	15	-1
diffeq	1599	954	5,8	849	6,2	11	-6
dsip	1796	934	6,9	934	6,1	0	11
elliptic	3848	2883	12,6	2652	12	8	4
ex1010	4589	3763	16,2	3424	15	9	7
ex5p	1135	915	7,1	786	7,5	14	-5
frisc	3691	3287	12,2	2695	11	18	9
misex3	1425	1085	7,5	987	6,8	9	9
pdc	4575	4889	18,1	4497	18,2	8	0
s298	1940	1192	13,8	1060	13	11	5
s38417	6406	4662	9,6	4102	7,2	12	25
s38584	6447	4590	10,4	3855	8,1	16	22
seq	1826	1411	10,1	1411	10	0	0
spla	3690	3448	15,3	3068	11	11	28
tseng	1220	665	5,2	665	5,3	0	-1
Average	3009,2	2420,7	10,6	2139,4	9,81	10	6

TABLE I

THE IMPACT OF CONGESTION AWARE PLACEMENT ON AREA AND PERFORMANCES

obtained thanks to timing closure improvement. In fact with less congestion the timing driven router has more flexibility and choices to route critical signals.

V. CONCLUSION AND FUTURE WORK

In this paper we present an adaptation of a technique of congestion aware placement for Mesh-based FPGA architecture containing local routing resources. We tuned experimentally the objective function parameters to get a good tradeoff between congestion optimization and the wire length reduction.

We conclude that by increasing the logic blocks number, we can better spread out congestion and utilize the interconnect resources more efficiently. Most of the benchmarks were routed with a channel width smaller than the one used to route a circuit placed to optimize wire length.

REFERENCES

[1] R. C. et al., "A New Approach to Single Event Effect Tolerance Based on Asynchronous Circuit Technique," in *Journal Electron and Test*, 2008, pp. 57–65.

[2] Q. Zhou and K. Mohanram, "Gate Sizing to Radiation Harden Combinational Logic," in *IEEE Transactions On Computer-Aided Design of Integrated Circuits and Systems*, 2006, pp. 155–166.

[3] A. DeHon, "Balancing interconnect ans computation in a reconfigurable computing array," in *Proceedings of the 1999 ACM/SIGDA seventh international symposium on Field programmable gate arrays*. ACM Press New York, NY, USA, 1999, pp. 69–78.

[4] D. C. D. Yeager and G.Lemieux, "Congestion estimation and localization in fpga: a visual tool for interconnect prediction," *In International Workshop on System Level Interconnect Prediction*, pp. 33–40, 2007.

[5] A. M. et al., "An Integrated Technology Mapping Environment," in *in International Workshop on Logic and Synthesis,*, 2005, pp. 383–390.

[6] V. Betz and J. Rose, "VPR: A New Packing Placement and Routing Tool for FPGA research," *International Workshop on FPGA*, pp. 213–22, 1997.

[7] A. S.-V. M. Huang, F. Romeo, "An Efficient General Cooling. Schedule for Simulated Annealing," in *Proc. Of the IEEE ICCAD*, 1986, pp. 381–384.

[8] H. L. Yue Zhuo and S. Mohanty, "A congestion driven placement algorithm for fpga synthesis," *In international Conference on Field Programmable Logic and Application*, pp. 1–4, 2006.

[9] D.Chiu, "Congestion-driven re-clustering cad flow for low-cost fpga," 2009.

[10] "Layout synthesis benchmark set, microelectronics center of north carolina, research triangle park, nc, may 1990."