# A Power-Aware Algorithm for the Design of Reconfigurable Hardware during High Level Placement

Wing On Fung[1]
*School of Electronics and Engineering1,*
*University of Edinburgh, King's Buildings,*
*Mayfield Road, Edinburgh, EH9 3JL, UK*
*wing.fung@ed.ac.uk*

Tughrul Arslan[1,2]
*Institute for System Level Integration2,*
*The Alba Centre, Alba Campus,*
*Livingston, EH54 7EG, UK*
*tughrul.arslan@ed.ac.uk*

## Abstract

*The popularity of reconfigurable logic devices and portable hardware demands ever increasingly power saving schemes for low power designs. This paper looks at the CAD design process of reconfigurable devices and presents a novel method to gain power savings during the placement stage of the CAD flow. The proposed system modeled the number of switches used in the circuit and employed simulated annealing algorithm to reduce the overall routing power. The system was tested against 8 large benchmark circuits. It was able to achieve a routing power saving of up to 18% compared with cases without modeling the switches.*

## 1. Introduction

Current design tools for reconfigurable architectures take measures of criteria such as total wirelength, critical paths and chip area to reduce power consumption and increase timing performance. Most of the optimisations were done late into the design stage such as routing and thus sometimes failed to model the architecture in a more accurate light in earlier stages of the design flow. Power analysis of configurable switches is usually implemented during the routing and mapping stages and has been largely ignored during the placement stage of the design due to the inaccuracy associated with power estimation at high level design process. This paper introduces a novel scheme which estimates the number and types of switches that the circuit is likely to use during high-level placement. The system then uses this information to search for an architecture that could potential use less routing resources and hence produce devices with lower power consumptions.

The following describes the methods used and investigates whether and how effective this approach is in terms of power saving. Section 2 gives an overview of the related works in the field. Section 3 describes the design model in which the technique is based on. Section 4 explains the principle of the algorithm. Section 4 presents the experimental result and analysis of the data. Finally, section 6 concludes the paper and discusses future works.

## 2. Background

The use of heuristics such as simulated annealing [1] to search for good placements has been widely exploited. The goals of these heuristics can be categorised into three main areas: timing driven, congestion driven and wirelength driven placements. There are in general two types of timing-driven algorithms, path-driven and net-driven timing placement.

Path-driven [2, 3, 4] timing placement tries to minimize the longest path delay directly. For an initially placed circuit, the critical path is found. The length of the path is the cost function in which the heuristic is trying to reduce. The critical paths will then be updated repeatedly for a number of iterations of placements. The algorithm stops when no further improvement can be made to its cost function. It provides a good estimation of critical path delay, however it will also consume a lot of computation time to keep track of all the critical paths.

Net-driven [5, 6] timing placement on the other hand is less complex and can be more efficient. It does not account for physical length of the net and assign each net with a weight that reflects its data dependence or timing criticality. Although it is less accurate in wirelength estimation, programmes such as [7] make use of this to derive more effective cost functions.

There are also routability-driven [8, 9] placement tools. The cost function can be modeled as topology free, where no routing is done during evaluation, such as bounding box and Rent‘s rule based modeling [10].

Second NASA/ESA Conference on Adaptive Hardware and Systems(AHS 2007)
0-7695-2866-X/07 $25.00 © 2007 **IEEE**

It can also be topology based modeling. A Steiner tree is built for every net and the fall in efficiency is compensated by the more accurate estimation of routing congesting.

There are currently commercial [11] and academic [5, 7] placement tools available which are driven by varies algorithms. VPR [5] uses bounding box [12] to estimate the total wire length, and uses path-driven timing analysis with simulated annealing to generate its placement. PATH [5] is an enhancement to VPR. It uses a different net weighting algorithm to VPR, and is implemented by accurately counting the number of paths and assigning the weight of the net according to the number of paths the net is in.

The performances of those tools were tested on T-PEKO [13]. Its algorithm creates circuit like hierarchies, in which the optimal delays are known to test those placement tools. It was found that the delay produced by the current tools are between 10% to 18% away from the optimal critical path delay, hence a significant improvement can still be made from the current available tools.

Work [14] has also been done on power aware algorithms. During the packing stage and placement stages, information on switching activities as well as bounding box is exploited to reduce power consumption. This however does not distinguish the routing resource from wire to switches during placement but does produce a more power aware design than preview works.

There are also other low power techniques for the design of reconfigurable logics. [19] uses voltage scaling within their routing architecture to reduce interconnect power. [20] employs different types of switches in different area of the same array to increase the efficiency of the circuit. [21] also optimises for power by the use of voltage scaling, but it is instead implemented during clustering.

## 3. Design Flow

The design flow of the system is illustrated in Figure 1. Firstly, the system inputs a technology-free netlist. It works out the switching activities on every node on the netlist and generates an activity file. This file will be used in later stages to evaluate power consumption of the circuit. Then the netlist is packed into clusters of logic elements, the size and architecture of which is specified by the designer. The clusters are then placed and routed with the aim of reducing wirelength and shortening the critical path.

### 3.1 The Model

More than 60% [15] of the total power consumed by FPGAs is due to power dissipation through routing resources. With the increasing popularity and ease of use of portable reconfigurable hardwires, this gives the motivation to seriously consider minimising power consumption as a primary objective. Accordingly, the model being used should reflect the task of minimising routing resource whether it is the length of wires or the number of switch boxes used. The placement architecture is assume to be an island-style [16] array with s-box switches and c-box switches connecting the tracks and pins. For the sick of simplicity, the switches used in the experiment are universal tri-state buffers.

### 3.2 Calculating Power Consumption using Powermodel

The Powermodel [17] evaluates the power of a placed and routed design principally from estimating its switching activities at every node. It uses an activity estimator to work out the possibility of a node switching, the information is passed on to a packing tool TV-pack before being placed and routed in VPR as described in Figure 1.

The placement tool uses simulated annealing to search for a placement for a minimum delay cost and shortest total wirelength of a design. The only function within the algorithm that could reduce power is the bounding box function. A bounding box [12] of a net is the rectangular box that contains the span of the net. The algorithm searches for small bounding boxes for all nets to ensure minimal wires will be used when the array is routed.

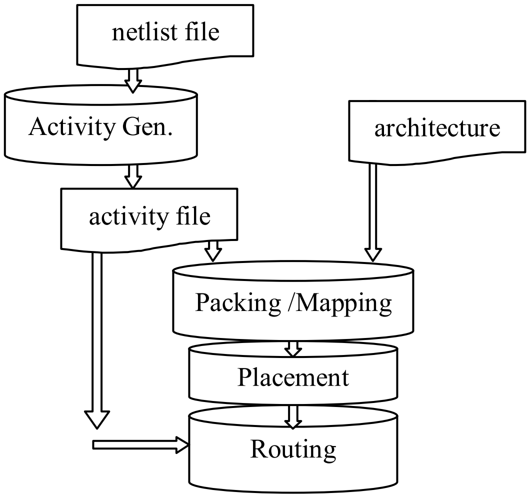

**Figure 1. System for evaluation power consumption**

Following placement and routing, in order to estimate the power consumption for a specific net, it records all the tracks that the net has to go through. For each of those tracks, the capacitances of the connected segments are recorded. The internal capacitance of the switches connecting the wire segments and switches connecting the pins and segments are also added to give an overall capacitance of the connection. In the case of the test we performed, all switches are tri-state buffers, and there was no distinction between s and c switch boxes. The power can then be found from the capacitance information and the switching activities on that net. The leakage power of the unused switches is also added to the overall routing power.

## 4. Modelling Switches in Placement

To take account of power consumption during placement stage, we consider the routing architecture in more details during placement. In order to reduce the capacitance of a net, apart from reducing the length of the net, it is also necessary to reduce capacitance of the switches used in the net. The new function looks at the placement of the pins of each connection and deduces how many switches it is likely to use. In reducing the number of connecting switches or the number of high capacitance switches used, it can prevent wastage in the routing stage and hence routing power can be reduced.

In the case of Figure 2, the connecting pins are directly opposite each other, hence they can be connected using only one c-box switch.

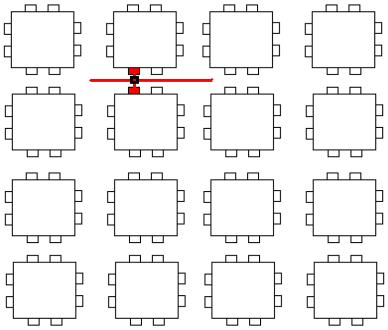

**Figure 2. switch usage – case 1**

In the case of Figure 3, the locations of the pins of the two connecting blocks are on the same axis of wires. For this connection, a minimum of two c-box switches are required to connect the pins together, provided the wire segments are long enough.

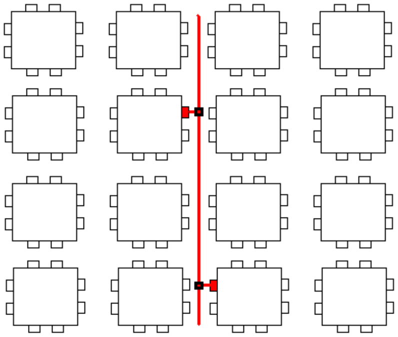

**Figure 3. switch usage – case 2**

In the case of Figure 4, two tracks are required to connect the pins. Hence using both c-box and s-box switches, the minimum number of switches required are three.

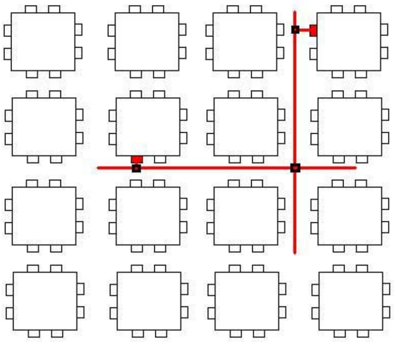

**Figure 4. switch usage – case 3**

In the case of Figure 5, the net requires at least three tracks to connect. Hence it uses a minimum of four switches, two of each type. This switch counting information for every connection is added during the placement stage for search.

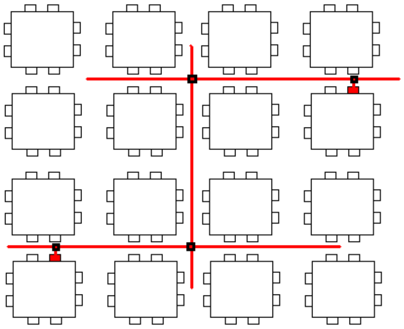

**Figure 5. switch usage – case 4**

The weighting of the switch boxes used are integrated into the bounding box function of the VPR. For a single net, the total weight is the bounding box value of the net, and bounding box times a weighting depending on the case and the type of switches used for each two point connections in the net.

$$Fitness = \sum_{i=1}^{inet} \left( bounding\ box \right) \cdot \left( 1 + C_1 n_1 + C_2 n_2 \right)$$

Where C1 and C2 are the weight coefficient associated with switch type 1 and 2, n1 and n2 are the number of switches that the net requires to route.

## 5. Experiment

The switch counting function was tested against eight circuits of the MCNC LGSynth93 [18] benchmark. The architecture uses tri-state buffers as switches for both s-box and c-box switches. The architecture contains the same ratio of track length. Each design's switching activities were estimated and packed using the previously mentioned activity generator and packing tool. They are each placed five times from different starting points in the algorithm search. The resulting placements are then routed using the same routing tool in VPR. For comparison, the circuits are also placed and routed from the same starting point without using the switch counting function. The results are shown in Figure 6 and 7.

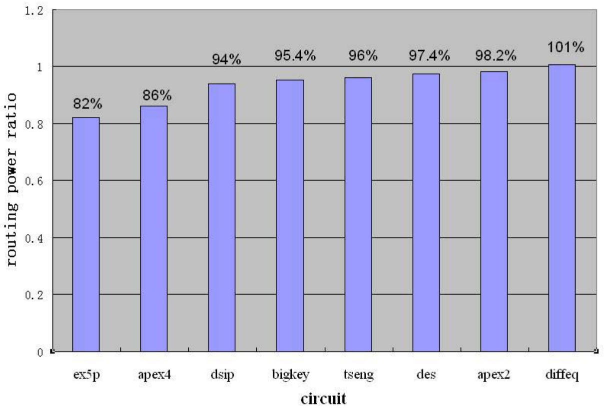

**Figure 6. comparison of minimum routing power of switch counting scheme with normal scheme**

Figure 6 shows the comparison between the routing power of placements with and without the switch counting function. The minimum power achieved from the two set of runs were selected and the graph shows the routing power from the switch counting scheme divided by normal scheme. A ratio of less than 100% would indicate a routing power saving using switch counting function.

In most cases, there are routing power savings with the exception of diffreq. The efficient of the switch counting function is depended upon the track ratio of the architecture. If the segment length of each track is one, the number of switches used will be directly proportional to the length of wire and hence the switch counting function becomes abandon. Therefore, no power saving can be achieved. Hence finding a suitable architecture for the design of each circuit is important when using this function.

Figure 7 shows the average of five runs for the 8 circuits. It is able to achieve a minimum average of 15.6% routing power saving for ex5p and a maximum of 2% for bigkey. The success of the switch counting function relies greatly on the coefficients chosen. If the coefficients are too large and the weightings of the switch counts are too high, it will disrupt the algorithm to a point where no significant wirelength improvement can be made. Hence finding a good balance for different switches and circuits prior to simulation is important for saving routing power and is part of an ongoing study.

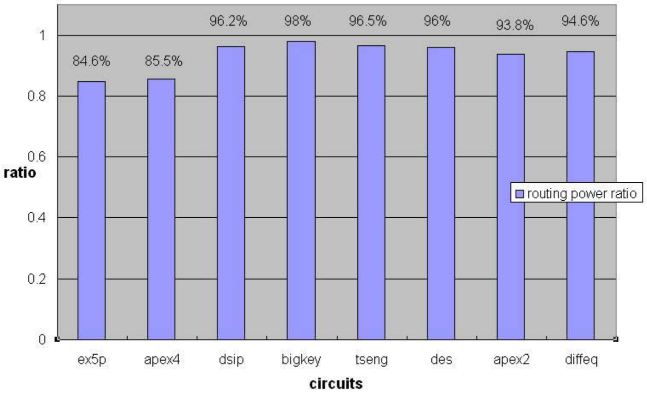

**Figure 7. comparison of average routing power of switch counting scheme with normal scheme**

## 6. Conclusion

This paper has presented a switch modeling technique where it examines the routing resource early in the placement stage of designing reconfigurable devices to save routing power. It has been tested against 8 benchmark circuits and a routing power saving of at most 18% was found. It demonstrated the advantages of modeling reconfigurable devices more accurately and early on in placement stage and results in higher performance and better designs.

## 7. References

[1] C-C. Chang, J. Cong, Z. Pan, and X. Yuan, Physical hierarchy generation with routing congestion control, in Proc. Int. Symp. on Physical Design, April 2002, pp. 36 - 41.

[2] Takeo Hamada, Chung-Kuan Cheng, and Paul M. Chau, A Timing-Driven Placement Tool Using A Piecewise Linear Resistive Network Approach, 30th Conference on Design Automated Conference, 14-18 June 1993, pp. 531 - 536.

[3] Mathur, A.; Liu, C.L., Compression-relaxation: A New Approach To Performance Driven Placement For Regular Architectures, IEEE/ACM International Conference on Computer-Aided Design, 6-10 November 1994, pp. 130 – 136.

[4] W. Swartz and C. Sechen, Timing-driven placement for large standard cell circuits, in Proc. ACM/IEEE Design Automation Conference, 1995, pp. 211-215.

[5] V. Betz and J. Rose,VPR: A New Packing, Placement and Routing Tool for FPGA Research, Seventh International Workshop on Field-Programmable Logic and Applications, 1997, pp. 213 - 222.

[6] H. Eisenmann and F. M. Johannes. Generic global placement and flootplannina, in Proc. 35th A CM/IEEE Design-Autom&on Conference, 1998, pp. 269-274.

[7] X Kong, T., A novel net weighting algorithm for timing-driven placement, IEEE/ACM International Conference on Computer Aided Design, 2002, ICCAD 2002, 10 - 14 Nov 2002, pp. 172 – 176.

[8] Chen Li, Min Xie, Cheng-Kok Koh, Cong, J., Madden, P.H., Routability-driven placement and white space allocation, IEEE/ACM International Conference on Computer Aided Design, 2004. ICCAD-2004, 7-11 Nov. 2004, pp. 394 – 401.

[9] C.-C. Chang, J. Cong, D. Pan, and X. Yuan, Multilevel global placement with congestion control, IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol. 22, no. 10, October 2003, pp. 1424 - 1431.

[10] X. Yang, R. Kstner, and M. Sarrafzadeb, Congestion estimation during top-down placement, IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol 21, Issue 1, Jan. 2002 pp. 72 –80.

[11] W. Swartz and C. Sechen, Timing-driven placement for large standard cell circuits, in Proc. ACM/IEEE Design Automation Conference, 1995, pp. 211 - 215..

[12] C.-L.E. Cheng, RISA: accurate and efficient placement routability modeling, in Proc. IEEE/ACM International Conference on Computer-Aided Design, Nov 1994. pp. 690-695.

[13] Chin-Chih Chang, Cong, J., Romesis, M., Min Xie, Optimality and scalability study of existing placement algorithms, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Volume 23, Issue 4, April 2004 pp. 537 – 549.

[14] Lamoureux, J., Wilton, S.J.E., On the interaction between power-aware FPGA CAD Algorithms, International Conference on Computer Aided Design, 9-13 Nov 2003, pp. 701 - 708.

[15] Balancing interconnect and computation in a reconfigurable computing array, ACM/SIGDA Int. Symp. on Field Programmable Gate Arrays, 1999, pp. 69 - 78.

[16] Yen-Tai Lai, Ping-Tsung Wang, Hierarchical interconnection structures for field programmable gate arrays, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, June 1997, Pages:186 – 196.

[17] K. Poon, A. Yan, S.J.E. Wilton, A Flexible Power Model for FPGAs, International Conference on Field-Programmable Logic and Applications, Sept 2002, pp. 48-58.

[18] LGSynth93 MCNC Benchmarks. Obtained from http://www.eecg.toronto.edu/~lemieux/sega/ccts_blif.tar

[19] Mondal, S.; Memik, S.O., A Low Power FPGA Routing Architecture, IEEE International Symposium on Circuits and Systems, 23-26 May 2005, pp.1222 - 1225 Vol. 2.

[20] Siozios, K.; Soudris, D.; Thanailakis, A., A Novel Methodology for Designing High-Performance and Low-Power FPGA Interconnection Targeting DSP Applications, IEEE International Symposium on Circuits and Systems, 21-24 May 2006, pp. 4383 – 4386.

[21] Deming Chen; Cong, J., Delay Optimal Low-Power Circuit Clustering for FPGAs with Dual Supply Voltages, Proceedings of the 2004 International Symposium on Low Power Electronics and Design, 9-11 Aug 2004, pp. 70-73.

0-7695-2866-X/07 $25.00