PIN COUNT PREDICTION IN RATIO CUT PARTITIONING FOR VLSI AND ULSI

Dirk Stroobandt

University of Ghent, ELIS Department
Sint-Pietersnieuwstraat 41
B-9000 Gent, Belgium
dstr@elis.rug.ac.be

ABSTRACT

Partitioning is an important step in computer-aided design. The 'ratio cut' bipartitioning algorithm [1] is known to be one of the best partitioning algorithms. It partitions a circuit into two (disjoint) modules by cutting some of its nets. Based on theoretical arguments, the cost function that is minimized for finding the best partitioning structure in the circuit is set to be the ratio of the number of nets cut to the product of the module sizes. In previous work [2], we showed that the ratio should contain the number of pins created by all cuts instead of the number of cuts itself to include multi-terminal nets properly. In this paper, we propose a change of the ratio denominator by introducing an accurate estimate of the number of pins resulting from the cuts. We suggest to use Rent's rule for pin count prediction. Results show improvements over the classical 'ratio cut' algorithm.

1. INTRODUCTION

The production of VLSI and ULSI computer chips requires the layout (placement and routing) of the chip design on a carrier. The huge amount of components in present-day circuits requires a partitioning of the circuit into smaller modules for being able to cope with the "size" of the circuit in the layout steps. Partitioning is also necessary to implement circuits that are too large to fit in a single chip, in multiple chips (MCM's) or multiple FPGA components. Apart from this high-level use of partitioning for floorplanning, an equally important application of partitioning is to be found in the placement and routing steps. For the high demands put on system performances these days and for the computer-aided design tools (CAD tools) for placement and routing to be good enough, accurate predictions of system performances are badly needed to limit the search in the vast solution space [3, 4]. These CAD tools therefore use estimator tools [5], usually based on partitioning methodologies [6]. The better the partitioning tool, the better the estimates.

Literature on partitioning is abundant. A very good general overview of partitioning methodologies is presented by Alpert and Kahng [7] and a comparison between several partitioning methods has been presented by Hagen et al [6]. They concluded that the so called 'ratio cut' partitioning method gave the best results. An implementation of 'ratio cut' has been presented by Wei and Cheng [1].

Although the 'ratio cut' algorithm of Wei and Cheng is far better than most other algorithms, it still has a few deficiencies. The method tries to minimize the number of nets "cut" in order to find the intrinsic clustering structure of the circuit. In [2], Stroobandt showed that minimizing the number of cuts does not take multi-terminal nets into account correctly. The real objective of partitioning is to minimize the information flow between the partitioning modules, hence the number of new pins generated by the cut. For point-to-point interconnections, both criteria (minimizing the number of cuts versus minimizing the number of new pins generated by the cuts) are equivalent. This no longer holds for multi-terminal nets.

The ratio used in 'ratio cut' is based on the prediction of the number of nets cut for a circuit with uniform interconnection structure. However, we know that circuits do not have uniformly distributed interconnection structures. In fact, the hierarchical clustering, intrinsicly present in any real circuit, shows an interconnection structure that follows the well-known Rent's rule [8]. In this paper, we use Rent's rule to obtain a more accurate estimate of the number of pins generated by cutting nets and we apply this pin count prediction in the ratio cost function.

The remainder of this paper is organized as follows. The circuit and the partitioning models are explained in the next section. The introduction of the pin count cost function is briefly described in section 3 and section 4 elaborates on taking into account the Rent behaviour of circuits. The results are presented in section 5 and a conclusion follows at the end of this paper.

2. MODEL FOR THE CIRCUIT AND FOR THE PARTITIONING PROCESS

A circuit can be represented by a set of interconnected blocks as in figure 1 (the blocks can be the representation of transistors, gates, or even entire circuits). An interconnection between blocks is called a *net*. A net that is connected to more than two blocks is called a *multi-terminal net*. Some of the nets are also connected to the outside of the circuit. These are called *external nets* (as opposed to *internal nets* which only connect blocks within the circuit). In order to model these external nets properly, we introduce a new kind of block and call it a *pin*. A pin models the external terminal for the net. The other blocks are called *logic blocks*. Every external net is connected with exactly one pin. Note that the number of pins thus equals the number of external nets.

Partitioning a circuit means dividing this circuit into disjoint subcircuits (called *modules*), each containing a subset of the blocks (figure 2). This partitioning is done using some kind of criterion. Generally, the criterion is to minimize the number of nets crossing the borders of modules in the partition. Nets that are cut by mod-

The author is Postdoctoral Fellow of the Fund for Scientific Research of Flanders (Belgium).

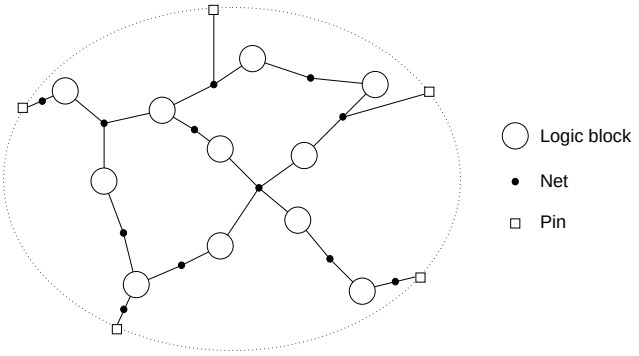

Figure 1: Model of a circuit.

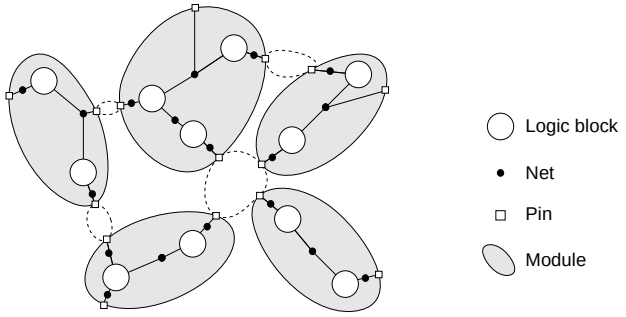

Figure 2: Partitioning the circuit of figure 1 into modules.

ule boundaries are shared between two or more modules and are said to be external to the modules. Therefore, the net is split into a number of subnets, one for each module that shares the net. A new pin is assigned to each subnet. Each module can then itself be seen as a circuit and can be partitioned further. A partitioning process where the modules themselves are recursively partitioned is called a *hierarchical partitioning method*.

3. THE RATIO CUT ALGORITHM WITH A RATIO VALUE BASED ON THE PIN COUNT

The 'ratio cut' algorithm is based on the notion that circuits are generally designed in a hierarchical way and thus hold an intrinsic partitioning structure. Finding this structure is important since the quality of the partitioning result depends on the final integration of all partition levels, from the basic subcircuits to the entire system. A greedy approach to find a minimum cut at a specific level may sacrifice the quality of cuts for the following levels. For this reason, Wei and Cheng suggested a bipartitioning algorithm (the circuit is partitioned in two parts) that uses what they call the "*ratio cut*". The *ratio* of a cut is defined as $R_{A,A'} = C_{A,A'}/(|A||A'|)$, where A and A' denote the partitioned modules (with cardinalities $|A|$ and $|A'|$) and $C_{A,A'}$ denotes the *cut capacity*, i.e., the number of nets that are cut.[1] The *ratio cut* is the cut that generates the minimum ratio among all cuts in the network. In [1], Wei and Cheng showed that this cost function alleviates the hidden size effect (unbalanced partitioning modules lead to a significantly lower cut capacity but

[1] In the most general case, the net weights are also included.

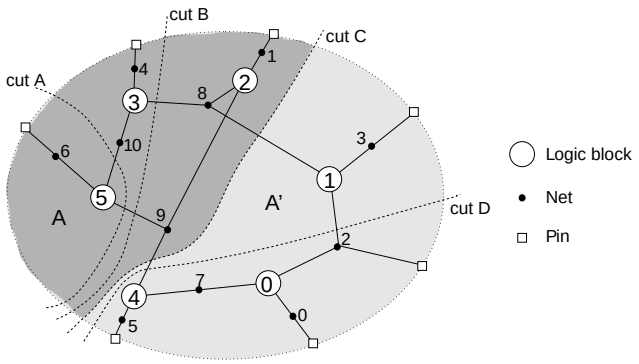

Figure 3: Moving gates 3 and 2 from A' to A (from cut A, over cut B, to cut C) results in the best cut for this simple example circuit.

generally do not follow the intrinsic clustering structure).

Wei and Cheng described a heuristic algorithm for finding the ratio cut in a netlist of a circuit [1]. The algorithm is move-based, i.e., it iteratively moves logic blocks from one module to the other. Each time a logic block is moved, the ratio changes. The next block to be moved, is chosen to be the one that results in the lowest ratio after being moved[2]. The algorithm then performs a number of "sweeps" through the circuit and the cut which results in the lowest ratio value forms the final choice for partitioning (figure 3).

As all partitioning methodologies, 'ratio cut' uses the cut capacity as a measure for the quality of the cut. The reason for this obviously is that one wants as few connections between different partitioned modules as possible. In fact, the real problem is to minimize the information flow through each of the module boundaries, since a low flow implies a good clustering. We now recall that partitioning leads to new pins on every net that has been cut (figure 2). One can easily see that the minimal flow condition is synonymous to minimizing the number of pins. Since the number of pins that is already present before the partitioning begins, is fixed, it is also synonymous to minimizing the cut, if every net that is cut leads to exactly two pins, one for each module. This is true for point-to-point interconnections but not for multi-terminal external nets [9]. The different behaviour between external and internal nets can be observed from figure 4. For the external multi-terminal net, the already existing pin can be reused which saves us one new pin. Although the cut capacity still equals 2 in this example, the total number of new pins generated by the cut is 3 instead of 4. Therefore, in [2], we suggested to use the number of new pins generated by the cut (call it the *pincut capacity*) as the cost function instead of the cut capacity, with noticeably better results.

4. PIN COUNT PREDICTION

Up to now, the ratio is based on the prediction of the cut capacity for a circuit with uniform interconnection structure [1]. However, we know that the hierarchical clustering, intrinsicly present in any real circuit design, shows a very different interconnection structure. In this section, we want to capture the notion of the interconnection complexity.

[2] Note that this new ratio can be higher than the one before the moving of the logic block. This helps the algorithm to move away from local minima.

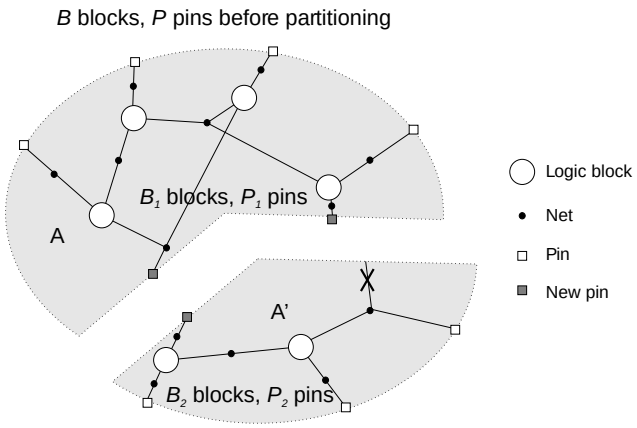

Figure 4: The cutting of external multi-terminal nets reduces the pin count (partitioning of figure 3 according to cut D).

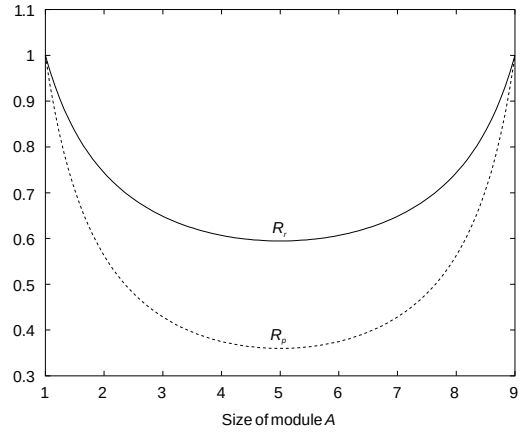

Figure 5: Comparison between different ratio values for circuits of 10 logic blocks.

4.1. Modelling the connection hierarchy

Circuits can be classified on the basis of their interconnection complexity. This *interconnection complexity* of a circuit is based on the notion that some circuits have a totally different structure of interconnections than others. These differences in interconnection complexity over all circuits have been experimentally observed by Rent and his observations led to the well-known Rent's rule [8]. It is a relationship between the number of logic blocks B in a module of a partitioned circuit, and the number of the module's external connections (pins) P:

$$P = T_b\, B^r, \tag{1}$$

where T_b is the average number of terminals per logic block, and r is called the *Rent exponent*. This exponent is a measure of the interconnection complexity of the circuit, with increasing values for increasing interconnection complexity. Generally, r ranges from 0.47 for regular circuits (such as Random Access Memories), up to 0.75 for complex circuits (such as fast full custom VLSI circuits) [10]. The validity of Rent's rule is a result of the fact that designers tend to build their circuit designs hierarchically, imposing the same complexity at each level of hierarchy. This leads to the observed "self-similarity" of circuits. Rent's rule can be observed by partitioning a circuit and plotting the number of pins per module versus the number of blocks per module (as in figure 6).

4.2. Using Rent's rule for pin count prediction

Rent's rule thus predicts the number of pins P for a module of certain size B. We can use this prediction in the ratio cut cost function. Therefore, consider figure 4 where a module containing B blocks is partitioned into two submodules of B_1 and B_2 blocks, respectively. According to Rent's rule, the whole module has $E(P) = T_b\, B^r$ pins and both submodules have $E(P_1) = T_b\, B_1^r$ and $E(P_2) = T_b\, B_2^r$ pins, respectively. The total number of new pins P_n, generated by the cut (in the partitioning process), therefore can be predicted to be

$$\begin{aligned} E(P_n) &= E(P_1) + E(P_2) - E(P) \\ &= T_b\, \left(B_1^r + B_2^r - (B_1 + B_2)^r\right). \end{aligned} \tag{2}$$

So, in order to alleviate the hidden size-effect in our partitioning algorithm, we should use the following ratio

$$R_r = \frac{P_n}{B_1^r + B_2^r - (B_1 + B_2)^r}. \tag{3}$$

Let us compare the new ratio R_r (based on Rent's rule) to its 'ratio cut' counterpart R_p (based on the uniform graph model but with the pincut capacity),

$$R_p = \frac{P_n}{B_1 B_2},$$

in figure 5, for a circuit of 10 gates (the figures for larger circuits are comparable except for the fact that the edges are a lot steeper). For the ease of comparison, we multiplied both ratios by an appropriate constant such that their value is 1 if only one gate is separated from the rest by the cut. One can see that the ratio R_r is always larger than R_p in the centre (balanced cut), relative to the value at the edges. This means that the ratio value R_p underestimates the cost for a balanced cut (and alleviates the hidden size effect too much). This forces the partitioning program to find more balanced cuts at the cost of not finding the best clusters. We therefore suggest to use R_r instead of R_p in order to recover the real hierarchical clustering structure of the circuit. The Rent exponent r can be estimated from a first (rough) partitioning or can be set to a fixed value (0.6 generally is a good first choice).

5. RESULTS

Since the quality of a partitioning tree has a direct bearing on the quality of the resulting layout, we would like to find the partitioning algorithm that generates the best tree of subcircuits. Therefore, we support the proposal of Hagen et al to use the Rent exponent as a quality measure for the underlying partitioning algorithm itself [6].

For extensive comparison between the 'ratio cut' method of Wei and Cheng and our improvement, we used 100 benchmark circuits: 20 circuits from the ISCAS85 benchmark set, 40 from the ISCAS89 benchmark set, 19 benchmarks made public by Alpert [11], 19 synthetic benchmark circuits generated by our own benchmark generation program [12], and 2 synthetic benchmarks composed by Darnauer's benchmark generation program [13].

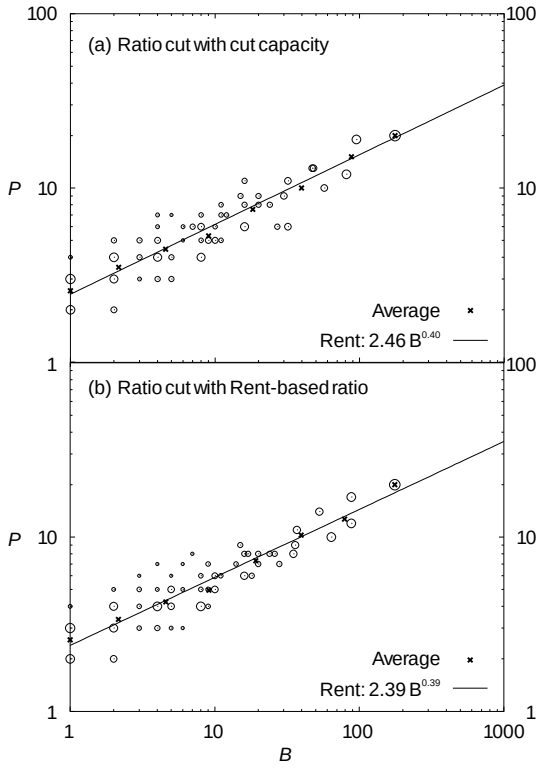

Figure 6: Comparison of the Rent behaviour of the ISCAS89 benchmark `s349` for both 'ratio cut' as implemented by Wei and Cheng (a) and our algorithm with the ratio value predicted by Rent's rule (b).

In figure 6, the Rent behaviour is shown for an example circuit. The two different ratio values are compared: the ratio as implemented by Wei and Cheng (figure 6(a)) versus our new ratio value predicted by Rent's rule (figure 6(b)). The circle sizes scale with the number of modules with the corresponding P and B that was found, and is relative to the total number of modules around an average number of blocks per module. The figures look alike but a fit of Rent's rule to the data shows that the data points generally are situated somewhat lower in figure 6(b) then in figure 6(a).

For all 100 benchmark circuits, we compared Rent's curve for both ratio values. For 37 circuits, our Rent's curve was situated under the 'ratio cut' one for all modules; for only 1 circuit, it was the other way around; for 20 circuits, our curve was situated under the other for more than half the (logarithmic) x-axis; and it was the other way around for 42 circuits. In conclusion, our results were better than the classical 'ratio cut' algorithm presented by Wei and Cheng [1], for 57 circuits out of 100.

The fact that the prediction seems to have a rather negative effect for some circuits can be explained because all algorithms described in this paper still have a problem with pin balancing (this is out of the scope of this paper but is the topic of [14]).

6. CONCLUSION

In this paper, we suggested to change the ratio value in 'ratio cut' partitioning in order to follow the intrinsic interconnection complexity of the circuit more closely. Rent's rule is the obvious choice for estimating the number of pins per module. The results show that this prediction has a positive result on the overall partitioning tree and hence is assumed to follow the intrinsic clustering structure more closely.

7. REFERENCES

[1] Y.-C. Wei and C.-K. Cheng. "Ratio cut partitioning for hierarchical designs". *IEEE Trans. Comput.-Aided Des., Integrated Circuits & Syst.*, vol. 10 (no. 7): pp. 911–921, Jul. 1991.

[2] D. Stroobandt. "Improvements to ratio cut partitioning for VLSI and ULSI". *Technical Report DG 98-11*, University of Ghent, Belgium, ELIS Department, Dec. 1998. Available at http://www.elis.rug.ac.be/~dstr/dstr.html.

[3] F. J. Kurdahi and A. C. Parker. "Techniques for area estimation of VLSI layouts". *IEEE Trans. Comput.-Aided Des., Integrated Circuits & Syst.*, vol. 8: pp. 81–92, 1989.

[4] D. Stroobandt and J. Van Campenhout. "Estimating interconnection lengths in three-dimensional computer systems". *IEICE Trans. on Inf. & Syst., Special Issue on Synthesis and Verification of Hardware Design*, vol. E80–D (no. 10): pp. 1024–1031, Oct. 1997.

[5] D. Stroobandt. "Analytical methods for a priori wire length estimates in computer systems". *Ph.D. thesis*, University of Ghent, Faculty of Applied Sciences, Nov. 1998. Available at http://www.elis.rug.ac.be/~dstr/dstr.html.

[6] L. Hagen, A. B. Kahng, F. J. Kurdahi, and C. Ramachandran. "On the intrinsic Rent parameter and spectra-based partitioning methodologies". *IEEE Trans. on Comput.-Aided Des., Integrated Circuits & Syst.*, vol. 13 (no. 1): pp. 27–37, Jan. 1994.

[7] C. J. Alpert and A. B. Kahng. "Recent directions in netlist partitioning: A survey". *Integration: the VLSI Journal.*, vol. 19 (no. 1–2): pp. 1–81, 1995.

[8] B. S. Landman and R. L. Russo. "On a pin versus block relationship for partitions of logic graphs". *IEEE Trans. on Comput.*, vol. C–20: pp. 1469–1479, 1971.

[9] D. Stroobandt and F. J. Kurdahi. "On the characterization of multi-point nets in electronic designs". In M. A. Bayoumi and G. Jullien, editors, *Proc. of the 8th Great Lakes Symposium on VLSI*, pp. 344–350. IEEE Computer Society Press, Feb. 1998.

[10] R. L. Russo. "On the tradeoff between logic performance and circuit-to-pin ratio for LSI". *IEEE Trans. Comput.*, vol. C–21: pp. 147–153, 1972.

[11] C. Alpert. http://vlsicad.cs.ucla.edu/~cheese/.

[12] D. Stroobandt. "Generating new benchmark designs for evaluation of CAD tools and new computer architectures". *Technical Report DG 98-05*, University of Ghent, Belgium, Apr. 1998. Available at http://www.elis.rug.ac.be/~dstr/dstr.html.

[13] J. Darnauer and W.W. Dai. "A method for generating random circuits and its application to routability measurement". *Proc. 1996 ACM/SIGDA Intl. Symp. Field Programmable Gate Arrays*, pp. 66–72, Feb. 1996.

[14] D. Stroobandt. "Pin balancing in ratio cut partitioning". *Proc. Swiss Conf. on CAD/CAM*, Feb. 1999.