Pin Balancing in Ratio Cut Partitioning

Dirk Stroobandt*

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

Abstract

Partitioning is a fundamental step in the computer-aided design process. One of the best algorithms for partitioning is `ratio cut` *[10] but, as many others, it does not take into account specific properties of multi-terminal nets, especially with regard to module pin count. In this paper, we show that the number of pins per module can be unbalanced if not taken care of properly. Since this unbalance can have important implications, we present a way to improve the pin balance without losing the best partitioning qualities.*

Keywords: Partitioning, Ratio cut, Balanced pin count, Multi-terminal nets, Rent's rule.

1 Introduction

The production of VLSI and ULSI[1] computer chips requires the layout (partitioning, 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. This partitioning step is necessary for being able to cope with the "size" of the circuit in the next layout steps or to implement circuits in multiple chips (MCM's)[2] or multiple FPGA[3] 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, computer-aided design tools (CAD tools) for placement and routing need accurate predictions of system performances to limit the search in the vast solution space [9]. CAD tools therefore use estimator tools [5], usually based on partitioning methodologies [2]. The better the partitioning tool, the better the estimates.

Literature on the partitioning problem is abundant. A very good general overview of partitioning methodologies is presented by Alpert and Kahng [1]. The authors divide the partitioning methods into four categories: move-based approaches, methods that construct a geometric representation of the partitioning problem, combinatorial approaches, and clustering-based methods. Most partitioning algorithms are move-based because these are the easiest to implement. A comparison between some of the partitioning methods has been presented by Hagen et al [2] and 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 [10].

All existing partitioning methodologies minimize the number of nets cut during the partitioning process. In [6], Stroobandt showed that the number of new pins, generated by the cut, is a better criterion since some nets generate less new pins than others. In this paper, we show that this also has a profound impact on the pin balance over the partitioned modules. We present an additional cost function that tries to keep both modules balanced. The results show that pin balance can benefit from the introduction of this additional cost function without losing the overall quality of the cut. This result can have a large impact in situations where pin capacity is limited or where a high pin balance is required. In any case, it improves the overall partitioning results.

The remainder of this paper is organized as follows. The partitioning model is explained in the next section. Section 3 briefly describes the `ratio cut` algorithm with pin count cost function. The reason for using an additional cost function is explained in section 4 and section 5 shows some results.

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

[1] Very Large-Scale and Ultra Large-Scale Integration.

[2] Multi-chip modules.

[3] Field Programmable Gate Array.

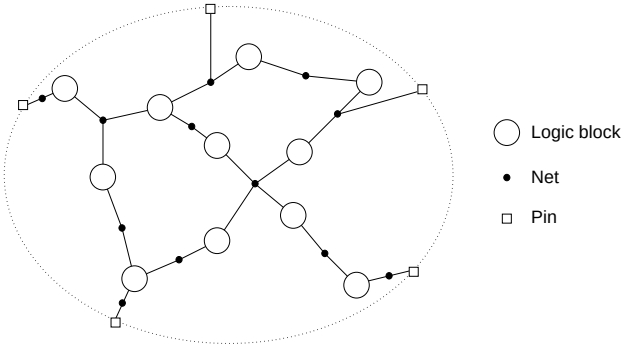

Figure 1: Model of a design.

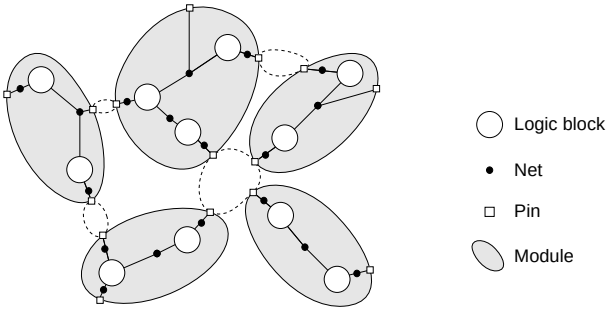

Figure 2: Partitioning the design of figure 1 into modules.

2 Model for the design and for the partitioning process

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

Partitioning a design means dividing this design into disjoint subdesigns (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 module 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 design 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 pin count

The `ratio cut` algorithm, presented by Wei and Cheng [10], is based on the notion that circuits are generally designed in a hierarchical way and thus hold an intrinsic partitioning structure. Finding this intrinsic hierarchical partitioning information is important since the quality of the partitioning result depends on the final integration of all partition levels, from the basic subcircuits to the whole 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 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.[4] The *ratio cut* is the cut that generates the minimum ratio among all cuts in the network.

3.1 The clustering property of the ratio cut

The clustering property of the ratio cut can be interpreted by a random graph model [10]. The probability of having an edge connecting each pair of nodes in this graph is equal to an identical value f. Consider a cut which partitions the circuit into two subsets A and A'. The expected capacity $C_{A,A'}$ of this cut equals the probability f multiplied by the number of possible edges between A and A':

$$E(C_{A,A'}) = f|A||A'|. \qquad (1)$$

Since the product of module sizes grows when the cut is more balanced, i.e. the size of the modules differs less, the expected capacity is lowest at the edges, i.e. when only a few nodes are separated from the rest. This effect, called *the size effect*, is unwanted since we generally like to obtain balanced partitions. To alleviate the hidden size effect, the ratio value is used.

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

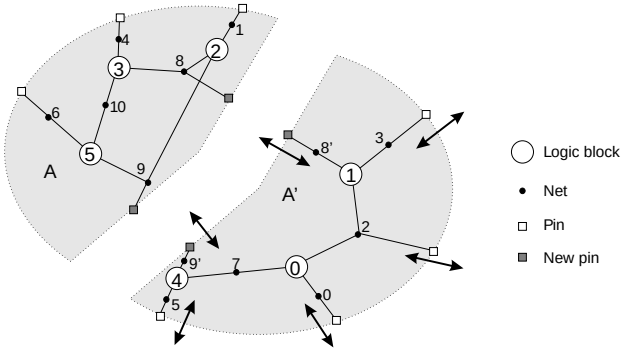

Figure 3: Minimizing the information flow or the number of pins results in good clustering

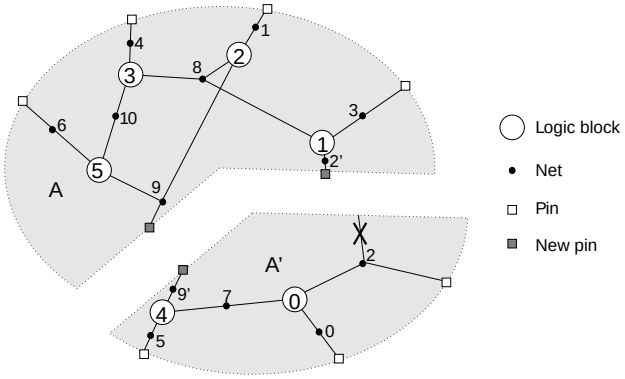

Figure 4: The cutting of external nets reduces the pin count.

As a consequence, the expected value of this ratio is a constant with respect to different cuts:

$$E(R_{A,A'}) = E\left(\frac{C_{A,A'}}{|A||A'|}\right) = f. \qquad (2)$$

Thus, if the edges of the graph are uniformly distributed, all cuts have the same ratio value. In other words, the choice of the cuts does not make a difference in such a uniformly distributed random graph. However, in a general circuit, different cuts generate different ratios. Cuts that go through weakly connected groups correspond to smaller ratios. The minimum of all cuts according to their corresponding ratios defines the sparsest cut since this cut deviates the most from the cuts of a uniformly distributed graph. Using the ratio value therefore results in finding the intrinsic hierarchical partitioning structure of the circuit.

3.2 Using a pin count criterion

As all partitioning methodologies, `ratio cut` uses the cut capacity as a measure for the quality of the cut. The reason for this obviously lies in the fact 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 (figure 3). We now recall that partitioning leads to new pins on every net that has been cut. 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), 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 [8]. The different behaviour between external and internal nets can be observed from figure 4. For the external multi-terminal net 2, 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 [6], we suggested to use the number of new pins P_n, generated by the cut (call it the *pincut capacity*) instead of the cut capacity. The ratio used in the cost function for the partitioning algorithm then becomes

$$R_p = \frac{P_n}{B_A B_{A'}}. \qquad (3)$$

Wei and Cheng described a heuristic algorithm for finding the ratio cut in a netlist of a circuit [10]. The algorithm is move-based, i.e. it iteratively moves nodes from one module to the other. Each time a node is moved, the ratio changes. The next node to be moved, is chosen to be the node that results in the lowest ratio after moving the node[5]. 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. In [6], we suggested to use the ratio R_p (based on the pincut capacity) as a cost function with noticeably better results.

4 Finding a more balanced cut

Since external multi-terminal nets create only one new pin instead of two, we should also check to which module the new pin is assigned. Up to now, we only minimized the total number of new pins. We did not worry about a good balance in pin count for the two new

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

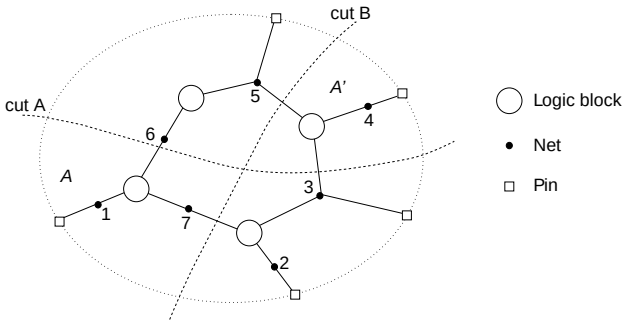

Figure 5: The difference between a balanced cut (cut A) and an unbalanced cut (cut B).

modules. In this section, we will show that this balance can be heavily effected by the choices made and we will present a way to obtain a more balanced cut in terms of pin count.

4.1 Example

Consider the very simple example of figure 5. Two possible cuts are shown on the figure (cut A and cut B). They both split the circuit in two equally sized modules by cutting two nets. Both cuts generate 3 new pins since they both cut one internal and one external net. The ratio cut cost function (whether it is based on cut capacity or pincut capacity) cannot discriminate between the two cuts. However, if we look at the number of pins per module, we observe a difference. For cut A, module A and module A' both have four pins after cutting. For cut B, module A only has three pins (on nets 1, 5, and 7) whereas module A' has five (on nets 2, 3, 4, 5, and 7). It is clear that we would choose cut A as the best cut since it results in a more balanced partitioning, but the `ratio cut` algorithm could as well have chosen cut B.

The previous example shows that the choice of a new cut can have an influence on the pin balance between the two modules, even if the number of new pins is the same for two competing cuts. Instead of only taking into account the number of new pins, we should also consider the pin balance. This can be done by bearing Rent's rule in mind, one of the most simple but, at the same time, most important rules in design characterization.

4.2 Rent's rule

Designs can be classified on the basis of their *interconnection complexity*. Differences in interconnection complexity have been experimentally observed by Rent and led to the well-known *Rent's rule* [3]. It is a relationship between the average number of elementary blocks B in a module of a partitioned design, and the average number of the module's external connections (pins) P:

$$P = T_b\, B^r, \tag{4}$$

where T_b is the average number of terminals per logic block, and r is called the *Rent exponent*. This exponent is bounded by 0 and 1, with increasing values for increasing interconnection complexity. Generally, r ranges from 0.47 for regular designs (such as Random Access Memories), up to 0.75 for complex designs (such as fast full custom VLSI designs) [4]. 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).

Rent's rule thus predicts the number of pins [7] for each module separately as

$$E(P_A) = T_b\, B_A^r \tag{5}$$

$$E(P_{A'}) = T_b\, B_{A'}^r. \tag{6}$$

In a balanced cut, the ratios P_A/B_A^r and $P_{A'}/B_{A'}^r$ should both equal T_b. An unbalanced cut with the same total number of pins ($P_A + P_{A'}$ constant) will have a higher ratio in one module and a lower one in the other. One can easily see that the degree of unbalance is reflected by the function

$$R_b = \left(\frac{P_A}{B_A^r} - \frac{P_{A'}}{B_{A'}^r} \right)^2,$$

which we call the *balance ratio*. This balance ratio can now be used as an additional cost function to discriminate between different cuts with the same pincut ratio value R_p, in order to find the most balanced one. The Rent exponent 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

Figures 6 and 7 show the P versus B plots for all modules during the top-down hierarchical partitioning of the ISCAS85 benchmark `c1355`. In figure 6, the `ratio cut` algorithm of Wei and Cheng [10] was used. Figure 7 shows the result for our new algorithm, based on pincut capacity and with pin balancing. 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 show that the pin balancing generally results in less variation in number of

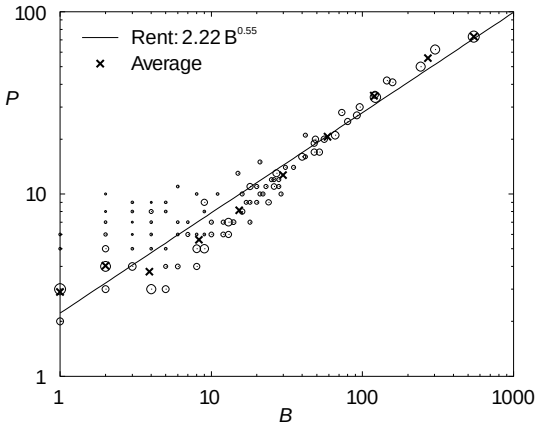

Figure 6: Partitioning results for the `ratio cut` algorithm.

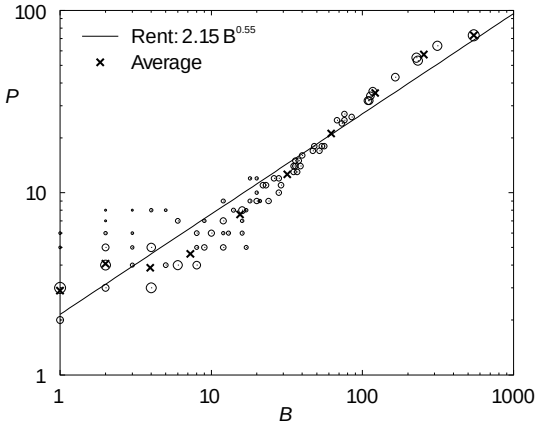

Figure 7: Partitioning results for the `ratio cut` algorithm with balanced number of pins.

pins for the same module size, as could be expected. Also, the minimization of pincut capacity instead of cut capacity results in slightly lower pin count values. This is reflected in the fact that the best fit for Rent's rule lies a bit lower with our algorithm. Similar results are obtained for the other benchmark circuits.

6 Conclusion

In this paper, we showed that the `ratio cut` partitioning algorithm can lead to an unbalanced pin count for the partitioned modules. We presented an extension to this algorithm that balances pin count and minimizes the total number of pins instead of the number of nets cut. The results, obtained with the new algorithm, show less variation in pins per module and generally a lower pin count value. Therefore, the new algorithm outperforms `ratio cut`, which is known as one of the best partitioning algorithms.

Apart from the fact that the general partitioning results improve with our method, a balanced pin count while partitioning can be very important in practical situations since pin limitation (the problem of not having enough pin capacity) remains a major issue in electronic design. While partitioning a design into parts that fit into one chip or one FPGA component, it is important to keep the number of pins as low as possible to prevent pin limitation problems.

References

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

[2] 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): pages 27–37, January 1994.

[3] 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: pages 1469–1479, 1971.

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

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

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

[7] D. Stroobandt. Pin count prediction in ratio cut partitioning for VLSI and ULSI. Accepted for the IEEE Intl. Symp. on Circuits and Systems, May 1999.

[8] 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*, pages 344–350. IEEE Computer Society Press, February 1998.

[9] 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): pages 1024–1031, October 1997.

[10] 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): pages 911–921, July 1991.