

3-2005

# A Minimum Cut Based Re-synthesis Approach

M. Welling

*Southern Illinois University Carbondale*

Spyros Tragoudas

*Southern Illinois University Carbondale*

Haibo Wang

*Southern Illinois University Carbondale*, [haibo@engr.siu.edu](mailto:haibo@engr.siu.edu)

Follow this and additional works at: [http://opensiuc.lib.siu.edu/ece\\_confs](http://opensiuc.lib.siu.edu/ece_confs)

Published in Welling, M., Tragoudas, S., & Wang, H. (2005). A minimum cut based re-synthesis approach. Proceedings of the Sixth International Symposium on Quality Electronic Design (ISQED'05), 202-207. doi: 10.1109/ISQED.2005.9 ©2005 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE. This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reposted without the explicit permission of the copyright holder.

---

## Recommended Citation

Welling, M.; Tragoudas, Spyros; and Wang, Haibo, "A Minimum Cut Based Re-synthesis Approach" (2005). *Conference Proceedings*. Paper 41.

[http://opensiuc.lib.siu.edu/ece\\_confs/41](http://opensiuc.lib.siu.edu/ece_confs/41)

This Article is brought to you for free and open access by the Department of Electrical and Computer Engineering at OpenSIUC. It has been accepted for inclusion in Conference Proceedings by an authorized administrator of OpenSIUC. For more information, please contact [opensiuc@lib.siu.edu](mailto:opensiuc@lib.siu.edu).

# A Minimum Cut Based Re-synthesis Approach

M. Welling, S. Tragoudas, H. Wang

Electrical & Computer Engineering Dept., Southern Illinois University at Carbondale

*m\_welling@hotmail.com/spyros@engr.siu.edu/haibo@engr.siu.edu*

## Abstract

A new re-synthesis approach that benefits from min-cut based partitioning is proposed. This divide and conquer approach is shown to improve the performance of existing synthesis tools on a variety of benchmarks.

## 1. Introduction

Multiple-level networks are best for modeling large circuit designs and they allow for degrees of freedom in the optimization process. Exact optimization methods [1] have been found for multi-level networks but are considered impractical by current standards due to high computational complexity. Most current packages, in turn, use more efficient heuristic-based optimizers which benefit from logical transformations. Basic transformations including elimination, decomposition, extraction, simplification and substitution are used in combination to find more optimal solutions [2].

These transformation operators can be performed in an algorithmic approach where a script is written with specific transformations which have been found to optimize. This approach is used in the well-known MIS/SIS packages [2]. Rule based systems such as the IBM's LSS package use data-bases to store many different transformations [3]. Pairs of implementations are stored within the entries of the data-base. Each entry has first a circuit which is to be detected within a combinational circuit and a second more efficient implementation for replacement. One of the major advantages of such a system is that the data base can be modified to allow more transformations. The design techniques and optimization tricks of a particular designer can be saved in the data-base for further use. This leads to genuine designs based on the rules which are stored within a data-base.

Different models have been used in the implementation of heuristic multi-level circuit

optimizers. The major models presently being used are the algebraic and Boolean models.

The algebraic model represents the functionality of network nodes as algebraic expressions. Algebraic expressions are polynomials over the set of network variables with unit coefficients and the rules for polynomial algebra are applied. Kernels are cube-free quotients of an expression. Extraction is performed after computing the set of kernels for each expression in a network and finding common kernels. Substitution and decomposition can also be performed by using algebraic division. The MIS package takes advantage of the algebraic model [2].

The Boolean model uses Boolean functions and don't care sets (external or satisfiability) are computed to represent the functionality of logic network nodes. The don't care sets allow for degrees of freedom with respect to the optimization process [4, 5].

The heuristic nature of the above synthesis techniques typically returns sub-optimal designs. Re-synthesis methods consider a synthesized netlist and redesign it for further optimization. Many well-known methods for re-synthesis are based on automatic test pattern generation (ATPG). In particular, redundancy detection and removal is based on the identification of undetectable or un-testable single stuck-at faults within the circuit. An untestable fault indicates that a logical redundancy exists and simplification takes place. Circuits may contain multiple undetectable stuck-at faults but once a circuit is simplified according to a redundancy, the faults must be recomputed for the new circuit. The process is repeated until no further redundancies are found [6]. More elaborate methods based on the same principle have been proposed which insert redundancy in the circuit in order to cause one or more irredundant lines to become redundant [7,8]. The method in [9] benefits from the detection and correction of design errors using ATPG.

Simultaneous addition and removal of many redundant wires can be obtained with the global flow

method [10]. This method uses the concept of mandatory assignments and a graph-theoretic problem formulation to redesign the fan-out branches of a single stem. Although technically it does not rely on ATPG, it is related to the above methods since the redesign is based on redundancy observations.

This paper introduces a non-ATPG re-synthesis method for reducing the transistor overhead of a synthesized circuit. The method recursively partitions the circuit into sub-circuits which are synthesized independently. It is described in Section 2 and its experimental evaluation is given in Section 3. It is experimentally observed that it benefits when combined with some of the previously mentioned methods.

Another non-ATPG re-synthesis method known as retiming repositions the flip-flops in the circuit in order to minimize the delay along the longest combinational path or the number of flip-flops [11]. This method does not re-design the combinational logic and is orthogonal to the previous approaches as well as the approach presented in this paper.

## 2. The proposed re-synthesis method

The input is a synthesized net-list that represents the combinational core of a synchronous sequential circuit. Typical objectives in re-synthesis include minimization of the total number of transistors and the maximum number of gates along any sensitized path. The circuit is abstracted as a directed acyclic graph.

The proposed method proposes that the net-list is partitioned into two sub-circuits. The output functions of each sub-circuit are co-synthesized independently. The re-synthesized sub-circuits are then merged to provide with the re-synthesized circuit.

The output functions of each sub-circuit are co-synthesized using any state-of-the-art synthesis tools that combine existing synthesis and re-synthesis methods from Section 1. The described process is in fact implemented recursively. In particular, each sub-circuit may be recursively partitioned into two parts whose functions are co-synthesized independently and are then merged to form the re-synthesized sub-circuit. Each such level of recursion is called a *pass* of the proposed method.

The rationale for this divide and conquer method is that the output functions of each sub-circuit may be co-synthesized with a different sequence of operations by any of the existing heuristic methods mentioned in Section 1. This may occur since the original circuit is partitioned into two sub-circuits so that all inputs of the first sub-circuit are inputs of the original circuit and

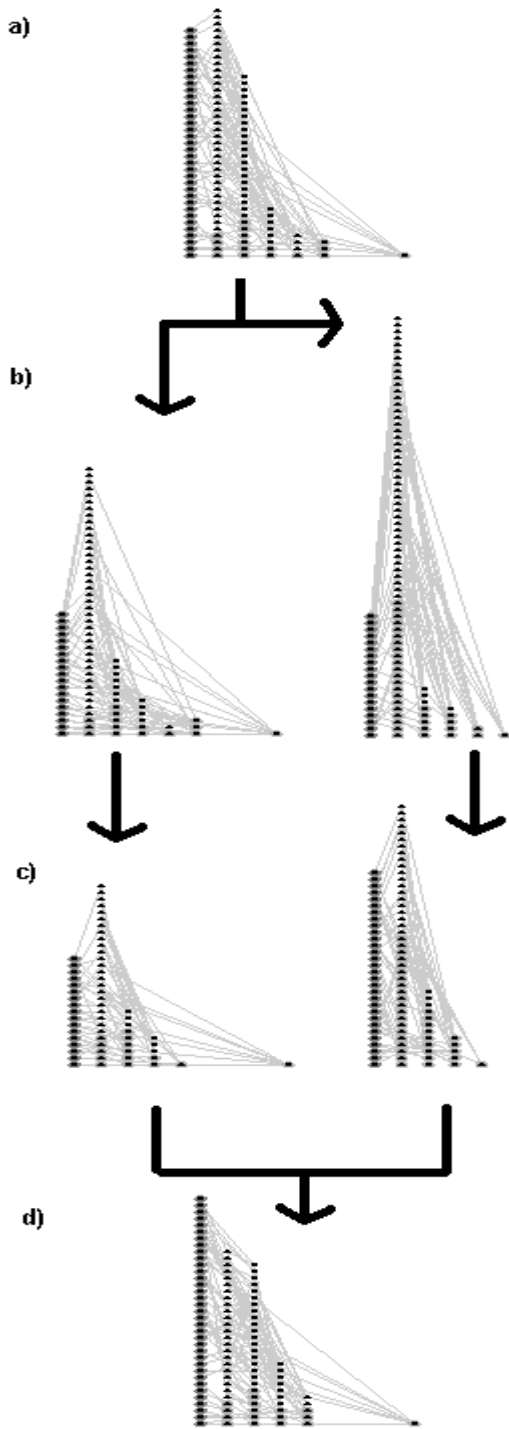
some inputs of the second sub-circuit are outputs of the first sub-circuit.

There are many ways to partition a circuit in two sub-circuits, and different partitions may result into different outcomes. The proposed method insists that the number of outputs from the first sub-circuit that are inputs to the second sub-circuit be kept minimum. This is obtained using the maximum-flow minimum cut algorithm in [12]. That way, the number of functions that need to be co-synthesized in the first sub-circuit is as small as possible, a desirable feature in synthesis, and the number of input variables in the supporting set of each output function in the second sub-circuit tends to be low, another desirable feature in synthesis.

This partitioning methodology was evaluated against several alternatives and was found to outperform them. For example, another partitioning methodology that was examined insisted that the two sub-circuits have equal number of gates prior to their re-synthesis. This partitioning scenario prevents that one of the two sub-circuits resembles the original circuit but does not always improve the design. Inferior results were also obtained when we did not insist that the number of interconnects between the two partitions be kept minimum using max-flow minimum-cut algorithms.

The following example demonstrates how the proposed bi-partitioning method works. A single pass of this algorithm is graphically depicted in Figure 1. This example shows a smaller design, the combinational core of s298, from the ISCAS'89 collection, which benefits from this re-synthesis technique. All figures assume a direction from left to right. Each column of nodes corresponds to a topological level in the acyclic graph that abstracts the net-list. The larger nodes, the nodes in the first topological level, correspond to the inputs. The nodes in the remaining columns are either gates within the design or outputs. The outputs must be specified or they could possibly be reduced from the design. In this example, each gate is a two-input NAND, and the co-synthesis of any set of functions has been obtained using the Buildgates synthesis tool of Cadence. The interconnects are drawn with gray lines.

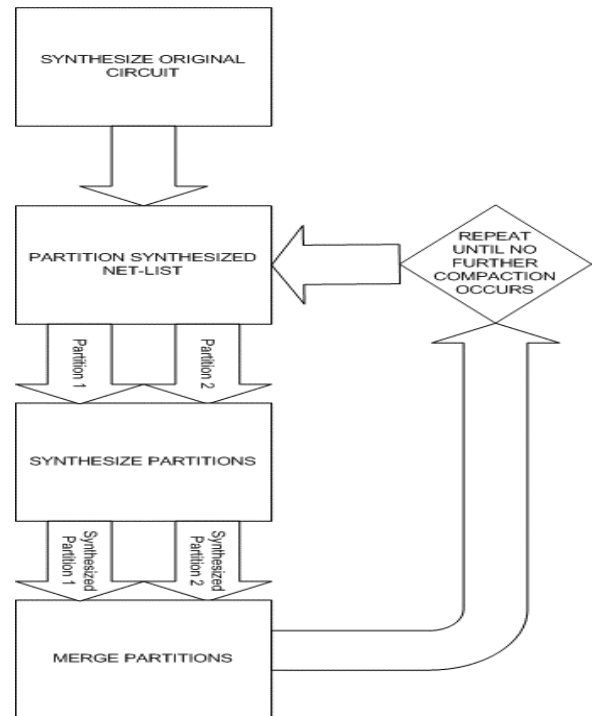
The original synthesized circuit is shown in Figure 1a. This circuit is directly partitioned resulting in the circuits in Figure 1b. Each of the two partitions is then passed through the synthesis tools separately. The separately synthesized partitions are depicted in Figure 1c. Once the partitions are optimized they are merged back into a single circuit (Figure 1d).



**Figure 1. Single pass of algorithm on s298.**  
 a) original circuit, b) partitioned circuit, c) synthesized partitions, d) merged circuit

The number of the two-input NAND gates in the original net-list is more than the number of the two input NAND gates in the final net-list. It is also clear that the interconnect structure is significantly reduced. Finally, the topological depth of the circuit (maximum number of gates along any path) is reduced by one. Therefore the final design requires less area, and has reduced propagation delay. It is also expected to consume less power since the number of transistors is reduced and the interconnect structure is simplified.

The outline of the proposed algorithm is given in Figure 2.



**Figure 2. Flowchart of proposed method**

### 3. Experimental Results

The proposed re-synthesis method has been implemented on top of the Buildgates synthesis tools in Cadence. We consider only two-input NAND gates in the library. We use simple gates so that the transistor count is a more realistic representation of the circuit area. Table 1 gives the transistor count for two passes of the proposed algorithm. Part (a) gives results for the combinational ISCAS'85 benchmarks, part (b) for the combinational core of the ISCAS'89 benchmarks and part (c) for the combinational core of the ITC'99 benchmarks.

**Table 1. The transistor counts for 2 passes of the proposed method using Buildgates of Cadence**

**(a) ISCAS'85 benchmarks**

Benchmark	Original	Pass 1	Pass 2
C1355	1892	-	-
C1908	1918	1890	-
C2670	3450	3306	-
C3540	4426	4370	4298
C432	1146	1128	-
C499	1902	-	-
C5315	6378	6282	6270
C6288	9458	-	-
C7552	9224	8926	8854
C880	1650	-	-

**(b) ISCAS89 benchmarks**

Benchmark	Original	Pass 1	Pass 2
S1196	1982	1976	1970
S1269	1966	-	-
S1423	2292	-	-
S1488	1620	1612	-
S1494	1556	1528	1512
S1512	2150	1680	1614
S208.1	326	-	-
S298	442	314	274
S3384	4990	4122	4008
S344	524	500	464
S6669	10112	10016	9730
S349	536	500	-
S386	662	-	-
S420.1	630	600	594
S444	618	-	-
S499	890	-	-
S510	860	-	-
S526	848	-	-
S820	1276	1244	-
S832	1154	-	1222
S967	1730	1722	-
S991	1262	-	-

**(c) ITC99 benchmarks**

Benchmark	Original	Pass 1	Pass 2
B01	154	136	130
B03	562	540	456
B04	2250	2242	2260
B06	210	174	170
B07	1646	1592	1292
B08	524	500	482
B09	632	-	578
B11	1940	-	-
B12	4280	4172	4166
B13	1126	1104	1098
B14	27424	26828	25892

It is shown that the transistor count reduces in many circuits. An “-“ indicates no improvement over the previous pass or the initial circuit, which is pass 0. We observe reductions on the transistor count on the majority of the circuits we experimented with.

Reductions on the transistor count occur in almost all of the larger benchmarks we experimented with. For example, the number in the name of the ISCAS'85 benchmarks indicate the number of lines in the original net-list.

In many cases where the transistor count is reduced we also gain on the number of topological levels. For example, in c2670, pass 1 reduces the number of levels from 32 down to 31 whereas in c3540 pass 2 drops the number of levels from 57 to 54. In some case we observed that the number of topological levels drops although no transistor count reduction is observed. For example, in c880 pass 1 does not reduce the transistor count but the re-synthesized circuit has 31 levels, down from 33. We did not observe a case where the transistor count is reduced but the number of topological levels is increased.

**Table 2. A single pass of the proposed method using minimum and balanced cuts on the ISCAS'85 benchmarks**

Benchmark	Original	Single Cut	Bal. Cut
C1355	1892	1900	1934
C1908	1918	1890	1924
C2670	3450	3306	3316
C3540	4426	4370	4412
C432	1146	1128	1192
C499	1902	1918	1998
C5315	6378	6282	6380
C6288	9458	9474	9586
C7552	9224	8926	9142
C880	1650	1690	1664

Table 2 compares the transistor count by the first pass of the implemented maximum-flow minimum-cut based partitioning method and the balanced bi-partitioning method where the partitioning of the original circuit results in two sub-circuits with almost equal transistor count. Results are listed for the ISCAS'85 benchmarks but similar results hold for the remaining circuits. The table clearly shows that the latter approach is inferior to the proposed one. In c880 the latter method has slightly less transistor count but none of the two partitioning methods reduces the original transistor count.

Table 3 gives some more detailed results on the ISCAS'85 benchmarks with the proposed method. Results from six passes of the proposed method are given. The general observation (in all benchmarks) is

that in larger circuits reductions on the transistor count occur as the number of passes increases. We note that the CPU overhead of the proposed method (when compared to the execution time of Buildgates) is insignificant when the number of passes is kept to a small constant (less than 10).

**Table 3. Results for 6 passes of the proposed method on the ISCAS'85 benchmarks**

Bench	Orig.	P1	P2	P3	P4	P5	P6
C1355	1892	-	-	-	-	-	-
C1908	1918	1890	-	-	-	-	-
C2670	3450	3306	-	-	-	-	-
C3540	4426	4370	4298	4254	4206	3912	-
C432	1146	1128	-	-	-	-	-
C499	1902	-	-	-	-	-	-
C5315	6378	6282	6270	-	-	-	-
C6288	9458	-	-	-	-	-	-
C7552	9224	8926	8854	8788	-	8758	-
C880	1650	-	-	-	-	-	-

Experiments were also taken using the X SIS package for completeness purposes. This a newer version of the Berkley MIS package [3] with a graphical user interface. The results were taken for a single pass of the proposed algorithm and only for the ISCAS'85 benchmarks. X SIS does not use implication-based or ATPG-based redundancy-related re-synthesis techniques, and in general does not produce as compact designs as Buildgates. The method did not produce as much reduction on the transistor count as with Buildgates. This may imply that the proposed method benefits when combined with the latter re-synthesis approaches. Nevertheless, reductions were observed in some circuits. For example, we synthesized c432 using up to four-input NAND and NOR gates, and we were able to see a reduction from 1034 transistors down to 968.

## 4. Conclusions

Through experimentation with partitioning and synthesis tools, it has been found that partitioning may be useful as a re-synthesis technique. The experiments found that partitioning the netlist using a min-cut algorithm and synthesizing independently the sub-circuits with a commercial synthesis package (Cadence Buildgates) the transistor count may be reduced significantly.

The minimum cut partitioning methodology seemed to produce better results than balanced cuts and other alternatives. This can be justified when considering that the first partition of a minimum cut has the least number of lines that need to be co-synthesized

and the second partition has the least number of lines to incorporate into its functionality. It is experimentally shown that this methodology generally works best with larger designs and is not guaranteed to work in every situation.

## 5. References

- [1] Brayton, R.K. Somenzi, F., *An exact minimizer for Boolean relations*, IEEE International Conference on Computer-Aided Design (ICCAD), 5-9 Nov. 1989, pp. 316 – 319
- [2] Brayton, R.K., Rudell, R., Sangiovanni-Vincentelli, A., Wang, A.R , *MIS: A Multiple-Level Logic Optimization System*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 6 , Issue 6 , November 1987, pp. 1062 - 1081
- [3] J.A Darringer, W.H. Joyner, C.L. Berman, L. Trevillyan, *Logic Synthesis through Logic Transformation*, IBM Journal of Research and Development, pp. 272-280, July 1981.
- [4] R.K. Brayton, Sentovich, E.M., Somenzi, F, *Don't cares and global flow analysis of Boolean networks*, IEEE International Conference on Computer-Aided Design (ICCAD), 7-10 Nov. 1988, pp. 98 - 101
- [5] Savoj, H., Brayton, R.K, *Observability relations and observability don't cares*, IEEE International Conference on Computer-Aided Design (ICCAD), 11-14 Nov. 1991, pp. 518 - 521
- [6] S.-C. Chang, D. I. Cheng, C.-W. Yeh , *On removing multiple redundancies in combinational circuits*, Proceedings of the Conference on Design, Automation and Test in Europe (DATE), February 1998.
- [7] Entrena, L.A., Kwang-Ting Cheng, *Combinational and sequential logic optimization by redundancy addition and removal*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 14 , Issue 7 , July 1995, pp. 909 - 916
- [8] Entrena, L.A., Espejo, J.A., Olias, E., Uceda, J., *Timing optimization by an improved redundancy addition and removal technique*, Proceedings EURO-DAC '96, European Design Automation Conference, 16-20 Sept. 1996, pp. 342 – 347
- [9] Veneris, A., Abadir, M.S, *Design rewiring using ATPG*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 21 , Issue 12 , Dec. 2002, pp. 1469 - 1479
- [10] Berman, C.L., Trevillyan, L.H, *Global Flow Optimization in Automatic Logic Design*, IEEE Transactions

on Computer-Aided Design of Integrated Circuits and Systems, Vol. 10 , Issue 5 , May 1991, pp. 557 – 564

[11] C. Leiserson, J. Saxe, Retiming Synchronous Circuitry, *Algorithmica*, vol. 6, pp. 5-35, 1991.

[12] Ford, L.R., Jr. and Fulkerson, D.R., *Flows in Networks*, Princeton University Press, 1962.