Extracting of High-level Structural Representation from VLSI Circuit Description Using Tangled Logic Structures

Andrey Trukhachev [1] and Natalia Ivanova [2]

[1]National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), Moscow, Russia

[2]Bauman Moscow State University, Moscow, Russia

AATrukhachev@mephi.ru, ineldin@gmail.com

Abstract

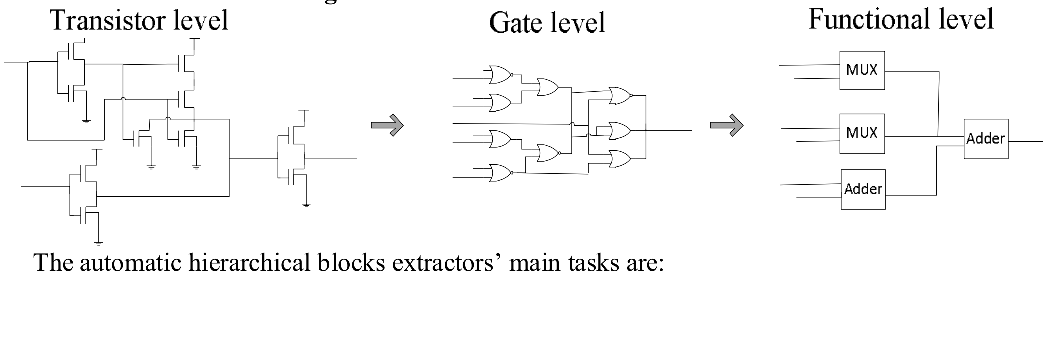
This paper proposes a method of automatic VLSI circuit analysis. On the first step transistors are grouped by their structure. Groups with irregular structure are highly interconnected to each other. Detecting Tangled Logic Structures (TLS) with a GTL-depended linear ordering and genetic algorithm divides the circuit due to its functional structure and forms the gate-level VLSI circuit. High-level functional blocks in circuit description consist of gate-level cells groups, which are also highly interconnected. After TLS-blocks extracting, it is possible to describe their function. TLS-blocks are smaller, represent a cell of high-level circuit, and are thus more suitable for further functional circuit analysis than a gate-level VLSI circuit.

Keywords: VLSI, Genetic Algorithm, Tangled Logic, Functional Circuit Analysis

1 Introduction

Due to the progress of the semiconductor industry, the scale of VLSI circuits dramatically increases, which makes hierarchical representation of circuit crucial for computer-aided design (CAD) tools and designer himself. The main levels of abstraction of VLSI circuit presented on Fig. 1. Transistor-level circuit describes VLSI as a set of transistors and their interconnections; gate-level consists of gates formed from transistors, each gate has its function; functional or behavioral level is represented by functional blocks, which could be grouped in to the blocks of the next hierarchical level.

Figure 1: Levels of VLSI circuit abstraction.



The automatic hierarchical blocks extractors' main tasks are:

- **Verification of a circuit layout** (M.S. Abadir, 1990). Due to the complexity of VLSI producing cycle there could be different fails on any step of the process. For the layout defects detection it is important to understand the functional meaning of each layout block.
- **Speeding up circuit simulation** (T.J. Thatcher, 1992), (K.-T. Huang, 1995). **Design-for-testability and built-in-self-test** (I. Parulkar, 1994), (S. Jolly, 2002), registers can be selected as test resources (test pattern generators, scan registers and test response compactors) which can reduce the complexity of achieving a testable design.
- **Floorplanning and resynthesis** (Kundu, 1998), (T. Jindal and et al, 2010).

Existing subcircuit extraction algorithms appearing in literature can be classified into three categories:

- **Structural recognition** (Boehner, 1988), (A. Lester, 1998);
- **Pattern matching** (F. Luellau, 1984), (H. Graeb, 2001), (M. Ohlrich, 1993);
- **Combination of structural recognition and pattern matching** (Lei Yang, 2006).

Structural recognition methods divide the CMOS circuits into channel connected-components (CCCs). Parallel and serial connections of the transistors inside a CCC, determine the logic function of a group. In general, structural recognition algorithms cannot handle irregular-structured blocks, such as flip-flops, latches, other high-level digital blocks, current mirrors, amplifiers, and other analog blocks with structures, it is hard to define them in terms of rules.

Pattern matching is the technology independent algorithms where circuit is represented as a graph, functional groups of elements are subgraphs, subgraph isomorphism is used to find the elements groups which are similar to previously defined functional pattern.

The combination of two previous technics allows speeding up the process by detecting regular logical structures and generating the hybrid graphs of transistors and gates netlist. Then pattern-matching algorithm is applying to extract the high-level structural representation.

Using the patterns is not possible if there is no information about structure or function of high-level hierarchy of the VLSI circuit. Another problem is pattern blocks represented by next hierarchical level from the gate-level. In this case much effort has been dedicated to matching large circuit with pattern which consists of ten or more vertices.

In this paper, we propose pattern-free, technology independent method for extracting of functional blocks with irregular structure. The first step is structural analysis of transistor-level VLSI circuit, then TLS are extracting from transistor-gate-level VLSI circuit. The third step is pattern matching with extracted (from gate-TLS-level VLSI circuit) TLS and library or manual TLS analysis. Then it is possible to obtain functional blocks from gate-level tangled logic structures by pattern matching or any other technics. Genetic algorithm for functional block initial vertices search could be used on this step.

The advantages of the method are possibility of pattern-free functional analysis of irregular structures. The first step could be performed without structural analysis which is dependent on technology of VLSI design.

The remainder of this paper is organized as follows. An overview of related work is given in Section 2. Section 3 describes method and its time complexity analysis. Section 4 presents experimental results on real sample. Concluding remarks are given in Section 5.

2 Related work

In (Lei Yang, 2006) FROSTY reads in a transistor-level CMOS netlist (object circuit) and a library file. The library file contains user-specified high-level functional blocks to be recognized from the object circuit and the netlist for each functional block (called subcircuit). The whole FROSTY flow is divided into two steps: in Step 1, the program tries to identify all CMOS gate with regular structure, converting the object circuit to mixed gate-level and transistor-level object graph; in parallel, the program transfers each library-defined subcircuit to a mixed gate-level and transistor-level subgraph too; in Step 2, pattern matching is applied to recognize all subgraphs from the object graph. After the above gate recognition and functional blocks extraction, FROSTY outputs VHDL or Verilog model with user-defined blocks in the library.

In practice, FROSTY pre-defines a library, which contains all kinds of functional blocks like flip-flops, adders, latches and so on. The disadvantages of this method are compilation of the functional block library and pattern matching process in Step 2. FROSTY tries to match all the library-defined subcircuits in the object circuit. Unfortunately, if one subcircuit does not exist in the object circuit, the program still spends CPU time in order to find it. In some cases library is unknown and this step could not be used.

In (T. Jindal and et al, 2010) T. Jindal et al. proposes a straightforward algorithm (tangled-logic finder) to identify GLTs. This method consists of three phases:

- Phase I: linear ordering generation.
- Phase II: initial candidate GTL generation.
- Phase III: GTL refinement and pruning.

The linear order generation initializes the group with a seed cell, which is randomly generated. Then, it iteratively adds one cell at a time to the group. The candidates for the cell addition are the cells outside of the group, but with direct edge connections with the group. Among these candidates, choose the one with the strongest connection with the group.

A cell group can be extracted from a linear ordering according to the metrics:

$$nGTLS(C) = \frac{T(C)}{A_G \cdot |C|^p}$$

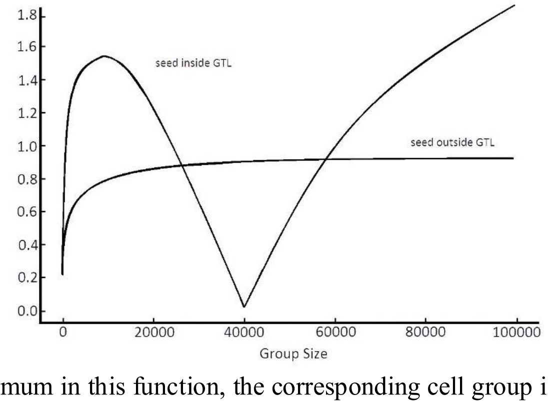
where $T(C)$ – ratio cut, A_G – is the average pin count of the cell, $|C|$ - scaled cost, p – Rent exponent.

$$GTLSD(C) = \frac{T(C)}{A_G \cdot |C|^{p \cdot A_C / A_G}}$$

where A_C is the ratio of the number of pins contained in C divided by $|C|$.

A group C of size $k = |C|$ is composed by the first k cells in the linear ordering. Then, the function $nGTLS(C)$ or $GTLSD(C)$ with respect to k is obtained like in Figure 2.

Figure 2: nGTLS score.



If there is a clear minimum in this function, the corresponding cell group is selected as a candidate GTL “B”. When computing the nGTLScore, the value of Rent exponent p needs to be chosen. This is obtained by averaging the Rent exponents for all groups obtained in the linear ordering. The Rent exponent of a group C can be estimated by $\frac{\ln T(C) - \ln A_C}{\ln|C|}$ where A_C is the average number of pins per cell in C. For each candidate Bi obtained in Phase II, another set of candidates $Bi_{,1}$, $Bi_{,2}$, ... , $Bi_{,l}$ using seeds inside Bi and the same procedure as Phase I and II is generated.

Tangled-logic finder does not need any library to find functional groups. The purpose of this method is to find TLS but not every TLS is a functional block. In this paper, we propose to set additional parameters to the Tangled-logic finder algorithm to specify our search for better functional partition. The combination of structural analysis, pattern matching and TLS extracting make up the deficiency of each method used separately.

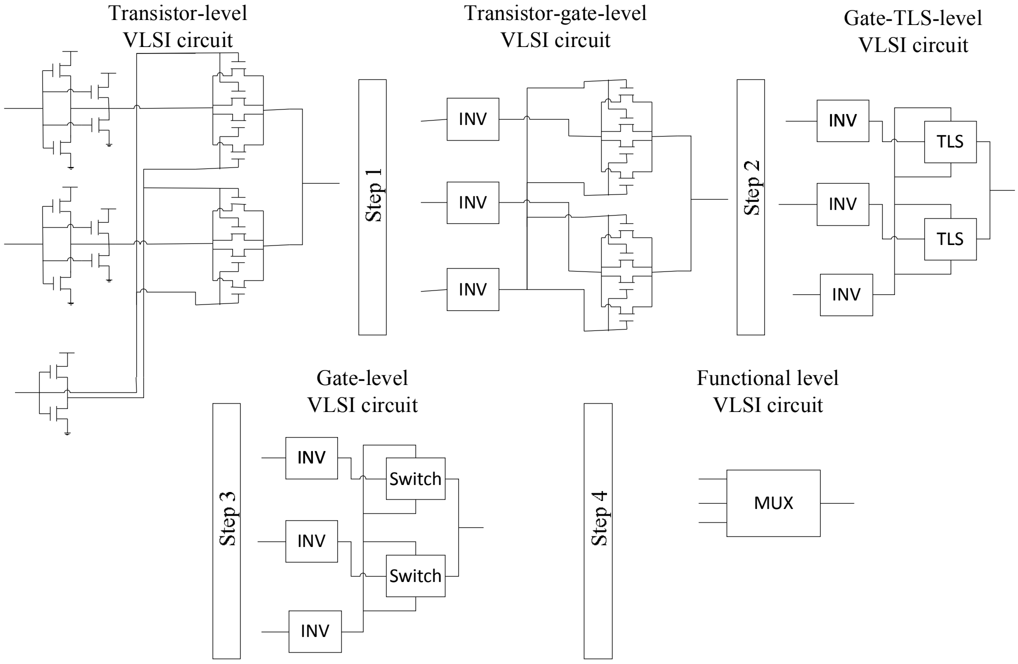
3 A Method To Find Functional Groups

To Based of methods and algorithms described in section 2, we propose an algorithm to find functional group from transistor-level VLSI circuit. This method consists of five steps:

- Step 1: Structural analysis of transistor-level VLSI circuit.
- Step 2: TLS extracting from transistor-gate-level VLSI circuit.
- Step 3: Pattern matching with extracted gate-TLS-level TLS and library or manual TLS analysis.
- Step 4: Pattern matching with extracted gate-level TLS and library or manual TLS analysis. Genetic algorithm for functional block initial vertices search. Genetic algorithm is used for functional block initial vertices search.
- Step 5: Repeat Step 4 until the necessary functional level of hierarchy will be obtained.

Figure 3 shows data structures for all steps.

Figure 3: Data structures for functional block extraction steps.



Step 1 is described in (Lei Yang, 2006). In **step 2** TLS extracting method (that is described in section 2) could be more efficient when the best candidate is chosen due to best GTL value.

Step 3 is the conversion of the gate-TLS-level VLSI circuit into the gate-level VLSI circuit. Obtaining library functional block from TLS is less complicated process then from transistor-gate-level VLSI circuit. If library of functional patterns is available, we do not need to match all patterns with all circuit. Each extracted TLS will have a set of structural parameters for example number of TLS elements, elements interconnections. Comparing TLS structural parameters with same parameters of pattern functional blocks simplifies this step. After the third step we will obtain the gate-level VLSI circuit.

Using common chains data on **step 4** the higher-hierarchical functional levels of VLSI circuit could be extracted. In the next section we will show modified steps of functional extraction on real VLSI circuit sample.

At this step, genetic algorithm performs the search of those initial vertices that will allow us to obtain functional blocks. The correct choice of the initial vertex determines the result of the functional block extraction. In (Riham Moharam, 2017) genetic algorithm uses on graph tasks. In (Kaiqi Zhang, 2017), a genetic algorithm is used to extract a subset of vertices in a graph.

The main steps of search with genetic algorithm are:

1. Analyses of scheme graph connectivity. Each of the connectivity components considers separately.

2. Random selected vertices are the individuals of the population. Blocks begin to be built from each vertex using the nGTLS metric.

3. The fitness function is the minimum value of the nGTLS function, which was calculated for a block constructed on the basis of the current individual after several steps.

4. The individuals of the population are ranked according to the fitness function.

5. The lower quarter of the population is excluded from consideration.

6. Among the better half, pairs are randomly formed. Those pairs complement the population after the crossing.

7. The crossing procedure:

- Finding the shortest way between two vertices.
- The new individual to be added to the population is the vertex situated in the middle of the shortest way.

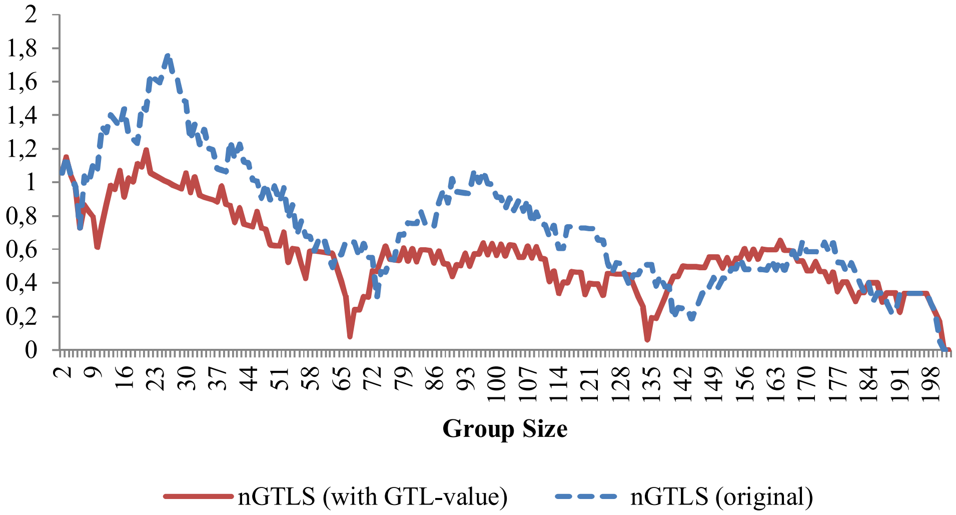
8. The supplemented population passes through a mutation. If an individual mutates, it is excluded from the population. New individual adds to its place corresponding to one of the neighboring vertices of the one that underwent a mutation.

9. The new generation goes from steps 3 to 8. The algorithm completes its work after the predetermined number of generations.

4 Experimental Results

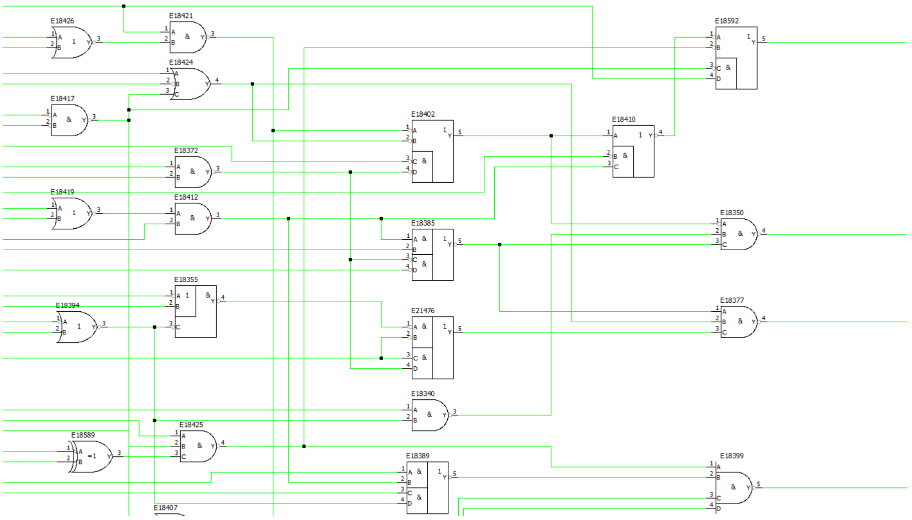
For nGTLS metrics obtained by old and new methods for transistor-level VLSI circuit TLS groups are shown on figure 4. New method curve on figure 4 has precise and more accurate minima than initial method curve.

Figure 4: Transistor-level VLSI circuit TLS groups.



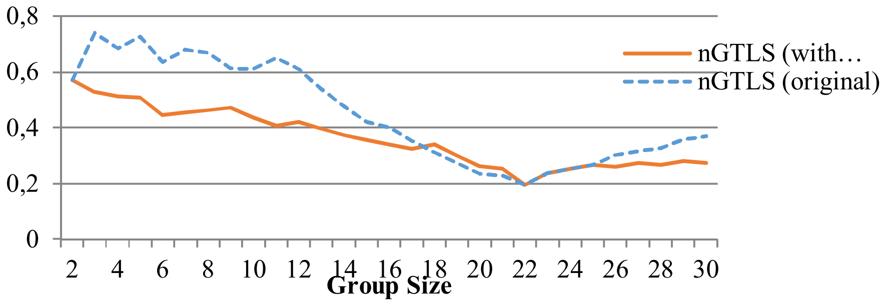
Functional group of gate-level VLSI circuit is shown on figure 5.

Figure 5: Functional group of gate-level VLSI circuit.



Curves on figure 6 describe the group of tangled logic (GTL) extracted from gate-level VLSI circuit. Both metrics curves have minima when group size is 22 elements. This group is real sample of functional block. New nGTLS metric curve is more plane than old metric at the beginning.

Figure 6: GTL from gate-level VLSI circuit.



Due to the new linear ordering the algorithm chooses more suitable candidate thus minima are more accurate. In the example on figure 4 original method minimum is group of 72 elements. The real size of group is 67 elements. The difference was caused by inaccurate select of candidates.

5 Summary

This paper introduces a new method of automatic transistor-level VLSI circuit analysis, which combines structural analysis, pattern matching and improved TLS extraction. The hierarchical representation of VLSI circuit can help with verification of a circuit layout, speeding up circuit simulation, design-for-testability and built-in-self-test, floorplanning and resynthesis of VLSI. Our method extracts higher hierarchical functional blocks from transistor-level VLSI circuit with regular and irregular structure. TLS blocks speed-up the pattern matching and allow to obtain functional block without a pattern. Due to GTL-based linear ordering and genetic algorithm TLS extracting routing has higher degree of accuracy. Future works seek to expand the technics of technology- and library-independent VLSI circuit analysis, and to figure out the new ways of pattern matching for TLS groups.

6 Acknowledgements

This work was supported by Competitiveness Growth Program of the Federal Autonomous Educational Institution of Higher Professional Education National Research Nuclear University MEPhI (Moscow Engineering Physics Institute)

References

A. Lester, P. B.-S. (1998). YAGLE, a second generation functional abstractor for CMOS VLSI circuits. *Proceedings of the Tenth International Conference on Microelectronics*, 265–268.

Boehner, M. (1988). LOGEX—an automatic logic extractor from transistor to gate level for CMOS technology. *Proc. IEEE/ACM Des. Automation Conf.*, 517–522.

F. Luellau, T. H. (1984). A technology independent block extraction algorithm. *Proc. IEEE/ACM Des. Automation Conf.*, 610–615.

G. Pelz, U. R. (1994). Pattern matching and refinement hybrid approach to circuit comparison. *IEEE Trans. Comput.-Aided Des.*, 264–275.

H. Graeb, S. Z. (2001). The sizing rules method for analog integrated circuit design. *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, 343–349.

I. Parulkar, M. B. (1994). Extraction of a high-level structural representation from circuit description with applications to DFT/BIST. *Proc. IEEE/ACM Des. Automation Conf.*, 345–350.

K.-T. Huang, D. O. (1995). A novel graph algorithm for circuit recognition. *Proc. IEEE Int. Symp. Circuits Systems*, 1695–1698.

Kaiqi Zhang, H. D. (2017). Maximizing influence in a social network: Improved results using a genetic algorithm. *Physica A: Statistical Mechanics and its Applications, Volume 478*, 20-30.

Kundu, S. (1998). GateMaker: a transistor to gate level model extractor for simulation, automatic test pattern generation and verification. *Proc. Int. Test Conf.*, 372–381.

Lei Yang, C.-J. R. (2006). FROSTY: A program for fast extraction of high-level structural representation from circuit description for industrial CMOS circuits. *INTEGRATION, the VLSI journal 39*, 311–339.

M. Ohlrich, C. E. (1993). SubGemini: identifying subcircuits using a fast subgraph isomorphism algorithm. *Proc. IEEE/ACM Des. Automation Conf.*, 31–37.

M.S. Abadir, J. F. (1990). An improved layout verification algorithm (LAVA). *Proc. Eur. Des. Automation Conf.*, 391–395.

Riham Moharam, E. M. (2017). Genetic algorithms to balanced tree structures in graphs. *Swarm and Evolutionary Computation, Volume 32*, 132-139.

S. Jolly, A. P. (2002). Automated equivalence checking of switch level circuits. *Proc. IEEE/ACM Des. Automation Conf.*, 299–304.

T. Jindal and et al. (2010). Detecting tangled logic structures in vlsi netlists. *Design Automation Conference (DAC)*, 603–608.

T.J. Thatcher, R. S. (1992). Automatic partitioning and dynamic mixed-mode simulation. *Proc. IEEE Custom Integrated Circuits Conf.*, 12.7.1–12.7.4.