Logic Gate Recognition in Guardian LVS

Introduction

This paper describes the recognition of regular CMOS gates in transistor-level circuits. The structural recognition algorithms using the rule-based techniques are the most efficient for logic gate identification [1, 2, 3, 4]. This class of algorithms is very fast and can easily find static logic gates, such as inverters, NAND, NOR, AOI and OAI gates. After the recognition, modified netlists can be compared in terms of logic gates and remaining transistors. Annotated graphs representing the connectivity of the netlists are much smaller in size and more distinguishable in structure than the graphs in transistor level. So the circuit comparison problem treated as an instance of graph isomorphism problem [5] is solved more efficiently.

Gate Recognition

Guardian LVS recognizes logic gates in CMOS circuits from their transistor-level. Groups of transistors that form the gates are represented by logical configurations. If connected transistor groups do not form logic gates, Guardian LVS can create logical configurations from parallel-series connected transistors of the same type. LVS verification is performed at the level of the logical configurations after these transformations.

Some settings should be adjusted in the Guardian LVS user interface to perform logic gate recognition:

- The option "**Logic gates**" should be chosen in "**Model Settings**" panel for MOS transistors.
- Power and ground nets should be specified in "**Net Settings**" panel. You can use the suffixes "**:P**", "**:G**" to specify power and ground nets in SPICE netlist. In this case you should turn on the option "**Remove Net Name Suffix starting from colon (:)**". The suffixes will be deleted from colon in net names and these nets will be interpreted as power and ground ones.

Note: it is not necessary to turn on the options "**Parallel reduction**" and "**Series reduction**" in "**Model Settings**" panel for MOS transistors to recognize inverters. But if you want to recognize more complex logic gates these options have to be turned on .

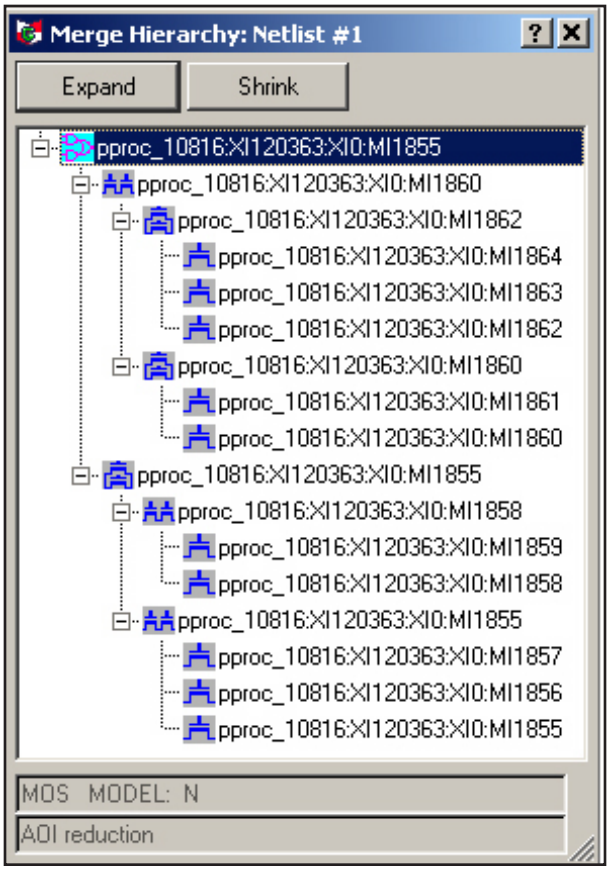

Figure 1. Structure of AOI gate

The logic gates are formed only for "correctly organized" MOS transistors:

- All transistors of logic gate must have the same number of pins;
- P-type and N-type components of logical gate must have the same number of elements after reduction operations;

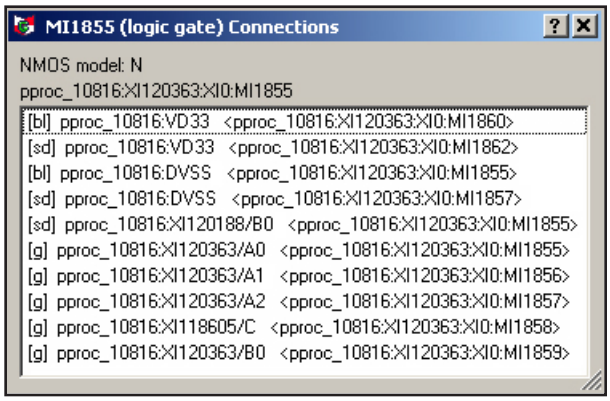

Figure 2. Connectivity of AOI gate

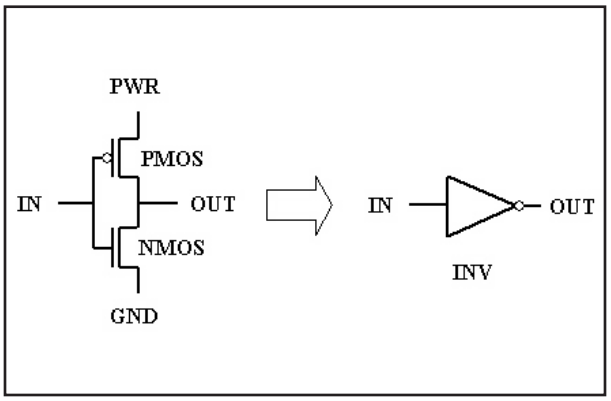

Figure 3. INV

- P-type and N-type components must have the same set of different nets connected to gate terminals of transistors;
- Bulk terminals of P-type (N-type) transistors must be connected to one power (ground) net.

The user can choose whether or not to mix the transistor models in P-type or N-type component using the option "**Match model**" in "**Model Settings**" panel.

Guardian LVS detects logical equivalence of pin groups in logical gates, so you can swap equivalent pins and devices at transistor level. For example, the user can swap the input pins of NAND or NOR gates.

Discrepancies and match nodes are reported on gate level, and the inspection tools can be used to investigate the structure or connectivity of logic gates. For example, the **Merge Hierarchy** tool (Figure 1) shows how an AOI gate has been created.

The **Connectivity Traversing** tool contains the logic gate connectivity information, i.e. the list of all nets connected to the terminals of this gate (see Figure 2).

Types of Recognized Logic Gates

Guardian LVS recognizes the following types of logic gates:

• INV — CMOS inverter (Figure 3.)

The pair of correctly organized (see previous section) transistors of P- and N- types (Figure 3) is replaced by a logical configuration **INV**

• **NAND_N** — CMOS **NAND** with N inputs (Figure 4)

Two groups of correctly organized PMOS and NMOS transistors connected in parallel and in series, correspondingly, can be replaced by **NAND** gate with N inputs and one output. Once they are reduced, the resulting NAND gate will have N swappable input terminals.

To produce NAND gates, the options "**Parallel reduction**" and "**Series reduction**" in "**Model Settings**" panel should be turned on.

- **NOR_N** — CMOS **NOR** with N inputs (Figure 5.)

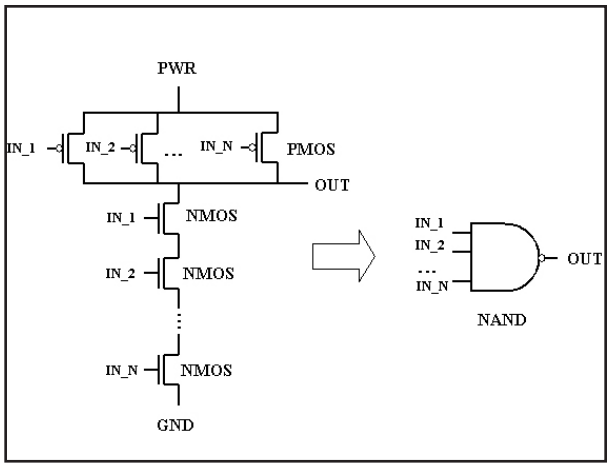

Figure 4. NAND_N

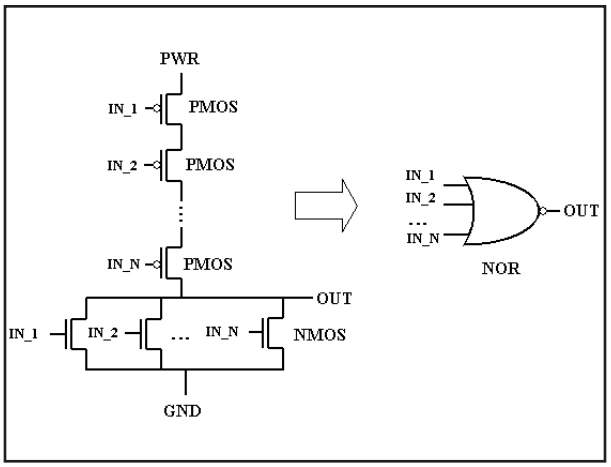

Figure 5. NOR_N

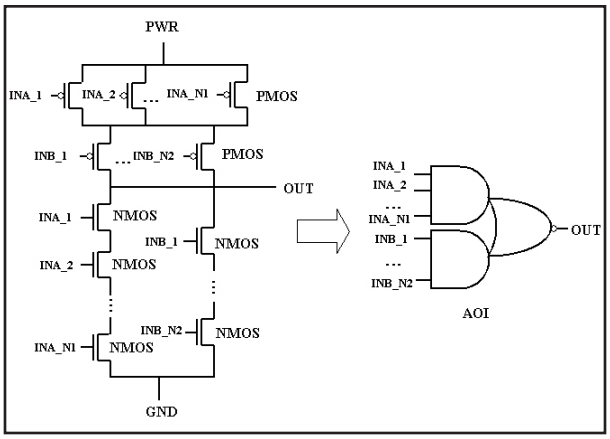

Figure 6. AOI_N1_N2

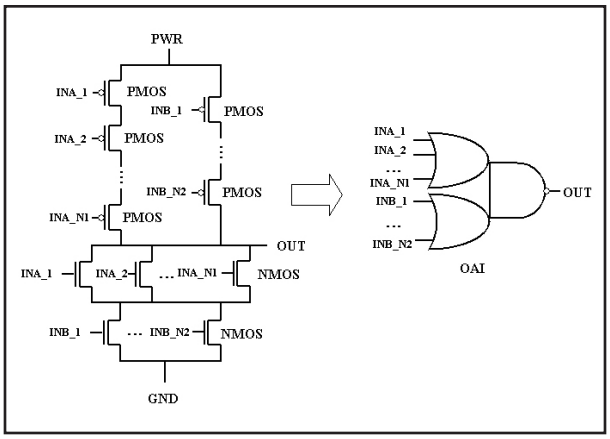

Figure 7. OAI_N1_N2

Two groups of correctly organized PMOS and NMOS transistors connected in series and in parallel, correspondingly, can be replaced by NOR gate with N inputs and one output. All N input terminals are swappable.

To produce NOR gates, the options "**Parallel reduction**" and "**Series reduction**" in "**Model Settings**" panel should be turned on.

- **AOI_N1_…_NK** — CMOS AND-OR-Invert logical configuration represented by K AND elements with N1, N2, …, NK inputs in each, accordingly, and OR-Invert block (Figure 6)

The input terminals inside each AND configuration of AOI gate are swappable, for example in Figure 6 the inputs INA_1, INA_2, …, INA_N1 are swappable, and the inputs INB_1, …, INB_N2 can be swapped. Also, the groups of PMOS transistors connected in parallel are interchangeable. So the parallel transistors connected to INB_1, …, INB_N2 nets can be placed above the group of transistors connected to INA_1, INA_2, …, INA_N1.

To produce AOI gates, you need to turn on the "**Parallel reduction**" and "**Series reduction**" options in "**Model Settings**" panel.

- **OAI_N1_…_NK** — CMOS OR-AND-Invert logical configuration represented by K OR elements with N1, N2, …, NK inputs in each, accordingly, and AND-Invert block (Figure 7)

The input terminals inside each OR configuration of OAI gate are swappable. For example, in Figure 7 the inputs INA_1, INA_2, …, INA_N1 are swappable, and the inputs INB_1, …, INB_N2 can be swapped. Also the groups of NMOS transistors connected in parallel are interchangeable. So the parallel transistors connected to INB_1, …, INB_N2 nets can be placed above the group of transistors connected to INA_1, INA_2, …, INA_N1.

To produce OAI gates, the "**Parallel reduction**" and "**Series reduction**" options should be turned on in the "**Model Settings**" panel.

Conclusion

Logic gate recognition in the transistor-level circuits has been proposed and realized in Guardian LVS. This recognition allows more successful LVS netlist comparison.

References

[1] A.Lester, P.Bazargan-Sabet, A.Greiner, "YANGLE, a second generation functional abstractor for CMOS VLSI circuits," *Proceedings of the10th International Conference on Microelectronics*, 1998, pp.265-268.

[2] M.Boehner, "LOGEX – an automatic logic extractor from transistor to gate level for CMOS technology," *Proc. IEEE/ACM Design Automation Conference*, 1988, pp.517-522.

[3] L.Yang, C-J.R.Shi, "FROSTY: A program for fast extraction of high-level structural representation from circuit description for industrial CMOS circuits," *Integration, the VLSI Journal,* V. 39, N 2, 2005.

[4] L.Yang, C-J.R.Shi, "FROSTY: A fast hierarchy extractor for industrial CMOS circuits," *UWEE Technical Report,* N UWEETR-2003-0011, 2003.

[5] C.Ebeling, "GeminiII: A Second Generation Layout Validation Program", *IEEE International Conference on Computer-Aided Design*, 1988, pp. 322-325.