Marcin SZELEST
DELPHI POLAND S.A. TECHNICAL CENTER KRAKOW,
30-399 Krakow, ul. Podgorki Tynieckie 2

An automatic formal verification tool for schematic-level design of an electrical project

M.Sc. Marcin SZELEST

Received M.Sc. degree in Electronic and Telecommunication from Silesian University of Technology, in 2004. From 2004 to 2008 he worked as Digital IC Designer at Evatronix. He has joined automotive industry in 2008. Since then, he has held multiple engineering positions in Delphi Technical Center Krakow starting from Electrical Engineering, through Advanced Electrical Engineering – Modeling and Simulations to Advanced Design Automation Eng. He is licensed HAM-radio operator since 1995 and IEEE Senior Member.

e-mail: Marcin.Szelest.PL@IEEE.ORG

Abstract

A schematic diagram can be interpreted as set of components connected together to build a network. The network is a structure whose topology can be read and analyzed. The automatic formal verification aims at detecting some non-trivial mistakes, made by an electrical engineer, connected with the structure of schematic, building blocks and typical functional sub circuits. The paper presents the approach for realization of an automatic formal detection tool starting from the definition of schematic structure, through connectivity with a schematic-capture tool, to a sub-circuit identification algorithm.

Keywords: Electrical Fault Detection, Formal Verification, Design Rule Checking, Electrical Rule Checking.

Narzędzie do automatycznej weryfikacji formalnej schematu ideowego sprawdzające poprawność schematu już we wczesnych etapach procesu projektowego

Streszczenie

Schemat ideowy układu elektronicznego może być postrzegany jako zbiór wzajemnie połączonych komponentów elektronicznych tworzących pewien graf połączeń. Graf ten jest strukturą, której topologia może być odczytana i przeanalizowana. Automatyczna weryfikacja formalna schematu ideowego ma na celu wykrycie nietrywialnych błędów (związanych ze strukturą tegoż schematu), popełnionych przez projektanta w trakcie opracowywania schematu. Błędy te są efektem pominięcia lub niewłaściwego wykorzystania elementów elektronicznych w typowych podukładach (blokach funkcjonalnych). W artykule opisano realizację komputerowego narzędzia wspomagającego proces automatycznej weryfikacji formalnej, począwszy od określenia sposobu zapisu schematu, przez integrację z komercyjnymi narzędziami EDA (ang. *Electronic Design Automation*) aż po określenie algorytmów identyfikujących podobwody. Opracowane narzędzie umożliwia dokonanie weryfikacji schematu ideowego już w bardzo wczesnych etapach procesu projektowego, co znacząco redukuje liczbę pętli w procesie projektowym, wpływając korzystnie na końcowy koszt opracowania produktu oraz skracając czas niezbędny do przygotowania urządzenia do produkcji.

Słowa kluczowe: wykrywanie błędów, weryfikacja formalna, sprawdzanie reguł projektowych, sprawdzanie reguł elektrycznych.

1. Introduction

Electrical circuits become more and more complex every year. The time necessary for testing and corrective actions grows exponentially with the design complexity at the development stage, where a certain error is detected [1]. That is why the verification process of design documentation should be started as early as possible. In the following design steps, in which the circuit description becomes more detailed, the verification should be extended by additional factors that reflect the increased description of the project. This approach makes early detection of a design error possible, which significantly reduces cost and complexity of corrective actions.

EDA (Electrical Design Automation) tools provide two basic methods of formal verification: ERC (Electrical Rule Checking) and DRC (Design Rule Checking). Unfortunately both of them are focused on – only – two electrical specializations: design of integrated circuits (IC design) [2] and design of printed circuit boards (PCB design) [3]. Furthermore, violations that may be detected by them are limited to trivial cases as: shorting to battery, shorting to ground, unconnected pin or inconsistency between schematic and circuit layout.

Lack of computer aided tools for formal verification, that can be done on a schematic level, made us create our own tool for this purpose. This paper describes its main principles as well as practical implementation.

2. Formal verification of a schematic

On the basis of a schematic diagram only (without the simulation models) simulation of its behavior is not possible, however structure of the circuit can be analyzed for compliance with design rules. The following example rules may be checked in this aspect:

- necessity of using gate resistors in MOSFET keys [4];
- necessity of using discharge capacitors (ESD) on each input pin of IC [5];
- necessity of using decoupling capacitors in certain sub circuits [6];
- obligation of using clamping circuits on each ADC [7];
- necessity of placing varistor on each pin of external supply connector [8].

Please note, that basing on commercially existing ERC and DRC tools, none of the mentioned above problems may be easily verified.

3. Example problem – oscillations in a MOSFET key

Keying circuits, built with MOSFET transistors, are prone to the phenomenon of unwanted fading oscillations (also called "ringing"), that occur during switching off the transistor. These oscillations are the result of the existence of parasitic capacitances and inductances of the key and their influence on transient waveforms [9].

This phenomenon, in addition to degradation of functionality, significantly worsens electromagnetic compatibility. The frequency of oscillations often reaches tens of MHz, which can interfere with radio reception in the FM broadcasting band. In addition, one should emphasize the fact that measurements of the oscillations are - due to high frequency and short duration - extremely difficult to perform. They can be taken only with the use of high-quality measurement equipment [10]. Fig. 1 presents the schematic diagram of a MOSFET key, while Fig. 2 shows the simulation result for this circuit with parasitic effects taken into consideration. As it can be seen, each transition of the key causes fading HF oscillations. These oscillations may be suppressed by reducing the Q-factor of the simulated circuit. The Q-factor is reduced by a series resistance connected between the driver and gates pin of the transistor. The value of this resistance is a compromise between the levels of oscillations and the desired switching time. Usually this value varies in the range of few Ohms.

Basing on the description above, the following design rule may be defined:

If the design contains keys realized with the use of MOSFET transistors, an electrical engineer must use gate resistance to suppress unwanted oscillations. The value of this resistance must be within a specified range.

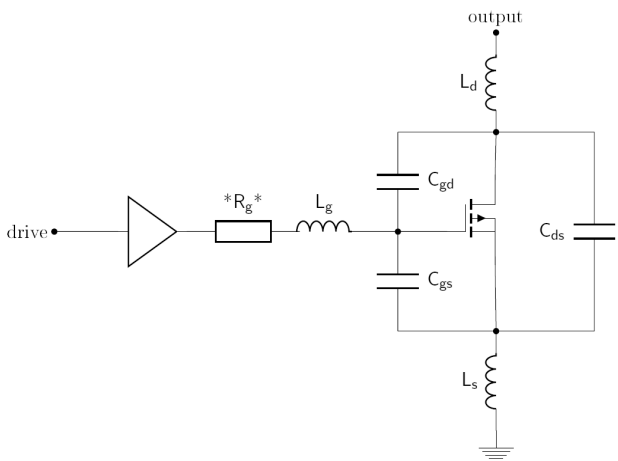

Fig. 1. Schematic of the example MOSFET key with parasitic capacitances (C_{gd}, C_{gs}, C_{ds}) and inductances (L_g, L_d, L_s)
Rys. 1. Schemat klucza tranzystorowego (MOSFET) z uwzględnieniem pasożytniczych pojemności (C_{gd}, C_{gs}, C_{ds}) i indukcyjności (L_g, L_d, L_s)

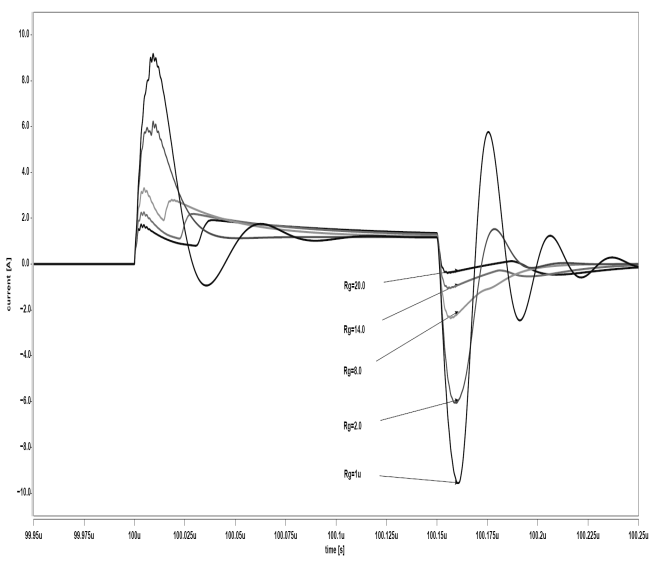

Fig. 2. Fading HF oscillations, caused be transient state during transition, as a function of series gate resistance R_g
Rys. 2. Gasnące oscylacje wielkiej częstotliwości, w zależności od rezystancji w obwodzie bramki R_g, wywołane stanami przejściowymi w trakcie przełączania klucza

4. Storing electrical schematic on a computer

The schematic diagram of an electrical circuit may be written and stored in many ways. In the most trivial case it is a handmade picture, however it is obvious that this picture cannot be directly used in EDA tools. Currently used methods of writing and storing the schematic evolved from simple netlist (Spice simulators), through Hardware Description Languages (HDL) to strictly object-oriented description. Each of the mentioned methods has pros and cons. The notation, that is intuitive for a human, may be very difficult for a computer and vice versa.

Looking at the hand-made drawing of the circuit diagram it can be concluded that:

- each component that exists on the schematic is an object;
- the objects are of different types (resistor, capacitor, inductor, IC, connector, etc.);
- objects are connected together with the use of nets;
- pins of objects cannot be connected directly - nets must be used for this purpose;
- net is not a straight line - it can consist of multiple straight lines called segments;
- ending of segments form a group called a junction;
- each component and net may have a label associated with them.

The characteristics mentioned above indicate, that each electrical schematic may be successfully described with object-oriented techniques.

Data Model is a description of the objects represented by a computer system together with their properties and relationships [11]. It can be created in many ways, however the usage of dedicated tools is preferred. These tools provide a convenient graphical interface for creation of a model structure and contain mechanisms for model validation and automatic source code generation. One such a tool is the Eclipse Modeling Framework (EMF) [12]. Figure 3 shows the Data Model for an electrical schematic created with the use of EMF. According to Fig. 3:

- electrical schematic consists of at least one sheet (notation 1..* at arrowhead denotes that);
- sheet contains exactly one frame and any number of components and nets;
- each component and net may have associated no more than one label;
- each net consists of at least one segment;
- each component has its own unique identifier, type definition, list of attributes and some number of pins;
- components may be connected by nets (groups of segments).

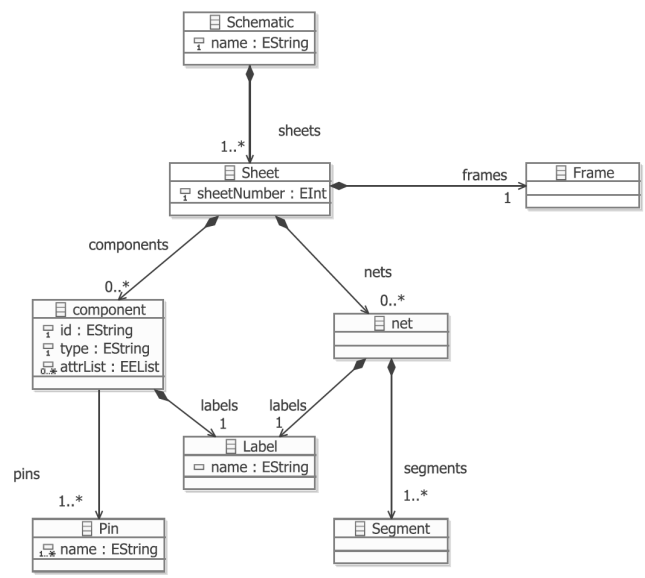

Fig. 3. EMF Data Model that represents a general electrical schematic
Rys. 3. Model danych EMF reprezentujący strukturę schematu ideowego

Basing on the structure defined above, automatic generation of a source code is possible. The generated code consists of definitions of classes with a set of methods to handle the objects constructed with the use of these classes. Please note that it is a generic model of a schematic and this model does not contain any usable information of any schematic - it is a data model only without values and information stored inside.

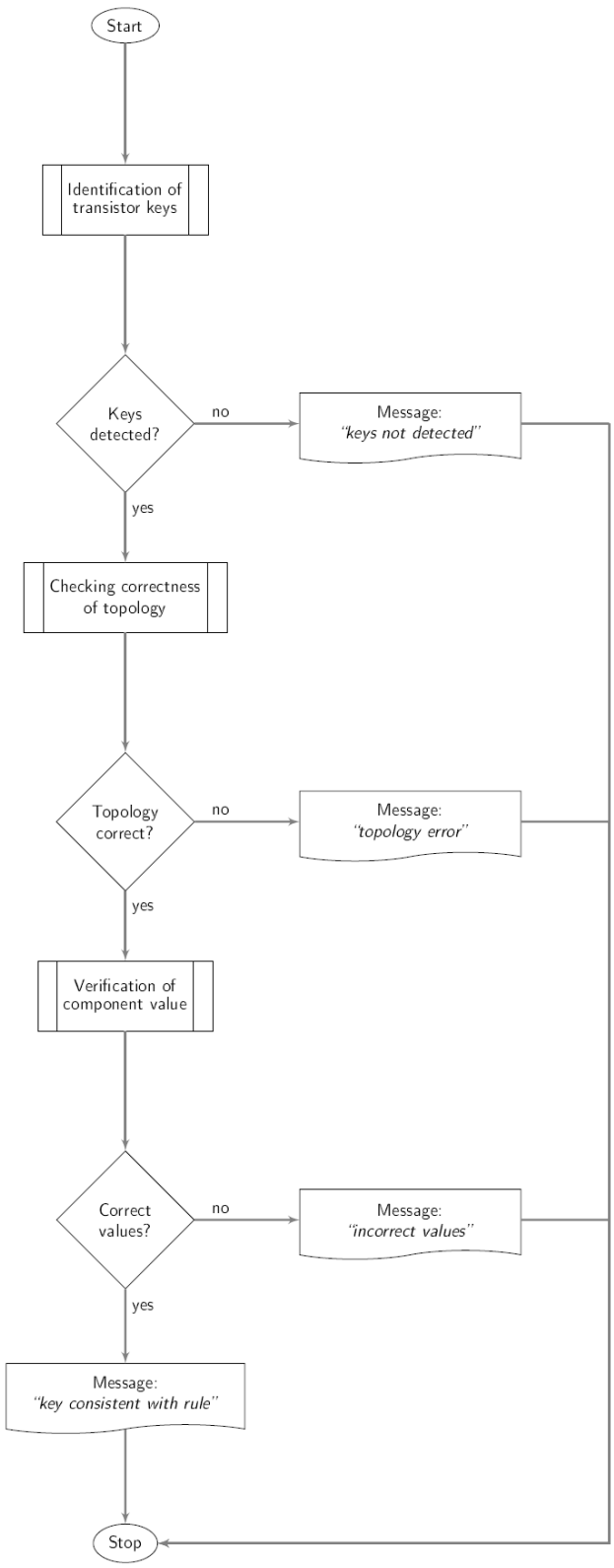

Fig. 4. High-level block diagram of the detection algorithm
Rys. 4. Wysokopoziomowy schemat blokowy algorytmu detekcji

The object-oriented description is also used in EDIF Standard (Electronic Design Interchange Format). The standard defines a method to write and store the schematic that can be easily reused by different EDA Tools [13].

5. Algorithm

The example design rule defined in Section 3 and the data model from Section 4 are the starting point for creation of an algorithm. The algorithm should be capable of validating the entire design for compliance with the mentioned design rule. The first step of automatic formal verification is identification of all sub circuits that performs key functionality. The obtained list must be reduced to sub-circuits that are compliant with the assumed topology - for example all keys that do not contain MOSFET transistors should be neglected. Finally, for each key from the reduced list, the value of gate resistance should be checked and compared with the demanded range. Fig. 4 shows the high-level diagram of the algorithm that realizes automatic formal verification for the example design rule.

A. Identification of a Key

First step of formal verification algorithm is key identification. Whole schematic must be analyzed and all sub-circuits that undergo to certain design rule should be extracted from them. For example, to extract MOSFET switches vulnerable to ringing phenomenon, following actions must be performed:

- identification of all MOSFETS that exist on analyzed schematic (because our rule is defined only for circuits build with these transistors);
- exclusion - from obtained list of MOSFETs – transistors that work in different configuration than expected;
- comparison of sub-circuit topology details with respect to generic MOSFET key circuit.

Basing on the schematic that is shown on Fig. 1, list of typical features of keying circuit may be created. The list, created by experienced engineer, should contain a set of simple and clear topologic rules that must be fulfilled to classify analyzed circuit as compliant with certain design rule. During creation of the list special care should be devoted to avoid exclusion of allowable deviation from the generic circuit. For example, two resistors connected in parallel instead of one should be allowed. The list is created only once, during implementation of certain rule.

B. Topology Verification for Sub-Circuit

Successful classification of MOSFET key does not guarantee that topology of the circuit is fully compliant with generic building block. Therefore additional check must be performed to validate specific features of topology. For our example design rule, two additional factors should be analyzed:

- existence of series resistor in gate circuit;
- lack of any additional components connected directly to gate of the MOSTET (in spite of mentioned earlier gate resistor).

C. Component's Value Verification

After making sure, that the analyzed sub-circuit undergoes to a certain design rule, verification of the component values may be performed. In the case of our example design rule, only one component should be validated in aspect of its value – the gate resistor. According to [12], if the resistance Rg is equal to 4.7Ω, the circuit may be classified as compliant with the design rule for avoiding unwanted damping oscillations.

6. Implementation

The presented algorithm was implemented as stand-alone application, written in C# language. The application controls the schematic capture tool (DxDesigner from Mentorgraphics, inc.). Communication with the schematic and reading process is realized with the use of COM+ interface. The application performs formal verification and returns a summary report with listing all detected violations. The final report is a Microsoft Excel Worksheet, what guarantee flexibility and post-processing possibility after the automatic verification process.

As test projects, three automotive electrical designs were chosen. Deliberately untagged versions of designs were chosen to obtain more violations of design rules. The test projects are characterized in Table I.

Tab. 1. Short characteristic of the test projects
Tab. 1. Krótka charakterystyka trzech testowych projektów

Name	Nb. of worksheets	Nb. of nodes	Nb. of resistors	Nb. of capacitors	Nb. Of transistors	Nb. of MOSFETS	Total 2-port components	Nb. of ICs
Project 1	21	1096	748	628	66	60	1474	22
Project 2	3	116	38	57	10	2	107	5
Project 3	15	397	245	271	31	27	581	22

In order to demonstrate benefits of using the developed tool, two aspects were analyzed: time savings and reduction of human interaction thanks to the automatic verification.

A. Reduction of time effort

For this purpose, four actions required for the hand-made analysis were specified. For each of the specified actions, a separate time effort equation was defined. According to these equations, the total time for three example projects was calculated. The time given in Table 2 is presented in [hh : mm : ss] form. The time of human-made analysis depends on many factors, so the assumptions made in Table 2 should be interpreted as example values.

Tab. 2. Estimated time effort for human-made formal verification of a schematic
Tab. 2. Szacunkowa złożoność czasowa dla analizy ręcznej

Action	Equation	Project 1	Project 2	Project 3
identification of MOSFET transistors	3mins/worksheet	1:03:00	0:09:00	0:45:00
identification of MOSFET keys	1min/MOSFET	1:00:00	0:02:00	0:27:00
topology check	1.5min/key	1:30:00	0:03:00	0:40:30
Preparation of a report	2mins/key	2:00:00	0:04:00	0:54:00
Total		5:33:00	0:18:00	2:46:00

The estimated time effort was compared with the time effort measured for automatic approach. The results of comparison are given in Table 3. Please note that the time savings are significant for each of the three example projects.

Tab. 3. Comparison of the estimated effort for human-made and automatic formal verification
Tab. 3. Porównanie złożoności czasowej między analizą wykonywaną ręcznie i podejściem automatycznym

Name	Manual verification	Automatic verification
Project 1	05:33:00	00:01:01
Project 2	00:18:00	00:00:07
Project 3	02:46:00	00:00:28

B. Reduction of human interaction

With the use of FMEA (Failure Mode and Effects Analysis) methodology, a table of potential errors and their effects was created [14]. According to the table, the risk factors before and after the use of the presented tool were calculated. The comparison of these factors is shown in Table 4.

Tab. 4. Summary of potential errors and the sum of risk factors before and after automatic formal verification
Tab. 4. Porównanie potencjalnych błędów i czynników ryzyka przed i po dokonaniu automatycznej weryfikacji formalnej

Cause of potential error	Risk factors before	Risk factors after
Mistake during edition of data	1110	716
Error in base documentation	558	270
Error propagated in documentation	273	196
Incorrect requirements	127	92

7. Conclusions

The automatic formal verification is particularly useful for large projects, where manual approach for checking of design rules is time-consuming and cumbersome. In this paper the author has proved that the automatic formal verification is possible and very beneficial because of:

- time-effort savings;
- reduction of human-interaction;
- possibility of early detection of design mistakes.

The described methodology of formal verification was practically implemented by the author and is already used in Delphi Automotive. The developed tool is treated as an extension of a set of expert tools that are used during the design process.

8. References

[1] Mitretek Systems, Inc., "Developing Functional Requirements for ITS Projects," Intelligent Transportation Systems Joint Program Office US Department of Transportation, Tech. Rep., April 2002.
[2] Assura Physical Verification User Guide, Cadence, January 2011.
[3] DxDesigner Users GuideFor Expedition Flow, MentorGraphics, 2011.
[4] AN11158 Understanding power MOSFET data sheet parameters, NXP, April 2012, application Note.
[5] Lin J., Duvvury C., Haroun B., Oguzman I., and Somayaji A.: A failsafe ESD protection circuit with 230 fF linear capacitance for high-speed/ high-precision 0.18 /spl mu/m CMOS I/O application, in Electron Devices Meeting, 2002. International, Dec. 2002, pp. 349 –352.
[6] Fan J., Knighten J., Orlandi A., Smith N., and Drewniak J.: Quantifying decoupling capacitor location, in Electromagnetic Compatibility, 2000. IEEE International Symposium on, vol. 2, 2000, pp. 761 –766 vol.2.
[7] TI, "TL7726 Hex Clamping Circuit," Texas Instruments, Tech. Rep., 1994.
[8] Shirai Y., Miyato Y., Taguchi M., Shiotsu M., Hatta H., Muroya S., Chiba M., and Nitta T.: Over-voltage suppression in a fault current limiter by a zno varistor, Applied Superconductivity, IEEE Transactions on, vol. 13, no. 2, pp. 2064 – 2067, June 2003.
[9] IOR, "The Dos and Donts of Using MOS-Gated Transistors, AN-936," International Rectifier, Tech. Rep.
[10] Application Bulletin AB-9, "Suppressing MOSFET Gate Ringing in Converters: Selection of a Gate Resistor," Fairchild Semiconductor, Tech. Rep., July 1998.
[11] Bezivin J., Jouault F., and Touzet D.: Principles, standards and tools for model engineering, in Engineering of Complex Computer Systems, 2005. ICECCS 2005. Proceedings. 10th IEEE International Conference on, June 2005, pp. 28 – 29.
[12] El Boussaidi G. and Mili H.: A model-driven framework for representing and applying design patterns, in Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International, vol. 1, July 2007, pp. 97 –100.
[13] Kahn H. and Goldman R.: The Electronic Design Interchange Format EDIF: present and future , in Design Automation Conference, 1992. Proceedings., 29th ACM/IEEE, Jun 1992, pp. 666 –671.
[14] Legg J. M.: Computerized Approach for Matrix-Form FMEA, Reliability, IEEE Transactions on, vol. R-27, no. 4, pp. 254–257, Oct.

otrzymano / received: 03.03.2014
przyjęto do druku / accepted: 02.05.2014 *artykuł recenzowany / revised paper*