Structure-Preserving Modeling of Safety-Critical Combinational Circuits

Feim Ridvan RASIM, Canan KOCAR, Sebastian M. SATTLER
Email: {feim.rasim, sebastian.sattler}@fau.de, canan.kocar@studium.uni-erlangen.de

Friedrich-Alexander University Erlangen-Nuremberg, Chair of Reliable Circuits and Systems
Paul-Gordan-Str. 5, 91052 Erlangen, Germany

Abstract— **In this work, a representative combinational circuit is abstracted from transistor level to gate level and a structure preserving transition is carried out into a signal flow graph.**
For creating a signal flow plan it is necessary to swap the nodes and the edges in the signal flow graph. After executing this action the result is a signal flow plan. A value table exhibits the coding of the whole circuit. Then the so called module view is used to get the familiar compact display and neighborhood relations are repeated once more, the resolution method is used.
It is observed that in digital circuits, undefined results can occur but these must be avoided in safety critical circuits. These events have to be secured in practice by costly and expensive verification and testing. In order to deal with the problem now, the structure preserving modeling has to be understood, since this is the only way to achieve a one-purpose, qualitative and cost effective search for errors.

Keywords— **Combinational circuit, structure, modeling, symbolic analysis, signal flow graph, Boolean function**

I. INTRODUCTION

In order to ensure the functional safety of circuits or systems which are regarded as critical to safety, the mutual convert of models and functions is of great importance. The inconsistency problem is omnipresent; therefore the essential claim for conformity with the formal derived function and the function derived from the real structure has a present role [1].
The directed mode of operation of a system should be represented by a circuit or switching table, also called a table of values, one-to-one in the sense of the encoding can be reproduced. In safety-critical circuits it is necessary not defined results, which often occur in complex circuits, to avoid or to monitor. The transferability of circuits into additional and other display possibilities is therefore a necessary property to ensure the functional safety of safety-critical circuits.
In this work, a representative combinational circuit is visualized in various ways. In all these representations, however, it should be noted that the "structurally preserving modeling and transfer" is maintained. This means that the formally derived function must consistently match the function derived from the respective representation type. Both functions must in no case have inconsistencies, since only the fault-free function is included in the circuit [1].
Functional safety can be guaranteed by the *condition of the structure-based modeling and transfer*.

Organization of the paper: First, the theoretical foundations are briefly explained in Chapter II. They are regarded as basic knowledge in order to understand this work. Subsequently, the implementation is described in detail in chapter III and visualized by sketches and models. In the end, the results and the core outline of the work are summarized again and an outlook is given.

II. THEORETICAL FOUNDATIONS

A. *Structural changeover and modeling*

Structurally-faithful modeling unites function and structure one-by-one in the sense of a monomorphism injective - that is, the structure has at most one solution (this is the function) - and of a epimorphism surjective - that is, the function has at least one solution (that is the structure). Such a mapping enables a one-to-one (local-bijective) and understandable description of a generating system.
During transferring into various presentation possibilities the structurally-faithful modeling has a significant role. It is extraordinary important that the formally derived (modeled) function coincide with the function generated by the real structure. Consequently the function has to correspond to reality and shall not exhibit any inconsistencies. Only in this way the functionality of a circuit can be ensured. Structurally-faithfull therefore means that the relation to reality must never be lost during modeling.
During the transfer, it is also important that the function generated from the real structure consistently match the function derived from the signal flow graph or any other type of presentation. Only in this way the functionality of the generated circuit can be ensured.
In addition, there is a structure-based transfer only in the absence of inconsistencies. A transfer of the signal flow graph or of the circuit into a value table must also be structurally-faithfull. Thus the function derived from the evaluation table must correspond to the same function derived from the signal flow graph or the generated circuit.

B. *Signal flow graph*

The signal flow graph (SFG) is a vividly method to present the internal structure of a system or the interaction of several systems. This presentation allows a better understanding of the function as well as the interrelations of one or more systems. In addition, the signal flow graph is the appropriate tool for abstracting functions or connections to the category level (associativity and identity). The signal flow graph is a directed and weighted graph whose nodes represent objects (sets) and edges morphisms (functions). The edges of this graph can be understood in a dual view as small processing units which process incoming signals (edges) in a particular form and then send the result to all outgoing edges (signals). Signal flow graphs are formally defined graphs [2].

C. *RS Buffer*

In complex circuits, many structures exist that can create undefined results. These undefined results must not occur in safety-critical circuits, since otherwise the desired function of the circuit can not be guaranteed. For this reason, the RS Buffer structure is established [3]. It can intercept undefined cases in combination with a dual-rail approach. Thus, it is possible to stabilize a complex circuit in its function without glitch. These stabilized states do not produce unpredictable events and can therefore be processed by the circuit without causing errors. Fig. 1 shows the circuitry of the RS Buffer. The value at the node X in the circuit corresponds to the value at the pin Y, because of the two inverters. Thus, the X is neglected for the sake of clarity in the value table Tab. 1.
On closer examination of Tab. 1, it is noticeable that the RS Buffer triggers a switching process only during assignments $(Z, \bar{Z}) = (1,0)$ and $(Z, \bar{Z}) = (0,1)$. The old state is retained for assignments $(Z, \bar{Z}) = (1,1)$ and $(Z\ \bar{Z}) = (0,0)$.

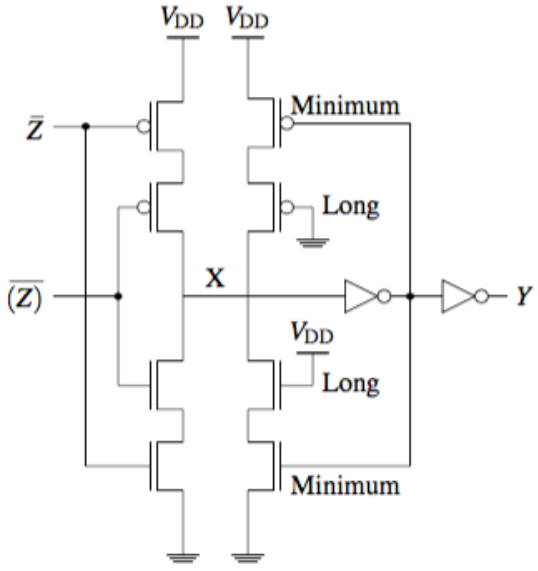

Figure 1: Circuit of the RS Buffer [3]

Z	$\bar{Z}$	Y
0	0	Y
0	1	0
1	0	1
1	1	Y

Table 1: Value table of the RS Buffer [3]

III. IMPLEMENTATION

In this chapter, a NAND2 is considered at the transistor level in Fig. 2. Afterwards the circuit is transferred from the transistor level into the gate level in a structured manner.
It should be noted that the analog circuit, that is, the circuit at the transistor level, is described at the gate level in propositional logic. Subsequently, the circuit is converted into a signal flow graph and signal flow plan at the gate level. Various possibilities for the representation of the output circuit, then such as the evaluation table, the module view or the resolves are presented. With all these possibilities of representation, it should be noted that the respective derived function must not have any inconsistencies that means, the formally derived function must agree with a function generated from a real structure.

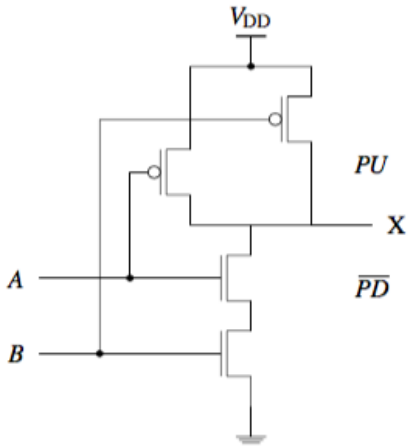

Figure 2: NAND2 at transistor level

Fig. 2 shows a NAND2 at transistor level. This circuitry is a combinational circuit because there are no feedback lines. It is a complex gate with two inputs and an output between which the logical link „NAND“ exists.
A NAND2 outputs „0“ at the output when both inputs are assigned a „1“. This means that if one of the two inputs is assigned a „0“, the output creates a „1“.

A. *Concretization from transistor level to gate level*

The analysis of a circuit at the transistor level is more detailed and more complex than viewing at the gate level, since the representation in gates is only a „model“ which allows a simplified and clear view of the circuitry. The transfer of a

structure at the transistor level into a structure at the gate level is therefore often called an abstraction and serves to increase the clarity and simplify the understanding of the structure.
Nevertheless, from propositional logic and category point of view transistor level is the abstraction (parent) of the gate level (child). This is important to keep in mind.

In the first step is now the NAND2 transferred to the gate level. First, the pull-up (PU) and the pull-down ($\overline{PD}$) are viewed separately from each other. The two transistors of the pull-up are connected in parallel, they must be concatenated.
Furthermore the operating voltage V_{DD} has to be considered in the pull-up. This runs in series with each of the two transistors. The pull-down transistors, on the other hand, are connected in series, which are catenated. The mass potential runs serially to both the two transistors of the pull-down.
Since in the last section it was explained how the transistors in the PU and $\overline{PD}$ are switched, it is now explained how PU and $\overline{PD}$ are related to each other. PU and $\overline{PD}$ are connected by a composition (concatenation). This composition ensures the RS Buffer in Fig. 3.

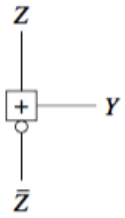

Figure 3: Block view of the RS Buffer [3]

After the pull-down, a switch "¬" is installed, in order to meet the propositional composition. The function of the RS Buffer has already been explained in the basic chapter. The operating voltage is usually designated with V_{DD}, the reference point is the mass with the low-active input ($\overline{\mathrm{GND}}$).

In the second step sub circuits are described as concrete mathematical functions. It should also be noted that each partial circuit is basically at least disjunct, in this case even complementary, this mean a pull-up (PU) and a pull-down ($\overline{PD}$). Pull-Up means that the output is pulled up to the operating voltage V_{DD}. This part of the circuit is low-active since a logical „0“ must be present at the primary input in order to trigger the switching process. The pull-down draws the output to the mass potential. It is referred to as high-active, since a logical „1“ must be present at the primary input, so that a switching process is triggered.
For example, Tab.1 shows the functionality of the RS Buffer. The operation is assumed to be already known. Thus, the circuit consisting of a pull-up and a pull-down has the following equation:

$$Y = PU + \neg\,\overline{PD} = \;\; Y = Z + \neg\,\bar{Z} \qquad (1)$$

Equation (1) will play an important role in the later course of the work.

After all steps, as explained above, have been carried out, a structure preserving model at the gate level results. Important is, that during all steps inconsistencies must not occur.
In Fig. 4, the NAND2 is now displayed in propositional logic at gate level. As described above, the PU and $\overline{PD}$ are summarized using the composition. In addition, a switch is built between $\overline{PD}$ and the composition. Its task is also to highlight the low-active input of the RS Buffer. It is important that the stars here in the continuation only serve as a "monitor" for checking. If the circuit is designed correctly, each star (*) can only supply a "0".

In the third step, the node X (pin X) is expressed in a function: The PU and $\overline{PD}$ is connected to the composition „+“. The operating voltage V_{DD} flows in series with the inputs in the PU. The mass runs serially to all inputs in the $\overline{PD}$.
The inputs in the PU are connected in parallel, while the inputs in the $\overline{PD}$ are connected in series. The switch between composition and pull-down "switches" the last part of the term in Equation (1). In summary, it should be emphasized that viewing at the gate level (in propositional logic) allows a more simplified view, which makes the derivation of the function at node X easier.

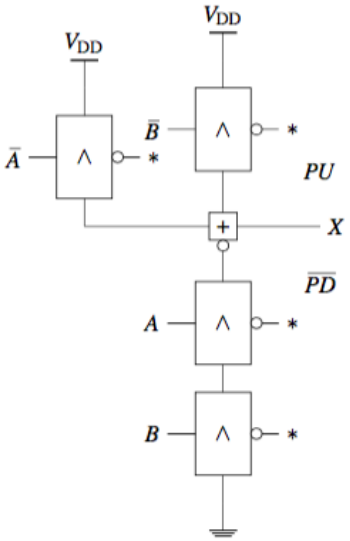

Figure 4: NAND2 at gate level

$$X = V_{\mathrm{DD}} \cdot (\bar{A} + \bar{B}) + \; \neg(\overline{\mathrm{GND}} \cdot B \cdot A) \qquad (2)$$

The above circuit at gate level shows the concreteness of the output circuit (Fig. 4). As described above, it was transferred from the transistor level to the gate level in a structure preserving manner. The functions derived at the transistor level and the functions derived at the gate level must be identical with respect to their partial order, this means the function of the circuit must not be changed by the transfer. Only then is the transfer a structurally-faithful one. The gate level can be viewed as a model view. It serves to increase clarity as well as contribute to an understanding of the circuit.

B. Transfer to a signal flow graph (SFG)

With the basic knowledge from chapter II, the function X is now transferred step by step into an SFG at Fig. 5. The operating voltage V_{DD} is catenated with the concatenated inputs $\bar{A}$ and $\bar{B}$, which means that these two primary inputs must be represented as edge weight. The second edge receives the catenated weight $B \cdot A$ and the mass represents the node. The switch, which must be installed here, is not to be forgotten. These two edges concatenate to the node X.

$$X = V_{\mathrm{DD}} \cdot (\bar{A} + \bar{B}) + \neg(\overline{\mathrm{GND}} \cdot B \cdot A)$$

It is important that the function X generated from the real structure is consistent (in the direction) with the functions derived at the gate level and the signal flow graph. The SFG allows a further comprehensible and simple visual consideration of the problem. The system is represented simply and visually by weighted, directed graphs. In the dual sense, edges are small processing units that process incoming signals (pins) in a certain form and send the result to all outgoing pins (signals).

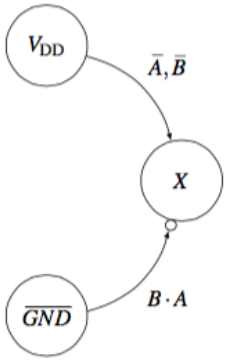

Figure 5: SFG of the NAND2

C. Deriving a signal flow plan (SFP)

As already explained, the node can be interpreted in the dual sense as a partition, a signal, and an edge over its weight as processing (operation) of the signal. Thus it generates a new signal. The states of the output circuit are to be found in the nodes. The edges are supplemented with their weights. If the NAND2 circuit is now considered more closely at gate level, the background knowledge of this work can be used to derive a signal flow plan. It is important to know that the function which is derived from the signal flow graph has to coincide with the function which is derived at the gate level. For only then the modeling has been done in a structured way and the relation to reality has not been lost.

For the derivation of the signal flow plan it is determined which signals are visualized in a node and which are represented by an edge. The edge is a directed one line, which connects two nodes and effects the processing of a signal via its weight. The mass as well as the operating voltage represent the nodes in the SFG. The primary inputs are shown as edges.

The signal flow plan (action plan) is used to determine the complexity of a system. The nodes (blocks) in a signal flow plan are small processing units (blocks) that receive incoming signals on edges (that are nodes in the corresponding SFG) in outgoing signals on edges (that are nodes in the corresponding SFG). By changing nodes in edges and edges in nodes results from one signal flow graph are a signal flow plan, and vice versa.

Fig. 6 shows the signal flow plan derived from the signal flow graph of the NAND2 circuit. The signals, in this case the nodes (edges) of the signal flow plan, are found on the nodes of the signal flow graph. The edges of the signal flow graph are found in the edges (blocks) of the signal flow plan.

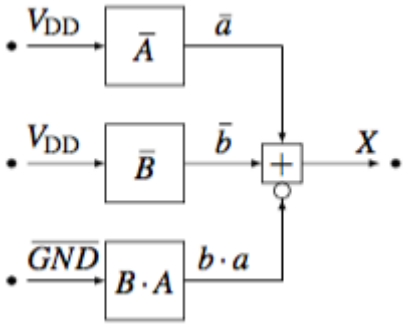

Figure 6: SFP of the NAND2

D. View in module view

Afterwards the gate level of node X has been transferred to a signal flow plan, the signal flow plan is displayed in a module view. This digital structural element serves for further simplified viewing of the real system.

This further simplifies understanding of the structure.

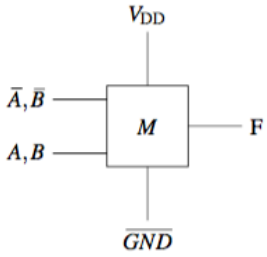

Figure 7: Module-view of the NAND2

The module-view for node X has:

- an input vector (inputs): $\bar{A}, \bar{B}, A, B$
- a programming vector (states): $\overline{\mathrm{GND}}$ and V_{DD}
- an output vector (output): X

E. Evaluation table of the NAND2

In the next step the output circuit is shown in the form of a truth table (value table, switching table). The function in Equation (3) can be expressed as follows:

$$\begin{aligned} X &= (X, \bar{X}) \\ &= (V_{\mathrm{DD}} \cdot (\bar{A} + \bar{B}), \neg(\overline{\mathrm{GND}} \cdot B \cdot A)) \\ &= PU + \neg\overline{PD} \end{aligned} \tag{3}$$

The relationship between PU and $\overline{PD}$ is the same as already shown in Equation (1). Not to forget, the composition „+“ between PU and $\overline{PD}$ is the RS Buffer. Tab. 1 is also required in order to be able to set up the evaluation table. We are in the propositional logic, but a 0 is written for the sake of simplicity instead of $\bar{1}$.

The operating voltage V_{DD} depends on the pull-up. It is important to know that if only the operating voltage supplies a „1“, a reliable switching process can be present in the PU. The pull-down depends on the mass $\overline{\mathrm{GND}}$.

	PU				$\overline{PD}$					
$\overline{GND}$	$\bar{A}$	$\bar{B}$	V_{DD}	V_{DD}	A	B	$\overline{GND}$	PU	$\overline{PD}$	X
*	1	1	1	*	0	0	1	1	*	1
*	1	0	1	*	0	1	1	1	*	1
*	0	1	1	*	1	0	1	1	*	1
*	0	0	1	*	1	1	1	*	1	0
*	1	1	0	*	0	0	0	*	*	[a]X
*	1	0	0	*	0	1	0	*	*	[a]X
*	0	1	0	*	1	0	0	*	*	[a]X
*	0	0	0	*	1	1	0	*	*	[a]X

Table 2: Total switching table of the NAND2

A switching operation in the $\overline{PD}$ can only take place when the mass is at „1“. Tab. 2 shows the total switching table of node X.

On closer examination of the table it can be seen that this table is composed of three divisional tables. The result of the PU and the $\overline{PD}$ and the node X represent a further table. The coding universe for the PU consists of the operating voltage V_{DD} and $\bar{A}$, $\bar{B}$. For the $\overline{PD}$, the coding universe consists of A, B and the mass $\overline{\mathrm{GND}}$. Thus the table for the PU and $\overline{PD}$ has a total of $2^3 = 8$ assignments.

The output X represents the respective state for the one-to-one coding of the table. The lower part of the table (last four lines) represents assignments which tend not to trigger any switching operations. The reason for this is that during the pull-up the operating voltage V_{DD} can only assume the value „1“. The mass of PU does not matter. For pull-down, the mass $\overline{\mathrm{GND}}$ must only have the value „1“, so that a switching operation is triggered. The operating voltage may be in the pull-down does not matter.

The results of the pull-up and the pull-down are calculated by the formula (2). In order to determine the resulting node X, Tab. 1 is to be considered. It represents the relationship between PU and $\overline{PD}$.

F . Partial evaluation table for NAND2

In the next step, the total switching table Tab. 2 is transferred to a partial switching table Tab. 3.

The correctness of the partial switching table is only given because a structurally correct transformation and modeling has taken place. This ensures that the circuit has been designed without errors and thus a partial switching table can be applied without errors. Tab. 3 shows the partial switching table of node X.

The operating voltage is unsignificant for the $\overline{PD}$, whereas in the PU it can trigger a switching operation only with a „1“. The mass is meaningless for the PU, whereas only one „1“ of the mass in the $\overline{PD}$ triggers a switching process. Thus the last four lines of the total switching table is lost.

For the safety of a defect-free structure it can be firmly assumed that in these assignments, the output X assumes the states in Tab. 3.

$\neg\bar{A}$ für PU A für $\overline{PD}$	$\neg\bar{B}$ für PU B für $\overline{PD}$	V_{DD}	$\overline{GND}$	PU	$\overline{PD}$	X
0	0	1	1	1	*	1
0	1	1	1	1	*	1
1	0	1	1	1	*	1
1	1	1	1	*	1	0

Table 3: Partial switching table of the NAND2

G . Resolution method for the NAND2

In the last step, the resolution method is repeated. By resolving the neighborhood relation is used, while the essentials, the satisfiability, remain unchanged. In order to be able to resolve the Equation (3), it is first embedded in a KV diagram (Fig. 8).

$$V_{\mathrm{DD}} \wedge \bar{B} \vee V_{\mathrm{DD}} \wedge \bar{A} \vee \neg(A \wedge B \wedge \overline{GND}) \vee V_{\mathrm{DD}} \wedge \overline{GND} \tag{4}$$

In order to make resolutions visible, the equations for the individual blocks are set up in the KV diagram. The green part of the Equation (4) is a redundant prime implicant in the KV diagram. This is now used in the resolution method for resolving, as shown in Figure 9.

The resolution method obeys the following neighborhood relations:

- The resolver $\{y, z\}$ (largest common cover) can be generated from the clauses $\{x, y\}$ and $\{\bar{x}, y, z\}$.
- From the clauses $\{x, y\}$ and $\{\bar{x}, \bar{y}\}$ the resolvers $\{x, \bar{x}\}$ and $\{y, \bar{y}\}$ can be generated (largest joint cover) [4].
- The resolver $\{\ \}$ can be generated from the clauses $\{x\}$ and $\{\bar{x}\}$

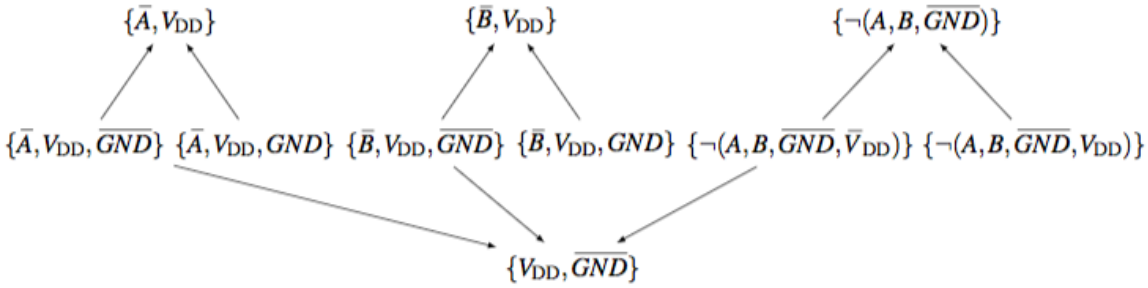

Figure 9: Resolution method for the NAND2

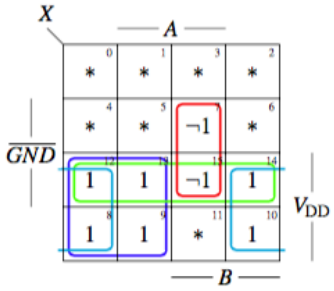

Figure 8: KV diagram for the NAND2

Figure 9 shows the resolution method for node X. The top line originates from Equation (2). For this purpose, the Equation (2) must be converted into a DNF system and subsequently represented as clauses, as can be seen in the following.

In order to be able to determine the six clauses, which are in the middle of the graph, the ones in the KV diagram (Fig. 8) have to be considered. When looking at the individual ones it can be seen that there is more information in them than when viewed as a block. By using this "additional" information it is possible to form resolutions. The green block in the KV diagram (Fig. 8) represents the lowest clause (prime implicant) in Fig. 9.

In order to be able to resolve these clauses, an assumption must be made. This assumption implies that the switch „¬“ has to be neglected, since only in this way can a resolvent be formed: Equation (2) is an equation concretized to category level.

In propositional logic, the switch can be ignored. Each axiom shows the necessity to accept certain fundamental statements as axioms of a theory without proving it [5]. They are given per se.

IV. CONCLUSION

The transferability of circuits into other possibilities of representation is a necessary property to ensure the functional safety of safety critical circuits. In this work an output circuit has been visualized in various display possibilities. Each type of presentation has its advantages and disadvantages. Moreover, each type of representation has a depth of accuracy, clarity and compactness. However, all of these representations are common in that their "structurally-faithful modeling and transition" must be preserved. This means that a formally derived function has to match consistently with the function derived from the respective representation type. Both functions must in no case have inconsistencies, because only then the fault-free function of the circuit can be maintained.

REFERENCES

[1] G. Uygur, F. R. Rasim, M. Özgül, S. M. Sattler: Strukturtreue Modellierung sicherheitskritischer Systeme, AmE 2015, 24-25 Februar, 2015, Dortmung, Deutschland
[2] R. Dorf, R. Bishof: Moderne Regelungssysteme. 2007
[3] G. Uygur, S. M. Sattler: Structure Preserving Modeling for Safety Critical Systems. Mixed-Signal Testing Workshop (IMSTW), 2015
[4] C. Meinel, M. Mundhenk: Math. Grundlagen der Inf.
[5] S. Goebbels, S. Ritter: Mathematik verstehen und anwenden. Springer Spektrum