Automatic Reliability Analysis of Electronic Designs using Fault Trees

Peter Liggesmeyer
Hasso-Plattner-Institute for
Software System Engineering
University of Potsdam
D-14471 POTSDAM

Oliver Mäckel
Siemens AG
Corporate Department Technology
CT PP 2E
D-81730 MUNICH

Abstract. In embedded systems development safety and reliability are important quality characteristics. Fault tree analysis is often used to determine these characteristics. Generalized fault trees improve the readability and thus prevent the insertion of bugs, e.g., during manual modification. Fault tree analyses are usually performed manually. Taking into account the high complexity of large, technical systems this is often not possible with respect to the existing time and budget constraints. Furthermore, completeness and precision of the results are neither guaranteed nor systematically provable. The automatic generation of generalized fault trees permits to save time and effort and it increases the quality of the results. Completeness, correctness, and consistency of the generated fault trees are guaranteed. We developed a fault tree generation tool for electronic circuits that generates fault trees based on the standardized EDIF 300 interchange format. This tool has already been used to analyze failure situations in industrial automation applications.

1 Motivation

The development of embedded systems faces an ever growing demand for quantified reliability and safety. Dependable results are required. Safety integrity of a system in quantifiable terms is more often required by customers, assessors, and licensing bodies.

Fault tree analysis (FTA), Markov models, and failure modes and effects analysis (FMEA) are commonly used methods to analyze potential hazards, and their potential influence on system reliability and safety [1, 11]. These methods are proven and accepted in reliability and safety engineering. By means of Boolean logic, fault trees represent the relationship between causes and undesired or hazardous events. Fault tree analysis can be started early in the design process. Typical questions are: Does the chosen design meet the reliability and safety requirements? What are the most critical components? Design options can be judged reasonably.

Fault trees are usually generated manually. Highly skilled, experienced engineers analyze the system based on existing documents that describe the system. Considerable knowledge, system insight and overview are necessary to consider many failure modes and their consequences at a time. This manual work is error-prone,

costly, and usually incomplete. Furthermore, there is usually no single person that is capable to analyze software as well as hardware.

Our approach is the automatic generation of fault trees.

It is required to answer the following questions:

- How reliable is a system including hardware and software with respect to particular failures, that may cause certain safety critical situations?
- What are appropriate means to reduce the risks, e.g., software test cases, hardware design modifications?

2 Fault Tree Analysis

2.1 Introduction

Fault tree analysis is a common technique for the analysis of particular failure situations of a system. A specific failure situation is the so-called top-event of the fault tree. Fault tree analysis is particularly used in complex systems to analyze safety critical effects of failures. It produces an easy-to-read model of the fault combinations of a system and their consequences.

The concept of fault tree analysis was developed by Bell Telephone Laboratories in 1961 [8], to improve the reliability of the Minuteman Launch Control System.

Fault tree analyses can be used for all kinds of systems, e.g., software [9], electrical and mechanical components [7]. The purpose of fault tree analysis is to analyze the dependencies between failures of components and subsystems and the fault tree's top-event. Fault tree analysis is a standardized technique [2, 5].

Qualitative and quantitative analyses can be carried out with fault tree analyses. The qualitative analysis aims at identifying all possible failure combinations that lead to a predefined undesired event. The quantitative analysis provides data about reliability characteristics, e.g., the frequency of the undesired event or the unavailability of the system.

2.2 Safety and Reliability Modeling of Embedded Systems with Fault Trees

Fault trees are a suitable technique for safety and reliability modeling of embedded systems. Accepted principles for handling complexity are available in fault trees. They are modular and they support abstraction. Furthermore, fault trees are a compassable technique if certain preconditions hold. This permits to perform separate fault tree analyses for system components, e.g., electronic components and software, and to combine the fault trees in order to get a valid analysis result for the system. Fault trees are a formal technique that supplies quantified results, and very large fault trees can be evaluated quantitatively.

They can be used to detect safety and reliability problems during the development of the system. In safety-critical applications fault tree analyses are often required by certification institutes usually based on guidelines and standards (see [6, 13]). Completeness and precision of manually generated fault trees are neither guaranteed

nor systematically provable. Furthermore, manual fault tree analysis requires substantial effort.

These disadvantages can be avoided by automation of fault tree analyses. During the development of a system, documents are produced that contain information about the system's behavior and structure. These are, e.g., specifications and architecture descriptions in early phases and, e.g., software source code and electronic circuit designs in later phases. These documents implicitly contain information about misbehaviors of the system as well as information about the behavior of the system in the normal mode of operation. They can be used to automate at least a part of the fault tree analysis.

3 Automatic Generation of Fault Trees

3.1 Motivation

On the one hand automation of safety and reliability analyses permits to save time and effort. On the other hand, it increases the quality of the results. This includes the generation of fault trees from already available documents (e.g., software source code, control logic or electronic circuit diagrams) and the systematic use of fault tree libraries. In the following, the generation of fault trees from electronic circuits is described. Fault tree generation techniques based on control logic descriptions or software source code are also available (see [10] and [12]).

In order to get a valid system fault tree, the generated component fault trees may be connected afterwards. This requires to generate fault trees with consistent interfaces, e.g., between hardware and software fault trees. It is important to select the correct top-event for the subsequent fault tree construction at an interface.

Fault tree analyses for electronic circuits are often used in safety and reliability analyses of safety-critical systems. Electronic circuits are usually developed using computer-aided design tools. For this reason, the corresponding data is available in electronically processible form. It is thus possible to use these data in order to automate the fault tree analyses. A manual fault tree analysis of complex electronic circuits is almost impossible. It is very difficult to keep a uniform perspective on failure mechanisms during a manual fault tree analysis, which is a substantial precondition for consistency.

In manual fault tree analysis for electronic circuits the fault tree's top-event usually is defined to be a faulty signal level at a certain output pin. In principle, also other effects can be analyzed. These include, e.g., timing problems. However, we do not take into account such problems in the following. Only virtually digital effects are allowed as top-events of the fault tree. Thus the top-events taken into account are faulty high and low levels at output pins. This limitation is usually accepted in a variety of application domains, e.g., industrial automation, and it simplifies the fault tree generation procedure.

3.2 Fault Tree Generation based on Hardware Component Fault Trees

Fault tree generation based on component fault trees uses the interconnection of the electronic components of a circuit as the basis for the generation of a fault tree with respect to a certain top-event. This is performed by combining basic fault trees for the electronic components. These components are elementary electronic components, blocks, that may contain components and nested blocks, or components describing logical functions, e.g., an AND-gate. There are two refinement hierarchies: The refinement of the cause-effect relation from effects to causes and the hierarchical refinement of blocks.

A procedure for generating fault trees for electrical circuits is described in [4]. This procedure does not use any hierarchical refinement however.

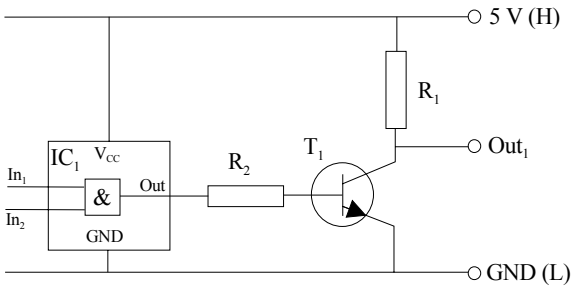


Fig. 1: Example circuit

The component fault trees are created by hand. They are stored in a library. It is important to provide component fault trees for all failure modes under consideration. We take into account faulty high and low levels of signals.

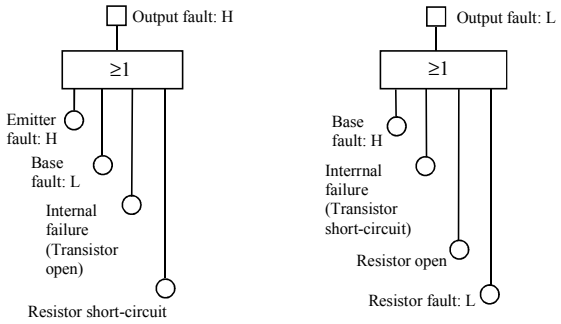


Fig. 2: Component fault trees for the common emitter-connection

Fig. 1 shows an example of a circuit to be analyzed [11]. Fig. 2 contains component fault trees for the two failure modes under consideration. These component fault trees describe the effects of T_1 and R_1 in the common emitter-connection.

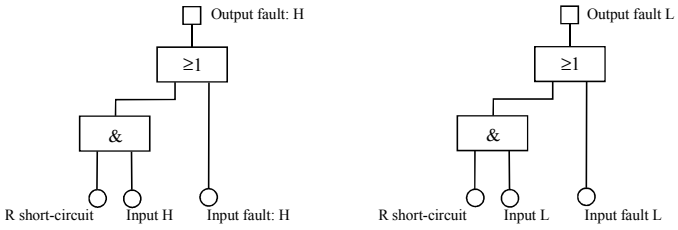


Fig. 3: Component fault trees for a resistor

Fig. 3 contains component fault trees for a resistor.

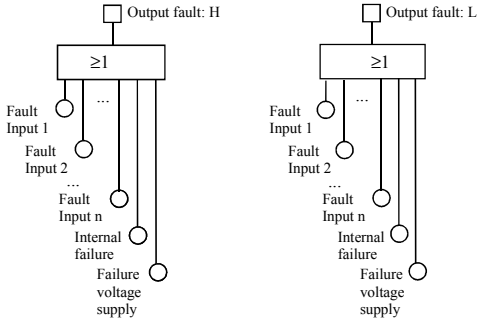


Fig. 4: Component fault trees for a block

The behavior of a block with n inputs and voltage supply can generally be described with the component fault trees shown in fig. 4.

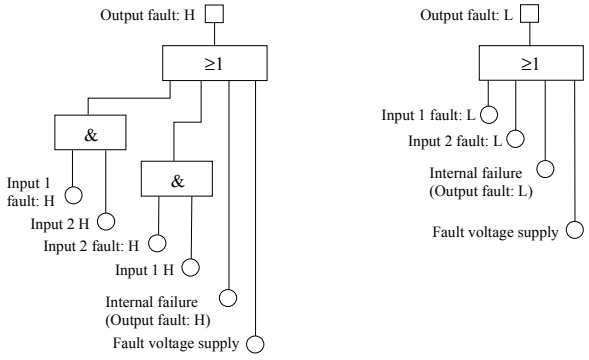


Fig. 5. Component fault tree for an AND-gate

The fault tree is generated by backward tracing starting at the selected top-event (Out_1 fault: H) and searching possible causes. During this process the appropriate component fault trees are joined. Since the block contains an AND-gate (fig. 1) the

appropriate component fault tree for an AND-gate (fig. 5) can be used. This produces the fault tree shown in fig 6.

3.3 Hierarchical design of electronic circuits

Complex electronic circuits are usually designed hierarchically. Blocks, e.g., amplifiers or voltage supplies are defined at the beginning. They are designed later in the development. The refinement process produces hierarchic levels, that should be used in order to generate appropriate abstraction levels within the corresponding fault tree. It is thus possible to analyze the electronic circuit at different levels of detail - a rough level for the first estimation and a detailed level in order to produce accurate analysis results. It is also possible to examine separate sections of the circuit in different degrees of detail. Critical parts of an electronic circuit may be analyzed more thoroughly. It is usually difficult to assign criticalities to sections of an electronic circuit. Rough estimations at block level may help to identify critical parts of the design, that may be analyzed in detail later. This approach focuses the analysis effort on critical parts and thus improves the applicability to large designs.

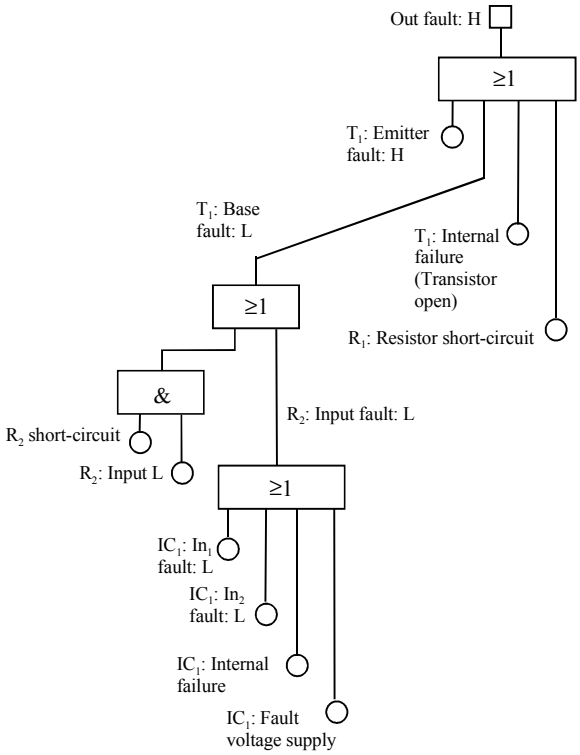
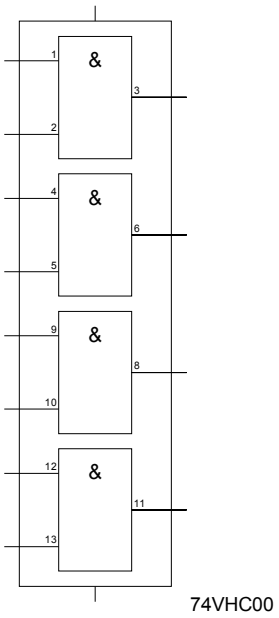


Fig. 6. Fault tree composed from component fault trees

3.4 Identification of integrated circuits

Electronic circuits often include components, that contain more than one basic function, e. g., four NAND-gates in the chip 74VHC00 (fig. 7), or two optocouplers in the chip ILD217 (fig. 8). These may be used at separate sections of an electronic circuit. Independent signals, that are associated with different functions on a single chip may become dependent (fig. 9). These common causes cannot be detached in the schematic diagram. For this reason components, that are integrated into a single chip, should be put into a block (fig. 10). This permits to assign common cause failures of all functions to the block, and to describe these with a fault tree module associated with the block (fig. 11 for the fault tree of the basic components and fig. 12 for the fault tree of the block at the higher abstraction level). This is an important improvement with respect to the analyses of safety-critical systems.



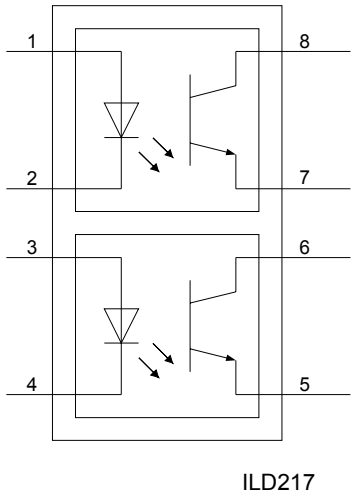


Fig. 7. components integrated into a single chip (four NAND-gates of IC 74VHC00)

Fig. 8. integrated components (two opto-coupler of ILD217)

Components, that are integrated into a single chip may belong to different logical blocks. If this situation occurs, the logical blocks are decomposed until it is possible to move the single chip components into an appropriate block.

3.5 Detection of standard patterns

Fundamental patterns occur very often, e.g., transistors in common base-, collector- and emitter-connection. Automatic recognition of such patterns permit to use appropriate fault tree modules in order to model their specific behavior. A new block is generated. Using fault tree modules for fundamental circuit patterns leads to more accurate results.

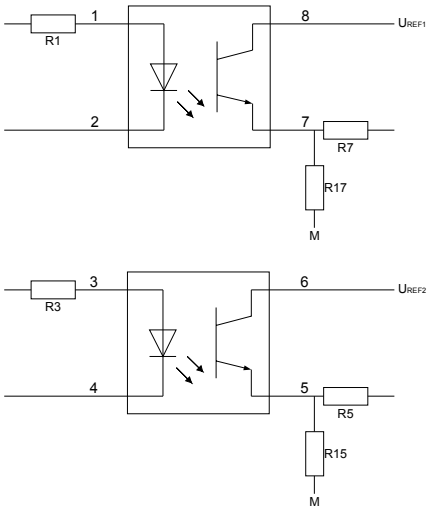


Fig. 9. (independent) input signals

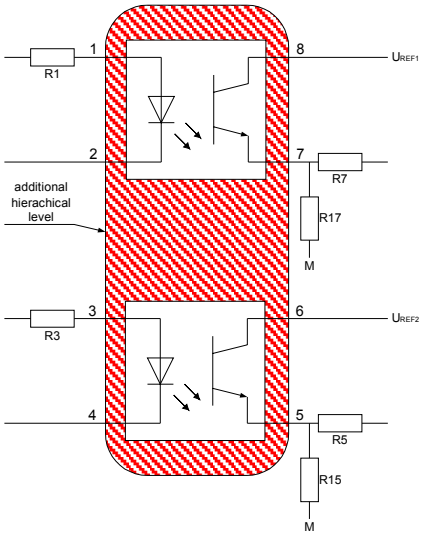


Fig. 10. two components on a single chip as a block

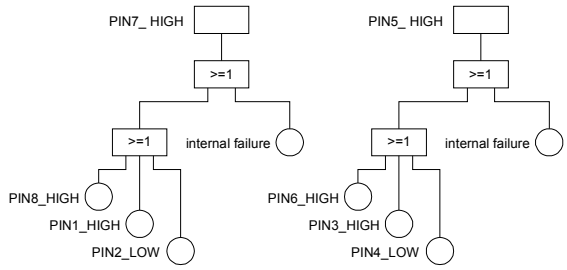


Fig. 11. fault trees for independent input signals (fig. 9)

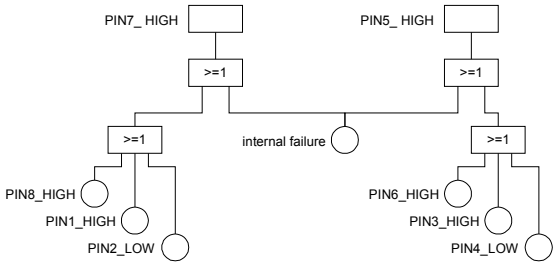


Fig. 12. fault tree for the correct not independent input signals (fig. 10)

3.6 Failure modes

The previous version of the tool FATEGO distinguished two basic failure modes. These were incorrect high signals or incorrect low signals, e. g., *stuck at high.* It turned out that these are considered sufficient in safety critical systems. However, two additional failure modes were added to the present version of the tool. These describe the incorrect transition of a signal from *High to Low* or from *Low to High*.

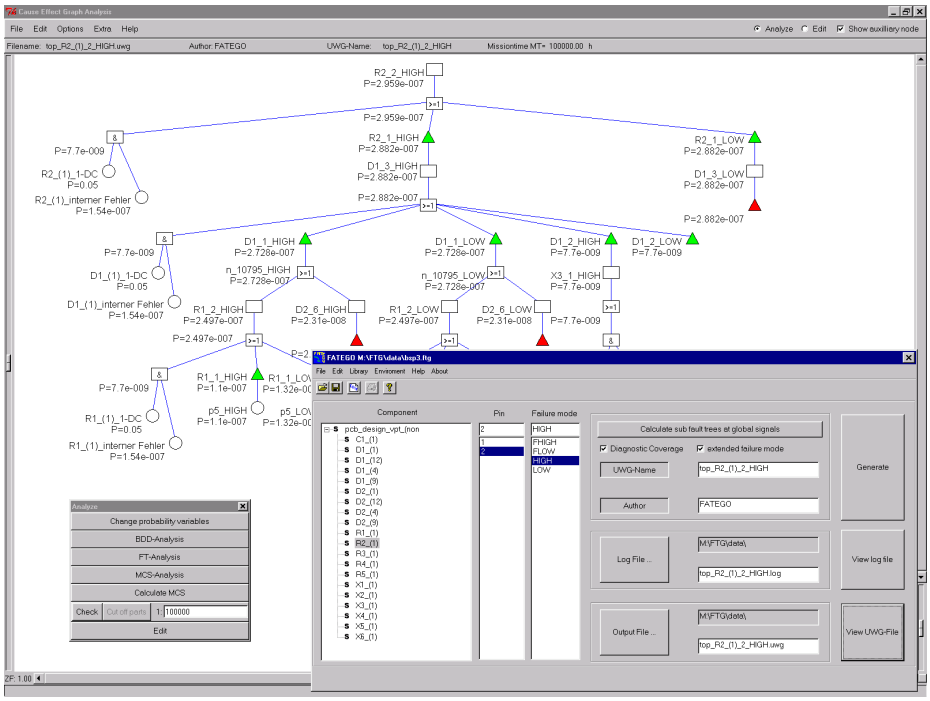


Fig. 13. The User-Interface of FATEGO

4 Summary

Fault tree analysis is an accepted method. Automatic construction of fault trees permits to analyze complex systems. It saves much effort and time and improves the quality of the results.

The tool FATEGO (fig. 13) was developed in the corporate technology department of Siemens. It automatically generates fault trees from electronic circuit data. FATEGO reads data in the EDIF 300 interchange format [3]. It permits the definition of the fault tree's top-event, and generates the appropriate fault tree. FATEGO also takes into account the influence of diagnostic routines on failure identification.

Acknowledgments

The authors wish to express their gratitude to Mr. M. Tschuwarditsch and Mr. B. Opgenoorth at Siemens Automation & Drives, where FATEGO is applied within the certification of fail-safe modules of the S7-F 400 programmable logic controller.

References

1. Birolini A., Qualität und Zuverlässigkeit technischer Systeme: Theorie, Praxis, Management, Berlin: Springer 1991
2. DIN 25424, *Fehlerbaumanalyse*: Teil 1, Methode und Bildzeichen; Teil 2, Handrechenverfahren, Berlin: Beuth Verlag GmbH.
3. FDIS IEC 61690*, Electronic Design Interchange Format (EDIF)*, International Electrotechnical Commission.
4. Fussell J.B., Computer aided fault tree construction for electrical systems, in: Barlow R.E., Fussell J.B. (Ed.), Reliability and Fault Tree Analysis, Philadelphia: Society for industrial and applied mathematics 1975
5. IEC 61025, *Fault Tree Analysis (FTA)*, International Electrotechnical Commission.
6. IEC 61508, Functional safety of electrical/electronical/programmable electronic safety/related systems, International Electrotechnical Commission.
7. Kececiogliu, Dimitri, *Reliability Engineering Handbook* Part 1 and Part 2, Englewood Cliffs: Prentrice Hall 1991.
8. Lees, Frank P.: *Loss Prevention in the Process Industries*, Vol. 1 and Vol. 2, London, Boston: Butterworths 1983.
9. Leveson, Nancy*: Safeware: system safety and computers*, Addison-Wesley, 1995.
10. Liggesmeyer, P.; Rothfelder, M.: *Improving System Reliability with Automatic Fault Tree Generation*, in: Proceedings 28th Annual Fault Tolerant Computing Symposium, Munich, June 1998, pp. 90-99.
11. Liggesmeyer, P.*: Qualitätssicherung softwareintensiver technischer Systeme*, Heidelberg: Spektrum-Verlag 2000
12. Liggesmeyer, P.: Quantified Software Risk Assessment and Test Support by Automated Fault Tree Generation, in: Proceedings EuroSTAR ´99, Barcelona, November 1999
13. prEN 50129*, Railway Applications: Safety related electronic systems for signalling*, European Committee for Electrotechnical Standardization, Bruessel.