



KfK 3190
EUR 7056e
August 1981

The Boolean Algebra with Restricted Variables as a Tool for Fault Tree Modularization

L. Caldarola, A. Wickenhäuser
Institut für Reaktorentwicklung
Projekt Schneller Brüter

Kernforschungszentrum Karlsruhe

KERNFORSCHUNGSZENTRUM KARLSRUHE

Institut für Reaktorentwicklung

Projekt Schneller Brüter

KfK 3190

EUR 7056e

THE BOOLEAN ALGEBRA WITH RESTRICTED VARIABLES

AS A TOOL FOR FAULT TREE MODULARIZATION

L. Caldarola

A. Wickenhäuser

Kernforschungszentrum Karlsruhe GmbH, Karlsruhe

Als Manuskript vervielfältigt
Für diesen Bericht behalten wir uns alle Rechte vor

Kernforschungszentrum Karlsruhe GmbH
ISSN 0303-4003

The boolean algebra with restricted variables as a tool for fault tree modularization.

Abstract

The number of minimal cut sets (m.c.s.) of very complex and highly interconnected fault trees can become extremely large (e.g. more than 10^7). In this case the usual analytical approach of dissecting the fault tree TOP variable into m.c.s. is not only computationally prohibitively expensive, but also meaningless because it does not offer any synthetic overview of system behavior. The method proposed in this paper overcomes the deficiencies of the analytical method. It is shown that, by applying boolean algebra with restricted variables (b.a.w.r.v.), the concept of fault tree modularization can be straightforwardly extended from a single gate to a set of gates. Thus, large fault trees are divided into smaller fault trees (modules), which are connected to each other according to a simple scheme. This scheme is represented by a block diagram in which each block is a module. The modules are analyzed separately by the m.c.s. method, and the results are combined according to the block diagram connections to calculate the occurrence probability of the TOP event. The method allows the calculation of very large fault trees in a short time and offers a synthetic overview of system behavior through the block diagram. Numerical examples are also included. Calculations have been carried out by using the computer code MUSTAMO, which is based on the theory developed in this paper.

Boolesche Algebra mit beschränkten Variablen als Mittel zur Fehlerbaum-Modularisierung

Kurzfassung

Die Anzahl der Minimalschnitte sehr komplexer und stark vermaschter Fehlerbäume kann extrem groß werden (beispielsweise mehr als 10^7). Für diesen Fall ist das übliche analytische Verfahren der Zerlegung der TOP-Variablen des Fehlerbaums in Minimalschnitte sowohl rechen-technisch prohibitiv teuer, als auch sinnlos, weil es keinen Überblick über das Systemverhalten liefert. Mit der hier vorgeschlagenen Methode werden diese Mängel der analytischen Methode überwunden. Es wird gezeigt, daß durch Einsatz der Booleschen Algebra mit beschränkten Variablen das Konzept der Fehlerbaum-Modularisierung von einem einzelnen Gatter ohne weiteres auf eine Menge von Gattern erweitert werden kann. Große Fehlerbäume werden dadurch in kleinere Fehlerbäume (Module) aufgeteilt, die nach einem einfachen Schema miteinander verknüpft sind. Dieses Schema wird durch ein Blockdiagramm dargestellt, in dem jeder Block ein Modul ist. Die Module werden nach der Methode der Minimalschnitte einzeln analysiert, und die Ergebnisse werden aufgrund der Verknüpfungen des Blockdiagramms zusammengefaßt, um die Eintrittswahrscheinlichkeit des TOP-Ereignisses zu berechnen. Die Methode erlaubt die Auswertung von sehr großen Fehlerbäumen in kurzer Zeit und liefert über das Blockdiagramm einen Überblick über das Systemverhalten. Die Methode wird auch an Hand von numerischen Beispielen erläutert. Die Berechnungen wurden mit Hilfe des Rechenprogrammes MUSTAMO durchgeführt, das auf der in diesem Bericht beschriebenen Theorie basiert.

Preface

The "ad hoc" european expert working group in reliability during their 11th meeting held at Ispra (Italy) on 15th and 16th October 1980 recommended to investigate the use of boolean algebra with restricted variables in future computer programs for fault tree analysis.

Following this recommendation a meeting was held at Karlsruhe on 1st April 1981. The participants were Messers A. Cross and R. Matthews from the Safety and Reliability Directorate, UKAEA (Warrington, Great Britain), Mr. A. Amendola from the European Joint Research Center of Ispra (Italy), Mr. C.A. Clarotti from the Comitato Ricerche Nucleari (Roma, Italy) and Messers L. Caldarola, A. Wickenhäuser, H. Knuth and H. Schnauder from Kernforschungszentrum Karlsruhe (Federal Republic of Germany).

At the end of the meeting the participants issued the following statement:

"In order to extend the current techniques of logical analysis to give a more complete system representation, it seems advisable to use boolean algebra with restricted variables (b.a.w.r.v.) in future computer programs for fault tree analysis.

The advantages of b.a.w.r.v. over the traditional boolean algebra techniques are as follows:

1. It handles components with more than two states.
2. It extends the concept of modules from that of a single gate to that of a set of gates. This has the potential for handling fault trees with large numbers of minimal cut sets. The extent of this potential should be further investigated.
3. Because of the modularisation of the fault tree, the logical information is presented in a more compact, understandable form. This is of particular importance when the number of minimal cut sets is very large.

With reference to points 1 and 2 above, b.a.w.r.v. is the common language which can be used, at the boolean level, in both fault tree analysis and state analysis, thus allowing the combination of the two techniques in a more manageable way. In addition there are no basic problems integrating b.a.w.r.v. with computer aided fault tree construction, common mode analysis and quantitative analysis (analytical and/or simulation methods). The development of these aspects should also be explored.

The above points are valid in all applications of fault tree analysis such as risk analysis, design optimisation, on line diagnostics etcetera."

During the meeting the authors showed the applications of b.a.w.r.v. for fault tree modularization. This paper is the authors presentation on the subject at the meeting.

Contents

Introduction	1
1. Generalities on the boolean algebra with restricted variables	6
2. Description of the method	11
3. An example	29
4. A second example	46
5. Conclusions	59
6. References	61

INTRODUCTION

The evaluation of the occurrence probability of the top event of a fault tree can be carried out by means of simulation methods (Monte Carlo-type methods) or by means of analytical methods.

Numerical simulation allows reliability information to be obtained for systems of almost any degree of complexity. While this method provides estimates it does not yield parametric relations. In addition, since the failure probability of a system is usually very low, precise results can be achieved only at the expense of very long computational times.

Analytical methods give more insight and understanding because explicit relationships are obtainable. The results are also more precise because these methods usually give the exact solution of the problem.

In 1970 Vesely /1/ gave the foundations of the analytical method for fault tree analysis. Vesely's theory was improved by the present author. A computer program for fault tree analysis was developed based on this theory / 2; 3 /. This computer program proved to be the best analytical program for fault tree analysis in the Federal Republic of Germany / 4 /.

Vesely's method can be applied only to coherent systems with binary (two states) components. Another important limitation of the method is that the boolean function which describes the TOP variable of the fault tree must not contain negated variables. Finally the theory does not give any indication on how to handle statistically dependent components.

Since there are components (e.g. a switch) which have more than two states, a theory was developed by the author in 1977 /5/ to handle systems with multistate components. Here the basic idea was introduced to associate the primary variables with the states of the primary components instead of with the primary components. In addition the basic boolean algorithms were described. In 1978 the author /6/ showed that the technique of multistate super-components can be used to remove statistical dependencies from a fault tree, by introducing supercomponents defined "ad hoc" with more than two states.

An interesting feature of the method proposed in /5/ and /6/ is that the boolean function which describes the TOP variable of the fault tree

does not necessarily need to be coherent. In addition boolean functions containing negated variables can be treated.

A formalization of the theory by means of the so called "boolean algebra with restricted variables" has been developed by the author in /7/, and /8/.

It is shown in /8/ that the boolean algebra with restricted variables (b.a.w.r.v.) is the common language which can be used in both fault tree analysis and state analysis, thus allowing the combination of the two techniques in a more manageable way. This feature is of particular value for handling statistical dependencies in fault trees. The importance of the b.a.w.r.v. was recognised in /9/, where it was said that the b.a.w.r.v. "will play the role that Vesely's paper played ten years ago" /9/.

In /10/ the coherent systems were defined for the more general case in which multistate (two or more than two states) primary components are contained in a system. Here the concept of "associated coherent function" of a given boolean function is introduced.

Based on the theory given in /7/; /8/ and /10/ the computer program MUSTAFA was developed to analyze fault trees of coherent and non coherent systems containing statistically independent as well as dependent components with two or more than two states.

In this paper another important application of the b.a.w.r.v. will be examined, namely fault tree modularization.

In the case of very large systems with many interconnections the total number of minimal cut sets (m.c.s.) of a fault tree may become extremely large (e.g. more than 10^7). In this case the usual m.c.s. approach is not only computationally impossible but also meaningless because it does not offer any synthetic overview of system behaviour. This deficiency was also pointed out in the german reactor risk study /11/.

For this reason attempts have been made /12; 13/ to modularize large fault trees. In order to briefly illustrate the previously available methods, let us consider the fault tree 1 of Fig. 1. The meaning of the symbols used in Fig. 1 are explained in Table 1. The primary components underneath gate G09 are different from the primary components located underneath the rest of the fault tree. The same holds for gate G10. One can therefore calculate the fault tree 1 by treating the gates G09 and G10 as primary variables.

Fault tree 1 can be dissected into three smaller fault trees, namely G09, G10 and the main fault tree in which G09 and G10 enter as primary variables (modules). The three resulting fault trees can be analyzed separately one after the other, and the results are properly combined to calculate fault tree 1.






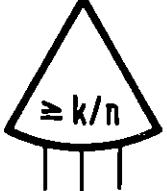
It is important to point out that the theory available from the literature allows the modularization based on single gates only.

With reference to fault tree 1 it is not possible to handle the gate G05 as a module because some of the primary components underneath G05 (F and H) are also underneath G06. The same holds for G06. Consider now the gate G05 and G06 together (as a set). The primary components underneath the set of gates G05 and G06 (E; F; H and K) are different from the primary components located underneath the rest of the fault tree. One could therefore try to modularize fault tree 1 by considering the gates G05 and G06 not individually but together as a set.

The theory presented in this paper allows the extension of the concept of modularization from that based on a single gate to that based on a set of gates.

Table 1

List of Symbols used in the Fault Trees.

Symbol	Meaning
	Primary Variable
	Non Primary Variable
	OR Gate
	AND Gate
	NOT Gate
	MAJORITY Gate (at least k out of n)

Note : A marked point at the input of a gate means that the variable is negated.
(see NOT Gate).

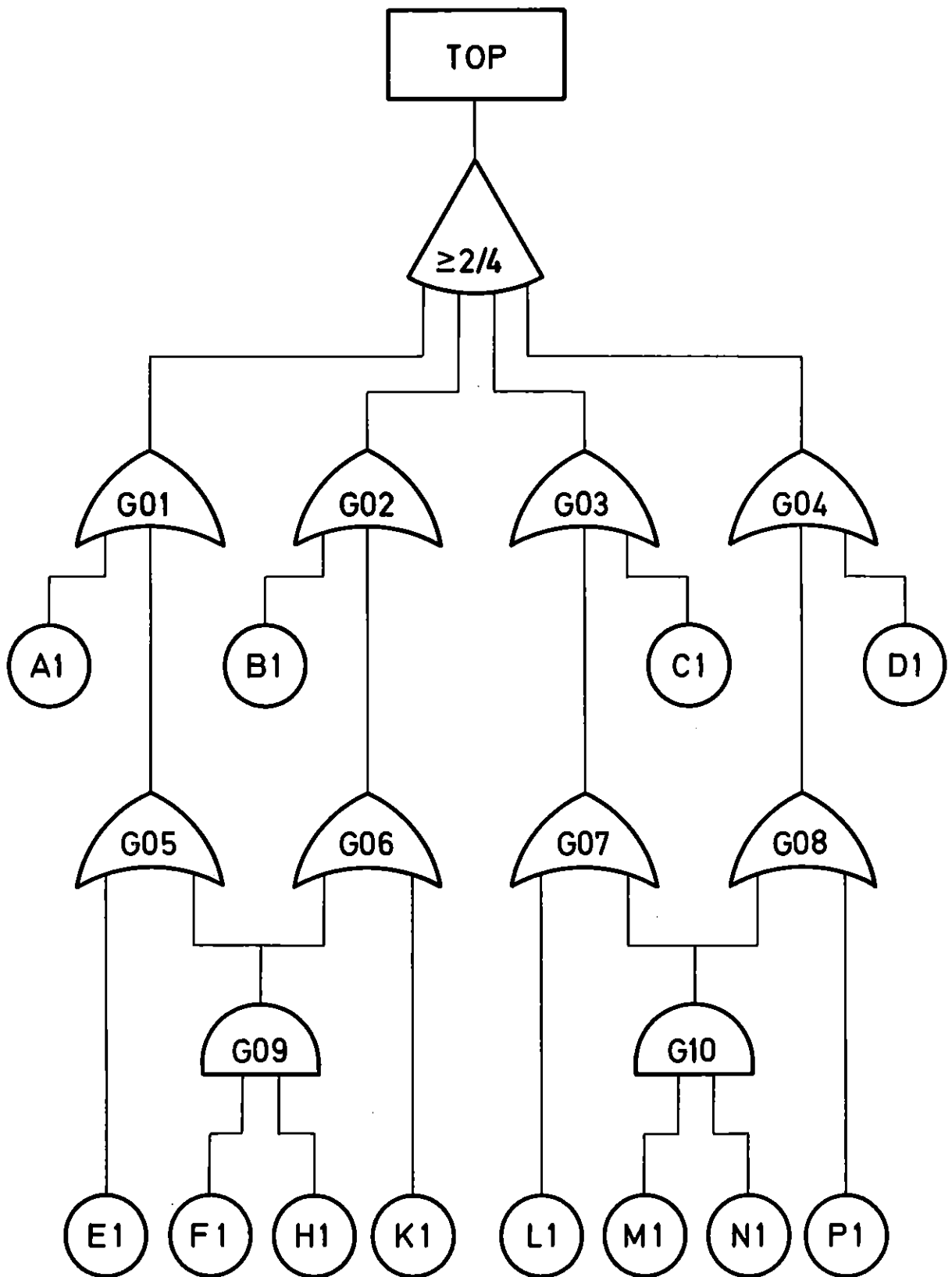


Fig. 1: Fault Tree 1

1. Generalities on the boolean algebra with restricted variables

According to what is said in the introduction, the basic idea of the boolean algebra with restricted variables is that of associating the primary variables (literals) with the states of the primary components instead of with the primary components.

A primary component will be indicated by a lower case letter. For instance a, b, c are components.

A state of a primary component will be indicated by the same notation as the primary component to which it belongs followed by a positive integer number (a0, a1, a2, etc.). In general we shall have aq with q= 0; 1; 2....; na - 1 where na is the total number of states belonging to primary component a.

We now associate with each state aq a boolean variable Aq (literal) which takes the value 1 if primary component "a" occupies state aq and the value 0 if "a" does not occupy aq.

The event

$$\{Aq = 1\} \longleftrightarrow aq \quad (1-1)$$

indicates that primary components "a" occupies state aq.

Conversely, the event

$$\{Aq = 0\} \longleftrightarrow \bar{a}q = \bigcup_{k=0}^{na-1} ak \quad (k \neq q) \quad (1-2)$$

indicates that primary component "a" does not occupy state aq and therefore occupies one of its other possible states (i.e. the union of all remaining states).

Note the one to one equivalence between state aq (small a) and boolean variable Aq (capital A) associated with it. We have

$$aq \longleftrightarrow \{Aq = 1\} \quad (1-3)$$

and

$$\bar{a}q \longleftrightarrow \{\bar{A}q = 1\} \longleftrightarrow \{Aq = 0\} \quad (1-4)$$

Since a primary component must occupy one of its states and can occupy only one state at a time, the variables A_q must obviously satisfy the following two types of restrictions.

Restriction Type 1 The disjunction of all binary variables associated with the same primary component is always equal to 1.

$$\bigvee_{q=0}^{na-1} A_q = 1 \quad (1-5)$$

Restrictions Type 2 The conjunction of two different binary variables associated with the same primary component is always equal to 0.

$$A_q \wedge A_k = 0 \quad (q \neq k) \quad q; k = 0; 1; 2; \dots; na-1 \quad (1-6)$$

Note that there is only one restriction type 1 and $na \cdot (na - 1) / 2$ restrictions type 2.

The complement rule is also important.

Complement rule A negated (complemented) literal is equal to the disjunction of all remaining literals belonging to the same primary component, that is

$$\overline{A_q} = \bigvee_{k=0}^{na-1} A_k \quad (k \neq q) \quad (1-7)$$

Note that the complement rule can be derived from the restrictions and viceversa /7/.

It has been shown in /7/8/ and /10/ that the boolean algebra with restricted variables allows one to operate on boolean variables in a way similar to the traditional boolean algebra, but with the additional rules given by Eqs. 1-5 to 1-7. These additional rules apply only among the primary variables (literals) which belong to the same primary component. There are no additional rules among primary variables which do not belong to the same primary component.

The following definitions have already been introduced in /7/, /8/ and /10/ and will be used throughout this paper.

Definitions

1. A monomial is a conjunction of literals.

Note that by definition a monomial does not contain negated literals.

2. A zero monomial is a monomial which is always equal to zero.
A monomial is identical with zero if it contains at least two different literals of the same primary component (restrictions type 2).

3. A literal is said to be obligatory if its deletion in a given monomial alters the truth table of the monomial.
Repeated literals are not obligatory.

$$B_i \wedge B_i = B_i \quad (1-8)$$

4. An irredundant monomial is a non zero monomial which contains only obligatory literals.

5. A complete monomial (minterm) is an irredundant monomial which has a number of literals equal to the number of primary components present in the system.

6. If two irredundant monomials are such that the first (say X) contains all literals of the second one (say Y), the first monomial implies the second one. The first monomial (X) is called subsuming monomial and the second one (Y) subsumed monomial.

7. A disjunctive form of a boolean function is any disjunction of monomials which is equivalent to the function.

8. The disjunctive canonical form of a boolean function is that disjunctive form of the function in which every monomial is complete.

9. A monomial belonging to a disjunctive form of a boolean function is said to be obligatory if its deletion in the disjunctive form alters the truth table of the function.

A monomial is not obligatory if (1) it is a zero monomial, or (2) it subsumes another monomial of the disjunctive form, or (3) it implies the disjunction of two or more other monomials of the disjunctive form.

10. A disjunctive form of a boolean function is called a normal disjunctive form if (1) all monomials are irredundant and (2) no subsuming monomial is contained in it.

11. An irredundant disjunctive form of a boolean function is a normal disjunctive form of the function which ceases to be a disjunctive form of the function if one of its monomials is removed (deleted).

The monomials of an irredundant disjunctive form are all obligatory.

12. An irredundant monomial (say X) is said to be a prime monomial (or prime implicant) of a boolean function (say TOP) if (1) X implies the TOP and (2) every subsumed monomial Y obtained from X by replacing one of its literals with 1 does not imply the TOP.

Prime monomials are also currently called minimal cut sets in the literature.

13. A base of a boolean function is any disjunction of prime monomials which is equivalent to the function.

14. The complete base of a boolean function is the disjunction of all its prime monomials.

15. An irredundant base of a boolean function is a base which ceases to be a base if one of its prime monomials is removed (deleted).

The prime monomials of an irredundant base are all obligatory.

16. The three simplification rules, which allow one to get a normal disjunctive form from a disjunctive form are the following:

1. Delete the repeated literals of a monomial (idempower law).
2. Delete zero monomials (exclusion law).
3. Delete subsuming monomials (absorption law).

17. We call intact literal (or intact primary variable) of a primary component that literal which is associated with the intact state of the primary component.

For convection the literal with the index "0" is the intact literal. For instance AO, BO, CO are the intact literals respectively of the primary components A, B, C.

18. A boolean function is said to be irredundant if it has only one base which is at the same time complete and irredundant.

19. A boolean function is said to be coherent if at least one literal (the intact literal) of each primary component does not appear in the complete base of the function.

It is important to point out /10/ that a coherent function is irredundant but that an irredundant function is not necessarily coherent.

20. The associated coherent function of a given boolean function TOP is that function Φ which is generated from any normal disjunctive form of the TOP by replacing all intact literals by 1.

Due to the way in which the function Φ is generated, one can easily verify that TOP implies Φ .

If a boolean function is coherent, its associated coherent function is identical with the boolean function. The reverse is also true.

The following rules on coherent boolean functions are important /10/.

Rule 1

If a normal disjunctive form of a boolean function is such that at least one literal of each primary component does not appear in it, the function is coherent, and the normal disjunctive form is the only base of the function.

Rule 2

If a boolean function is coherent, its base can be calculated from any of its normal disjunctive forms by replacing all intact literals by 1 and by applying the absorption law among the monomials.

2. Description of the method

According to /8/ the occurrence probability "P" of the event that a stochastic boolean variable takes the value 1 is equal to the expected value "E" of the stochastic boolean variable, that is

$$P \left\{ \text{TOP} = 1 \right\} = E \left\{ \text{TOP} \right\}$$

For more details about the above equation see chapter 2 of /8/.

In the following we shall speak of the expected value of a stochastic boolean variable and we shall mean by that the occurrence probability of the associated event.

In the following we shall use the symbols + and • to indicate the operations respectively of disjunctions (\vee) and conjunctions (\wedge) among boolean variables.

Note that the symbols + and • indicate the arithmetical operations respectively of addition and multiplication when they are used in conjunction with expected values of boolean variables.

The method will be described step by step by applying it to a fault tree.

Let us consider the already mentioned fault tree 1 (Fig. 1). The primary components of the fault tree are A, B, C, D, E, F, H, K, L, M, N and P.

The primary components are all binary, i.e. they have two variables, one associated with the failed state (failed variable) and one associated with the intact state (intact variable). So in the case of the primary component A we have the primary variable A1 which is associated with the failed state and the primary variable A0 which is associated with the intact state. The two primary variables A0 and A1 are restricted variables. We have:

$$A0 \cdot A1 = 0$$

$$A0 + A1 = 1$$

$$\overline{A0} = A1$$

$$\overline{A1} = A0$$

The fault tree of Fig. 1 contains only failed variables, namely A1, B1, C1, D1, E1, F1, H1, K1, L1, M1, N1 and P1. Since the fault tree does not contain any intact variable, the boolean function TOP is coherent.

The fault tree 1 is very simple and could be solved without any difficulty by applying the usual analytical methods.

However, due to its simplicity, fault tree 1 is suitable for introducing the method, because all operations can be carried out by hand.

We introduce first some terminology of fault tree analysis. It is a common practice in fault tree analysis to classify the variables (vertices) into two categories: primary variables and non primary variables. The non primary variables will be called gates here.

Definition 21

The input variables of a gate are called predecessors of the gate.

Definition 22

A successor of a variable is any gate to which the variable is an input.

Definition 23

A route in an ordered sequence of variables which (1) starts with a primary variable, (2) ends with the TOP variable and (3) in which each variable is a successor of the preceding variable and a predecessor of the following variable.

With reference to fault tree 1 of Fig. 1, observe for instance that each one of the two sequences

F1 - G09 - G05 - G01 - TOP

M1 - G10 - G08 - G04 - TOP

is a route of the fault tree..

Definition 24

A bundle is a set of routes.

For example the two routes listed above constitute a bundle.

Definition 25

The territory of a given bundle is the set of all primary components associated with the primary variables contained in the routes of the bundle.

Referring to the bundle composed of the two routes shown above, notice that the primary variables belonging to the routes of the bundle are F1 and M1. The primary components associated with these primary variables are therefore F and M. The set $\{ F;M \}$ constitutes the territory of the bundle.

Select now an arbitrary set (group) of gates of fault tree 1, for example G05 and G06. Consider the complete set of routes which contain either G05 or G06 or both.

They are:

E1 - G05 - G01 - TOP
F1 - G09 - G05 - G01 - TOP
H1 - G09 - G05 - G01 - TOP
F1 - G09 - G06 - G02 - TOP
H1 - G09 - G06 - G02 - TOP
K1 - G06 - G02 - TOP

Each of the above six routes is said to be internal with respect to the group of gates G05 and G06. The bundle made of these six routes is called the internal bundle and its territory the internal territory of the group of gates G05 and G06. By inspection, the internal territory is, in this case, the set $\{ E;F;H;K \}$.

Consider now all the remaining routes of fault tree 1. They are:

A1 - G01 - TOP
B1 - G02 - TOP
L1 - G07 - G03 - TOP
C1 - G03 - TOP
M1 - G10 - G07 - G03 - TOP
N1 - G10 - G07 - G03 - TOP
M1 - G10 - G08 - G04 - TOP
N1 - G10 - G08 - G04 - TOP
P1 - G08 - G04 - TOP
D1 - G04 - TOP

Notice that none of the above ten routes contains G05 and/or G06. These routes are said to be external with respect to the group of gates G05 and G06. The bundle made of the above ten routes is called the external bundle and its territory the external territory of the group of gates G05 and G06. By inspection, the external territory is, in this example, the set $\{A;B;L;C;M;N;P;D\}$.

In summary, given an arbitrary group of gates, each route of the fault tree is either internal or external with respect to the selected gates. The internal routes are those which contain at least one gate of the group, while the external routes do not contain any gate of the group. The set of all internal routes constitutes the internal bundle and similarly the set of all external routes constitutes the external bundle. The set of all primary components associated with the primary variables contained in the internal bundle constitutes the internal territory of the selected group of gates. Likewise, the set of all primary components associated with the primary variables contained in the external bundle constitutes the external territory.

The above definitions allow one to identify, for any arbitrary group of gates, the associated internal and external territories.

In the example the following table can be finally set up:

Selected Group of Gates	G05; G06
Associated Internal Territory	E; F; H; K
Associated External Territory	A; B; C; D; L; M; N; P

Notice that the two territories have no primary component in common. In this case we say that the two territories are disjoint.

Definition 26

Two territories are said to be disjoint if they have no primary component in common.

Since the internal and external territories of the selected group of gates (G05 and G06) are disjoint, we shall soon see that the group of gates can be analyzed separately. For this reason we say that the group of gates is logically independent.

Definition 27

A group of gates is said to be logically independent if its internal and external territories are disjoint.

Consider the internal bundle of G05 and G06. Notice that no route of the internal bundle contains both G05 and G06. We say that the group of gates G05 and G06 is linear.

Definition 28

A group of gates is said to be linear if each route of its internal bundle contains one and only one gate of the group.

If a group of gates is linear and logically independent, it is possible to build with them a supercomponent, whose variables can be treated as primary variables of the fault tree. Each variable of the supercomponent can be considered in turn as the TOP variable of a fault tree which can be analyzed separately from the main fault tree as well as from the fault trees of the other variables of the supercomponent. We shall illustrate this break down procedure by applying it to the group of gates G05 and G06.

We consider the complements of G05 and G06, namely $\overline{G05}$ and $\overline{G06}$. The following conjunctions can be constructed with the four variables G05; G06; $\overline{G05}$ and $\overline{G06}$.

$$Q1 = G05 \cdot \overline{G06} \quad (2-1)$$

$$Q2 = G05 \cdot G06 \quad (2-2)$$

$$Q3 = \overline{G05} \cdot G06 \quad (2-3)$$

$$Q0 = \overline{G05} \cdot \overline{G06} \quad (2-4)$$

The four variables Q0; Q1; Q2 and Q3 can be regarded as the variables of a component (supercomponent Q) because they satisfy the appropriate restrictions. In fact starting from the equations 2-1 to 2-4 it is easy to verify that

$$Q0 + Q1 + Q2 + Q3 = 1 \quad \text{Restriction 1st Type}$$

$$\left. \begin{array}{l} Q1 \cdot Q2 = 0 \\ Q1 \cdot Q3 = 0 \\ Q1 \cdot Q0 = 0 \\ Q2 \cdot Q3 = 0 \\ Q2 \cdot Q0 = 0 \\ Q3 \cdot Q0 = 0 \end{array} \right\} \text{Restrictions 2nd Type}$$

The equations 2-1 to 2-3 can be solved with respect to the variables G05 and G06 which are present in the fault tree.

We get

$$\begin{array}{ll} \text{and} & G05 = Q1 + Q2 \quad (2-5) \\ & G06 = Q2 + Q3 \quad (2-6) \end{array}$$

The Eqs. 2-1 to 2-3 and 2-5 to 2-6 can be used to "cut" the original fault tree into four fault trees. This is diagrammatically shown in Fig. 2. The equations 2-5 and 2-6 are used in the main fault tree (the upper part) in which the variables Q1, Q2 and Q3 enter as the primary variables of supercomponent Q. The equations 2-1 to 2-3 are used to define the variables Q1, Q2 and Q3 each one being a TOP variable of a separate fault tree. (see Fig. 2)

Notice that the group of gates G07 and G08 (Fig.1) is also linear and logically independent. We introduce here the supercomponent R with four states in a similar way as we have done in the case of Q. We finally obtain that the original fault tree has been cut into seven simpler fault trees (Fig. 3). The six new variables Q1 to Q3 and R1 to R3 enter as primary variables in the main fault tree (the upper fault tree). Each one of the six new variables is in turn a TOP variable of a separate fault tree. All seven fault trees are shown in Fig. 3.

The minimal cut sets of the main fault tree can be easily calculated by using the rules of boolean algebra with restricted variables. The algorithms are given in /7/; /8/ and /10/. The minimal cut sets are shown in Fig. 4 under the heading TOP.

We can group the minimal cut sets of the TOP with respect to all possible conjunctions among the primary variables of the supercomponents Q and R. By doing that, we get

$$\begin{aligned} \text{TOP} = & X\alpha + Q1 \cdot X\beta + Q3 \cdot X\gamma + Q2 + R1 \cdot X\delta + R3 \cdot X\epsilon + \\ & + R2 + Q1 \cdot R1 + Q1 \cdot R3 + Q3 \cdot R1 + Q3 \cdot R3 \end{aligned} \quad (2-7)$$

where

$$\begin{aligned} X\alpha = & A1 \cdot B1 + A1 \cdot C1 + A1 \cdot D1 + B1 \cdot C1 + B1 \cdot D1 + \\ & + C1 \cdot D1 \end{aligned} \quad (2-8)$$

$$X\beta = B1 + C1 + D1 \quad (2-9)$$

$$X\gamma = A1 + C1 + D1 \quad (2-10)$$

$$X\delta = A1 + B1 + D1 \quad (2-11)$$

$$X\epsilon = A1 + B1 + C1 \quad (2-12)$$

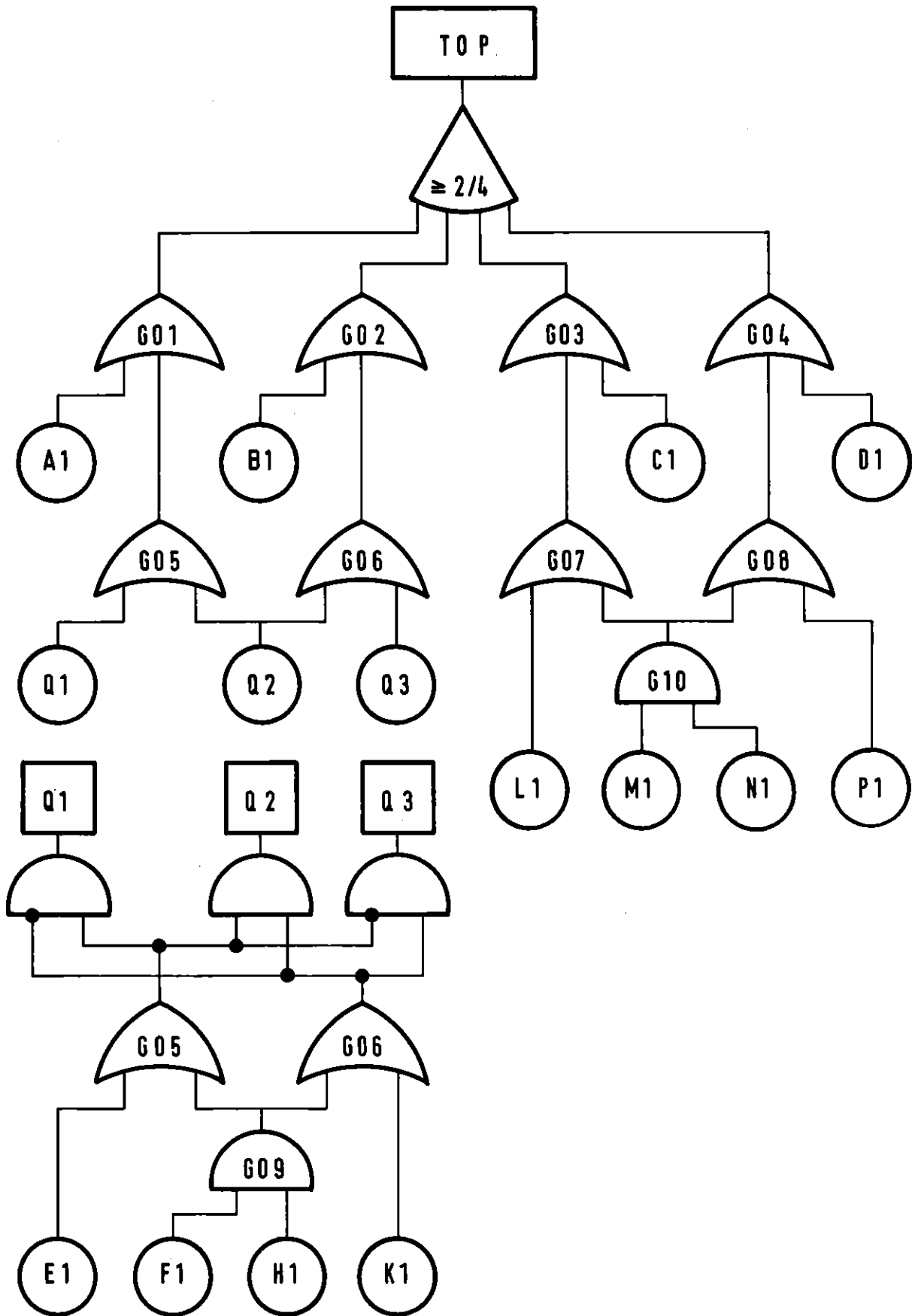


Fig. 2: Fault Tree 1. Modularisation with one Supercomponent (Q)

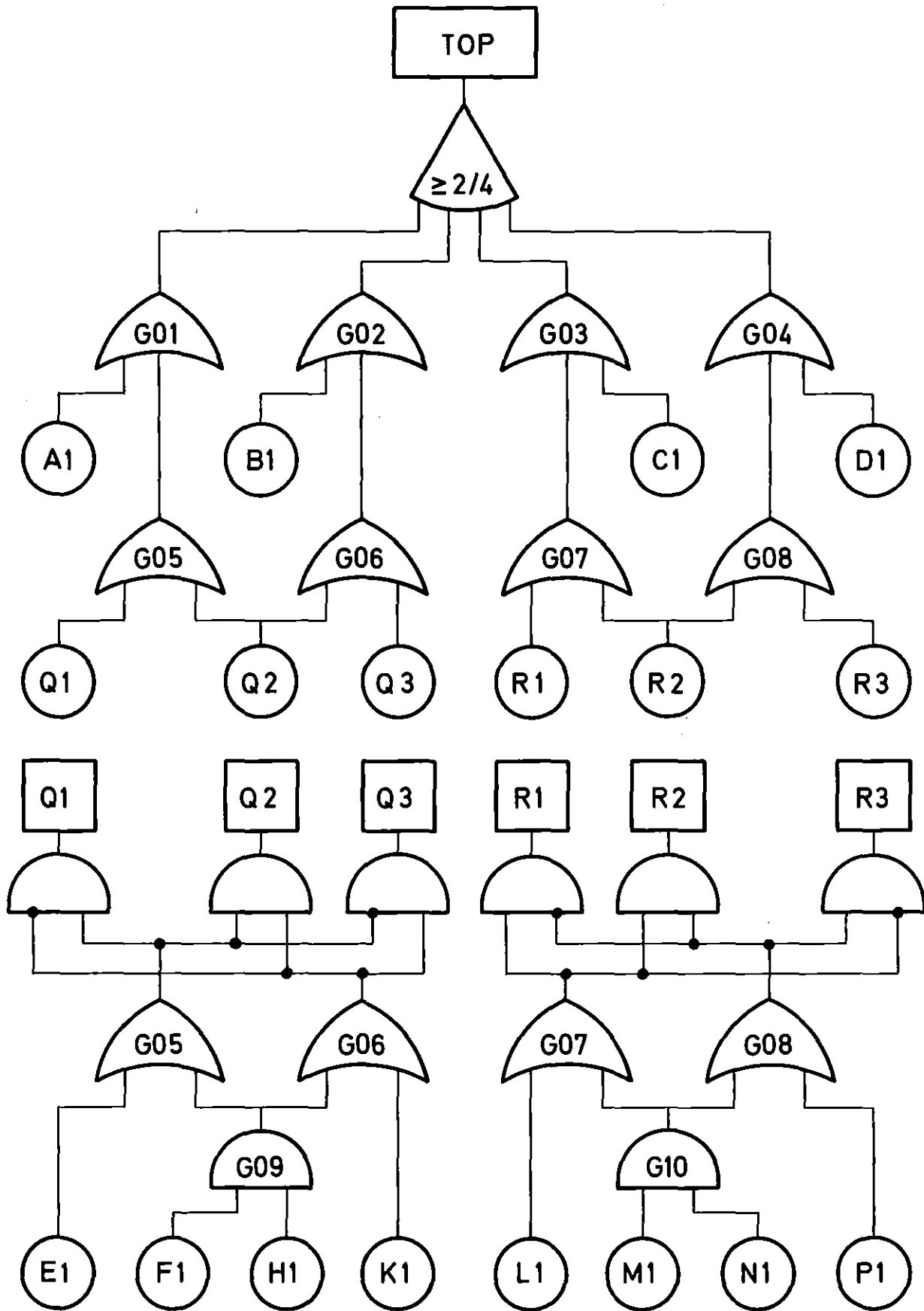


Fig. 3: Fault Tree 1. Modularisation with two Supercomponents (Q and R)

TOP

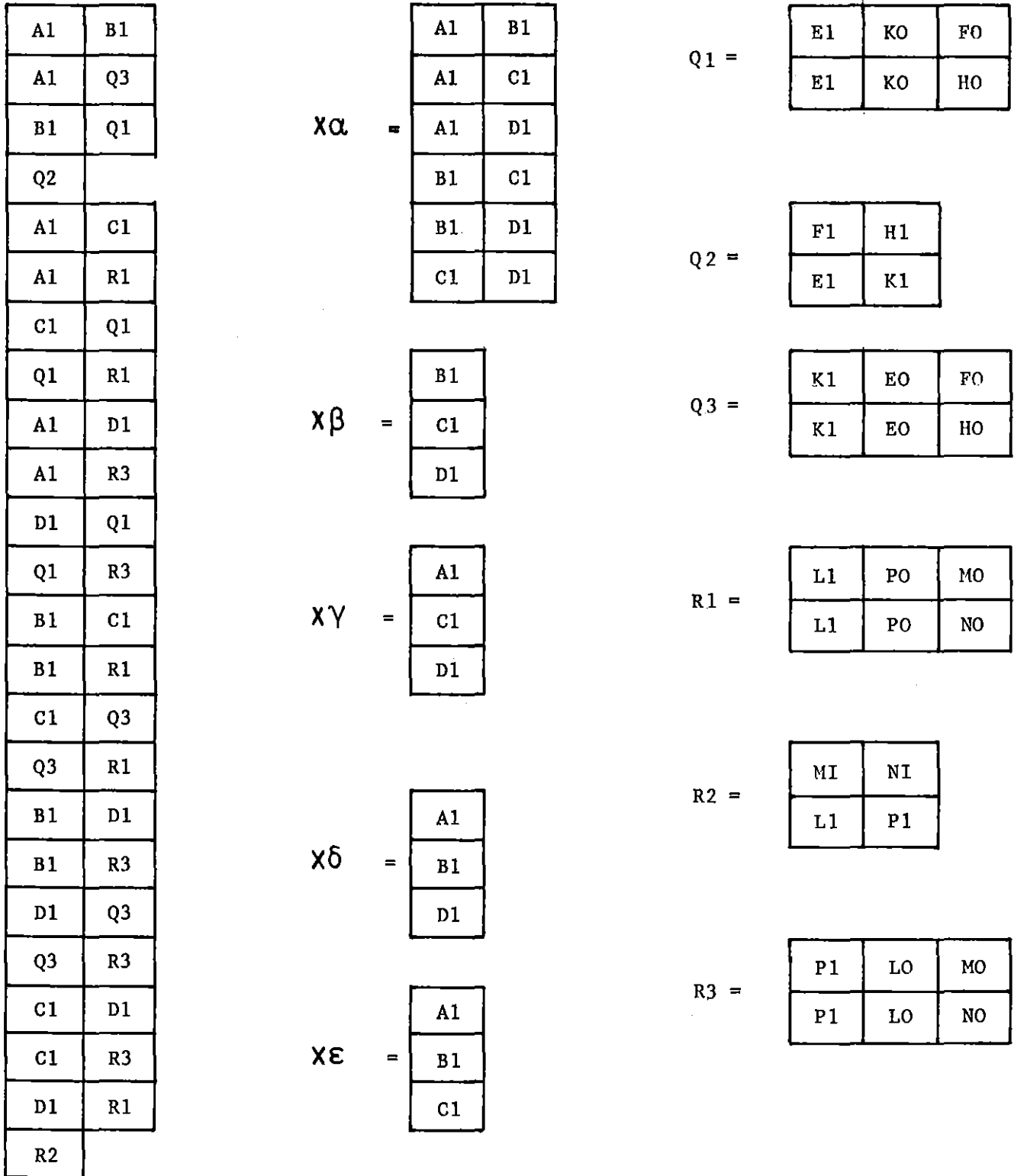


Fig. 4: Fault Tree 1. Minimal Cut Sets (M.C.S.) of the main fault tree and of the modules.

The block diagram of Fig. 5 shows the interconnections among the various boolean functions, that is Eq. 2-7. Each block (module) is a boolean function. Two blocks belonging to two different columns are pairwise each other logically independent, that is they have no primary component in common. The blocks belonging to the same row are pairwise each other logically independent.

We calculate now the minimal cut sets of the variables of the supercomponents Q and R, e.g. the fault trees Q1, Q2, Q3, R1, R2 and R3 in Fig. 3.

We calculate Q1. From Fig. 3 we get

$$Q1 = G05 \cdot \overline{G06} \quad (2-13)$$

$$G05 = E1 + G09 \quad (2-14)$$

$$G06 = K1 + G09 \quad (2-15)$$

$$G09 = F1 \cdot H1 \quad (2-16)$$

Taking into account Eqs. 2-14 to 2-16, Eq. 2-13 becomes

$$\begin{aligned} Q1 &= (E1 + F1 \cdot H1) \cdot \overline{K1} \cdot (\overline{F1} + \overline{H1}) = \\ &= (E1 + F1 \cdot H1) \cdot K0 \cdot (F0 + H0) = \\ &= E1 \cdot K0 \cdot F0 + E1 \cdot K0 \cdot H0 \end{aligned} \quad (2-17)$$

The minimal cut sets of the variables of all modules are also given in Fig. 4.

Let us now assume that all primary components of fault tree 1 are statistically independent. The expected values of the primary variables (that is the occurrence probabilities of the primary events) are assumed to be known and are given in Table 2.

With reference to the block diagram of Fig. 5, we can finally calculate the occurrence probability of the TOP event, that is the expected value of the TOP variable.

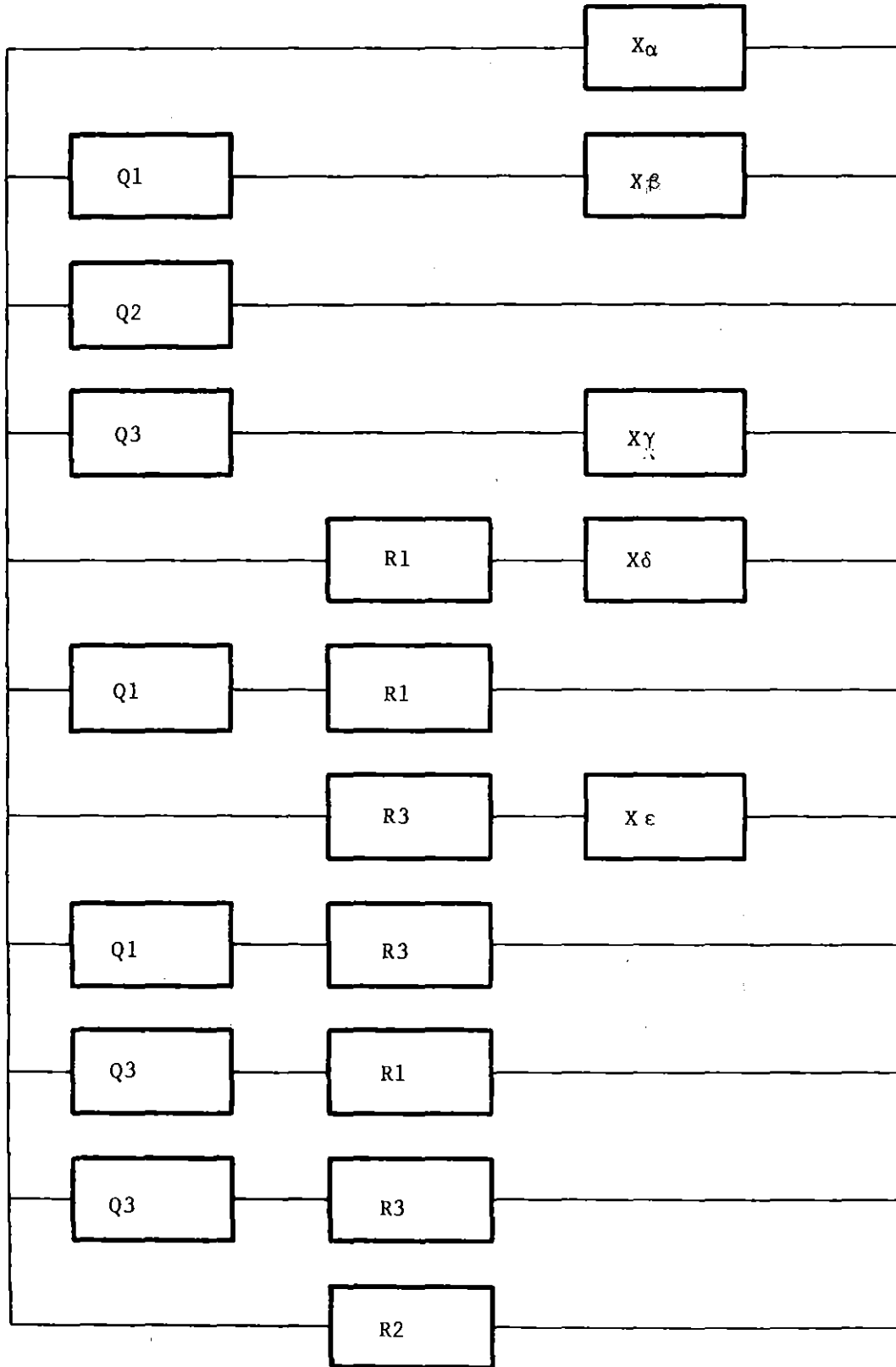


Fig. 5. Fault Tree 1. Block Diagram

Table 2

Fault Tree 1. Occurrence Probabilities of the Primary Variables

Primary Variable	Expected Value
A1	10^{-2}
B1	10^{-2}
C1	10^{-2}
D1	10^{-2}
E1	10^{-2}
F1	10^{-1}
H1	10^{-1}
K1	10^{-2}
L1	10^{-2}
M1	10^{-1}
N1	10^{-1}
P1	10^{-2}

We have

$$\begin{aligned}
 E \{ \text{TOP} \} \approx & E \{ X\alpha \} + E \{ Q1 \} \cdot E \{ X\beta \} + E \{ Q2 \} + E \{ Q3 \} \cdot E \{ X\gamma \} + \\
 & E \{ R1 \} \cdot E \{ X\delta \} + E \{ Q1 \} \cdot E \{ R1 \} + E \{ R3 \} \cdot E \{ X\epsilon \} + \\
 & E \{ Q1 \} \cdot E \{ R3 \} + E \{ Q3 \} \cdot E \{ R1 \} + E \{ Q3 \} \cdot E \{ R3 \} + \\
 & E \{ R2 \}
 \end{aligned}
 \tag{2-18}$$

The expected values of the boolean functions $X\alpha$ to $X\epsilon$, $Q1$ to $Q3$ and $R1$ to $R3$ can be calculated from the expected values of the primary variables. With reference to Fig. 4 and taking into account the numerical values of table 2, we get:

$$\begin{aligned}
 E \{ X\alpha \} \approx & E \{ A1 \} \cdot E \{ B1 \} + E \{ A1 \} \cdot E \{ C1 \} + E \{ A1 \} \cdot E \{ D1 \} + \\
 & E \{ B1 \} \cdot E \{ C1 \} + E \{ B1 \} \cdot E \{ D1 \} + E \{ C1 \} \cdot E \{ D1 \} = \\
 & = 6 \cdot 10^{-4}
 \end{aligned}
 \tag{2-19}$$

$$E \{ X\beta \} \approx E \{ B1 \} + E \{ C1 \} + E \{ D1 \} = 3 \cdot 10^{-2}
 \tag{2-20}$$

$$E \{ X\gamma \} \approx E \{ A1 \} + E \{ C1 \} + E \{ D1 \} = 3 \cdot 10^{-2}
 \tag{2-21}$$

$$E \{ X\delta \} \approx E \{ A1 \} + E \{ B1 \} + E \{ D1 \} = 3 \cdot 10^{-2}
 \tag{2-22}$$

$$E \{ X\epsilon \} \approx E \{ A1 \} + E \{ B1 \} + E \{ C1 \} = 3 \cdot 10^{-2}
 \tag{2-23}$$

$$E \{ Q1 \} \approx E \{ E1 \} = 10^{-2}
 \tag{2-24}$$

$$\begin{aligned}
 E \{ Q2 \} \approx & E \{ F1 \} \cdot E \{ H1 \} + \\
 & E \{ E1 \} \cdot E \{ K1 \} = 1.01 \cdot 10^{-2}
 \end{aligned}
 \tag{2-25}$$

$$E \{ Q3 \} \approx E \{ K1 \} = 10^{-2} \quad (2-26)$$

$$E \{ R1 \} \approx E \{ L1 \} = 10^{-2} \quad (2-27)$$

$$\begin{aligned} E \{ R2 \} &\approx E \{ M1 \} \cdot E \{ N1 \} + E \{ L1 \} \cdot E \{ P1 \} = \\ &= 1.01 \cdot 10^{-2} \end{aligned} \quad (2-28)$$

$$E \{ R3 \} \approx E \{ P1 \} = 10^{-2} \quad (2-29)$$

The expected values of the modules are written inside the corresponding block of the block diagram of Fig. 5. This has been done in Fig. 6 where the operations of Eq. 2-18 have been carried out. The expected value of the TOP is equal to $2.24 \cdot 10^{-2}$.

In order to compare the results of this method with those of other methods, one may be interested in calculating the minimal cut sets of the whole fault tree, starting from the minimal cut sets of each module.

We notice that the minimal cut sets of the variables Q1, Q3, R1 and R3 contain intact primary variables. On the other hand we know that the TOP is a coherent boolean function. Due to the theorems developed in /10/ and mentioned in section 2 of this paper, the coherent function TOP remains unaltered if in one of its normal disjunctive forms all intact variables are replaced by 1.

According to definition 20 of section 2, the associated coherent function of a given boolean function (say Q1) is that function (say CQ1) obtained from Q1 by replacing the intact variables by 1.

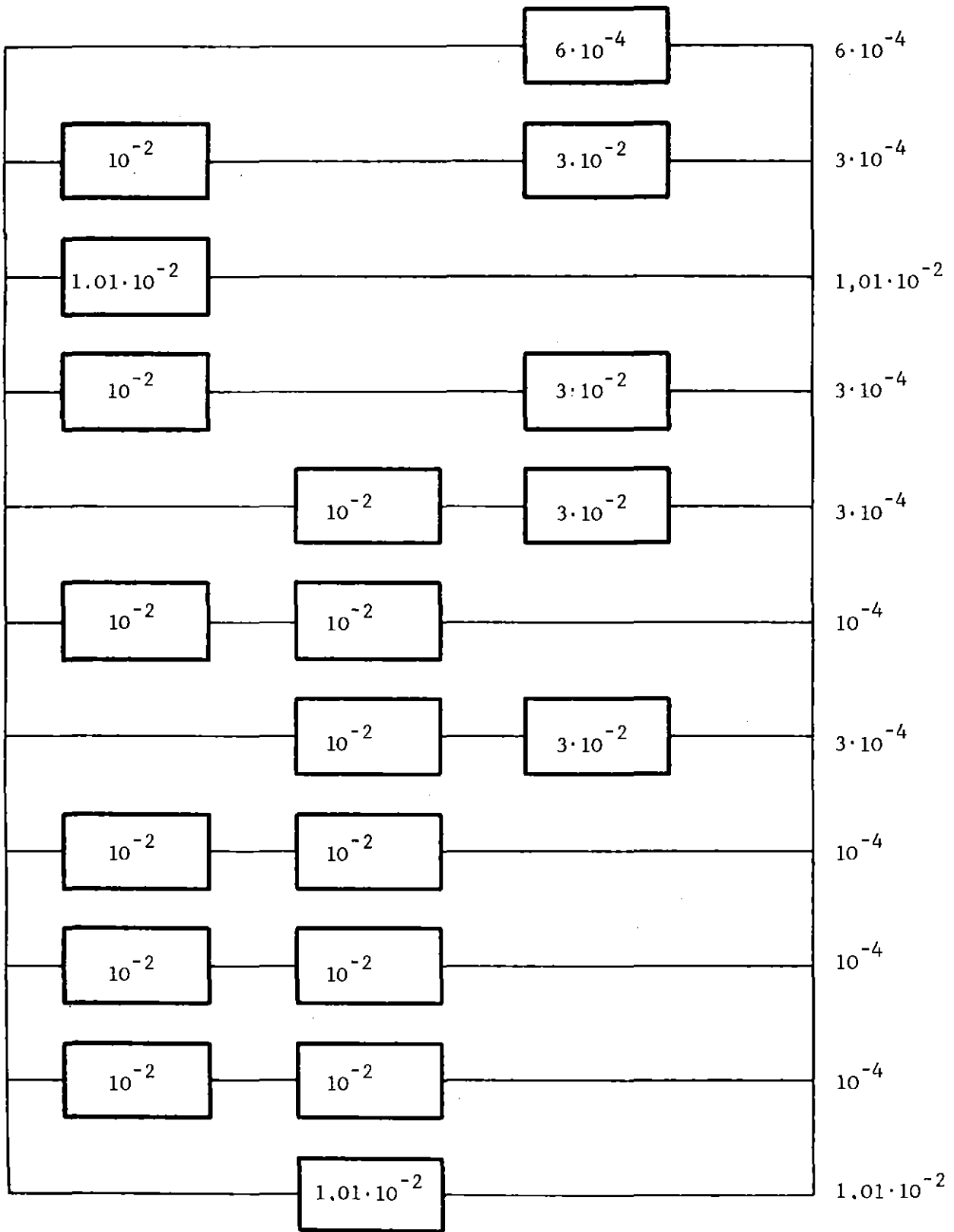
With reference to Fig. 4, we can calculate the associated coherent functions of Q1, Q3, R1 and R3. We have

$$CQ1 = E1 \quad (2-30)$$

$$CQ3 = K1 \quad (2-31)$$

$$CR1 = L1 \quad (2-32)$$

$$CR3 = P1 \quad (2-33)$$



Expected Value of TOP: $2.24 \cdot 10^{-2}$

Fig. 6 Fault Tree 1. Calculation of the Occurrence Probability of the TOP.

The functions Q2, R2 and X α to X ϵ are not effected by this operation because they do not contain any intact primary variable.

By replacing in Eq. 2-7 the boolean functions Q1, Q3, R1 and R3 by their associated coherent functions, we get

$$\begin{aligned} \text{TOP} = & X\alpha + CQ1 \cdot X\beta + CQ3 \cdot X\gamma + Q2 + CR1 \cdot X\delta + \\ & + CR3 \cdot X\epsilon + R2 + CQ1 \cdot CR1 + CQ1 \cdot CR3 + \\ & + CQ3 \cdot CR1 + CQ3 \cdot CR3 \end{aligned} \quad (2-34)$$

It is important to point out that the operation of replacing the intact variables by 1 alters the functions Q1, Q3, R1 and R3 respectively into CQ1, CQ3, CR1 and CR3 but leaves the function TOP unaltered.

The number of minimal cut sets of each associated coherent function is written inside the corresponding block of the block diagram (Fig. 7). The number of minimal cut sets (m.c.s.) of each row is simply given by multiplying the number of m.c.s. of all blocks belonging to the same row. Each result is written in correspondence of each row on the right side of the block diagram (see Fig. 7). The total number of minimal cut sets is simply given by summing up the number of m.c.s. of all rows. This operation is also shown in Fig. 7. The total number of m.c.s. of the fault tree 1 is equal to 26.

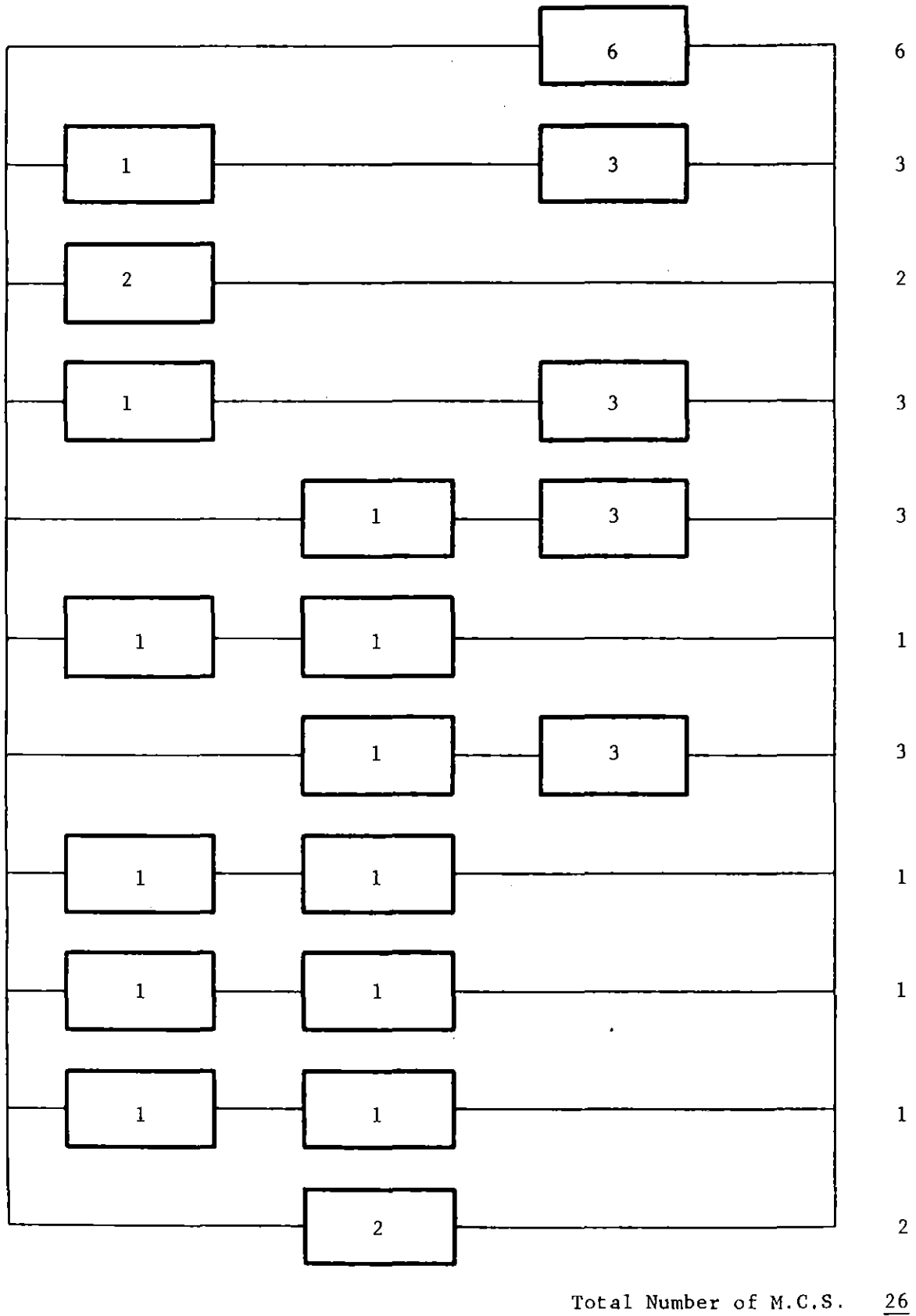


Fig. 7 Fault Tree 1. Calculation of the Number of Minimal Cut Sets (M.C.S.).

3. An example

Fig. 8 shows a larger fault tree (fault tree 2) which was proposed to the authors for test purposes by the Safety and Reliability Directorate (S.R.D.), UKAEA, Warrington, Great Britain. The TOP event of this fault tree is the failure of a part of a reactor protective system which is described in /14/. The occurrence probabilities of the primary events are given in Table 3. This fault tree has about $4.18 \cdot 10^{17}$ cut sets, 5630 of them being minimal cut sets.

By looking at the fault tree of Fig. 8, we notice that the group of gates G06 and G07 is linear and logically independent. We therefore introduce the supercomponent SC01 with $2^2 = 4$ states, namely

$$SC01-1 = G06 \cdot G07 \quad (3-1)$$

$$SC01-2 = \overline{G06} \cdot G07 \quad (3-2)$$

$$SC01-3 = G06 \cdot \overline{G07} \quad (3-3)$$

$$SC01-0 = \overline{G06} \cdot \overline{G07} \quad (3-4)$$

By applying the same procedure described in the previous section, one can cut the original fault tree into four smaller fault trees as it is shown in Fig. 9.

The computer program MUSTAMO executes this cut of a large fault tree into smaller fault trees and analyzes all the resulting fault trees separately one after the other.

The block diagram of fault tree 2 is shown in Fig. 10. Here the block characterized by the number 30 consists of the failed state of the primary component 30. The minimal cut sets of the modules M001 and M002 are listed respectively in Table 4 and 5. Fig. 11 shows the expected values of each module. These expected values are calculated by simply summing up the expected values of all minimal cut sets belonging to the module. It is known that this procedure overestimates the expected value of a module. The occurrence probability of the TOP is $1.947 \cdot 10^{-5}$. The CPU time for the complete analysis of fault tree 2 was 48.5 secs. on a IBM 3033 computer. From Fig. 11 it results that the modules SC01-1 (expected value: $1.275 \cdot 10^{-5}$) and M002 (expected value $6.699 \cdot 10^{-6}$) give by far the largest contributions to the occurrence probability of the TOP. For this reason the expected values of these two modules have been calculated by using the more precise method described in /8/. The results are written in Fig. 11 between brackets. The exact value of the occurrence probability of the TOP results to be $1.794 \cdot 10^{-5}$.

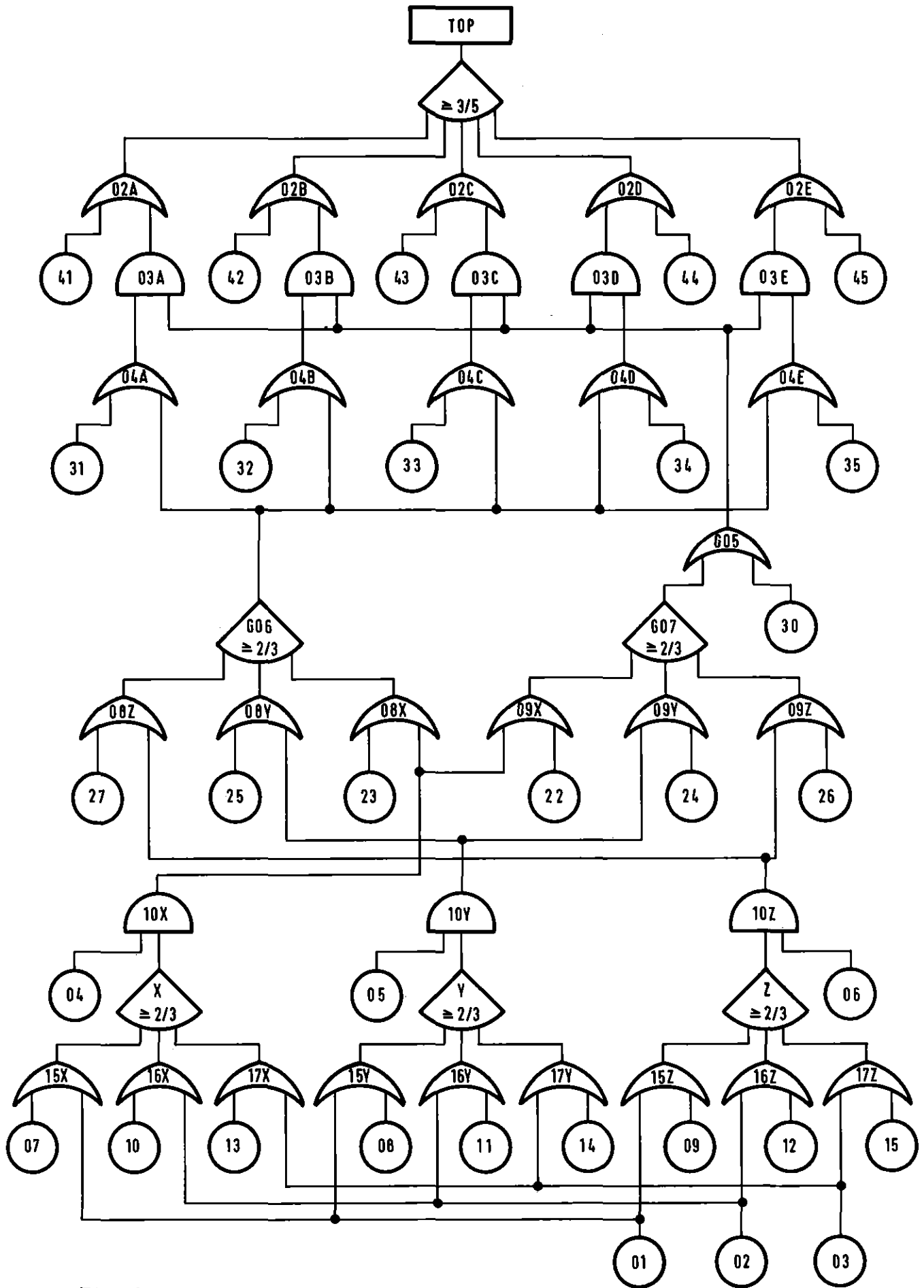


Fig. 8: Fault Tree 2

Table 3

Expected Values of the Primary Variables of Fault Tree 2

Primary Variable	Expected Value
From 01 to 03	$3.5 \cdot 10^{-2}$
From 04 to 06	$2.2 \cdot 10^{-2}$
From 07 to 15	$1 \cdot 10^{-1}$
From 22 to 27	$8.8 \cdot 10^{-4}$
From 30 to 35	$1.75 \cdot 10^{-3}$
From 41 to 45	$8.75 \cdot 10^{-3}$

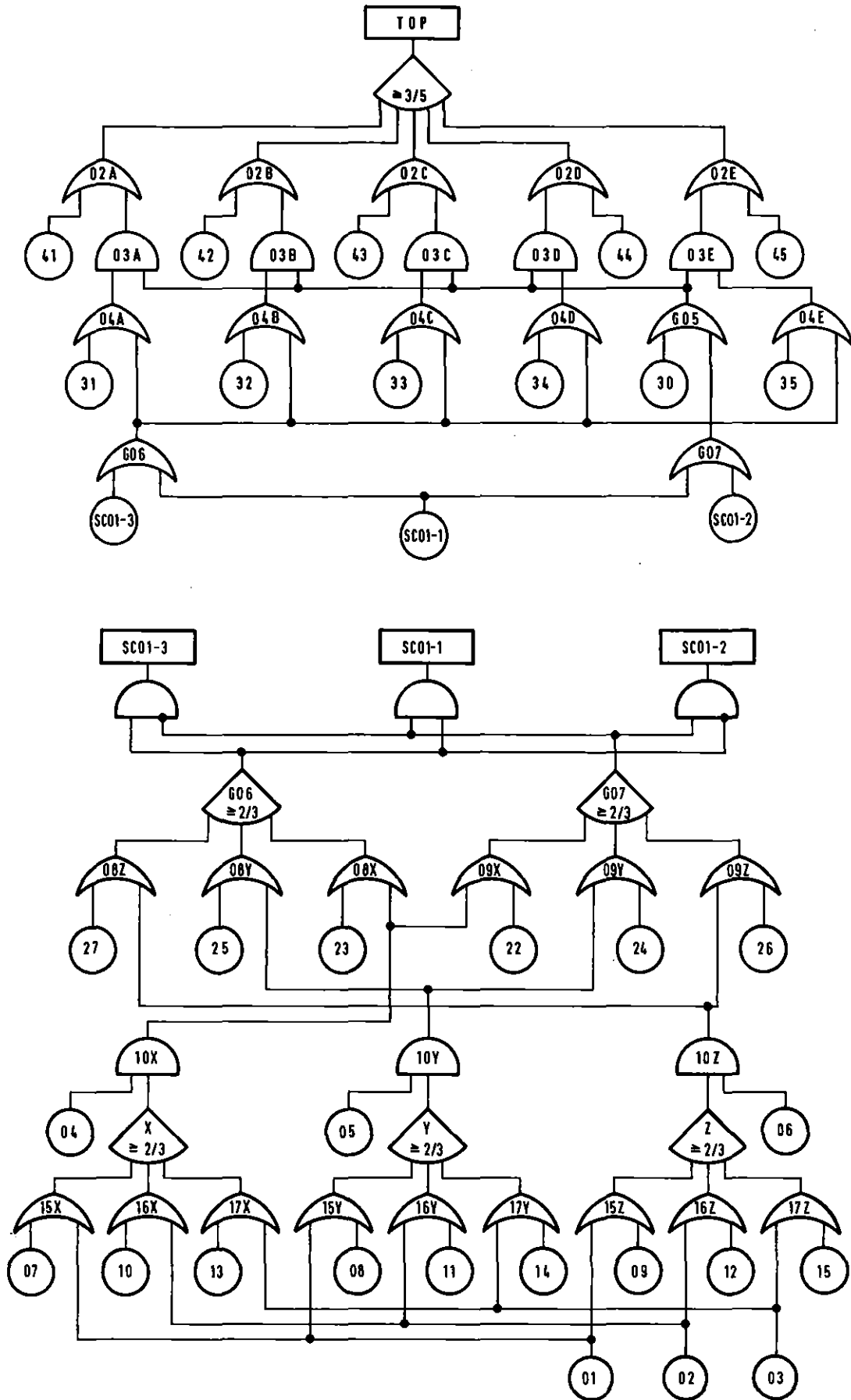


Fig. 9: Modularization of Fault Tree 2 by means of one Supercomponent (SC01)

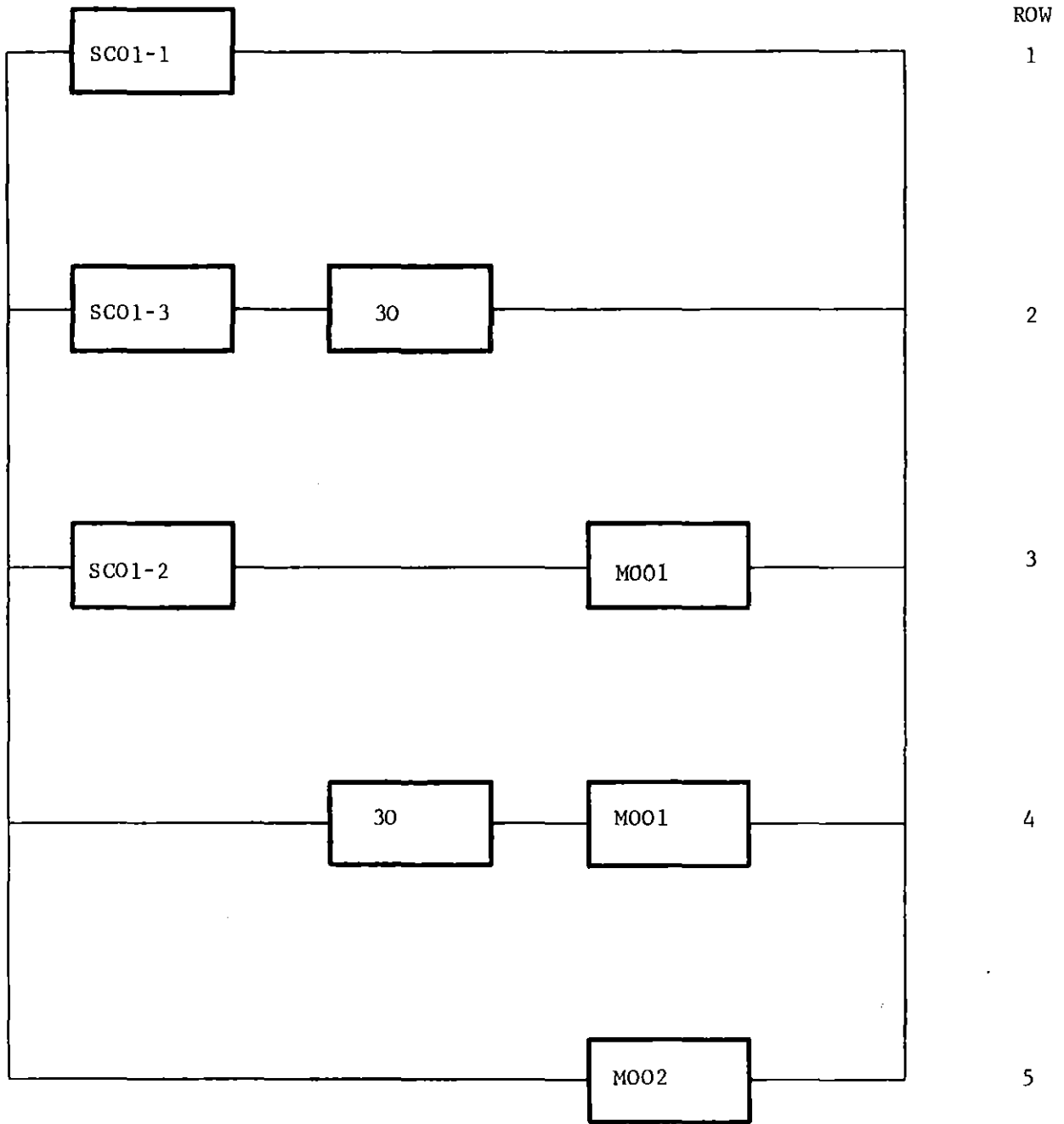


Fig. 10 Block Diagram of Fault Tree 2 , Modularization with one Supercomponent (SC01)

Table 4

Fault Tree 2. Minimal Cut Sets (M.C.S.) of Module M001

M.C.S.	Composition	M.C.S.	Composition	M.C.S.	Composition
1	31 · 32 · 33	25	31 · 43 · 44	49	42 · 43 · 34
2	31 · 32 · 43	26	41 · 33 · 34	50	32 · 33 · 35
3	31 · 42 · 33	27	41 · 33 · 44	51	32 · 33 · 45
4	31 · 42 · 43	28	41 · 43 · 34	52	32 · 43 · 35
5	41 · 32 · 33	29	31 · 33 · 35	53	32 · 43 · 45
6	41 · 32 · 43	30	31 · 33 · 45	54	42 · 33 · 35
7	41 · 42 · 33	31	31 · 43 · 35	55	42 · 33 · 45
8	31 · 32 · 34	32	31 · 43 · 45	56	42 · 43 · 35
9	31 · 32 · 44	33	41 · 33 · 35	57	32 · 34 · 35
10	31 · 42 · 34	34	41 · 33 · 45	58	32 · 34 · 45
11	31 · 42 · 44	35	41 · 43 · 35	59	32 · 44 · 35
12	41 · 32 · 34	36	31 · 34 · 35	60	32 · 44 · 45
13	41 · 32 · 44	37	31 · 34 · 45	61	42 · 34 · 35
14	41 · 42 · 34	38	31 · 44 · 35	62	42 · 34 · 45
15	31 · 32 · 35	39	31 · 44 · 45	63	42 · 44 · 35
16	31 · 32 · 45	40	41 · 34 · 35	64	33 · 34 · 35
17	31 · 42 · 35	41	41 · 34 · 45	65	33 · 34 · 45
18	31 · 42 · 45	42	41 · 44 · 35	66	33 · 44 · 35
19	41 · 32 · 35	43	32 · 33 · 34	67	33 · 44 · 45
20	41 · 32 · 45	44	32 · 33 · 44	68	43 · 34 · 35
21	41 · 42 · 35	45	32 · 43 · 34	69	43 · 34 · 45
22	31 · 33 · 34	46	32 · 43 · 44	70	43 · 44 · 35
23	31 · 33 · 44	47	42 · 33 · 34		
24	31 · 43 · 34	48	42 · 33 · 44		

Table 5

Fault Tree 2. Composition of the Minimal Cut Sets (M.C.S.) of Module M002.

M.C.S.	Composition
1	41 · 42 · 43
2	41 · 42 · 44
3	41 · 42 · 45
4	41 · 43 · 44
5	41 · 43 · 45
6	41 · 44 · 45
7	42 · 43 · 44
8	42 · 43 · 45
9	42 · 44 · 45
10	42 · 44 · 45

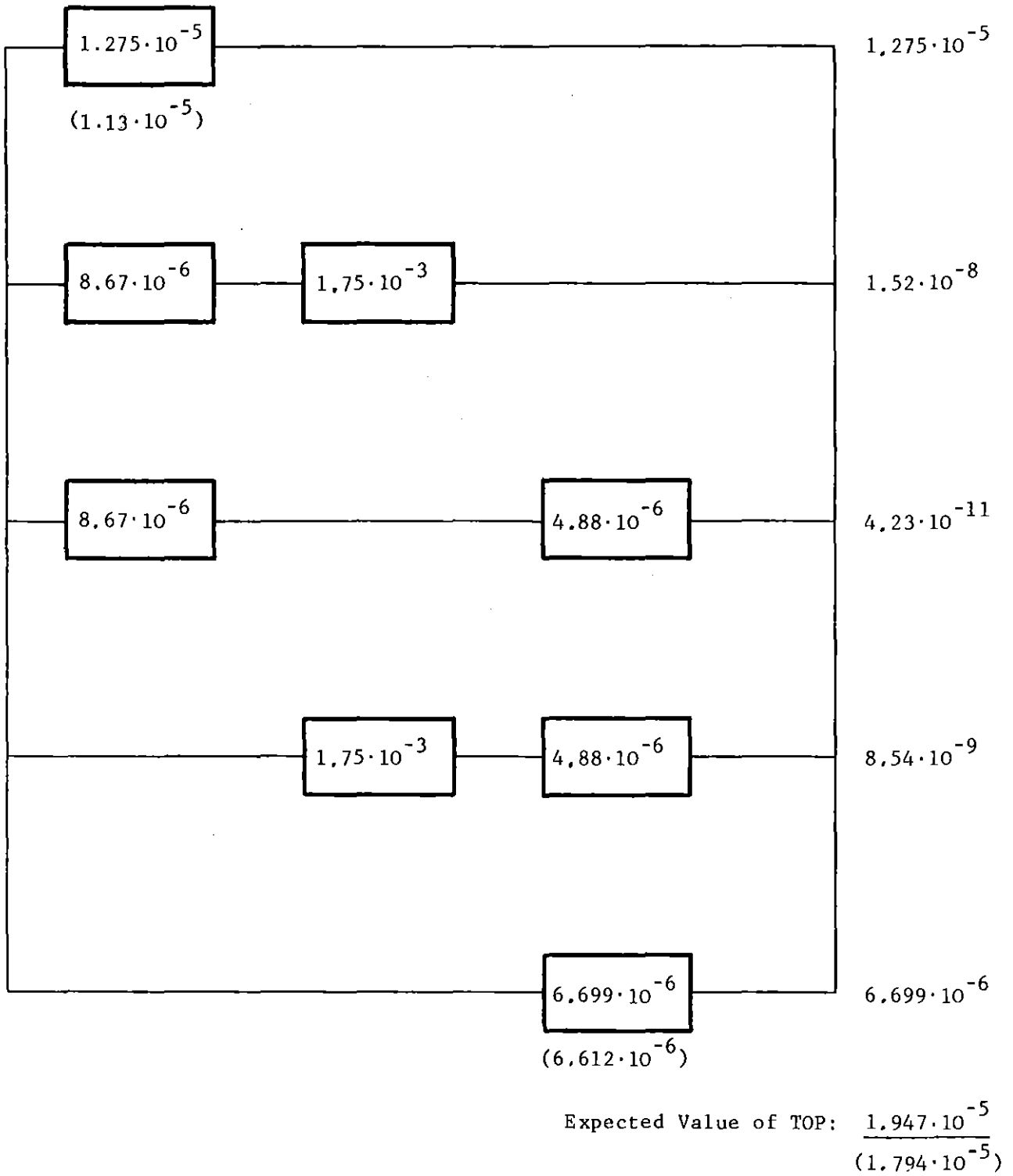


Fig. 11 Fault Tree 2. Modularization with one Supercomponent (SC01)
Calculation of the Occurrence Probability of the TOP.

In order to better compare the results obtained in this paper with those of S.R.D., the minimal cut sets (m.c.s.) have been divided into groups, each group being characterized by the length of the m.c.s. that is the number of primary variables contained in the m.c.s.

Table 6 gives the total number of m.c.s. contained in the associated coherent functions of each block ordered according to their length. Note that only the associated coherent functions CSC01-2 and CSC01-3 are different from the functions (SC01-2 and SC01-3) from which they have respectively derived.

The information contained in Table 6 has been used to calculate the total number of m.c.s. contained in each row of the block diagram ordered according to their length. This result is shown in Table 7, where the total number of m.c.s. ordered according to their length of the whole fault tree has been calculated (see last column of Table 7). The results up to the length 6 of the m.c.s. are identical with those of S.R.D. /15/. The remaining m.c.s. of order 7 could not be compared because the computer programs available at S.R.D. were not able to calculate all m.c.s. of the fault tree. From Table 7 one gets that the total number of m.c.s. of the fault tree 2 is equal to 5630.

The group of gates 10X, 10Y and 10Z of fault tree 2 (Fig. 8) is also linear and logically independent. One could introduce therefore an additional supercomponent with $2^3 = 8$ states, which is obtained by combining the three gates and their complements in all possible ways, that is

$$\begin{aligned} & 10 X \cdot 10 Y \cdot 10 Z \\ & \overline{10 X} \cdot 10 Y \cdot 10 Z \\ & 10 X \cdot \overline{10 Y} \cdot 10 Z \\ & 10 X \cdot 10 Y \cdot \overline{10 Z} \\ & \overline{10 X} \cdot \overline{10 Y} \cdot 10 Z \\ & \overline{10 X} \cdot 10 Y \cdot \overline{10 Z} \\ & 10 X \cdot \overline{10 Y} \cdot \overline{10 Z} \\ & \overline{10 X} \cdot \overline{10 Y} \cdot \overline{10 Z} \end{aligned}$$

It is possible however to reduce the number of states of the second supercomponent from 8 to 5 by condensing the first four states into a single macrostate.

Table 6

Fault Tree 2. Total Number of M.C.S. contained in each Block.

Length of M.C.S.	Block (associated coherent function)					
	SC01-1	CSC01-2	CSC01-3	30	MO01	MO02
1				1		
2		3	3			
3					70	10
4	18	72	72			
5	180					
6	27					
Total	225	75	75	1	70	10

Table 7

Fault Tree 2. Total Number of M.C.S. contained in each Row.

Length of M.C.S.	Row					Total
	1	2	3	4	5	
1						
2						
3		3			10	13
4	18			70		88
5	180	72	210			462
6	27					27
7			5040			5040
Total	225	75	5250	70	10	5630

If one calculates with the code MUSTAMO the fault tree of Fig. 8 (or those of Fig. 9) by considering the gates 10X, 10Y and 10Z as failed states of three different primary components, one gets a solution of the type

$$\begin{aligned} \text{TOP} &= 10X \cdot KX + 10Y \cdot KY + 10Z \cdot KZ + \\ &+ (10X \cdot 10Y + 10X \cdot 10Z + 10Y \cdot 10Z) \end{aligned} \quad (3-5)$$

where KX, KY and KZ are boolean functions which do not contain 10X, 10Y and 10Z.

Let us indicate with A1 the boolean function between brackets in Eq. 3-5, that is

$$A1 = 10X \cdot 10Y + 10X \cdot 10Z + 10Y \cdot 10Z \quad (3-6)$$

Eq. 3-5 can be written as follows:

$$\begin{aligned} \text{TOP} &= (10X + A1) \cdot KX + (10Y + A1) \cdot KY + \\ &+ (10Z + A1) \cdot KZ + A1 \end{aligned} \quad (3-7)$$

Eq. 3-7 means that the TOP remains unaltered if one replaces in the fault tree of Fig. 8 (or in those of Fig. 9) the variables 10X, 10Y and 10Z respectively with $(10X + A1)$, $(10Y + A1)$ and $(10Z + A1)$. This allows us to introduce the supercomponent SC02 with five states, namely

$$\text{SC02-4} = A1 = 10X \cdot 10Y + 10X \cdot 10Z + 10Y \cdot 10Z \quad (3-8)$$

$$\text{SC02-1} = (10X + A1) \cdot \overline{A1} = 10X \cdot \overline{10Y} \cdot \overline{10Z} \quad (3-9)$$

$$\text{SC02-2} = (10Y + A1) \cdot \overline{A1} = \overline{10X} \cdot 10Y \cdot \overline{10Z} \quad (3-10)$$

$$\text{SC02-3} = (10Z + A1) \cdot \overline{A1} = \overline{10X} \cdot \overline{10Y} \cdot 10Z \quad (3-11)$$

$$\text{SC02-0} = \overline{10X} \cdot \overline{10Y} \cdot \overline{10Z} \quad (3-12)$$

Note that the macrovariable (macrostate) SC02-4 results from the disjunction (condensation) of the four variables (states) $10X \cdot 10Y \cdot 10Z$, $\overline{10X} \cdot 10Y \cdot 10Z$, $10X \cdot \overline{10Y} \cdot 10Z$, and $10X \cdot 10Y \cdot \overline{10Z}$.

Fig. 12 shows fault tree 2 cut at two levels, namely G06 and G07 (supercomponent SC01 with four states) and 10X, 10Y and 10Z (supercomponent SC02 with five states).

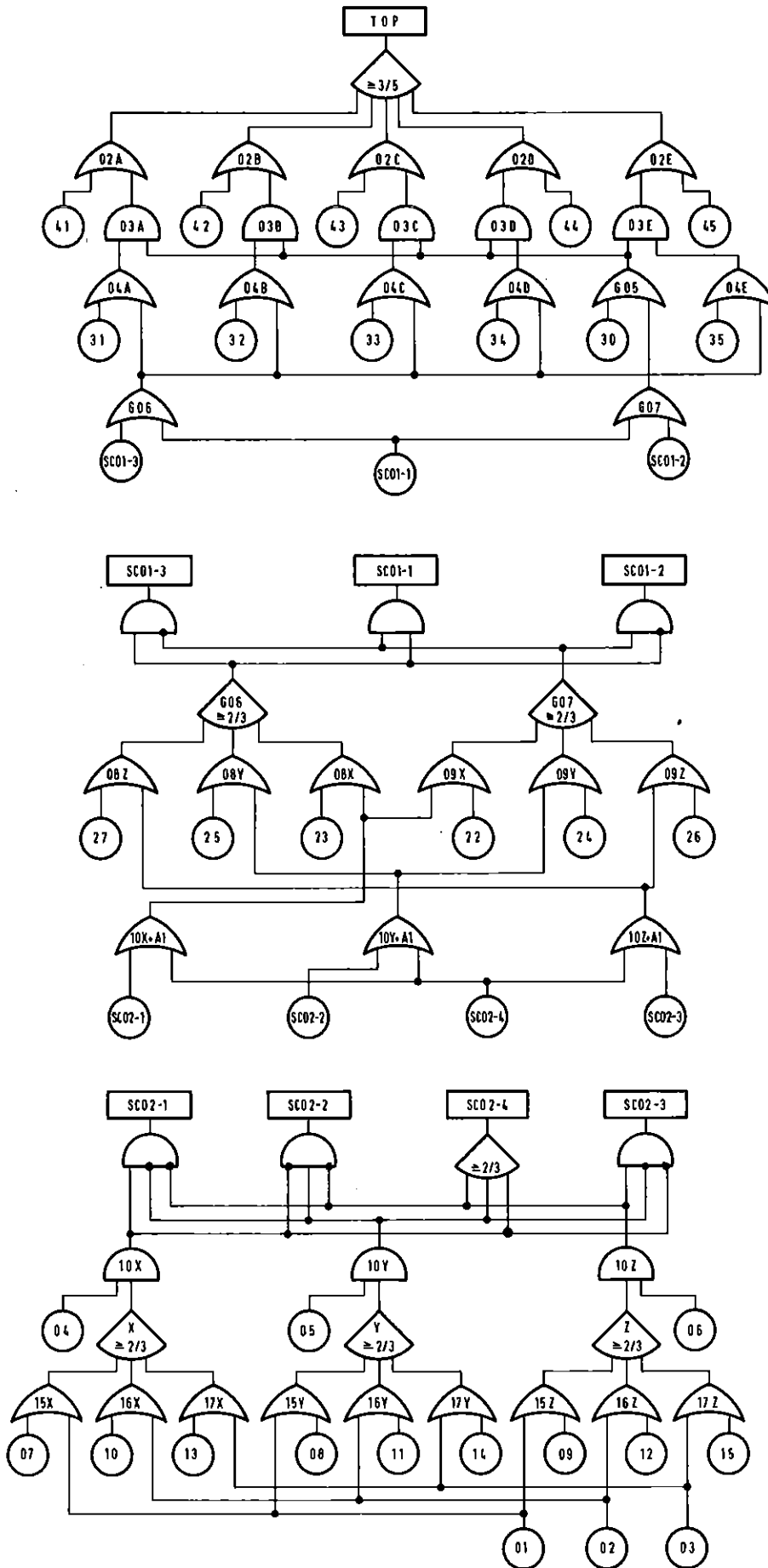


Fig. 12: Modularization of Fault Tree 2 with two Supercomponents (SC01 and SC02) in Cascade.

The computer program MUSTAMO executes the two cuts in cascade of fault tree 2 and analyses the resulting fault trees (Fig. 12) separately one after the other starting from the fault trees at the bottom.

Fig. 13 shows the block diagram of fault tree 2 with the two cuts in cascade. The m.c.s. of the modules NO01 to NO12 are given in Table 8. The m.c.s. of the functions MO01 and MO02 are given in Tables 4 and 5 respectively.

The expected values of each module are shown in Fig. 14, where the occurrence probability of the TOP has been calculated. The result is of course identical with that already obtained in the case of one supercomponent. From Fig. 14 one concludes that the modules SC02-4 (expected value: $1.274 \cdot 10^{-5}$) and the module MO02 (expected value $6.669 \cdot 10^{-6}$) give by far the largest contributions to the occurrence probability of the TOP.

Fig. 15 shows the calculation of the total number of minimal cut sets (m.c.s.). Note that the blocks SC02-0 have disappeared in the block diagram of Fig. 15 because the associated coherent function of SC02-0 is just 1 and does not give therefore any contribution to the m.c.s. of the fault tree. The notations CSC01-2 and CSC01-3 in Fig. 15 indicate the associated coherent functions respectively of SC01-2 and SC01-3.

We compare now the block diagram of Fig. 13 with that of Fig. 10. In the block diagram of Fig. 13 the modules SC01-1 to SC01-3 have been decomposed into smaller modules. The block diagram of Fig. 13 can be obtained from that of Fig. 10 just by carrying out this decomposition.

The block diagram of Fig. 13 is more complex but it gives also more insight into the importance of the various blocks. For instance we have already noticed that the module SC01-1 (225 minimal cut sets, Table 6) gives the largest contribution to the system unavailability (Fig. 11). This contribution is almost equal to that of the module SC02-4 (Fig. 14), which is a part of SC01-1, has only 72 m.c.s. (Fig. 15) and is therefore easier to analyze.

The CPU time for the complete analysis of fault tree 2 with two supercomponents in cascade (Fig. 12) was about 3 secs. This value is remarkably lower than the already mentioned value of 48.5 secs. of the CPU time of the case with only one supercomponent (Fig. 9).

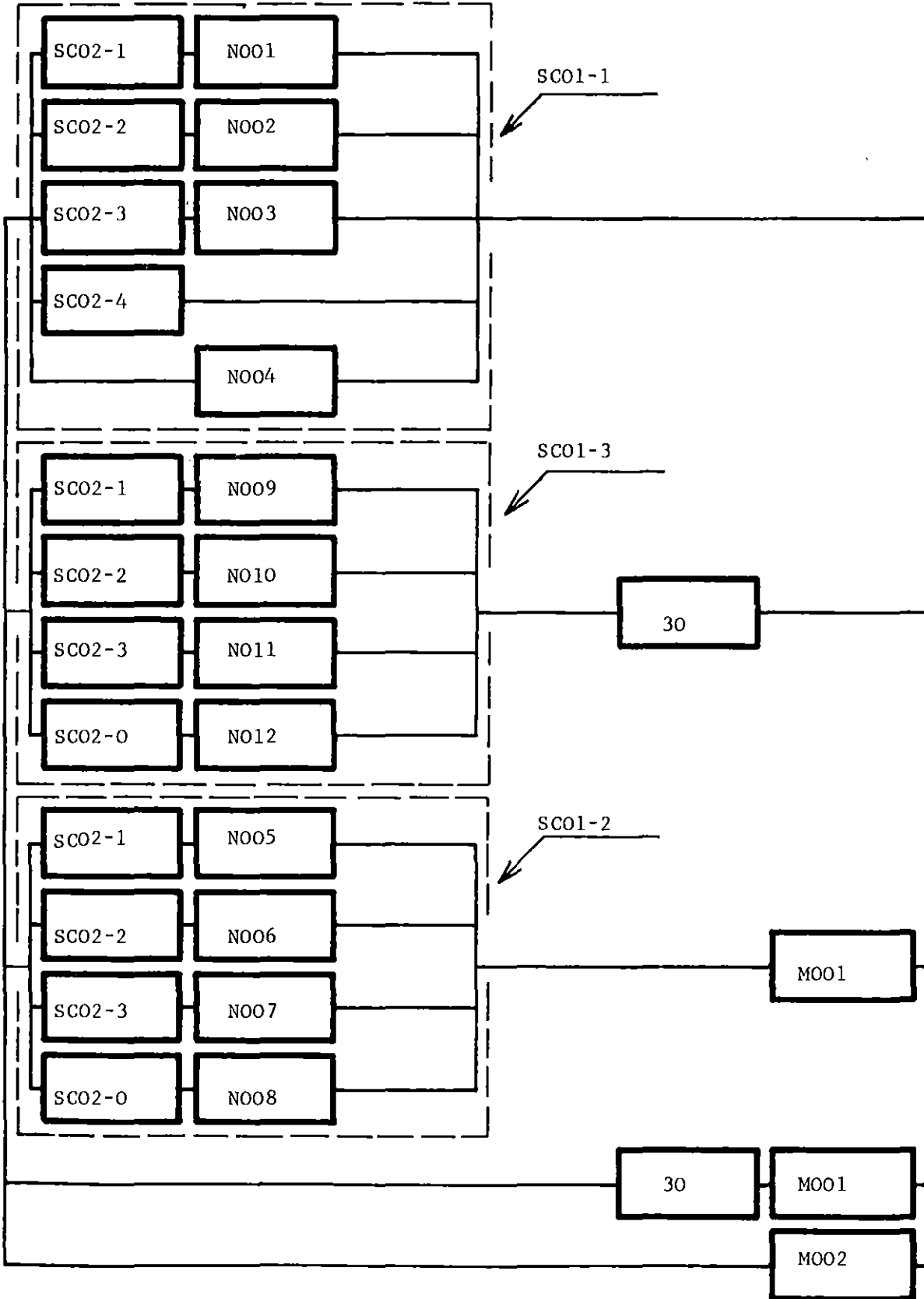


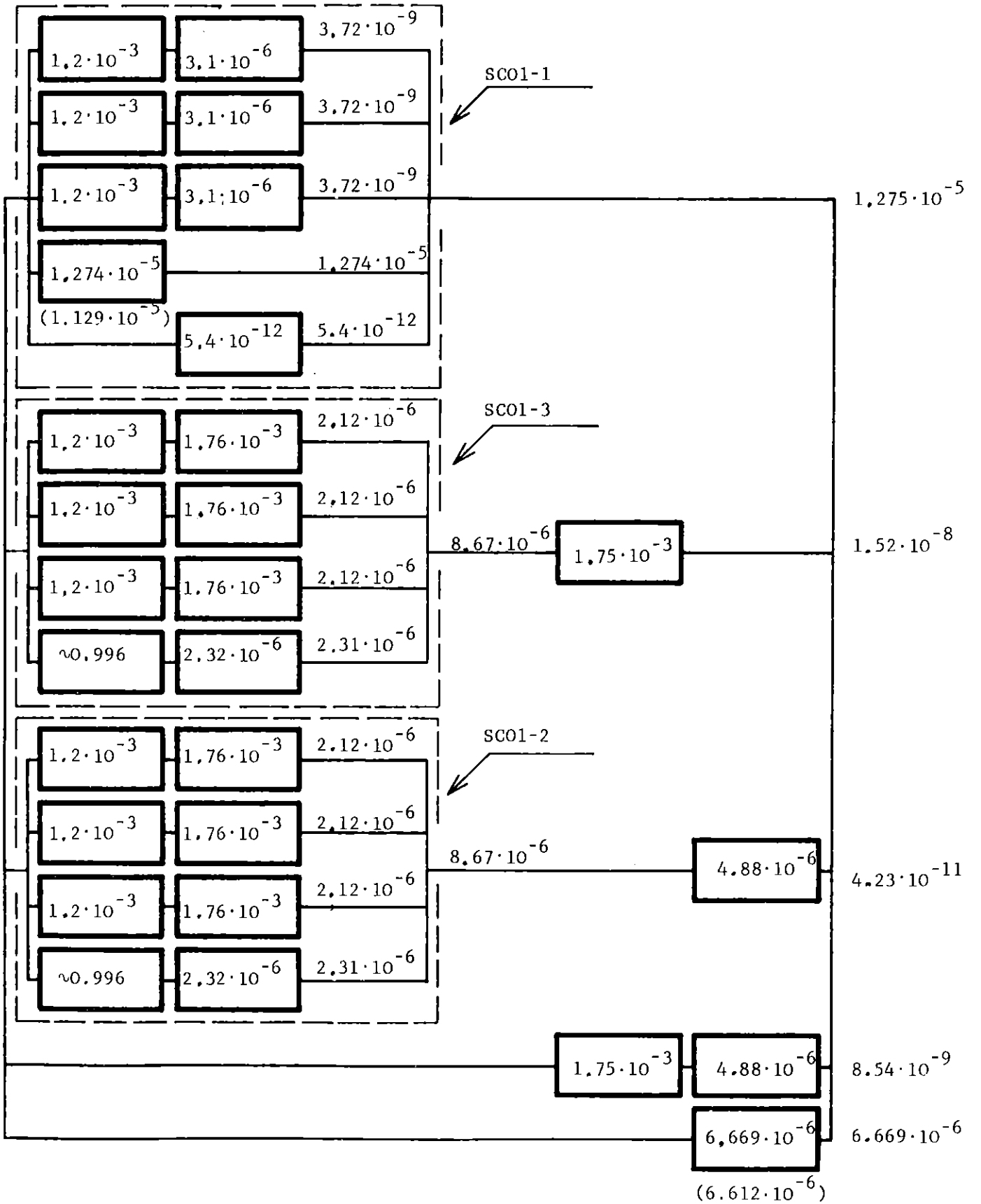
Fig. 13 Block Diagram of Fault Tree 2. Modularization with two Supercomponents (SC01 and SC02) in Cascade.

Table 8

Fault Tree 2. Two Supercomp. in Cascade - M.C.S. of NO01 to NO12

Block	Composition of M.C.S.	
	Module	Ass.Coh.Function
NO01	24 · 25	24 25
	24 · 27	24 27
	26 · 25	26 25
	26 · 27	26 27
NO02	22 · 23	22 23
	22 · 27	22 27
	26 · 23	26 23
	26 · 27	26 27
NO03	22 · 23	22 23
	22 · 25	22 25
	24 · 23	24 23
	24 · 25	24 25
NO04	22 · 24 · 23 · 25	22 24 23 25
	22 · 24 · 23 · 27	22 24 23 27
	22 · 24 · 25 · 27	22 24 25 27
	22 · 26 · 23 · 25	22 26 23 25
	22 · 26 · 23 · 27	22 26 23 27
	22 · 26 · 25 · 27	22 26 25 27
	24 · 26 · 23 · 25	24 26 23 25
	24 · 26 · 23 · 27	24 26 23 27
NO05	24 · $\overline{27}$ · $\overline{25}$	24
	26 · $\overline{27}$ · $\overline{25}$	26
NO06	22 · $\overline{27}$ · $\overline{23}$	22
	26 · $\overline{27}$ · $\overline{23}$	26

Block	Composition of M.C.S.	
	Module	Ass.Coh.Function
NO07	22 · $\overline{25}$ · $\overline{23}$	22
	24 · $\overline{25}$ · $\overline{23}$	24
NO08	22 · 24 · $\overline{25}$ · $\overline{23}$	22 · 24
	22 · 24 · $\overline{27}$ · $\overline{23}$	
	22 · 24 · 27 · 25	22 · 26
	22 · 26 · $\overline{25}$ · $\overline{23}$	
	22 · 26 · $\overline{27}$ · $\overline{23}$	
	22 · 26 · $\overline{27}$ · $\overline{25}$	
NO08	24 · 26 · $\overline{25}$ · $\overline{23}$	24 · 26
	24 · 26 · $\overline{27}$ · $\overline{23}$	
	24 · 26 · $\overline{27}$ · $\overline{25}$	
	24 · 26 · $\overline{27}$ · $\overline{25}$	
NO09	25 · $\overline{24}$ · 26	25
	27 · $\overline{24}$ · 26	27
NO10	23 · $\overline{22}$ · 26	23
	27 · $\overline{22}$ · 26	27
NO11	23 · 22 · 24	23
	25 · $\overline{22}$ · 24	25
NO12	23 · 25 · $\overline{22}$ · $\overline{24}$	23 · 25
	23 · 25 · $\overline{22}$ · $\overline{26}$	
	23 · 25 · $\overline{26}$ · $\overline{24}$	
	23 · 27 · $\overline{22}$ · $\overline{24}$	23 · 27
	23 · 27 · 22 · 26	
	23 · 27 · 24 · 26	
	25 · 27 · $\overline{22}$ · $\overline{24}$	
	25 · 27 · 22 · 26	
25 · 27 · $\overline{26}$ · $\overline{24}$	25 · 27	



Expected Value of TOP: $\frac{1.947 \cdot 10^{-5}}{(1.794 \cdot 10^{-5})}$

Fig. 14 Fault Tree 2. Modularization with two Supercomponents in Cascade. Calculation of the Occurrence Probability of TOP.

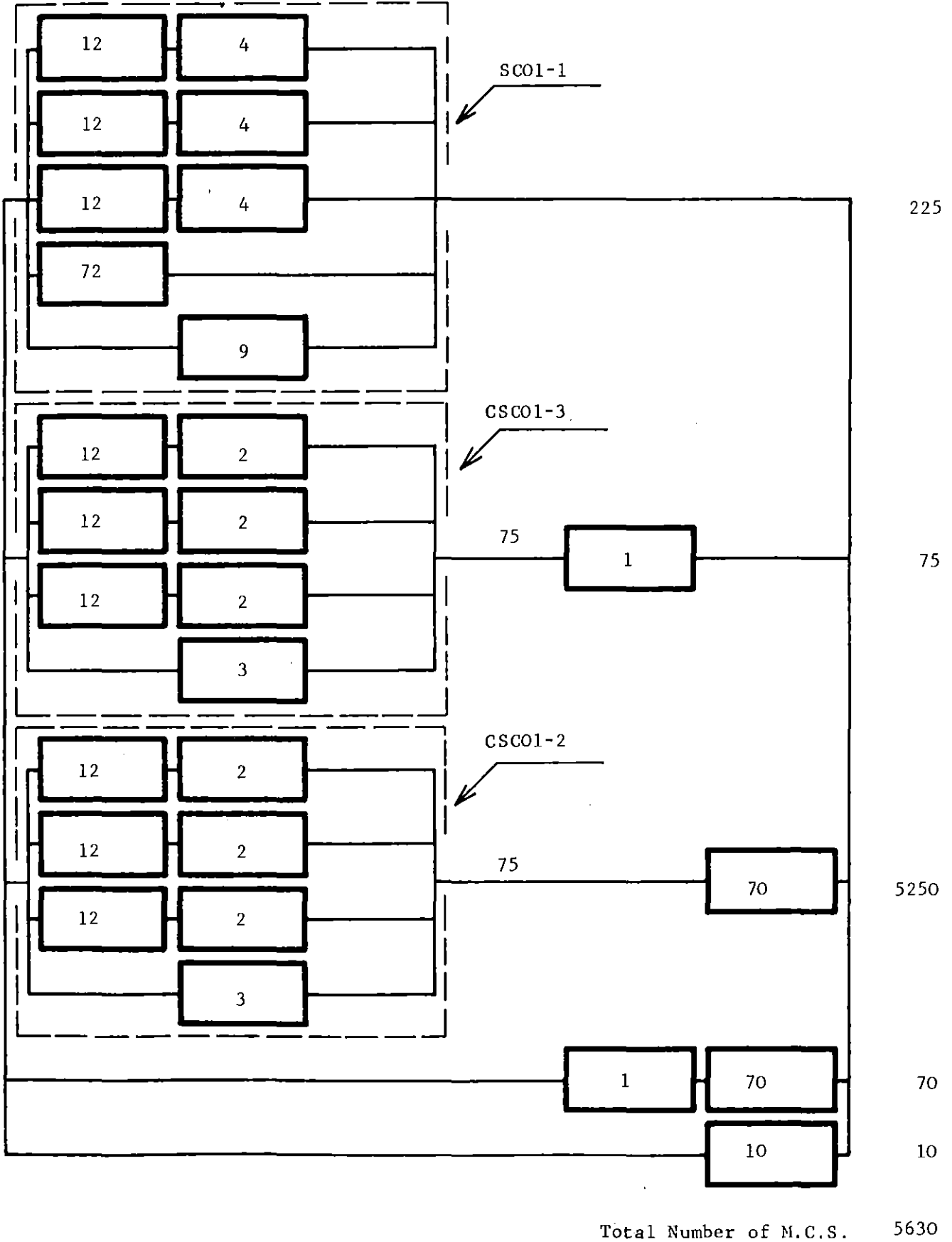


Fig. 15 Fault Tree 2. Modularization with two Supercomponents in Cascade. Calculation of the Total Number of Minimal Cut Sets (M.C.S.)

4. A second example

Fig. 16 shows a fault tree of a part of a reactor protective system. We call this fault tree: fault tree 3. All components of fault tree 3 are binary with the exception of P1, P2 and P3 which have each three states. Only the failed states P1-1 to P3-1 and P1-2 to P3-2 are present in the fault tree.

Table 9 gives the expected values of the primary variables. These expected values have been changed by orders of magnitudes from the original true values. Fault tree 3 has more than 10^{30} cut sets, about $1.1 \cdot 10^7$ of them being minimal cut sets (m.c.s.).

The two linear groups of gates C1, C2, E1, E2 and G1, G2, K1, K2 are both logically independent. We can define therefore two supercomponents, namely SC01 and SC02 each having $2^4 = 16$ states.

Since the fault tree is symmetrical with respect to the two supercomponents, we need to analyse only the first (SC01), the analysis of the second (SC02) being equivalent.

Table 10 shows the compositions of each state of supercomponent SC01. We note that the 7 variables SC02-9 to SC02-15 are equal to zero. This is found also automatically by MUSTAMO. For this reason the number of states of SC01 reduces to 9.

MUSTAMO breaks down the fault tree 3 into 17 fault trees. The main fault tree is shown in Fig. 17. MUSTAMO calculates the fault tree of Fig. 17. The solution is shown in the block diagram of Fig. 18. It is important that only four non zero variables of the supercomponents SC01 and SC02 are present in the solution. They are

- SC01 - 1
- SC01 - 2
- SC01 - 4
- SC01 - 5
- SC02 - 1
- SC02 - 2
- SC02 - 4
- SC02 - 5

The fault trees of the above variables are shown in Fig. 19 (SC01) and in Fig. 20 (SC02). The fault tree of module M046 is shown in Fig. 21. Fig. 22 shows the common structure of the fault trees of the modules M047 to M061. The composition of the variables U; V and Z for each module are given in the table in the same Fig. 22.

Fig. 23 shows the block diagram of fault tree 3 with the expected values of each module. The occurrence probability of the TOP event is $1.12 \cdot 10^{-8}$. Fig. 24 shows the total number of m.c.s. of the associated coherent functions of each module. The total number of m.c.s. of the fault tree is $1.1220036 \cdot 10^7$.

The CPU time for the complete analysis of fault tree 3 was 71 secs.

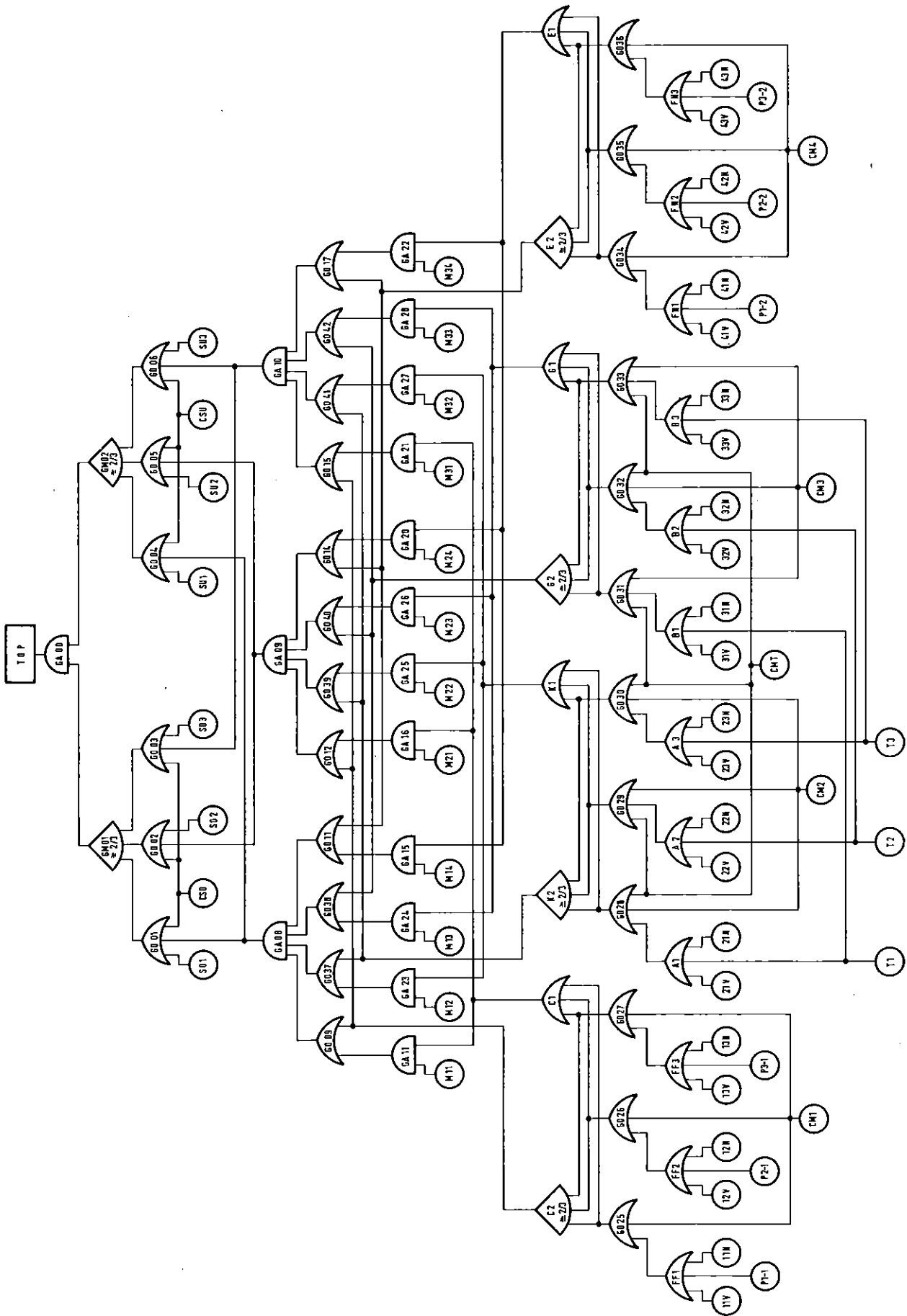


Fig. 16: Fault Tree 3

Table 9

Fault tree 3. Expected Values of the Primary Variables

Primary Variable	Expected Value
From M11 to M14	8.2569 · 10 ⁻²
From M21 to M24	8.2569 · 10 ⁻²
From M31 to M34	8.2569 · 10 ⁻²
From SU1 to SU3	2.5933 · 10 ⁻³
From SO1 to SO3	2.5933 · 10 ⁻³
CSO and CSU	5.9996 · 10 ⁻⁵
From 11V to 13V	9.99 · 10 ⁻⁴
From 21V to 23V	3.9984 · 10 ⁻⁴
From 41V to 43V	1.996 · 10 ⁻³
From 11N to 13N	9.901 · 10 ⁻³
From 21N to 23N	9.901 · 10 ⁻³
From 31N to 33N	1.9608 · 10 ⁻²
From 41N to 43N	9.901 · 10 ⁻³
From T1 to T3	2.991 · 10 ⁻³
From CM1 to CM4	10 ⁻³
From P1-1 to P3-1	9.98 · 10 ⁻⁴
From P1-2 to P3-2	9.98 · 10 ⁻⁴
From 31V to 33V	3.9984 · 10 ⁻⁴
CMT	10 ⁻³

Table 10

Fault Tree 3. Supercomponent SCO 1

STATE	COMPOSITION	COMMENT
1	$C1 \cdot C2 \cdot E1 \cdot E2$	Present in the result
2	$C1 \cdot \overline{C2} \cdot E1 \cdot E2$	Present in the result
3	$\overline{C1} \cdot \overline{C2} \cdot E1 \cdot E2$	
4	$C1 \cdot C2 \cdot E1 \cdot \overline{E2}$	Present in the result
5	$C1 \cdot \overline{C2} \cdot E1 \cdot \overline{E2}$	Present in the result
6	$\overline{C1} \cdot \overline{C2} \cdot E1 \cdot \overline{E2}$	
7	$C1 \cdot C2 \cdot \overline{E1} \cdot \overline{E2}$	
8	$C1 \cdot \overline{C2} \cdot \overline{E1} \cdot \overline{E2}$	
9	$\overline{C1} \cdot C2 \cdot E1 \cdot E2$	Z E R O
10	$C1 \cdot C2 \cdot \overline{E1} \cdot E2$	Z E R O
11	$\overline{C1} \cdot C2 \cdot \overline{E1} \cdot E2$	Z E R O
12	$C1 \cdot \overline{C2} \cdot \overline{E1} \cdot E2$	Z E R O
13	$\overline{C1} \cdot \overline{C2} \cdot \overline{E1} \cdot E2$	Z E R O
14	$\overline{C1} \cdot C2 \cdot E1 \cdot \overline{E2}$	Z E R O
15	$\overline{C1} \cdot C2 \cdot \overline{E1} \cdot \overline{E2}$	Z E R O
0	$\overline{C1} \cdot \overline{C2} \cdot \overline{E1} \cdot \overline{E2}$	

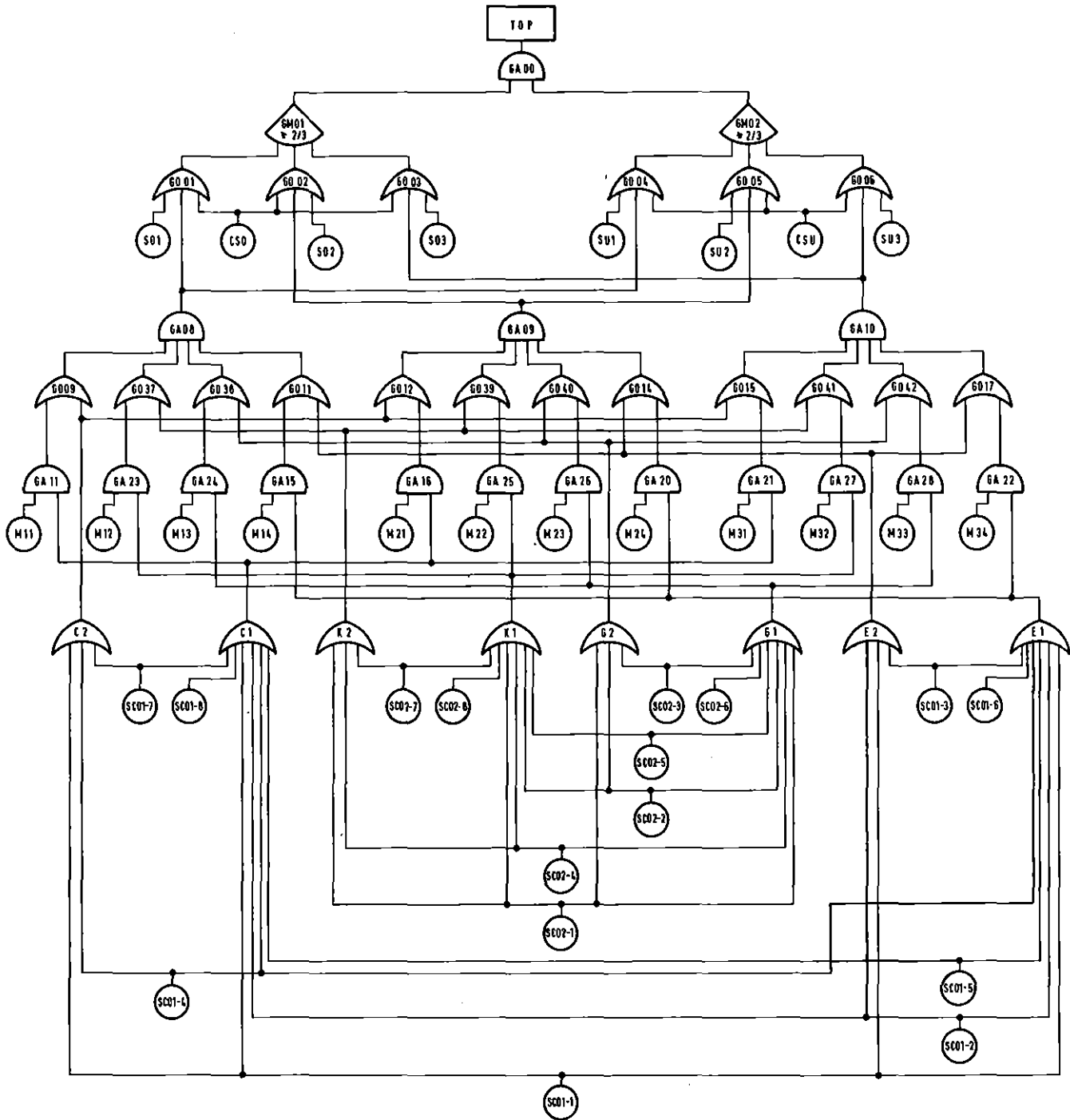


Fig. 17: Fault Tree 3. Main Fault Tree after Break Down.

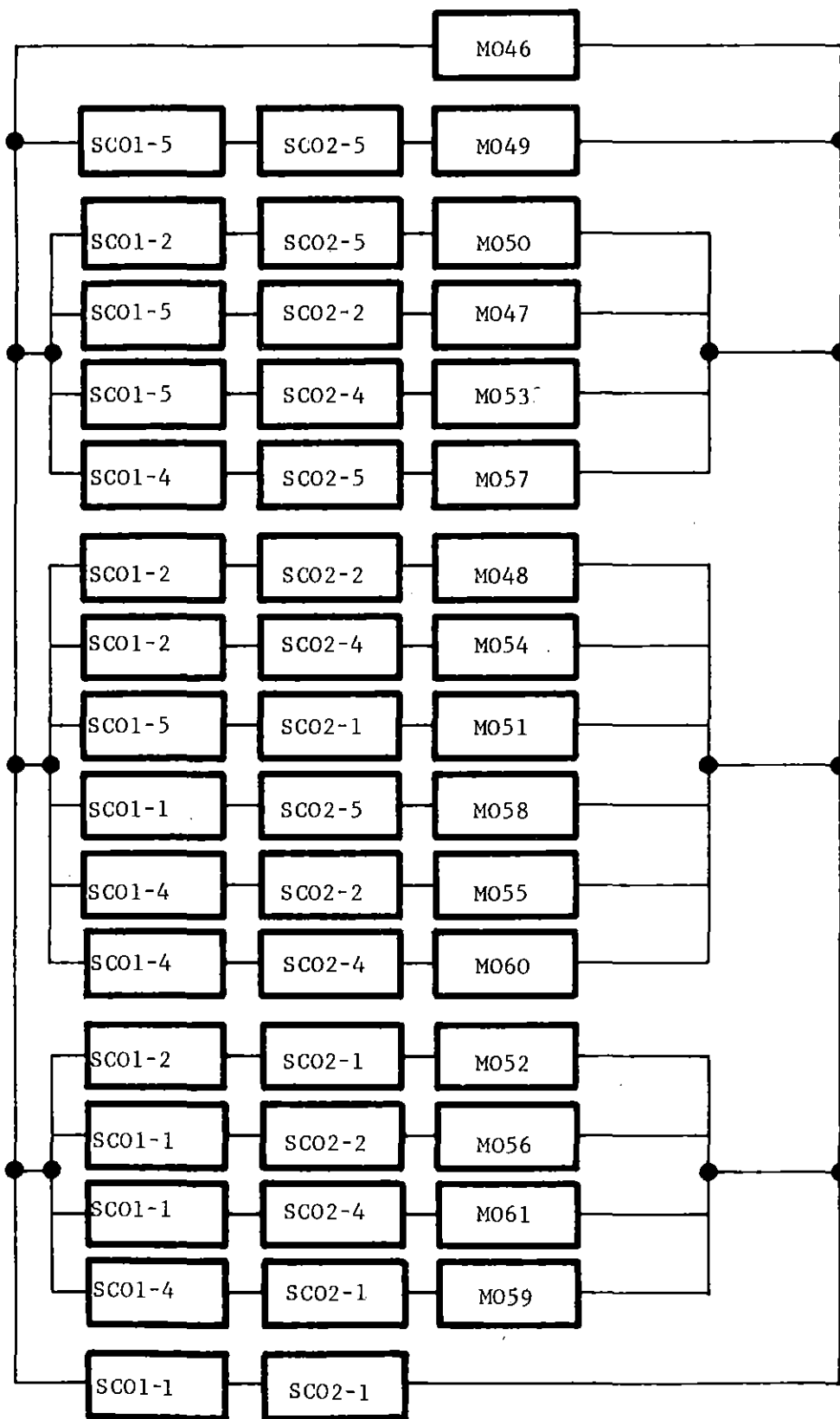


Fig. 18 Fault Tree 3. Block Diagram

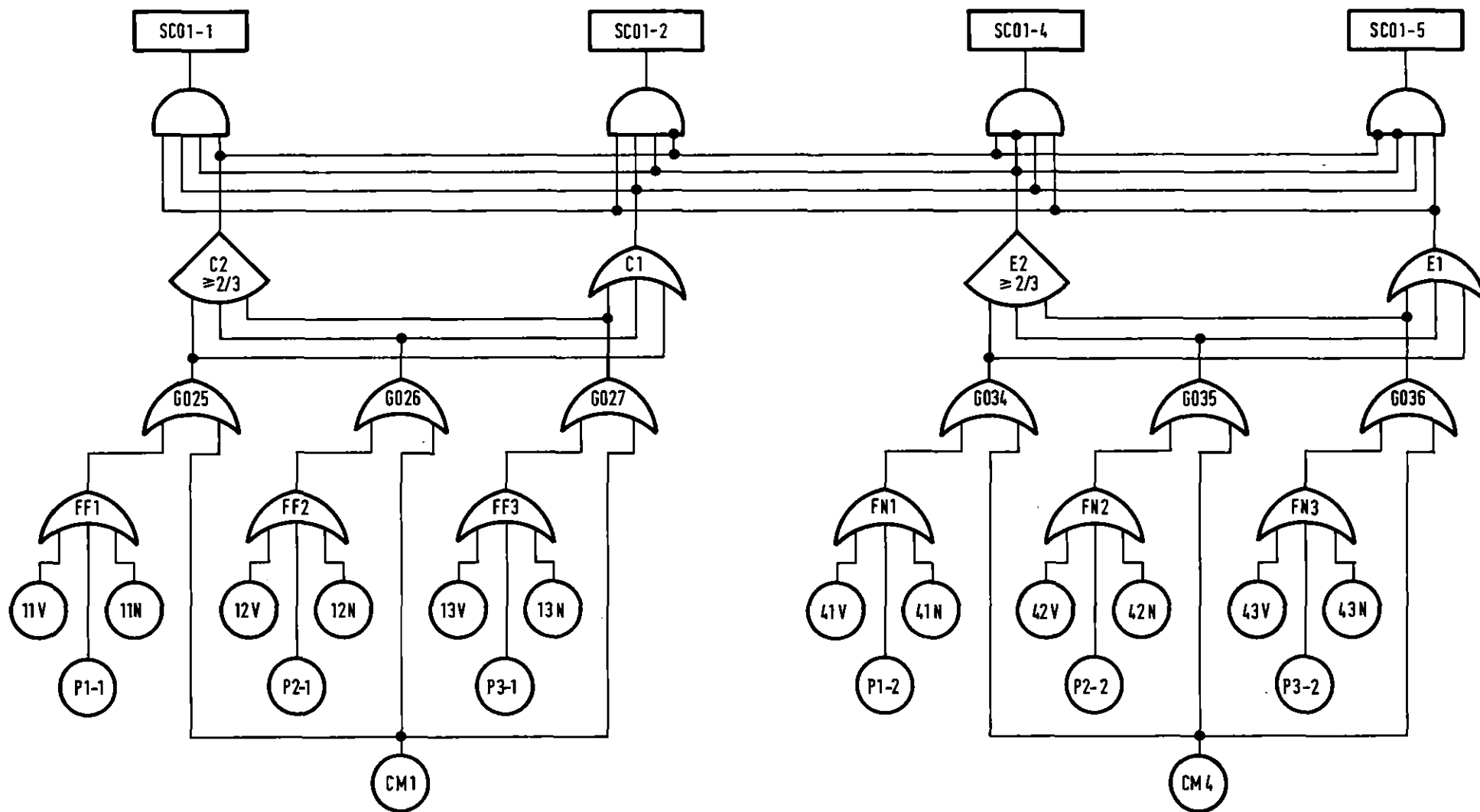


Fig. 19: Fault Tree 3. Fault Trees of the Variables of Supercomponent SC01.

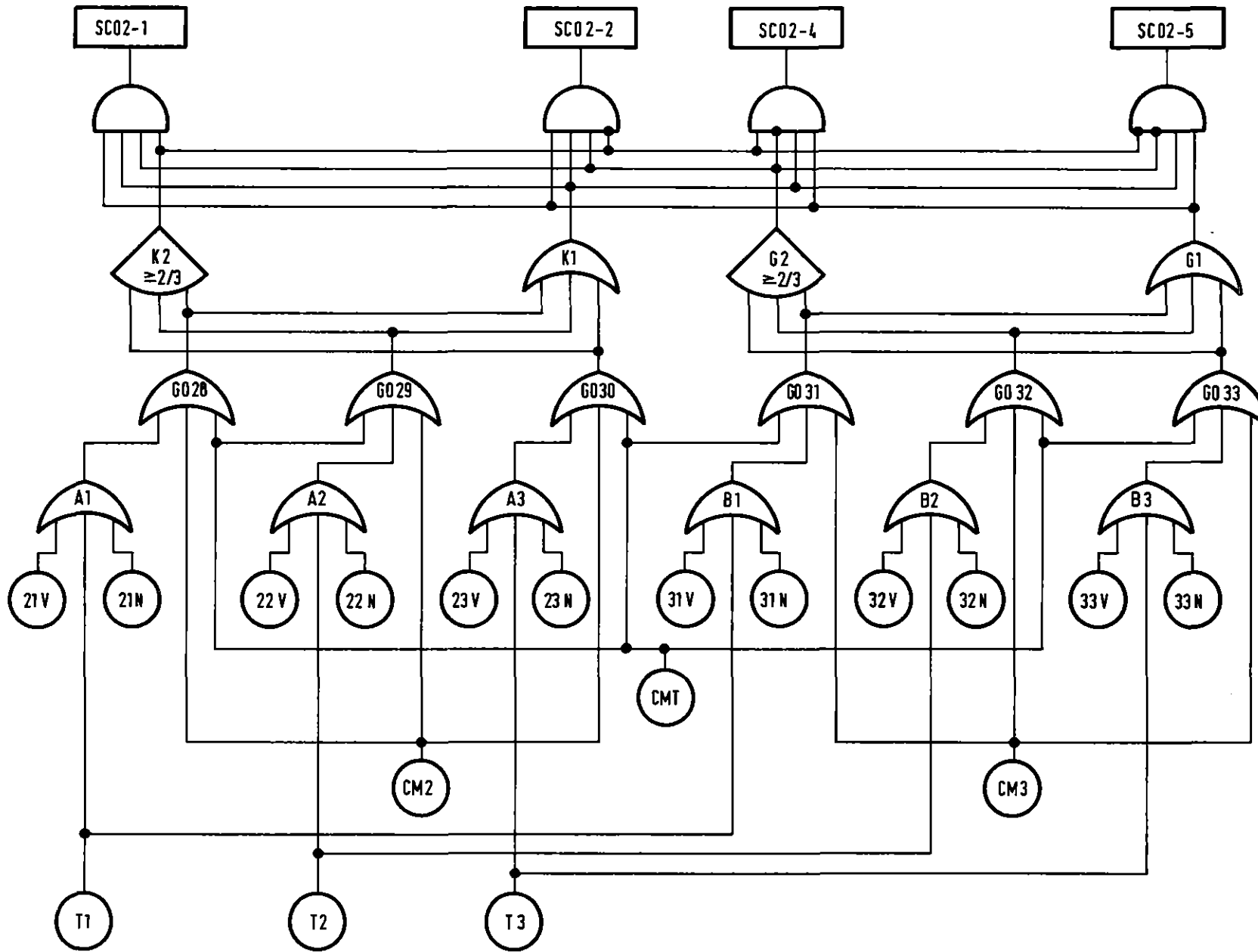


Fig.20: Fault Tree 3. Fault Trees of the Variables of Supercomponent SC02.

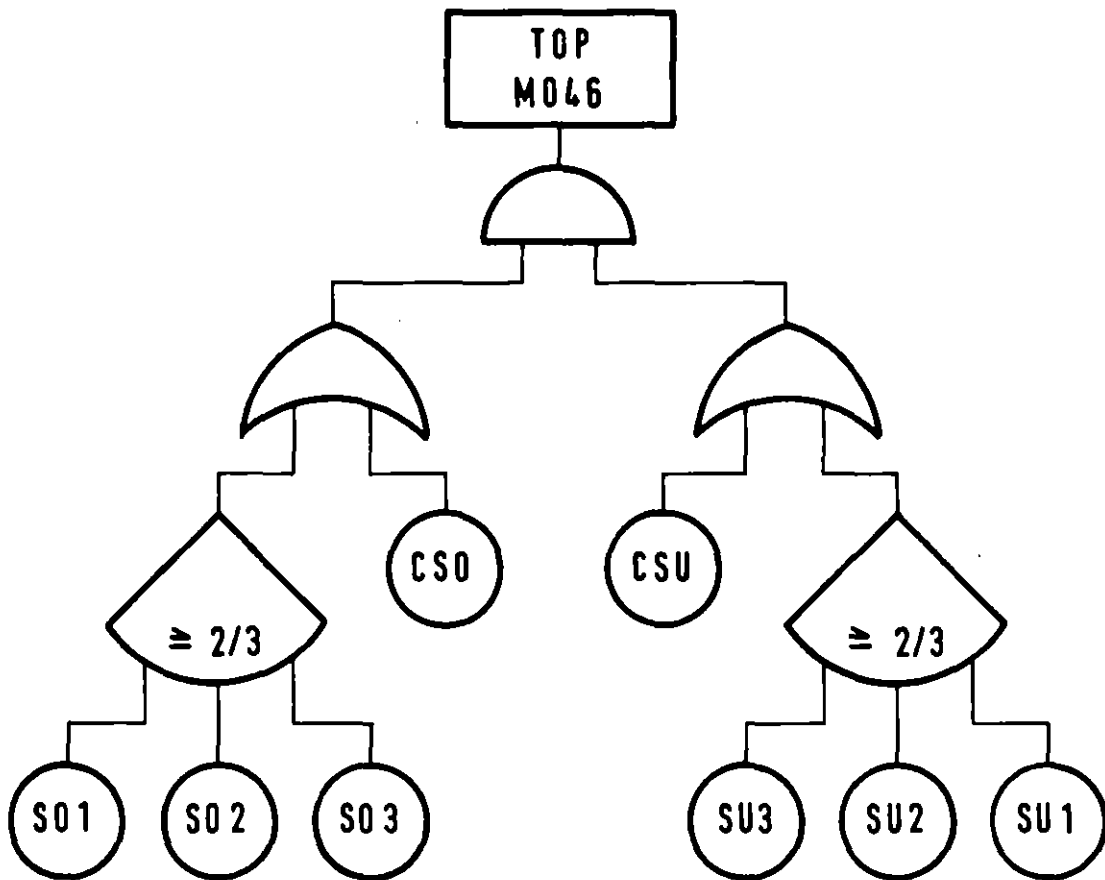
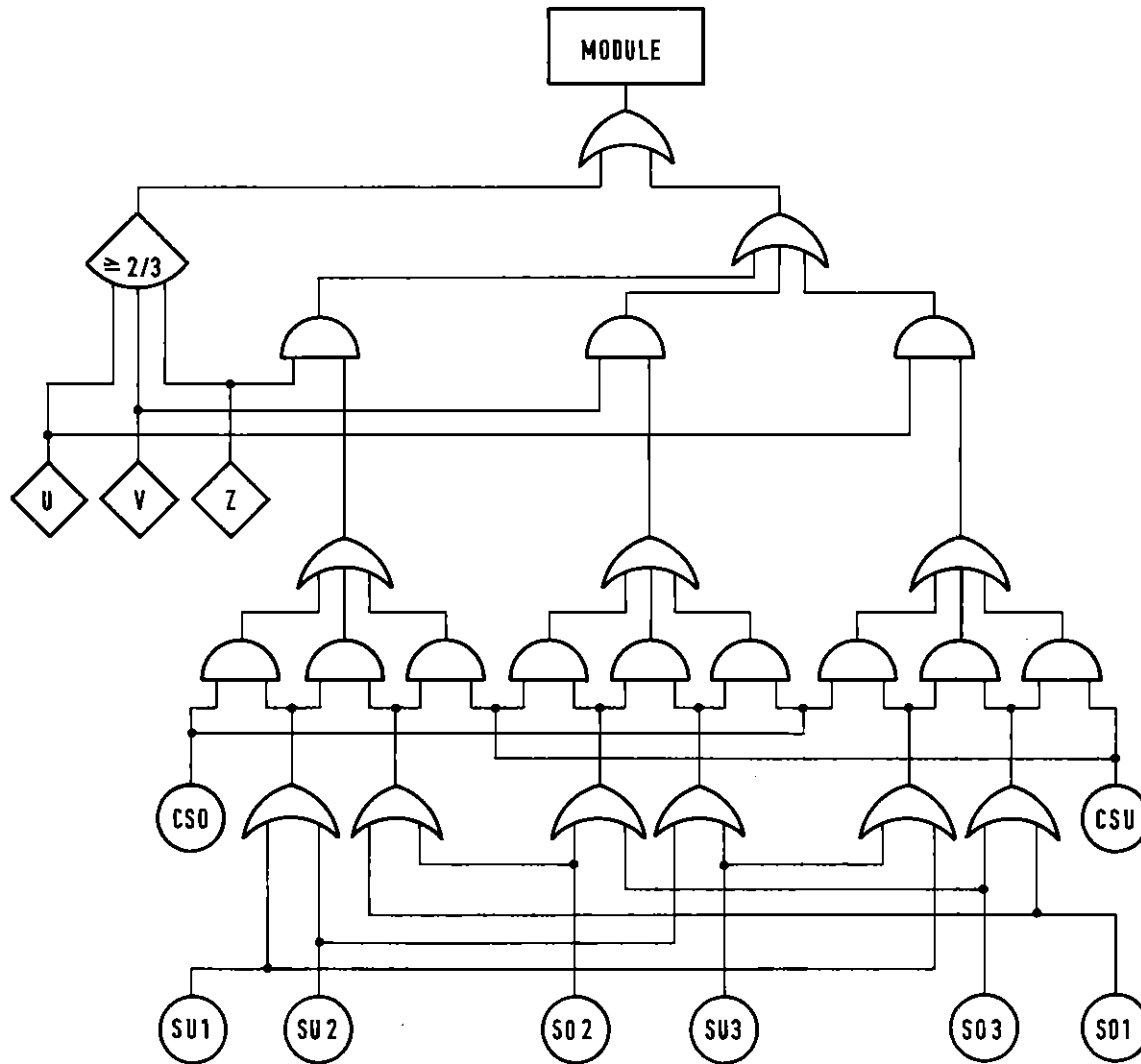


Fig. 21: Fault Tree 3.
Fault Tree of Module M046



MODULE	V	U	Z
M0 52	M11	M21	M31
M0 56	M12	M22	M32
M0 61	M13	M23	M33
M0 59	M14	M24	M34
M0 48	M11·M12	M21·M22	M31·M32
M0 54	M11·M13	M21·M23	M31·M33
M0 51	M11·M14	M21·M24	M31·M34
M0 58	M12·M13	M22·M23	M32·M33
M0 55	M12·M14	M22·M24	M32·M34
M0 60	M13·M14	M23·M24	M33·M34
M0 50	M11·M12·M13	M21·M22·M23	M31·M32·M33
M0 47	M11·M12·M14	M21·M22·M24	M31·M32·M34
M0 53	M11·M13·M14	M21·M23·M24	M31·M33·M34
M0 57	M12·M13·M14	M22·M23·M24	M32·M33·M34
M0 49	M11·M12·M13·M14	M21·M22·M23·M24	M31·M32·M33·M34

Composition of the Variables V, U and Z

Fig. 22: Fault Tree 3. Fault Tree of the Modules M0 47 to M0 61

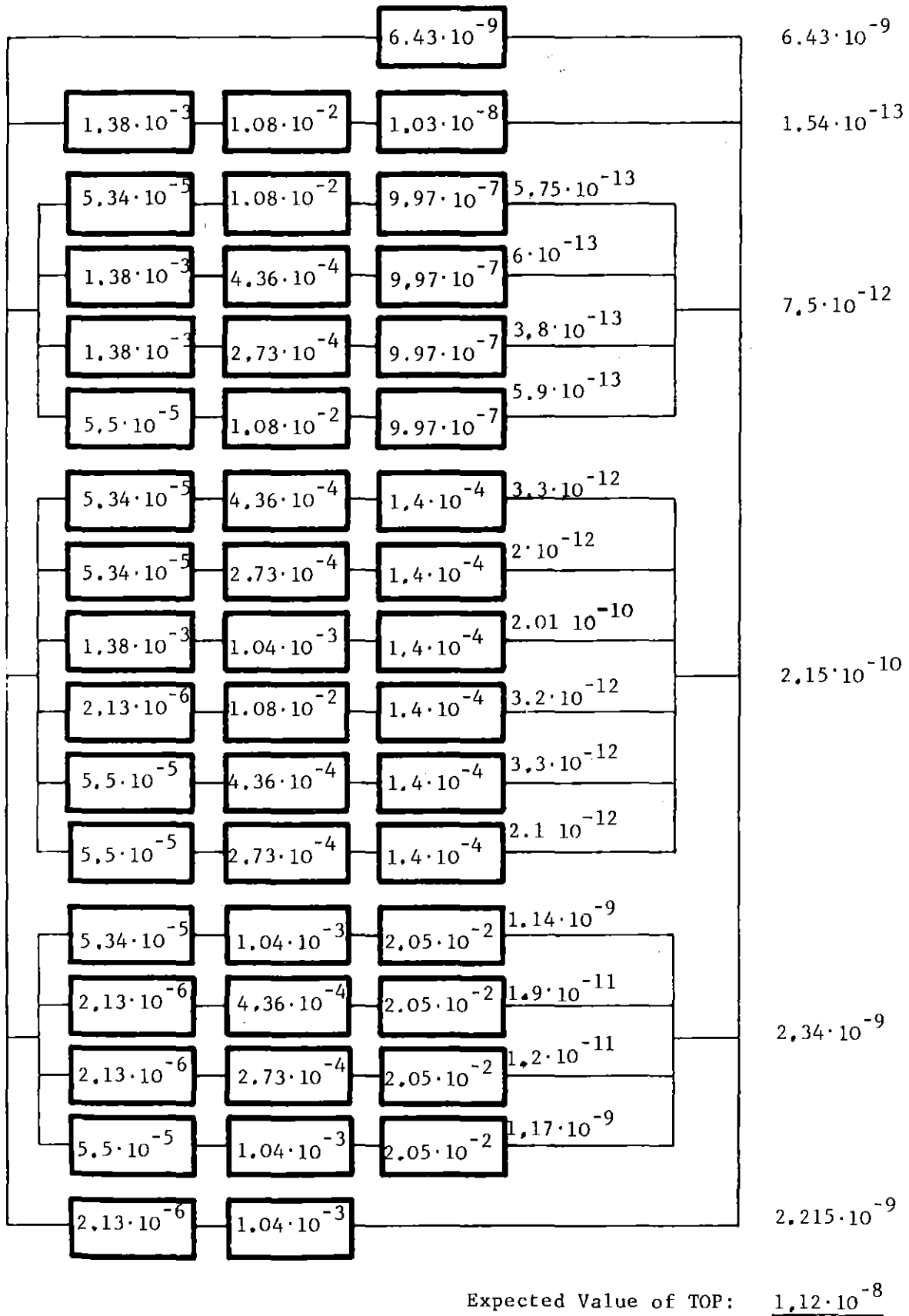


Fig. 23 Fault Tree 3. Calculation of the Occurrence Probability of the TOP

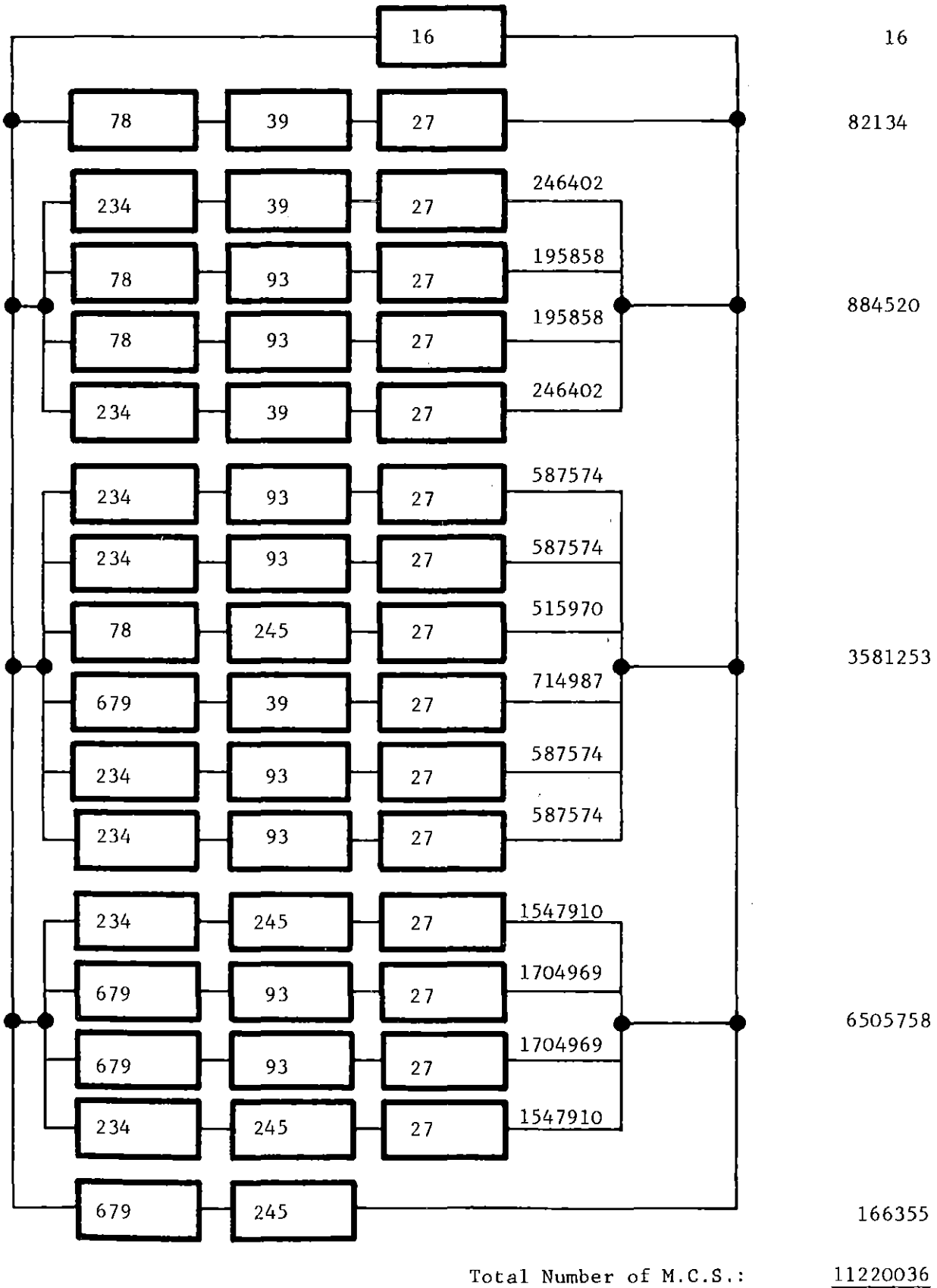


Fig. 24 Fault Tree 3. Calculation of the Total Number of Minimal Cut Sets (M.C.S.).

5. Conclusions

The number of minimal cut sets (m.c.s.) of very complex and highly interconnected fault trees can become extremely large (e.g. more than 10^7). In this case the usual analytical approach of dissecting the fault tree TOP variable into m.c.s. is not only computationally prohibitively expensive, but also meaningless because it does not offer any synthetic overlook of system behavior.

To emphasize this last point, a stack of paper 21 meters high would be required to print out all m.c.s. of fault tree 3 ($\sim 1.1 \cdot 10^7$ m.c.s.) from section 4. This is equivalent to the height of a six story building.

The above deficiencies were also pointed out in the german risk study /11/, where simulation methods were preferred to analytical methods.

The method suggested in this paper also overcomes the deficiencies of the analytical methods. By applying boolean algebra with restricted variables (b.a.w.r.v.), the concept of fault tree modularization can be straightforwardly extended from a single gate to a set of gates. Thus, large fault trees are divided into smaller fault trees (modules), which are connected to each other according to a simple scheme. This scheme is represented by a block diagram in which each block is a module. The modules are analyzed separately by the m.c.s. method, and the results are combined according to the block diagram connections to calculate the occurrence probability of the TOP event.

The method offers the following advantages:

1. Calculation of very large and highly interconnected fault trees within a reasonable computing time.

For example the CPU time on an IBM 3033 for the complete analysis of the already mentioned fault tree 3 was 71 secs.

2. A synthetic overview of system behavior.

Each block of the block diagram physically represents a failure mode of a part of the system (subsystem). The contribution of each subsystem failure mode to the occurrence probability of the TOP event can be read from the block diagram.

3. Calculation of the complete boolean function of the TOP variable in a compact form.

This is important for the following reasons:

- (a) Two or more fault trees of the same system can be compared at the boolean level in order to determine whether or not they are identical.

The comparison among different reliability analyses of the same system must be carried out not only at the level of probabilities (as it is usually done) but also at the level of events. In fact two TOP events, although they are different, could have the same occurrence probability. On the other hand two fault trees of the same system, although they look different, may be equal.

The problem of comparison among fault trees is becoming important because the confidence in the reliability analyses of systems will increase if the analyses are carried out by different and independent organizations.

- (b) For sensitivity studies the boolean calculation needs only be made once. The same holds for the evaluation of the confidence intervals of the TOP event occurrence probability.
- (c) Potential application to "bn line failure diagnosis". Here, in particular, a complete, clear and synthetic representation of system faults is required.

The analysis of fault tree 2 (section 3) with two supercomponents in cascade has shown that the most convenient supercomponent is not always that which has 2^n states, where n is the number of gates in the selected group. Efforts must be directed to find out more general rules for the definition of the most appropriate supercomponents.

Another interesting point for further developments is the removal of logical independence as a necessary condition for applying the method. The method being developed at Karlsruhe handles also linear groups of gates which are weakly logically dependent. The internal and external territories of a weakly logically dependent group of gates are not disjoint. They have only very few components in common.

6. References

1. W.E. Vesely, 1970, "A time dependent methodology for fault tree evaluation", Nucl. Eng. Des. 13, 337-360.
2. L. Caldarola, 1977, "Unavailability and failure intensity of components", Nucl. Eng. Des. 44, 147-162.
3. L. Caldarola, A. Wickenhäuser, 1977, "The Karlsruhe computer program for the evaluation of the availability and reliability of complex repairable systems", Nucl. Eng. Des. 43, 463-470.
4. K. Kotthoff, W. Otto, 1976, "Vergleich von Rechenprogrammen zur Zuverlässigkeitsanalyse von Kernkraftwerken", IRS-RS 172.
5. L. Caldarola, A. Wickenhäuser, 1977, "Recent Advancements in fault tree methodology at Karlsruhe", International Conf. on Nucl. Systems Reliability Engineering and Risk Assessment, Gatlingburg, SIAM, 518-542.
6. L. Caldarola, 1978, "Fault tree analysis of multistate systems with multistate components", ANS Topical Meeting on Probabilistic Analysis of Nuclear Reactor Safety, Los Angeles, California, Paper VIII.1, May 1978.
7. L. Caldarola, 1980, "Grundlagen der Booleschen Algebra mit beschränkten Variablen", KfK 2915.
8. L. Caldarola, 1980, "Generalized fault tree analysis combined with state analysis", KfK 2530.
9. G. Apostolakis, S. Garriba and G. Volta, 1980, "Synthesis and Analysis Methods for Safety and Reliability Studies", Plenum Press, 106.
10. L. Caldarola, 1980, "Coherent systems with multistate components", Nuclear Engineering and Design 58, 127-139.
11. Deutsche Risikostudie, 1979 - Kernkraftwerke, Fachband 2/1, Zuverlässigkeitsanalyse - Verlag TÜV Rheinland, 22.

12. P. Chatterjee, 1975, "Modularization of fault trees: a method to reduce the cost of analysis", Reliability and Fault Tree Analysis edited by R.E. Barlow and J.B. Fussell, SIAM, 101-126.
13. J. Olmos, L. Wolf, 1978, "PL - MOD - A computer code for modular fault tree analysis and evaluation", Proceedings of the ANS Topical Meeting on Probabilistic Analysis of Nuclear Reactor Safety, Los Angeles, California, Paper XIII - 4.1.
14. A.E. Green and A.J. Bourne "Safety assessment with reference to automatic protective systems for nuclear reactors" (in 3 parts), AHSB (S) R 117.
15. A. Cross, "Private Communication".
16. L. Caldarola, H. Schnauder and A. Wickenhäuser (in preparation).