ECSS-Q-40-04A Part 1

14 October 1997

EUROPEAN COOPERATION

ECSS

FOR SPACE STANDARDIZATION

Space Product Assurance

Sneak analysis - Part 1: Method and procedure

ECSS Secretariat
ESA-ESTEC
Requirements & Standards Division
Noordwijk, The Netherlands

Published by:	ESA Publications Division ESTEC, P.O. Box 299, 2200AG Noordwijk, The Netherlands
ISSN:	1028-396X
Price:	DFl 35

Printed in the Netherlands

Foreword

This standard is one of the series of ECSS standards intended to be applied together for the management, engineering and product assurance in space projects and applications. ECSS is a cooperative effort of the European Space Agency, National Space Agencies and European industry associations for the purpose of developing and maintaining common standards.

The application of Sneak Analysis is required by ECSS-Q-40 “Safety”.

This standard has been prepared by editing the ESA Standard PSS-01-411 Issue 1, reviewed by the ECSS Technical Panel and approved by the ECSS Steering Board.

(This page is intentionally left blank)

Contents list

Figures

Tables

1 Scope

The aim of Sneak Analysis is to identify "sneak circuits", i.e. unexpected paths for a flow of mass, energy, data or logical sequence that under certain conditions can initiate an undesired function or inhibit a desired function. Sneak circuits are not the result of failure, but are latent conditions, inadvertently designed into the system.

This standard establishes a procedure for performing sneak analysis and specifies the required output.

The standard is composed of two parts:

- part 1 (i.e. this document ECSS-Q-40-04 Part 1 "Sneak analysis - Part 1: Method and procedure") that contains the method and procedure for performing sneak analysis;
- part 2 (i.e. the document ECSS-Q-40-04 Part 2 "Sneak analysis - Part 2: Clue list") that contains a basic clue list to be used during sneak analysis.

This standard is applicable when the performance of sneak analysis is required by ECSS-Q-40 or by the contract between the customer and the supplier.

Alternative sneak analysis procedures proposed by the supplier may be accepted by the customer provided that equivalence, for the intended application, with the one presented in this standard is shown by the supplier.

(This page is intentionally left blank)

2 Normative references

This ECSS Standard incorporates by dated or undated reference, provisions from other publications. These normative references to the extent specified in the text are cited at the appropriate places and publications are listed hereafter. For dated references, subsequent amendments to or revisions of any of these apply to this ECSS Standard only when incorporated in it by amendment or revision. For undated references the latest edition of the publication referred to applies.

ECSS-P-001	Glossary of terms
ECSS-Q-40-04 – Part 2	Sneak analysis – Part 2: Clue list

(This page is intentionally left blank)

3
Definitions and abbreviations

3.1 Definitions

For the purposes of this standard, the definitions given in ECSS-P-001 apply. In particular, it should be noted that the following terms have a specific definition for use in ECSS standards.

Configuration Item

Failure

Severity

Software

System

The following terms and definitions are specific to this standard and shall be applied.

“**Clue**

A question pointing at a possible way through which design errors associated with one or more items of a system can lead to system malfunction.”

“**Design concern**

The result of a misapplication (or omission of application) of a design requirement or rule to one item of a system.”

“**Design Error**

A misapplication (or omission of application) of one or more requirements (i.e. the ones contained in requirement documents or specifications) or design rules (i.e. the rules that are used by the designers to synthesise a design that meets the design requirements) during the design process.”

NOTE Sneak circuits and design concerns are manifestations of design errors

“**Facilitation Condition**

A combination of states of system components (e.g. interlocks) that enables the condition mentioned in a clue to be triggered.”

“**Sneak Circuit**

An unexpected path for a flow of mass, energy, data or logical sequence that under certain conditions can initiate an undesired function or inhibit a desired function. Sneak circuits are not the result of failures, but are latent conditions, inadvertently designed into the system.

Sneak circuits include: sneak paths, sneak timings, sneak indications and sneak labels.”

“**Sneak indication**

Ambiguous or false display of system operating conditions that can cause the system or an operator to take undesired action.”

“**Sneak label**

Incorrect or imprecise labelling of system functions (e.g. controls, displays) that can cause an operator to apply incorrect stimuli to the system.”

“**Sneak path**

An unexpected path along which mass, energy, data or logical sequence flow in an unintended direction.”

“**Sneak timing**

Occurrences of events in an unexpected or conflicting sequence, or at an unexpected time, or for an unexpected duration. Therefore sneak timings could also occur if mass, energy, data or logical control flow along intended paths without respecting the intended dynamic behaviour of the system.”

“**Source**

An item of a system which contains mass, energy or data.”

“**Target**

An item of a system the unwanted activation or inhibition of which can trigger an undesired event.”

3.2 Abbreviations

The following abbreviations are defined and used within this standard.

Abbreviation	**Meaning**
FMECA	Failure Modes, Effects and Criticality Analysis
HW	Hardware
I/O Matrix	Input/Output Matrix
PA	Product Assurance
RAMS	Reliability, Availability, Maintainability and Safety
SADT	Structured Analysis and Design Technique
SART	Structured Analysis Real Time
SW	Software

4 Sneak analysis basic principles and application

4.1 Sneak analysis basic concepts

The basic sneak analysis concepts were set up following the observation that system failures can occur as a result of design errors and in the absence of component failures.

A common way to identify design errors is to perform detailed "reviews" of the design. During these reviews, check-lists derived from previous experience are generally used to supplement the reviewer's expertise and to structure the review. However, the results of a given design review are hardly reproducible by a different group of reviewers, since the review is a loose (and creative) process rather than an algorithmic one. To partially compensate for the above deficiencies and improve the effectiveness, both administrative procedures for the performance of the various review phases and analytical techniques are used.

Sneak analysis is a generic term used to indicated a group of analytical techniques employed to methodically identify sneak circuits and design concerns in a system.

Sneak path analysis is a sneak analysis technique that relies on the identification of paths between "targets" and "sources" and the use of clues.

Design concern analysis is a sneak analysis technique that is based only on the application of clues.

Clues are subdivided in the following three classes:

- "path clues", that are used during sneak path analysis and depend only on the kind of causal relation between sources and targets that is under investigation.
  An example is: Can the target be "off" when the source is "on"? That for electrical systems can also be worded as: Can the current coming from the source be diverted away from the target? Annex A explains how the path clues can be derived.
- "component+path" clues, that are dependent on the type of system (electronic, pneumatic, hydraulic, software), are applied to "system components" during Sneak Path Analysis. These clues are derived from experience and are related to those behaviours of a system component that can affect the flow of mass, energy, data or logical sequence between sources and targets. For switches an

example is: During change of state of switches, can transitory current paths exist?

- "component" clues, that are applied to system components during design concern analysis and are also derived from experience. They are dependent on the type of system. For an integrated circuit an example is: Have the maximum frequency conditions been taken into account?

4.2 Sneak analysis basic steps

The procedure for performing sneak analysis is specified in clause 5. In the following, its basic steps are outlined (for a graphical sketch, see Figure 1) in order to give an overview of its key aspects.

The aim of the **preparatory tasks** is to:

- define the analysis scope, i.e. to identify the boundaries and the mission phases of the part of the system that is subject to sneak analysis. For this purpose use should be made of the results of preliminary RAMS analyses such as preliminary hazard analysis and functional failure analysis. The depth of the analysis is also defined during this task;
- gather the data for the subsequent steps of the analysis;
- decompose the design in "blocks" according to the functions of the part of the system under analysis (if this is not already available as output of other RAMS or engineering analyses). The output of this task is used for subdividing the systems in parts that are easily manageable by the analyst and establishing a clear relation between functions and blocks of the design;
- document, in the "input/output matrix", the state of the functional inputs and outputs of the part of the system under analysis during the planned operational modes (if this is not already available as output of other RAMS or engineering analyses). This matrix is useful to screen out some paths during the path tracing.

The actual s**neak analysis** consists of:

- the sneak path analysis, the aim of which is to identify sneak paths, sneak timings and sneak indications through: identification of targets; identification of sources; tracing of paths between sources and targets; application of "component+path" clues to the components contained in the path;
- the design concern analysis, the aim of which is to identify sneak labels and design concerns through application of "component" clues;
- the assessment of the consequences of sneak circuits and design concerns up to the highest level of design decomposition that is of interest.

Finally, sneak circuits and design concerns are documented on "sneak circuit reports" together with the recommendations to eliminate them and a "sneak analysis final report" is produced that documents input data, interim results and conclusions.

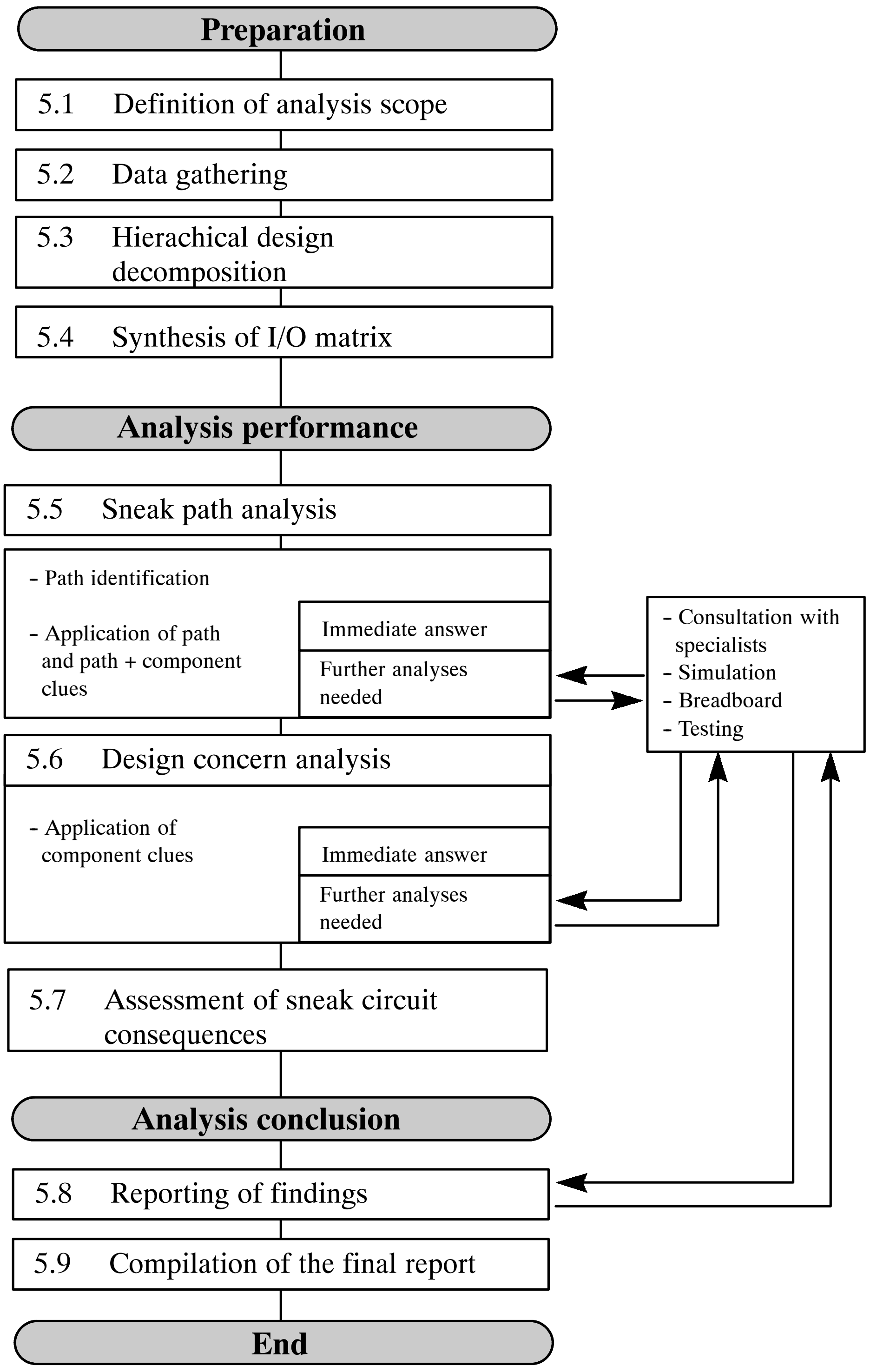

Preparation
5.1 Definition of analysis scope
5.2 Data gathering
5.3 Hierachical design decomposition
5.4 Synthesis of I/O matrix
Analysis performance
5.5 Sneak path analysis
- Path identification
- Application of path and path + component clues
Immediate answer
Further analyses needed
- Consultation with specialists
- Simulation
- Breadboard
- Testing
5.6 Design concern analysis
- Application of component clues
Immediate answer
Further analyses needed
5.7 Assessment of sneak circuit consequences
Analysis conclusion
5.8 Reporting of findings
5.9 Compilation of the final report
End

Figure 1: Outline of sneak analysis procedure

4.3 Input data for sneak analysis

The following data can be considered as inputs for sneak analysis:

System Level

- System requirements (including external interface requirements)
- System design
- Internal interface definition (including HW/SW interfaces)

- User manual (including operation procedures)
- Results of preliminary RAMS analyses (e.g. Functional failure analysis, preliminary hazard analysis)
- Results of functional analysis

Lower Level (for Hardware)

- Subsystem requirements
- Subsystem design
- Equipment requirements
- Equipment design (including drawings)
- Component specification (data sheets)
- (✖) Results of worst case analysis
- (✖) Results of part stress analysis
- (✖) Development testing results

Lower Level (for Software)

- User requirements
- Software requirements
- Architectural design
- Detailed design
- (✖) Results of software verification and validation activities

According to the scope of the analysis and its depth some of the above data might not be relevant.

The data marked with "(✖)", when available, might be useful to avoid duplication with other analyses/activities. However they do not generally contain the raw input data (e.g. requirements, drawings) for sneak analysis as the other listed documents do.

Most of the above quoted raw input data are generally contained in the system (subsystem, equipment) design specification and (for software) in the source code and detailed design.

For example, some of the inputs useful for sneak analysis that can be found in an (electrical) equipment design specification are:

- product description;
- top level diagram;
- functional characteristics (e.g. functions of each board);
- limitations (e.g. lifetime);
- external electrical interfaces;
- internal electrical interfaces;
- electrical schematics (including interface circuits);
- technical characteristics (not those required but those really implemented in the design, e.g. power line protection, grounding);
- parts list.

In the software detailed design the following data are of use for sneak analysis:

- software architecture describing the software decomposition in functions, their inter-relationships and sequencing;
- for each software item: function, subordinates, dependencies, interfaces, resources, processing, data;
- source code listing.

4.4 Sneak analysis application guidance

When planning the application of sneak analysis, it is important to take into account the following factors:

a. expertise required;

b. availability of computerised tools;

c. "delta" analysis due to design changes;

d. the characteristics of the application domain.

4.4.1 Expertise

It is important that the analysis team contain at least one design specialist in the domain (e.g. electrical, electronics) of the system to be analysed . In any case, a discussion on the preliminary findings of the sneak analysis is needed between the analysis team and the designers of the system concerned in order to screen-out possible false problems raised during the analysis and to synthesise the recommendations for the design changes needed to eliminate the sneak circuits.

4.4.2 Computerised tools

The availability of computerised tools, for performing one or more sneak analysis-related tasks (e.g. manipulation of drawings, identification of paths, application of clues), is useful to reduce the manpower effort needed for the application. In some cases, to resolve specific issues raised during the analysis (e.g. timing problems in digital circuits), either reference to analyses performed by the engineering function through the use of computerised simulators or the performance of some new simulations might be needed. The availability of the input data for the analysis (e.g. electrical schematics, component libraries) in an "electronic" format that is compatible with the one used by the available sneak analysis computerised tools allows to reduce the cost and time needed.

4.4.3 Delta analysis

Interim results of the analysis (e.g. hierarchical decomposition of the design, input/output matrices) should be clearly documented. This can reduce the cost of a "delta" analysis which could be required following changes in the design.

4.4.4 Application domain

The procedure specified in clause 5 has been worded in such a way that application in several domains is possible (provided that the used clue list covers these domains). For use of the procedure in specific domains, the following should be taken into account:

- when applying sneak path analysis to digital systems, the use of digital simulators is recommended in order to be able to tackle the complexity problems. The simulators should have the capability of identifying logic errors and timing problems. The application should be coordinated with the engineering function to avoid duplications;
- the application of sneak analysis to purely software systems (i.e. software without any HW/SW interface) is not recommended when inspections and static and dynamic analyses are already required;
- apply sneak analysis to hardware/software systems after the compliance with semantic and syntax rules of the software language has been checked by the compiler;
- when the system architecture is either very simple or is such complicated (at a low detailed level) that has to be represented in a simplified way (e.g. a system constituted by a couple of microprocessors represented as "black boxes"), then the sneak path analysis is not likely to identify significant problems. Only the design concern analysis should therefore be performed.

Finally, it is noted that sneak analysis is particularly well suited for electrical systems and electronics system composed by discrete components and relatively few integrated circuits.

5 Sneak analysis procedure

The Sneak Analysis procedure specified in this document is composed of tasks that can be grouped into three categories:

- preparation;
- analysis;
- reporting and conclusions.

The relationship between the various tasks is shown in Figure 1. The following pages specify the contents of the various tasks. Each **task description**, after an outline of the objective of the task, contains:

- **inputs**, where the information that is needed for the task is identified;
- **contents**, where the analytical steps that are required to carry out the task are specified;
- **outputs**, where the information that is expected to be produced as output from the task is identified.

5.1 Definition of the analysis scope

The aim of this task is to identify the items of the system and the mission phases that are to be analysed.

5.1.1 Inputs

The following inputs should be used:

a. the requirements on Sneak Analysis contained in the contract (if any);

b. the documents containing the design and operation data for the system concerned (e.g. see 4.3);

c. the results of other RAMS analyses such as preliminary hazard analysis and functional failure analysis;

d. the list of aspects to be considered included in Table 1.

5.1.2 Contents

To identify the items to be analysed, the following steps should be performed.

1. Check whether the Contract contains specific requirements on Sneak Analysis:
   - if yes, perform step 2 below;
   - if not, perform step 3 below.

2. Take into account the requirements contained in the contract (e.g. "Sneak Analysis to be applied to safety critical functions") and by means of the data contained in the documents quoted under 5.1.1.b and the results of the RAMS analyses quoted under 5.1.1.c., define a list of functions and/or items that are to be analysed. Define also the phases of the mission to be considered. Go to step 4.
3. By means of the result of RAMS analyses quoted under 5.1.1.c, identify the list of safety and reliability critical functions. Answer to the questions contained in Table 1 for each of the the items contained in the safety or reliability critical functions (in general a "yes"'answer to a question contained in table increases either the likelihood that the item could contain sneak circuits or the magnitude of the consequences of the manifestation of a sneak circuit). Synthesise the results obtained through the application of the questions contained in Table 1.
4. To define the level of depth of the analysis, take into account the results of step 2 (or 3) and, according to the available documentation, check whether the analysis is to be performed at lower levels (e.g. subsystem, assembly, equipment or component level). This can be done by using at the relevant level the questions of Table 1.

5.1.3 Outputs

The following information shall be identified:

a. the items that are to be analysed;

b. the mission phases to be considered;

c. the level of depth to be reached.

Table 1: Aspects to be considered when defining the analysis scope

Safety and Reliability Consequences

- Does the loss or inadvertent activation of the item lead to catastrophic or critical safety consequences?
- Does the loss or inadvertent activation of the item lead to loss, or considerable degradation, of the mission?

Design aspects: general

- Is testing under all operating modes impossible and/or not planned?
- Is it impossible or difficult to eliminate or control the consequences of a sneak circuit manifestation during system operations?
- Is the item involved in command-control or power functions?
- Is the item interfacing with several other items?
- Has the item several modes of operation?

Design aspects: electrical systems

- Is there functional interaction between the primary power sources?
- Is the "0 Volt" scheme complex?
- Is there functional interaction between the secondary power sources?

Programmatic aspects

- Are there several interfaces manufactured by different suppliers?
- Have many modifications occurred since the beginning of the programme?
- Are many modifications expected?

5.2 Data gathering

The aim of this task is to collect the input data necessary for the performance of a Sneak Analysis.

5.2.1 Inputs

The following inputs should be used:

a. list of items under analysis (see task "definition of analysis scope");

b. level of depth of the analysis (see task "definition of analysis scope");

c. the documents (e.g. see 4.3) containing the design and operations data for the above quoted items;

d. the clue list (see 2.1).

5.2.2 Contents

The following sub-tasks should be performed.

5.2.2.1 Screening of available documentation

Identify the parts of the documents quoted under 5.2.1.c that are relevant to the items under analysis at the required level of depth.

5.2.2.2 Homogeneity control

Check, from a configuration management point of view, the homogeneity of the documents that have been gathered under 5.2.2.1.

5.2.2.3 Familiarisation with the documentation

Become familiar with the documentation screened under 5.2.2.1.

During this process, there might be the need to ask the authors of the documents for clarification.

If not familiar with some key design or technology issues addressed in the documentation, perform a bibliographic research and/or consult "experts" on these issues.

5.2.2.4 Tailoring of the clue list

Taking into account the scope of the application, the list of items under analysis and the level of depth:

- supplement the clues contained in 5.2.1.d with other clues derived from the knowledge available at the Supplier; and/or
- tailor the list of clues contained in 5.2.1.d by discarding the ones that one not relevant for the application in order to improve the efficiency of the analysis.

5.2.2.5 Documentation of the findings

Document the outcome of the previous sub-tasks. In particular, make sure that:

- the points related to lack of documentation homogeneity or the requests for clarification are discussed with the product assurance and engineering personnel;
- the interfaces (between items or between elements of an item) are unambiguously described as pertains to labelling, kind of causality flow (e.g. current, logical control) crossing the interface, specified characteristics of the flow (e.g. current value, rise/fall time), characteristics of the flow that are to be considered during Sneak Analysis (if they are only a subset of the specified characteristics), timing constraints.

Give particular attention to the interfaces between hardware and software items. Identify real time issues, software asynchronous behaviour and constraints on control flow sequence.

If necessary aggregate information spread in several different documents (e.g. for a multi-board electronic equipment, if the data and control interfaces between between different boards are spread over several documents and drawings it might be useful to establish a synthetic sketch showing the above interfaces).

5.2.3 Outputs

The following information shall be identified:

a. the parts of the documentation that are relevant for the analysis;

b. the points where a lack of homogeneity has been identified;

c. the requests for additional information or clarification;

d. the definition of the interfaces of the items that are to be analysed;

e. the clue list to be used for the application.

5.3 Hierarchical design decomposition

The aim of this task is to produce a decomposition into "blocks" of the design of the items that are subject to Sneak Analysis. For each block the (sub)functions performed and the inputs and outputs are documented. This hierarchical decomposition is of use during the following tasks of the analysis because:

- it allows the systems to be subdivided into blocks that are easily manageable (both in terms of size and understandability) by the analysts;
- it establishes a clear cross-reference between the required functions and the actual design;
- it supports the assessment of the consequences of the sneak circuits (see subclause 5.7).

5.3.1 Inputs

The following inputs should be used:

a. the list of items that are included in the scope of the analysis (see task "definition of the analysis scope");

b. the depth of the analysis (see task "definition of the analysis scope");

c. the documentation relevant to the items under analysis (see task "data gathering");

d. the items' interfaces (see task "data gathering").

5.3.2 Contents

If a hierarchical decomposition compatible with the one described in this clause is already available as a result of other Product Assurance or engineering activities (e.g. in the form of SADT, SART, Data flow diagrams), this task should not be performed.

Otherwise the following sub-tasks should be performed on the items within the scope of the analysis.

5.3.2.1 Design decomposition

Take an item and, by looking at the documentation, identify its functions. Identify precisely the kind (e.g. data, electric current) and the origin of its inputs and outputs. Identify the part of the item that is associated mainly with a single item function. Define this part as a "design block". Repeat the previous step for all the items' functions. Once this has been done, identify the interfaces (e.g. in terms of data, electric current) between the various blocks. To avoid confusion, use the names of the interfaces mentioned in the documentation also in the blocks. Depict the above decomposition in graphical format. If necessary, go to the next level of decomposition and apply the above procedure to each block.

Repeat the above steps until the level that is above the lowest one that is of interest for Sneak Analysis is reached (e.g. if the analysis is to performed at component level for an electronic system, the decomposition is to arrive at board level).

During the stepwise decomposition, the results of the functional analysis (if performed during earlier phases of the programme) can be used to drive the identification of the boundaries of the blocks.

Figures 2A and 2B provide an example of the above decomposition for an electrical system F. This system receives control signals E1, E2, E3, power signals W1 and W2, has return current connections through signals M1, M2, M3 and generates output signals S1, S2, S3 (see Figure 2A). An initial decomposition could lead, for example, to a diagram such as that shown in Figure 2B, where 3 blocks, each associated to a function have been identified.

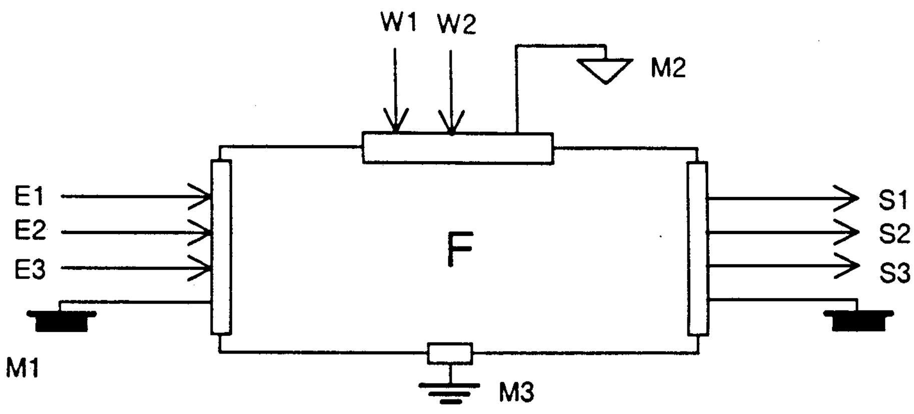
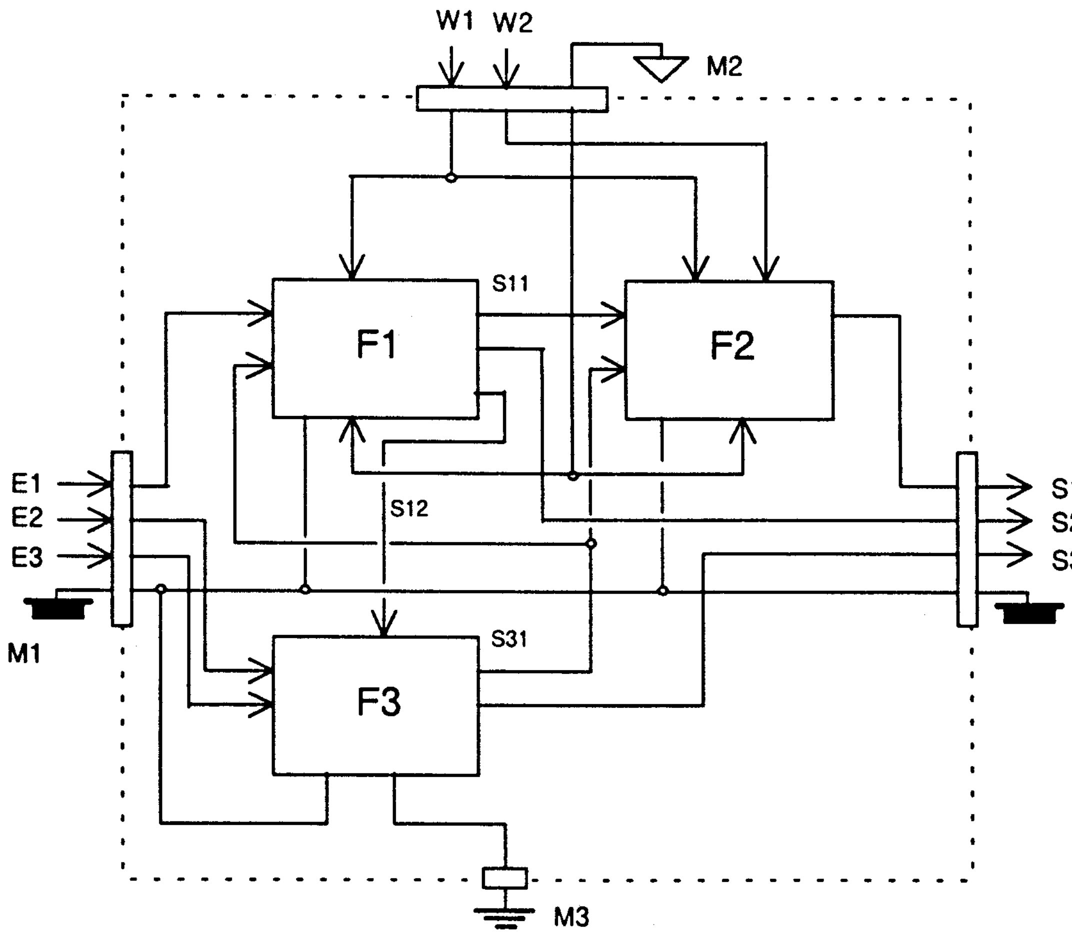


Figure 2A



Legend: In “Snm”, “n” is the block number and “m” is the signal number

Figure 2B

Figure 2: Illustration of design decomposition

5.3.2.2 Documentation of the design decomposition

Document the following information for each block:

- system concerned;
- description of the block;
- block diagram (e.g. see Figure 2B for block F);
- functions performed by the block;
- design characteristic of the block;
- interfaces with other blocks.

5.3.3 Outputs

Through performance of this task , a hierarchical decomposition into blocks of the items to be analysed shall be derived.

5.4 Synthesis of input/output matrix

The "input/output matrix" documents the elementary events (i.e. in this context the changes in the items' inputs) that trigger changes in the items' outputs under consideration.

5.4.1 Inputs

The following inputs should be used:

a. the list of items that are included in the scope of the analysis (see task "definition of the analysis scope");
b. the level of depth of the analysis (see task "definition of the analysis scope");
c. the mission phases that are to be considered for the various items (see task "definition of the analysis scope");
d. the documentation relevant for the items under a. (see task "data gathering");
e. the design blocks associated with the items under consideration (see task "hierarchical design decomposition").

5.4.2 Contents

If the input/output matrix and the operational sequence representation are already available as a result of product assurance or engineering activities in a format that is compatible with the one described in the following, this task should not be performed.

Otherwise the following sub-tasks should be performed.

5.4.2.1 Documentation of operational modes

Identify (by using inputs 5.4.1.c and 5.4.1.d) all the planned operational modes for the item during the mission phases to be analysed.

If simultaneous changes of operational modes for several items are planned, pinpoint them for further consideration in the next tasks (they are possible sources of sneak timings).

5.4.2.2 Identification of elementary switching events

Identify (by using inputs 5.4.1.d and 5.4.1.e) the elementary events that trigger the item outputs. These elementary events are for example:

- (for hardware) on/off commands, power source selection, configuration commands;
- (for software) reconfiguration, memory and/or register initialisation.

5.4.2.3 Input/output matrix at the highest level of design decomposition

Construct the input/output matrix as follows:

- enter in each row head a planned operational mode;
- enter in each column head the name of an item input or output;
- enter in the various matrix entries the state of each input or output for the various operational modes.

An example of the format of the matrix (for the item depicted in Figure 2A) is provided in Table 2.

Table 2: Example of input/output matrix

Input/Output Matrix	On/Off E1	Reset E2	Start E1	+5VDC W1	+40VDC W2	Reset lamp S1	Stand-by lamp S2	Heater S3
Off state	0	0	0	0	0	0	0	0
Reset mode	0	1	0	1	0	1	0	0
Wait mode	1	0	0	1	0	0	1	0
Heater transition	1	0	0	1	1	0	1	0
Heater mode	1	0	1	1	1	0	0	1
Safe mode	1	1	0	1	1	0	1	0

5.4.2.4 Input/output matrix at lower levels of design decomposition

In some cases, it can be useful to build an input/output matrix for some of the lower level blocks (the matrix at these low level of design decomposition is also called "switching matrix" at it documents the state of the "switches" contained in the design). The selection of these blocks is to be done on a case-by-case basis.

The criteria to be taken into account are:

- the complexity of the block architecture;
- the impact of a change in the block outputs on the output of the item.

The switching matrix can be built in a way similar to the one presented in the previous sub-tasks. The operational modes are the same as the ones identified there.

Exercise discipline when identifying switching matrices at low level of design decomposition in order to avoid a combinatorial explosion of the number of entries in these matrices.

5.4.2.5 Operational sequence representation

Using the outcome of subtasks 5.4.2.1 and 5.4.2.2, and the input 5.4.1.d, the sequence of operational modes for each item under analysis shall be identified and documented (e.g. in the form of timelines).

5.4.3 Outputs

Through performance of this task (or retrieval of outputs from other product assurance or engineering tasks) the following information shall be derived:

a. input/output matrix at the first level of design decomposition;

b. identification of instances of simultaneous switching;

c. input/output matrices for some of the lower level blocks (and list of relevant blocks).

d. operational sequence representation for the items under analysis.

5.5 Sneak path analysis

The aim of this task is to identify sneak circuits, mainly sneak paths, sneak timings and sneak indications.

5.5.1 Inputs

The following inputs should be used:

a. the list of items that are included in the scope of the analysis (see task "definition of the analysis scope");

b. the design decomposition of the above items (see task "hierarchical design decomposition");

c. the input/output matrix (see task "synthesis of input/output matrix");

d. the results of the top-level RAMS analyses (e.g. preliminary hazard analysis, functional failure analysis);

e. the documentation relevant for the items under analysis (see task "data gathering");

f. the instance of simultaneous switching of different items (see task "synthesis of input/output matrix");

g. the operational sequence representation (see task "synthesis of input/output matrix").

5.5.2 Contents

5.5.2.1 Identification of the targets

Within the items that are to be analysed, "targets" for the sneak path analysis shall be identified.

This can be done, for each planned operational mode, by identifying the safety or reliability critical outputs that are either required or to be inhibited.

In carrying this step the inputs 5.5.1.b and 5.5.1.d are used.

5.5.2.2 Identification of the sources

The resources (e.g. electrical current) and the associated "sources" (e.g. batteries) that are to be studied in connection with the targets shall be identified.

The dependence (if any) of the sources on the operational modes is to be noted.

5.5.2.3 Identification of the intended and undesired causal relationships ("path clues") between sources and targets

a. For each operational mode, the intended causal relations between the states of the sources and the states of the targets should be identified. Use the data contained in inputs 5.5.1.b and 5.5.1.c for this purpose.

b. Then the unintended causal relations (i.e. by definition all the relations in the source/target space that are different from the intended ones) should be identified.

c. Finally, the undesired causal relations (i.e. the subset of the unintended ones that lead to the loss or the inadvertent activation of the targets) shall be identified.

   It is noted that these "undesired causal relations" are the "path clues" (see also 4.1 and Annex A) that are relevant for the sources and targets under examination.

5.5.2.4 Identification of the resource paths

a. The paths that link the sources to the targets through the system structure (e.g. electrical drawings) should be identified for each operational mode. The analyst can thus follow the flow of the resource on the path.

b. For those parts of the items that have a complex structure, the task may be simplified in some cases by replacing the actual structure by the relevant block (see task "hierarchical design decomposition").

c. The paths should be identified through the following steps (see Figure 3 for an illustration of the process):

1. choose a target;
2. select an undesired relation ("path clue") between a target and one or more sources (e.g. target is "off" when sources are "on");
3. choose a source;
4. select an operational mode;
5. trace all the paths between the target and the source that are compatible with:
   * the input/output matrix;
   * the characteristics of the components between the source and the target (e.g. in electrical systems a diode allows current to flow in only one direction);
   * the undesired relation under consideration (e.g. for targets associated with required functions, the paths that can disconnect the target from the source[s] will be searched. For targets associated with undesired functions, any path that can connect any of the sources to the target will be searched);
6. repeat step 5 until there are no more operational modes.

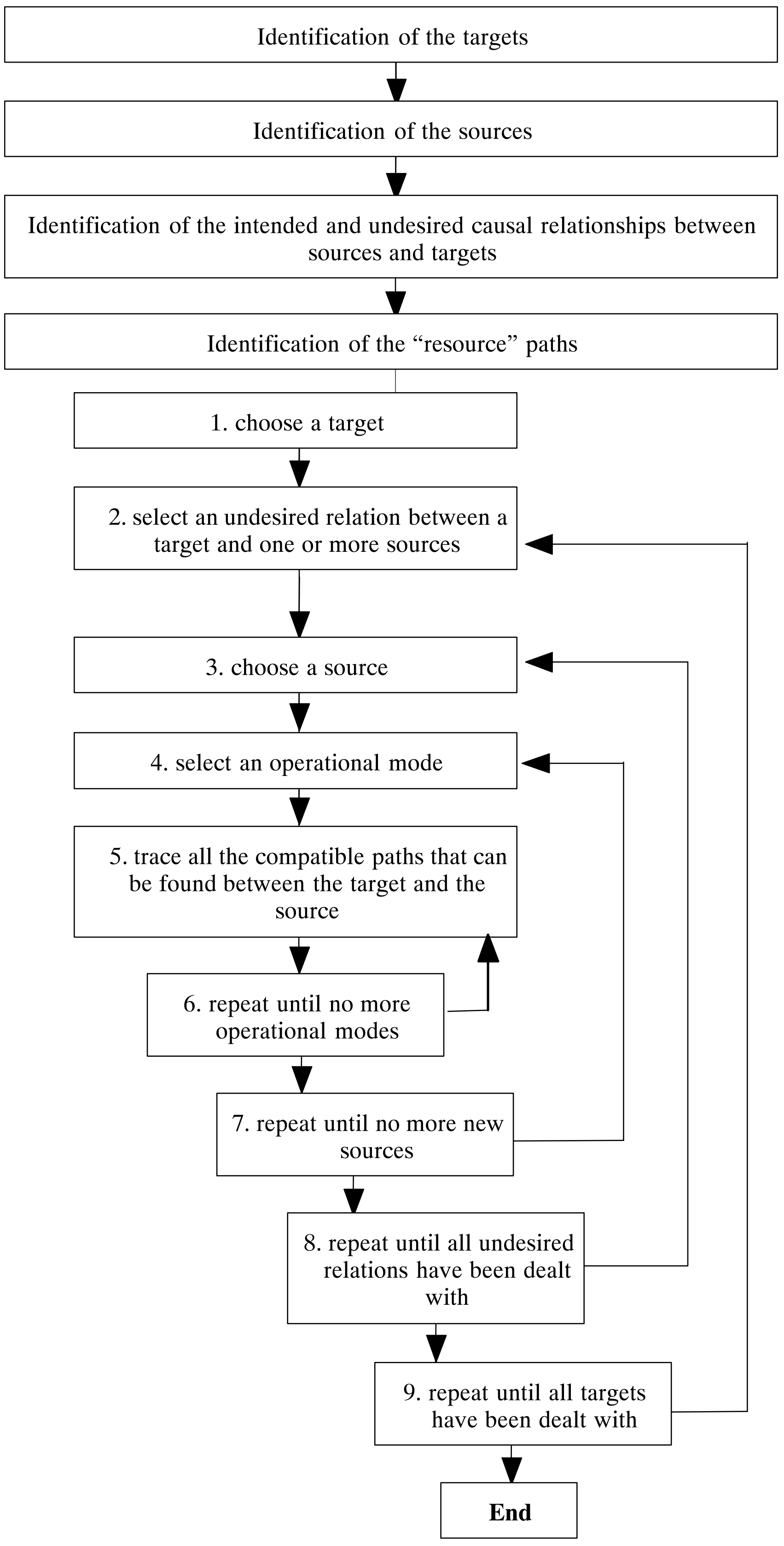

Identification of the targets
Identification of the sources
Identification of the intended and undesired causal relationships between sources and targets
Identification of the "resource" paths
1. choose a target
2. select an undesired relation between a target and one or more sources
3. choose a source
4. select an operational mode
5. trace all the compatible paths that can be found between the target and the source
6. repeat until no more operational modes
7. repeat until no more new sources
8. repeat until all undesired relations have been dealt with
9. repeat until all targets have been dealt with
End

Figure 3: Path identification process

7. select a new source and repeat steps 4 to 6 until all sources have been dealt with. At this stage:
   (a) if the identified paths do not provide a route for the actual occurrence of the undesired relation, go to the next undesired relation (step 8);
   (b) if the paths allow the undesired relation to occur then a potential sneak circuit has been found. The designers shall be consulted to check whether the problem is a real one. If this is the case, the sneak circuit consequences shall be assessed (see task "assessment of sneak circuit consequences");
   (c) if it cannot be decided whether there is a sneak circuit or not, perform the sub-task "Detailed path analysis" (see 5.5.2.5 below). Additionally if there are items on the paths that are controlled by software, perform the "hardware/software path analysis" (see 5.5.2.6 below).
8. repeat steps 2 to 7 until all undesired relations have been dealt with;
9. repeat steps 1 to 8 until all targets have been dealt with.

For manual identification of path, it is useful to have a number of copies of the drawings/diagrams/flow charts on which the various paths can be marked. The availability of a computer program is obviously beneficial for performing the path identification.

5.5.2.5 Detailed path analysis

1. Identify the components on the path under analysis that need to be studied in more detail. For these components the following steps should be repeated (see Figure 4 for an illustration of this process):
2. Pick-up a "component+path" clue;
3. Try and provide a direct answer by:
   * checking the switching matrix; or
   * inspecting the drawings, flow-charts or block diagrams containing the path; or
   * consulting the documentation gathered in the task "data gathering"; or
   * identifying "facilitation conditions", i.e. combinations of states of system components (e.g. interlocks) that enable the condition mentioned in the clue to be triggered. This identification can be done by tracing "facilitation paths" from the "facilitation" components backwards to the components that control them (see Figure 5 for an illustration of this concept); or
   * performing a simple quantitative analysis (bounding calculation).
4. If there is still no answer to the clue, carry out one or more of the following actions:
   * consult experts;
   * perform detailed quantitative analysis;
   * use testing on breadboard or prototype models.
5. If the clue under examination does not lead to a sneak circuit, go to the next clue;
6. If a potential sneak circuit is detected through performance of steps 1 to 5, the designers shall be consulted to check whether the problem is a real one). If this is the case, the sneak circuit consequences shall be assessed (see task "assessment of sneak circuit consequences").

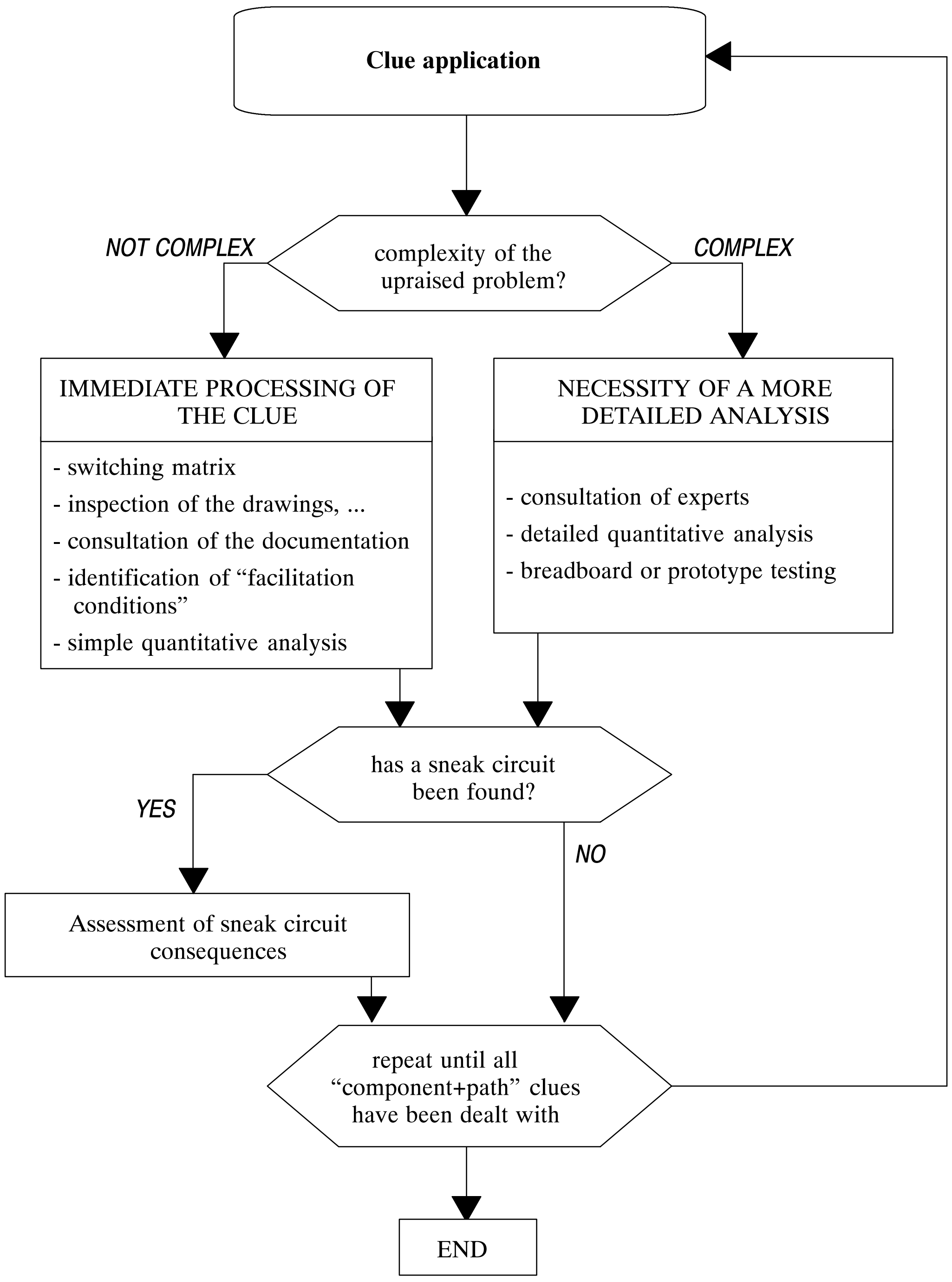
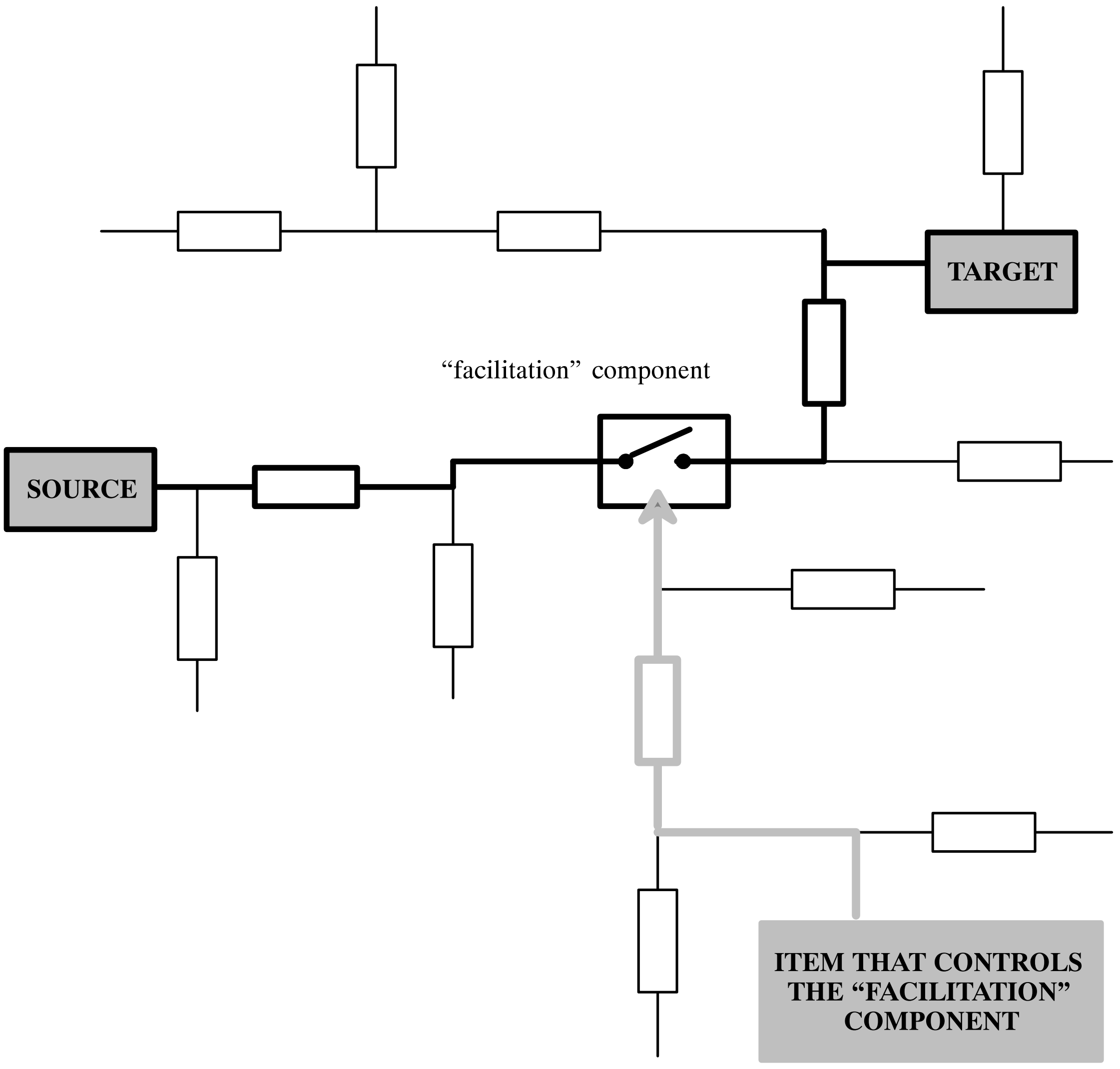

Clue application
complexity of the upraised problem?
NOT COMPLEX
COMPLEX
IMMEDIATE PROCESSING OF THE CLUE
- switching matrix
- inspection of the drawings, ...
- consultation of the documentation
- identification of "facilitation conditions"
- simple quantitative analysis
NECESSITY OF A MORE DETAILED ANALYSIS
- consultation of experts
- detailed quantitative analysis
- breadboard or prototype testing
has a sneak circuit been found?
YES
NO
Assessment of sneak circuit consequences
repeat until all "component+path" clues have been dealt with
END

Figure 4: Illustration of answering process to "component+path" clues



─	all the links that are part of the system
━	a "resource" path identified between the source and the target
▬	a "facilitation" path from the "facilitation" component backward to the item that controls it

Figure 5: Illustration of facilitation path search

5.5.2.6 Hardware/software path analysis

1. For items along the path that are controlled by software, an analysis of the software shall be performed to identify whether the software can lead to the following events:
   * to inadvertent or untimely activation or inhibition of the item;
   * to improper command sequence to the item(s) or improper input data to the item(s).
2. The procedure specified in Annex C should be used to carry out the above analysis.

3. It shall be assessed whether the events found during step 1 of 5.5.2.5, when considered with the path under analysis can lead to a sneak circuit (i.e. an undesired causal relation between source and target).
4. If a potential sneak circuit is detected through performance of steps 1 to 3, the designers shall be consulted to check whether the problem is a real one). If this is the case, the sneak circuit consequences shall be assessed (see task "assessment of sneak circuit consequences").

5.5.2.7 Sneak timing

Through the above process, those sneak circuits (i.e. sneak paths, sneak timings, sneak indications) that are a manifestation of undesired causal relations can be identified.

For those clues pointing out to timing problems (e.g. races, timing discrepancies between interfacing hardware and software items), a timing analysis should be performed. Use shall be made of inputs 5.5.1.f and 5.5.1.g.

This analysis, especially for digital items, will require in most cases the availability of relatively sophisticated simulation tools. Only in the simpler cases, the manual use of timeline diagrams is sufficient.

5.5.3 Outputs

The following information shall be derived:

a. list of targets;

b. list of sources;

c. intended relations between sources and targets;

d. undesired relations (path clues) between sources and targets;

e. sneak circuits.

5.6 Design concern analysis

The aim of this task is to identify design concerns and sneak labels.

5.6.1 Inputs

The following inputs should be used:

a. the list of "atomic items", i.e. the ones at the lower hierarchical level of design decomposition that is of interest (see task "definition of analysis scope");

b. the documentation relevant for the items under a. (see task "data gathering");

c. the list of operational modes (see task "definition of analysis scope");

d. the hierarchical design decomposition (see associated task).

5.6.2 Contents

The following steps 1 to 4 should be performed for all atomic items:

1. Select an operational mode.
2. Apply the "component" clues that match the characteristic of the item and are relevant during the selected operational mode. For each clue, try and provide an answer according to the process shown in Figure 6.
3. Treat drawing errors spotted during the application of the component clues as design concerns (or sneak labels if they are related to labelling of man-machine interfaces). Flag missing information instances to engineering.

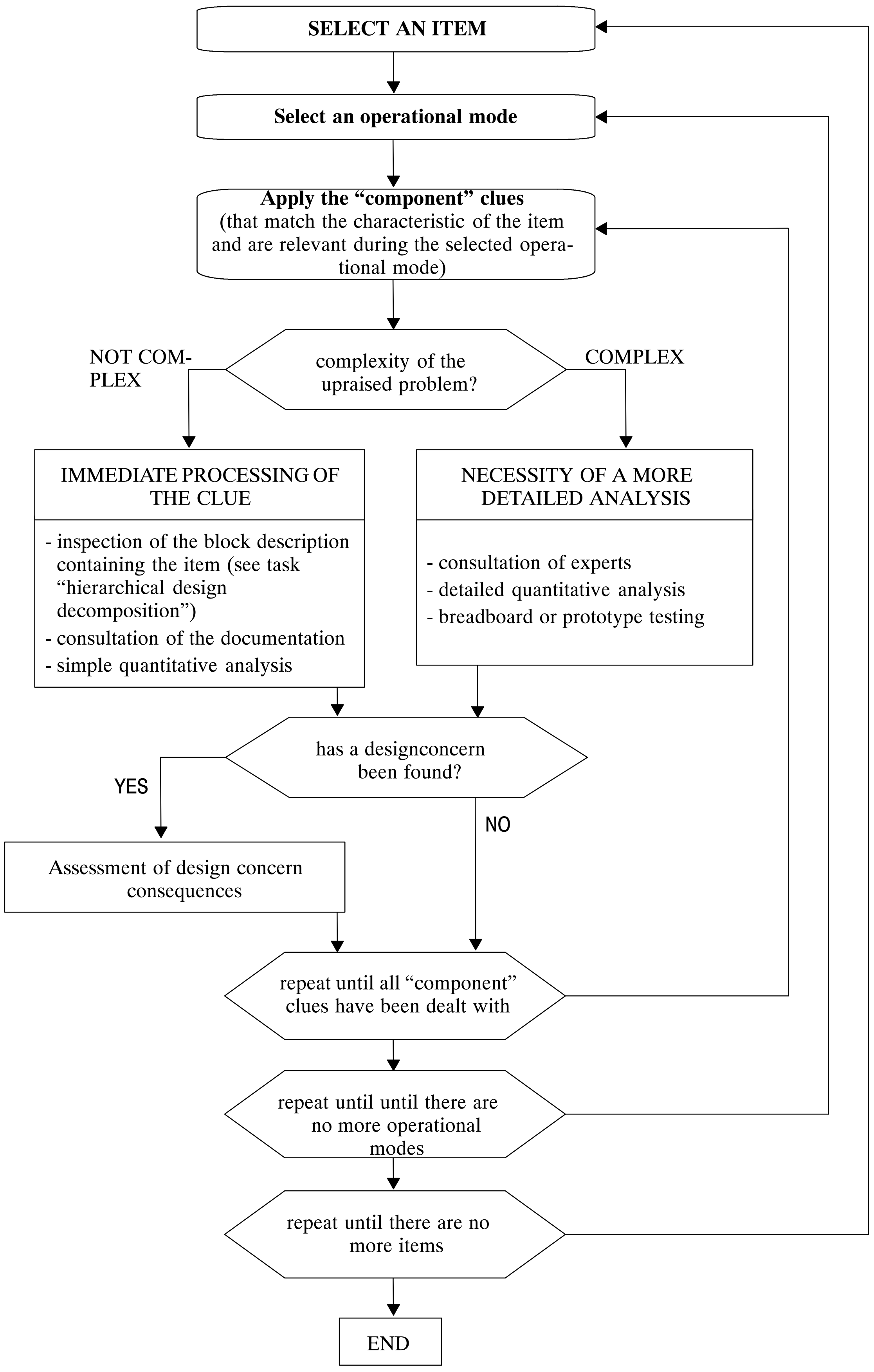

SELECT AN ITEM
Select an operational mode
Apply the "component" clues (that match the characteristic of the item and are relevant during the selected operational mode)
complexity of the upraised problem?
NOT COMPLEX
COMPLEX
IMMEDIATE PROCESSING OF THE CLUE
- inspection of the block description containing the item (see task "hierarchical design decomposition")
- consultation of the documentation
- simple quantitative analysis
NECESSITY OF A MORE DETAILED ANALYSIS
- consultation of experts
- detailed quantitative analysis
- breadboard or prototype testing
has a designconcern been found?
YES
NO
Assessment of design concern consequences
repeat until all "component" clues have been dealt with
repeat until until there are no more operational modes
repeat until there are no more items
END

Figure 6: Illustration of design concern analysis and answering process to "component" clues

4. Repeat steps 2 to 3 until there are no more operational modes
5. For all the candidate design concerns and sneak labels detected, consult the designers to check whether the problem is a real one. If this is the case, their consequences shall be assessed (see task "assessment of sneak circuit consequences").

5.6.3 Outputs

The following outputs shall be produced:

a. design concerns;

b. missing information instances;.

c. sneak labels.

5.7 Assessment of sneak circuit consequences

The aim of this task is to assess the consequences of a sneak circuit or design concern up to the higher level of design decomposition that is of interest.

5.7.1 Inputs

The following inputs should be used:

a. the list of items subjected to the sneak analysis (see task "definition of the analysis scope");

b. the list of operational modes (see task "definition of the analysis scope");

c. the design blocks (see task "hierarchical design decomposition");

d. the list of sneak circuits (see tasks "sneak path analysis" and "design concern analysis");

e. the list of design concerns (see task "design concern analysis").

5.7.2 Contents

For each sneak circuit or design concern identified during the previous tasks the following steps shall be performed.

1. Retrieve the operational mode(s) under which the sneak circuit or design concern was identified.
2. Assess the consequences of the sneak circuit or design concern on the next higher hierarchical level of design decomposition. In doing so, take into account:
   * the operational mode;
   * the characteristic of the design as described in the relevant blocks (see task "hierarchical design decomposition").
3. Repeat step 2 for the next higher decomposition level until the highest level under consideration is reached. Document the consequence on safety and/or reliability associated with the sneak circuit or design concern under consideration.
4. Repeat steps 1 to 3 until there are no more sneak circuits or design concerns.

NOTE In performing step 2 use can be made of the results of other RAMS analyses (e.g. FMECA). It is also beneficial at this stage to integrate the results of sneak analysis with the ones obtained by other analyses (e.g. FMECA, hazard analysis) that are performed in parallel. This will ensure consistency of the recommendations for the elimination of the sneak circuits or design concerns with the ones issued by the other RAMS analyses.

5.7.3 Outputs

The consequences on safety and/or reliability of each sneak circuit and design concern shall be documented.

5.8 Reporting of findings

The aim of this task is to report about the identified sneak circuits and design concerns, and the recommendations for their elimination.

5.8.1 Inputs

The following inputs should be used:

a. the sneak circuits and associated consequences (see task "assessment of sneak circuit consequences");

b. the design concerns and associated consequences (see task "assessment of sneak circuit consequences");

c. the documentation identified during the task "data gathering";

d. the list of items subject to sneak analysis (see task "definition of analysis scope");

e. the mission phases to be considered (see task "definition of analysis scope");

f. the design blocks (see task "hierarchical design decomposition").

5.8.2 Contents

5.8.2.1 Sneak circuit reports

For each sneak circuit or design concern, a "sneak circuit report" containing the entries (up to the "problem identification" one) included in the form contained in Annex B shall be prepared. Use information 5.8.1.c to 5.8.1.f for this purpose.

5.8.2.2 Grouping of sneak circuit reports

The reports related to the same items (e.g. subsystems, assemblies, equipments, components) shall be gathered and it shall be checked whether a correlation can be established between the problems mentioned in different reports.

This approach makes it possible, in some cases, to identify new problems that are due to the synergic effect of several sneak circuits.

In these cases, new sneak circuit reports shall be raised.

5.8.2.3 Issue of recommendations

a. In collaboration with engineering, the options for elimination of the identified sneak circuits and design concerns shall be studied and the sneak circuit reports shall be completed by adding the appropriate recommendation(s) for elimination.

b. Eventually, for each identified recommendation, evidence of its implementation and verification shall be obtained and documented in the sneak circuit report.

c. Concurrence of the supplier's product assurance manager and the project manager on both the recommendations and their implementation and verification shall be obtained.

5.8.3 Outputs

The sneak circuit reports containing the information required in Annex B shall be issued.

5.9 Compilation of the final report

The sneak analysis report is intended:

- to list all the documents that have been used and/or issued during the previous tasks;
- to describe all the problems that have been identified during the analysis;
- to describe the recommendations that have been issued to solve these problems.

5.9.1 Inputs

The inputs that should be used for this task are all the outputs from the previous tasks.

5.9.2 Contents

5.9.2.1 Compilation

The supplier shall compile the sneak analysis report, including as a minimum the information listed in clause 6.

5.9.2.2 Issue of the report

It may be acceptable to issue the sneak analysis report before all the recommendations issued by the analysis have been implemented and verified.

In these cases the supplier shall explain to the customer how the recommendations that are not yet implemented and verified will be followed-up.

5.9.3 Outputs

The sneak analysis report shall be issued.

(This page is intentionally left blank)

6 Sneak analysis report

6.1 Content of the report

The supplier shall produce a sneak analysis report containing:

- document number, issue, revision and date of reference and applicable documents (including reference to the sources of clue list);
- an introduction recalling and justifying the analysis scope (items to be analysed, operational modes to be considered, depth of analysis);
- identification of the parts (section, paragraph, page) of the documents (number, issue, revision, date) that were considered relevant as input information for the analysis;
- summary of the requests for clarification issued, their answers and status;
- the results of the hierarchical decomposition of the design (or reference to the document containing it);
- the input/output matrices (or reference to the document containing them);
- the list of targets, list of sources, intended relations between sources and targets, undesired relations between sources and targets considered during the analysis;
- the list of cases where lack of data was found in the input data;
- the sneak circuit reports, their associated recommendations and status;
- other problems that although not classifiable as sneak circuits or design concerns could lead to an undesirable impact on system safety and/or reliability and have not been identified by other RAMS analyses;
- the new clues synthesised during the analysis (if any);
- conclusion (with a summary table of sneak circuits and design concerns and their status) outlining the major problems found.

6.2 Approval of the report

The sneak analysis report shall be signed by the analyst(s) and shall be approved by the supplier's product assurance manager and project manager.

(This page is intentionally left blank)

Annex A (informative)

Derivation of path clues

The following illustrates how the path clues can be derived.

If a static relation between sources and targets is considered and their states can be modelled as binary variables, then all the path clues can be derived from the following two (generic) clues:

a. Can an undesired causal flow switch-on the targets?

b. Can an undesired causal flow switch-off the targets?
If the relation is time-dependent (i.e. the targets have to be "on" or "off" only during a certain time interval) then the following generic clues have to be added to the previous ones:

c. Can an intended causal flow switch-on the target at the wrong time?

d. Can an intended causal flow switch-off the target at the wrong time?

To identify the (specific) path clues for a given number of binary targets and sources the following steps have to be followed:

1. build the state table for the set identified by the targets and the sources;
2. identify all the unwanted set states in which at least one of the targets is "on". To these states apply the generic clue "a" (and "c" if the relation is time dependent) to derive the specific clues;
3. identify all the unwanted set states in which at least one of the targets is "off". To these states apply the generic clue "b" (and "d" if the relation is time dependent) to derive the specific clues.

For example, if the intended relation between a target T and two sources S1 and S2 is an "AND" and all the other relations are undesired, then clue a) generates the three following specific clues:

- Can T be on when S1 is ON and S2 is OFF?
- Can T be on when is S1 is OFF and S2 is ON?
- Can T be on when both S1 and S2 are OFF? and clue b) now gives the following specific clue:
- Can T be off when both S1 and S2 are on?

A similar approach can be followed to derive the path clues when targets and sources are not binary, but can assume only a finite number to states.

The above approach for the identification of path clues is a viable one if the total number of binary sources (N) and targets (M) considered is reasonably small. Otherwise the coverage of all the possible path clues (2**[N+M]) becomes unwieldy (if not impossible) for complex systems. In this last case, the analyst should limit the number of relevant path clues by:

- checking whether some targets are related only to a subset of sources; and
- using path clues that are related only to the most critical targets (according to the results of preliminary hazard analysis and functional failure analysis).

Annex B (normative)

Sneak circuit report form

SNEAK CIRCUIT REPORT

Reference:	PROJECT:
Issue:	FUNCTION:
Revision:	Subsystem:
Date:	Equipment:
Page:	Phase:

HAZARD CONSEQUENCE SEVERITY

RELIABILITY FAILURE EFFECT SEVERITY CATEGORY

CONFIGURATION ITEM REFERENCE:

DESCRIPTION OF FUNCTION/ITEM:

PROBLEM TITLE:

PROBLEM TYPE: ☐ SNEAK PATH ☐ SNEAK TIMING ☐ SNEAK INDICATION ☐ SNEAK LABEL ☐ DESIGN CONCERN

PROBLEM IDENTIFICATION (including CAUSES and ESTIMATED EFFECTS):

ANALYST
prepared by Name, date & signature

RECOMMENDATIONS:

IMPLEMENTATION AND VERIFICATION OF RECOMMENDATOINS:

ANALYST	APPROVAL	
prepared by Name, date & signature	PA Manager Name, date & signature	Project Manager Name, date & signature

Annex C (normative)

Sneak analysis applied to computer software

The aim of this task is to identify “facilitation conditions” related to the software which can lead to the unwanted activation or deactivation of equipment.

C.1 Inputs

The following inputs should be used:

a. the list of items that are included in the scope of the analysis (see task “definition of the analysis scope”);

b. the design decomposition of the above items (see task “hierarchical design decomposition”);

c. the relevant input/output matrices (see task “synthesis of input/output matrix);

d. the results of the “top level” RAMS analysis (e.g. preliminary hazards analysis, functional failure analysis);

e. the documentation relevant for the items under analysis (see task “data gathering”).

f. the targets, sources, the undesired relations (and the corresponding paths) between target and sources, that were identified at HW/SW level (see task “sneak path analysis).

C.2 Contents

C.2.1 Preparation

a. By reviewing the results (see C.1.f) of sneak path analysis at HW/SW level, the hardware items that are controlled by software shall be identified;

b. using the documentation (see C.1.e) on the HW/SW interfaces and on the software, the control commands and data issued by the software to the hardware items shall be identified;

c. the inputs to the software under analysis coming from the other hardware or software items shall also be identified.

C.2.2 Sneak path analysis in computer software

The following procedure for sneak path analysis in computer software should be followed:

1. choose as "intermediate targets" for the analysis in the software the commands and data outputs identified by performing sub-task C.2.1;
2. pick-up one intermediate target in the scope of the analysis;
3. trace a (facilitation) path in the software flow chart or data flow diagram backwards from the intermediate target to the software inputs (e.g. input registers, data initialisation instructions, operator commands, program start). Record the "logical condition" necessary for the path to be followed (e.g. if a path traverses the statement "IF x>0 THEN ...", the logical condition for the "YES" branch is "x>0");
4. check the path as it is built up by comparing it with the input/output matrix. If there is no operating mode which allows the path to be activated, abandon the path. Also abandon the path if the logical condition for the path simplifies to "FALSE";
5. continue the path trace as far as the software inputs;
6. apply software "component+path" clues to the software instructions (e.g. conditionals, loops, function calls, assignments) along the path;
7. assess whether the software can lead to the unwanted or untimely activation (or deactivation) of the intermediate target.
8. repeat steps 3 to 7 for the various paths through the software under analysis. It is noted that the number of paths to be analysed is limited for software where the two following requirements apply (as it generally the case nowadays):
   * the software shall be composed of modules;
   * each software shall have low cyclomatic complexity (e.g. less 10).

   If for the software under analysis the two above quoted requirements are not applicable, then appropriate heuristic criteria might need to be defined, in coordination with the software engineering and product assurance functions, to keep under control the number of paths to be analysed.
9. repeat steps 3 to 8 for all intermediate targets;
10. by reviewing the results of the above steps check that the software commands are provided according to the required sequence and the software output data is within the allowed ranges.

C.3 Outputs

The following outputs shall be derived:

a. list of intermediate targets;

b. facilitation conditions in the software that lead to unwanted or untimely activation (or deactivation) of the intermediate targets;

c. list of cases where improper sequence of commands or improper output data is produced by the software.

Annex D (informative)

Bibliography

A list of papers, available in the public literature, that contain either useful information about aspects of the sneak analysis procedure or examples of application of the procedure is included in this Annex. A short comment has been added after certain papers to explain the objective of the paper (if this is not clear from the title).

a. Taylor, J.R., Sneak analysis course notes, ITSA 90-11-1, October 1991.

Section 3, 4, 5 of this report contain guidance and examples anout the application of the the sneak path analysis (see section 5.5 of this standard). The report is available through the ESA/ESTEC/QP division.

b. Proceedings of the sneak analysis workshop, ESA/ESTEC WPP-033, Noordwijk, June 1992.

The results of several applications of sneak analysis in Europe are summarized in these proceedings. They could be of use mainly with respect to the tasks described in sections 5.1, 5.2, 5.3, 5.6, 5.7, 5.8 of this standard. These proceedings are available through the ESA/ESTEC Technical documentation center.

c. Sneak circuit analysis. A means to verify design integrity, USA Departement of the Navy, NAVSO P-3634, August 1987

This report provides an overview about sneak analysis benefits and applicability, an outline of results of previous applications of sneak analysis in the USA and guidance on its implementation. It is noted that the sneak analysis procedure is different from the one contained in this standard. This report could be of use with respect to the tasks described in sections 5.1, 5.2, 5.4 and 5.6 of this standard.

d. Dore, B., Lessons learned from pilot applications of sneak analysis in space projects, ESA SP-337, pp. 359-363, Noordwijk, May 1996.

e. Dore, B. & Norstrom, J.G., Pilot application of sneak analysis on computer controlled satellite equipment, Proceedings of Probabilistic Safety Assessment and Management Conference, pp. 1590-1596, Springer, London, 1996

This paper could be of use with respect to the task described in Annex C of this standard.

f. De Mateo, G., Application of sneak analysis to hydraulic systems, ESA/ ESTEC EWP-1801, Noodwijk, October 1994.

This report contains the results of an application of Sneak path analysis to hydraulic (e.g. propulsion) systems. It could be of use with respect to the tasks described in sec-

tions 5.4 and 5.5 of this standard. This report is available through the ESA/ESTEC Technical documentation center.

ECSS Document Improvement Proposal

1. Document I.D. ECSS-Q-40-04A Part 1	**2. Document Date** 14 October 1997	**3. Document Title** Sneak analysis - Part 1: Method and procedure
4. Recommended Improvement (identify clauses, subclauses and include modified text and/or graphic, attach pages as necessary)		
5. Reason for Recommendation		
6. Originator of recommendation		
Name:	Organization:	
Address:	Phone: Fax: E-Mail:	**7. Date of Submission:**
8. Send to ECSS Secretariat		
Name: W. Kriedte ESA-TOS/QR	Address: Keplerlaan 1 2200AG Noordwijk Netherlands	Phone: +31-71-565-3952 Fax: +31-71-565-6839 E-Mail: wkriedte@estec.esa.nl

Note: The originator of the submission should complete items 4, 5, 6 and 7.

This form is available as a Word and Wordperfect-Template on internet under
http://www.estec.esa.nl/ecss/improve/

(This page is intentionally left blank)