ECSS-Q-TM-40-04 Part 1A
16 April 2010

Space product assurance

Sneak analysis - Part 1: Principles and requirements

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

Foreword

This document is one of the series of ECSS Technical Memoranda. Its Technical Memorandum status indicates that it is a non-normative document providing useful information to the space systems developers' community on a specific subject. **It is made available to record and present non-normative data, which are not relevant for a Standard or a Handbook.** Note that these data are non-normative even if expressed in the language normally used for requirements.

Therefore, a Technical Memorandum is not considered by ECSS as suitable for direct use in Invitation To Tender (ITT) or business agreements for space systems development.

Disclaimer

ECSS does not provide any warranty whatsoever, whether expressed, implied, or statutory, including, but not limited to, any warranty of merchantability or fitness for a particular purpose or any warranty that the contents of the item are error-free. In no respect shall ECSS incur any liability for any damages, including, but not limited to, direct, indirect, special, or consequential damages arising out of, resulting from, or in any way connected to the use of this Document, whether or not based upon warranty, business agreement, tort, or otherwise; whether or not injury was sustained by persons or property or otherwise; and whether or not loss was sustained from, or arose out of, the results of, the item, or any services that may be provided by ECSS.

Published by: ESA Requirements and Standards Division
ESTEC, P.O. Box 299,
2200 AG Noordwijk
The Netherlands

Copyright: 2010© by the European Space Agency for the members of ECSS

Change log

ECSS-Q-40-04A Part 1 14 October 1997	First issue
ECSS-Q-TM-40-04 Part 1A 16 April 2010	Second issue Conversion of ECSS-Q-40-04A - Part 1: Method and procedure (14 October 1997) into a Technical Memorandum according to ECSS drafting rules for Technical Memorandum.

Table of contents

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 Technical Memorandum establishes a procedure for performing sneak analysis and specifies the required output.

The Technical Memorandum is composed of two parts:

- Part 1 (i.e. this document ECSS-Q-TM-40-04 - Sneak analysis - Methods and procedures) that contains the methods and procedures for performing sneak analysis;
- Part 2 (i.e. the document ECSS-Q-TM-40-04 - Sneak analysis - Clue list) that contains a basic clue list to be used during sneak analysis.

This technical memorandum is applicable when the performance of sneak analysis is required by ECSS-Q-ST-40 or by the business agreement 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 Technical Memorandum is shown by the supplier.

This Technical Memorandum may be tailored for the specific characteristic and constrains of a space project in conformance with ECSS-S-ST-00.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of this ECSS Technical Memorandum. For dated references, subsequent amendments to, or revision of any of these publications do not apply, However, parties to agreements based on this ECSS Technical Memorandum are encouraged to investigate the possibility of applying the more recent editions of the normative documents indicated below. For undated references, the latest edition of the publication referred to applies.

ECSS-ST-S-00-01	ECSS system - Glossary of terms
ECSS-Q-ST-30	Space product assurance - Dependability
ECSS-Q-ST-40	Space product assurance - Safety

3 Terms, definitions and abbreviated terms

3.1 Terms from other standards

For the purpose of this document, the terms and definitions from ECSS-S-ST-00-01 apply, in particular for the following terms:

configuration item

failure

severity

software

system

For the purpose of this document, the terms and definitions from ECSS-Q-ST-30 and ECSS-Q-ST-40 apply.

3.2 Terms specific to the present document

3.2.1 clue

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

3.2.2 design concern

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

3.2.3 design error

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.

3.2.4 Facilitation condition

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

3.2.5 Sneak Circuit

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.

NOTE 1 sneak circuits are not the result of failures, but are latent conditions, inadvertently designed into the system.

NOTE 2 sneak circuits include: sneak paths, sneak timings, sneak indications and sneak labels.

3.2.6 sneak indication

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

3.2.7 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.

3.2.8 sneak path

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

3.2.9 sneak timing

occurrences of events in an unexpected or conflicting sequence, or at an unexpected time, or for an unexpected duration.

NOTE therefore sneak timings can also occur if mass, energy, data or logical control flow along intended paths without respecting the intended dynamic behaviour of the system.

3.2.10 source

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

3.2.11 target

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

3.3 Abbreviated terms

For the purpose of this Document, the abbreviated terms from ECSS-S-ST-00-01, ECSS-Q-ST-30 and ECSS-Q-ST-40 and the following apply:

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	structures 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 indicate 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 sneak 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.

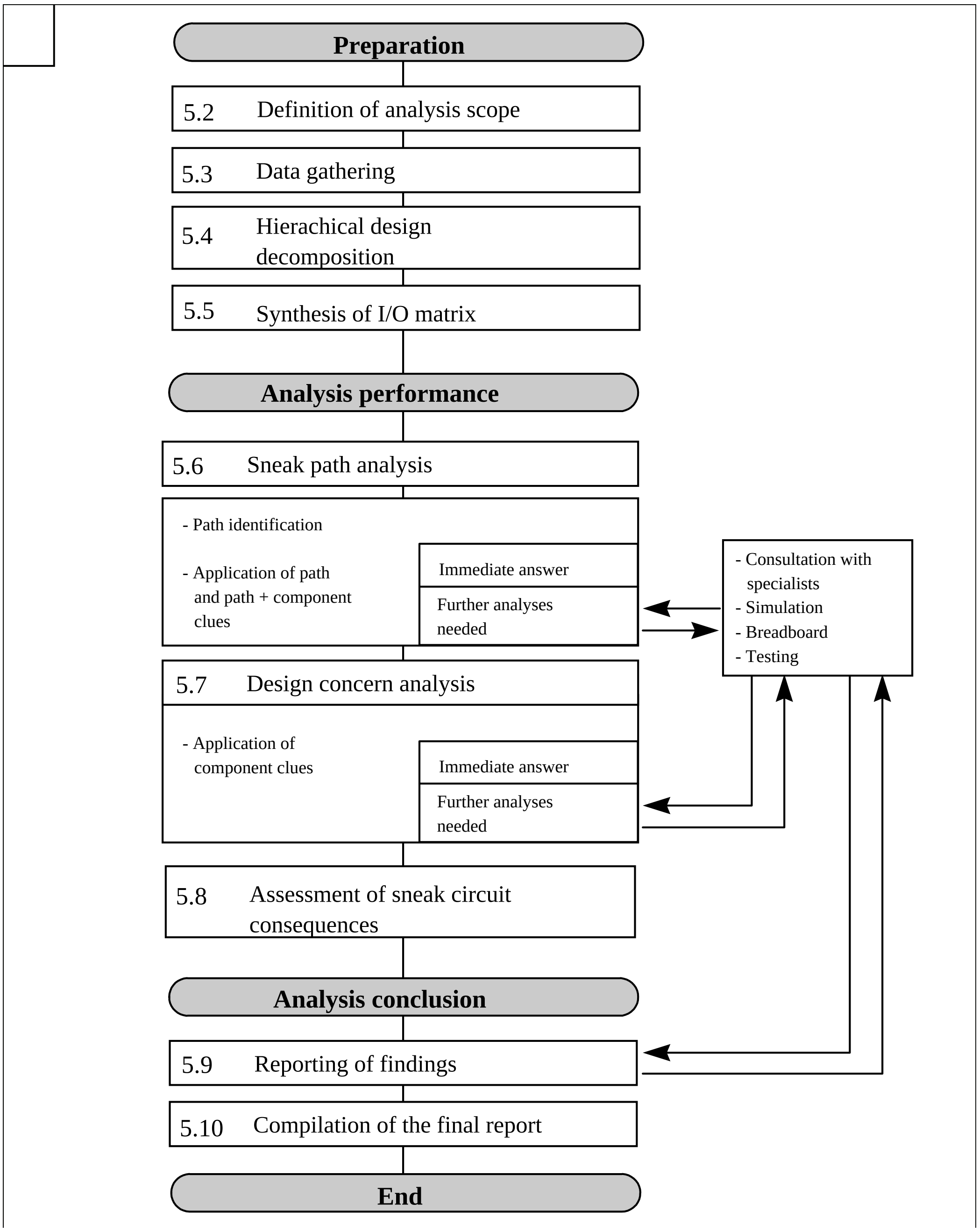

Preparation
5.2 Definition of analysis scope
5.3 Data gathering
5.4 Hierachical design decomposition
5.5 Synthesis of I/O matrix
Analysis performance
5.6 Sneak path analysis
- Path identification
- Application of path and path + component clues
Immediate answer
Further analyses needed
- Consultation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

Figure 4-1: Outline of sneak analysis procedure

4.3 Input data for sneak analysis

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

a. System Level

1. System requirements (including external interface requirements)
2. System design
3. Internal interface definition (including HW/SW interfaces)
4. User manual (including operation procedures)

5. Results of preliminary RAMS analyses (e.g. Functional failure analysis, preliminary hazard analysis)
6. Results of functional analysis

b. Lower Level (for Hardware)

1. Subsystem requirements
2. Subsystem design
3. Equipment requirements
4. Equipment design (including drawings)
5. Component specification (data sheets)
6. Results of worst case analysis (*)
7. Results of part stress analysis (*)
8. Development testing results (*)

c. Lower Level (for Software)

1. User requirements
2. Software requirements
3. Architectural design
4. Detailed design
5. 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:

a. product description;

b. top level diagram;

c. functional characteristics (e.g. functions of each board);

d. limitations (e.g. lifetime);

e. external electrical interfaces;

f. internal electrical interfaces;

g. electrical schematics (including interface circuits);

h. technical characteristics (not those required but those really implemented in the design, e.g. power line protection, grounding);

i. parts list.

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

a. software architecture describing the software decomposition in functions, their inter-relationships and sequencing;

b. for each software item: function, subordinates, dependencies, interfaces, resources, processing, data;

c. 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 Computerized 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:

a. 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;

b. 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;

c. 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;

d. 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

5.1 Overview

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

a. preparation;

b. analysis;

c. reporting and conclusions.

The relationship between the various tasks is shown in Figure 4-1. The following pages specify the contents of the various tasks. Each task description, after an outline of the objective of the task, contains:

a. **inputs**, where the information that is needed for the task is identified;

b. **contents**, where the analytical steps that are required to carry out the task are specified;

c. **outputs**, where the information that is expected to

5.2 Definition of the analysis scope

5.2.1 Overview

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

5.2.2 Inputs

a. The following inputs should be used:

1. the requirements on sneak analysis contained in the contract (if any);

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

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

4. the list of aspects to be considered included in Table 5-1

Table 5-1: Aspects to be considered when defining the analysis scope

Safety an 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?
Programmatic aspects
Are there several interfaces manufactured by different suppliers?
Have many modifications occurred since the beginning of the programme?
Are many modifications expected?

5.2.3 Contents

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

1. Check whether the contract contains specific requirements on sneak analysis:

   (a) if yes, perform step 2 below;

   (b) 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 5.2.2a.2 and the results of the RAMS analyses quoted under 5.2.2a.3., 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 5.2.2a.3, identify the list of safety and reliability critical functions. Answer to the questions contained in Table 5-1 for each of the items contained in the safety or reliability critical functions (in general a "yes" answer to a question contained in this Table 5-1 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 5-1

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 5-1.

5.2.4 Outputs

a. The following information shall be identified

1. Items that are to be analysed
2. the mission phases to be considered
3. the level of depth to be reached

5.3 Data gathering

5.3.1 Purpose

The purpose of this task is to collect the input necessary for the performance of a sneak analysis

5.3.2 Input

a. The following inputs should be used:

1. list of items under analysis (see task “definition of analysis scope”);
2. level of depth of the analysis (see task “definition of analysis scope”);
3. the documents (e.g. see 4.3) containing the design and operations data for the above quoted items;
4. the clue list (see ECSS-Q-TM-40-04 Part 2).

5.3.3 Contents

5.3.3.1 General

a. The following sub-tasks should be performed:

1. Screening of available documentation
2. Homogeneity control
3. Familiarisation with the documentation
4. tailoring of the clue list

5. Documentation of the findings

5.3.3.2 Screening of available documentation

a. The parts of the documents quoted under 5.3.2a.3 that are relevant to the items under analysis at the required level of depth .shall be identified.

5.3.3.3 Homogeneity control

a. The homogeneity of the documents that have been gathered under 5.3.3.2 shall be checked from a configuration management's point of view.

5.3.3.4 Familiarisation with the documentation

a. The sneak analyst shall :

1. Become familiar with the documentation screened under 5.3.3.2.

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

3. 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.

5.3.3.5 Tailoring of the clue list

a. Taking into account the scope of the application, the list of items under analysis and the level of depth, the sneak analyst shall:

1. supplement the clues contained in 5.3.3.1a.4 with other clues derived from the knowledge available at the Supplier; and/or

2. tailor the list of clues contained in 5.3.3.1a.4 by discarding the ones that one not relevant for the application in order to improve the efficiency of the analysis.

5.3.3.6 Documentation of the findings

a. The sneak analyst shall document the outcome of the previous sub-tasks as defined in 5.3.3.1a

b. In particular, the sneak analyst shall make sure that:

1. the points related to lack of documentation homogeneity or the requests for clarification are discussed with the product assurance and engineering personnel;

2. 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.

c. The sneak analyst shall give particular attention to the interfaces between hardware and software items in identifying:

1. real time issues,

2. software asynchronous behaviour and constraints on control flow sequence.

d. If necessary, the analyst shall aggregate information spread in several different documents (e.g. for a multi-board electronic equipment,

e. if the data and control interfaces between different boards are spread over several documents and drawings it might be useful to establish a synthetic sketch showing the above interfaces).

5.3.4 Outputs

a. The following information shall be identified:

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

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

3. the requests for additional information or clarification;

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

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

5.4 Hierarchical design decomposition

5.4.1 Objectives

The aim of this task is to produce 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:

a. it allows the systems to be subdivided into blocks that are easily manageable (both in terms of size and understandability) by the analysts;

b. it establishes a clear cross-reference between the required functions and the actual design;

c. it supports the assessment of the consequences of the sneak circuits (see subclause 5.8).

5.4.2 Inputs

a. The following inputs should be used:

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

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

3. the documentation relevant to the items under analysis (see task “data gathering”);

4. ‘items’ interfaces (see task “data gathering”).

5.4.3 Contents

5.4.3.1 General

a. 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.

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

5.4.3.2 Design decomposition

a. The following operations shall be performed:

1. 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.

2. Identify the part of the item that is associated mainly with a single item function. Define this part as a “design block”.

3. 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.

4. To avoid confusion, use the names of the interfaces mentioned in the documentation also in the blocks.

5. Depict the above decomposition in graphical format. If necessary, go to the next level of decomposition and apply the above procedure to each block.

6. 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).

NOTE During this 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.

Figure 5-1 and Figure 5-2 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 5-1). An initial decomposition could lead, for example, to a diagram such as that shown in Figure 5-2, where 3 blocks, each associated to a function have been identified.

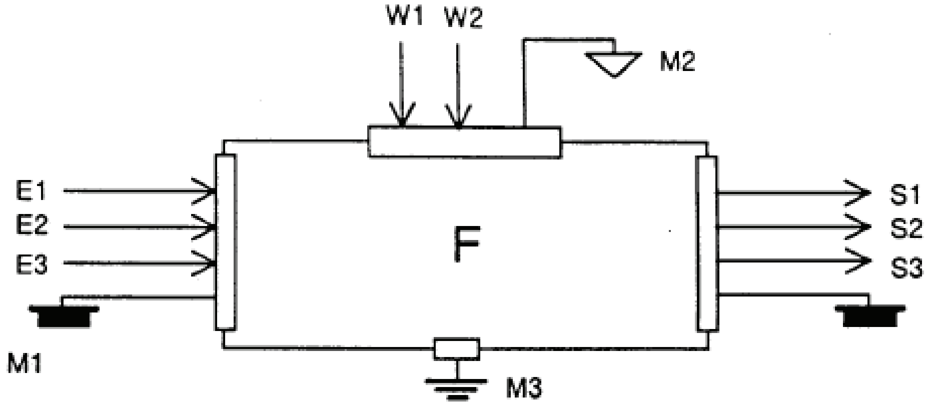
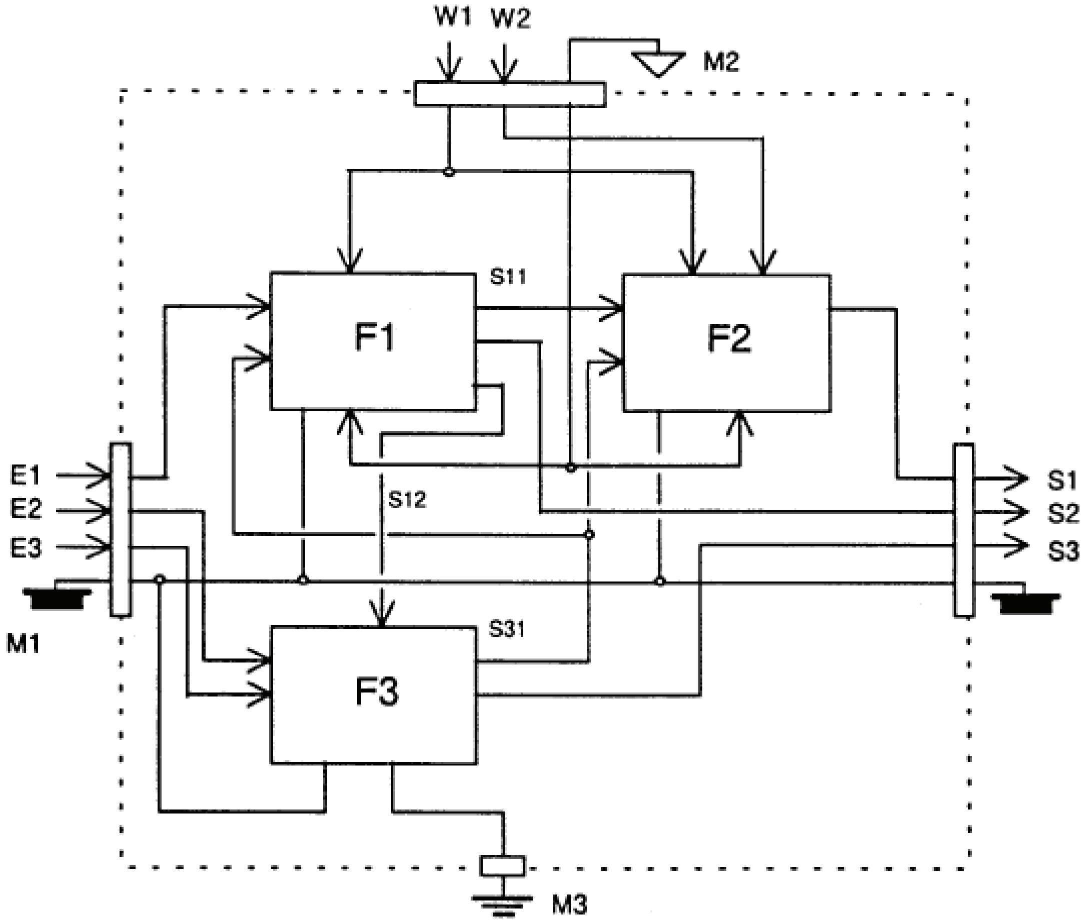


Figure 5-1: Illustration of design decomposition - Top level



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

Figure 5-2: Illustration of design decomposition - Lowest level

5.4.3.3 Documentation of the design decomposition

a. The following information shall be documented for each block:

1. system concerned;
2. description of the block;
3. block diagram (e.g. see Figure 5-2 for block F);
4. functions performed by the block;
5. design characteristic of the block;
6. interfaces with other blocks

5.4.4 Outputs

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

5.5 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.

5.5.1 Inputs

a. The following inputs should be used:

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

5.5.2 Contents

5.5.2.1 General

a. 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.

b. Otherwise the following sub-tasks should be performed.

5.5.2.2 Documentation of operational modes

a. All the planned operational modes for the item during the mission phases to be analysed shall be identified (by using inputs 5.5.1a.3 and 5.5.1a.4).

b. All simultaneous changes (if any) of operational modes for several items are planned shall be pinpointed for further consideration in the next tasks (they are possible sources of sneak timings).

5.5.2.3 Identification of elementary switching events

a. Elementary events that trigger the item outputs shall be identified (by using inputs 5.5.1a.4 and 5.5.1a.5).

NOTE These elementary events are for example:

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

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

a. the input/output matrix shall be built as follows:

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

NOTE An example of the format of the matrix (for the item depicted in Figure 5-1) is provided in Table 5-2.

Table 5-2: 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

5.5.2.5 Input/output matrix at lower levels of design decomposition

5.5.2.5.1 General

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).

5.5.2.5.2 Requirements

a. The selection of these lower level blocks shall be done on a case-by-case basis.

b. The selection of the shall be based on the following criteria to be taken into account are:

1. the complexity of the block architecture;

2. the impact of a change in the block outputs on the output of the item.

NOTE 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.

c. 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.

5.5.2.6 Operational sequence representation

a. Using the outcome of subtasks 5.5.2.2 and 5.5.2.3, and the input 5.5.1a.4, the sequence of operational modes for each item under analysis shall be identified and documented (e.g. in the form of timelines).

5.5.2.7 Outputs

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

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

2. identification of instances of simultaneous switching;

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

4. operational sequence representation for the items under analysis.

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

5.6 Sneak path analysis

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

5.6.1 Inputs

a. The following inputs should be used:

1. the list of items that are included in the scope of the analysis (see task "definition of the analysis scope");
2. the design decomposition of the above items (see task "hierarchical design decomposition");
3. the input/output matrix (see task "synthesis of input/output matrix");
4. the results of the top-level RAMS analyses (e.g. preliminary hazard analysis, functional failure analysis);
5. the documentation relevant for the items under analysis (see task "data gathering");
6. the instance of simultaneous switching of different items (see task "synthesis of input/output matrix");
7. the operational sequence representation (see task "synthesis of input/output matrix").

5.6.2 Contents

5.6.2.1 Identification of the targets

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

NOTE 1 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.

NOTE 2 Inputs of 5.6.1a.2 and 5.6.1a.4 are used.

5.6.2.2 Identification of the sources

a. 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.

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

5.6.2.3 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 in using the data contained in inputs 5.6.1a.2 and 5.6.1a.3 for this purpose.

b. 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. 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.

NOTE 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.

5.6.2.4 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.

NOTE 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 5-3 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:

   (a) the input/output matrix;

   (b) 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);

   (c) 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.

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 5.6.2.5). Additionally if there are items on the paths that are controlled by software, perform the "hardware/software path analysis" (see 5.6.2.6).

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.

   NOTE 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.

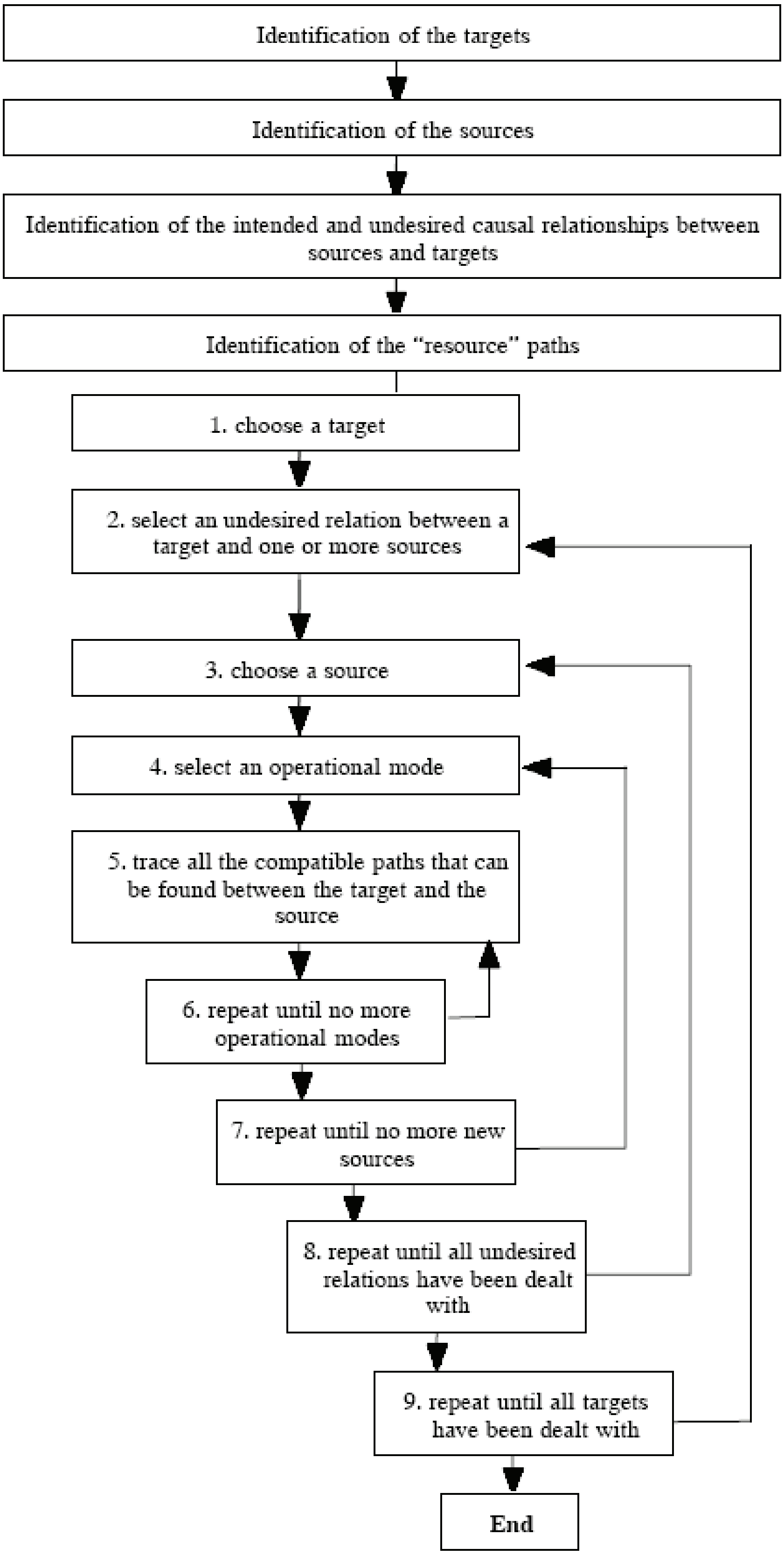

Identification of the targets
Identification of the sources
Identification of the intended and undesired causal relationships between sources and targets
Identification of the "resource" paths
1. choose a target
2. select an undesired relation between a target and one or more sources
3. choose a source
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 5-3: Path identification process

5.6.2.5 Detailed path analysis

a. The detailed path analysis shall consist of the following steps :

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 5-4 for an illustration of this process):

2. Pick-up a "component+path" clue;

3. Try and provide a direct answer by:

   (a) checking the switching matrix; or

   (b) inspecting the drawings, flow-charts or block diagrams containing the path; or

   (c) consulting the documentation gathered in the task "data gathering"; or

   (d) 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 5-5 for an illustration of this concept); or

   (e) 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:

   (a) consult experts;

   (b) perform detailed quantitative analysis;

   (c) 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").

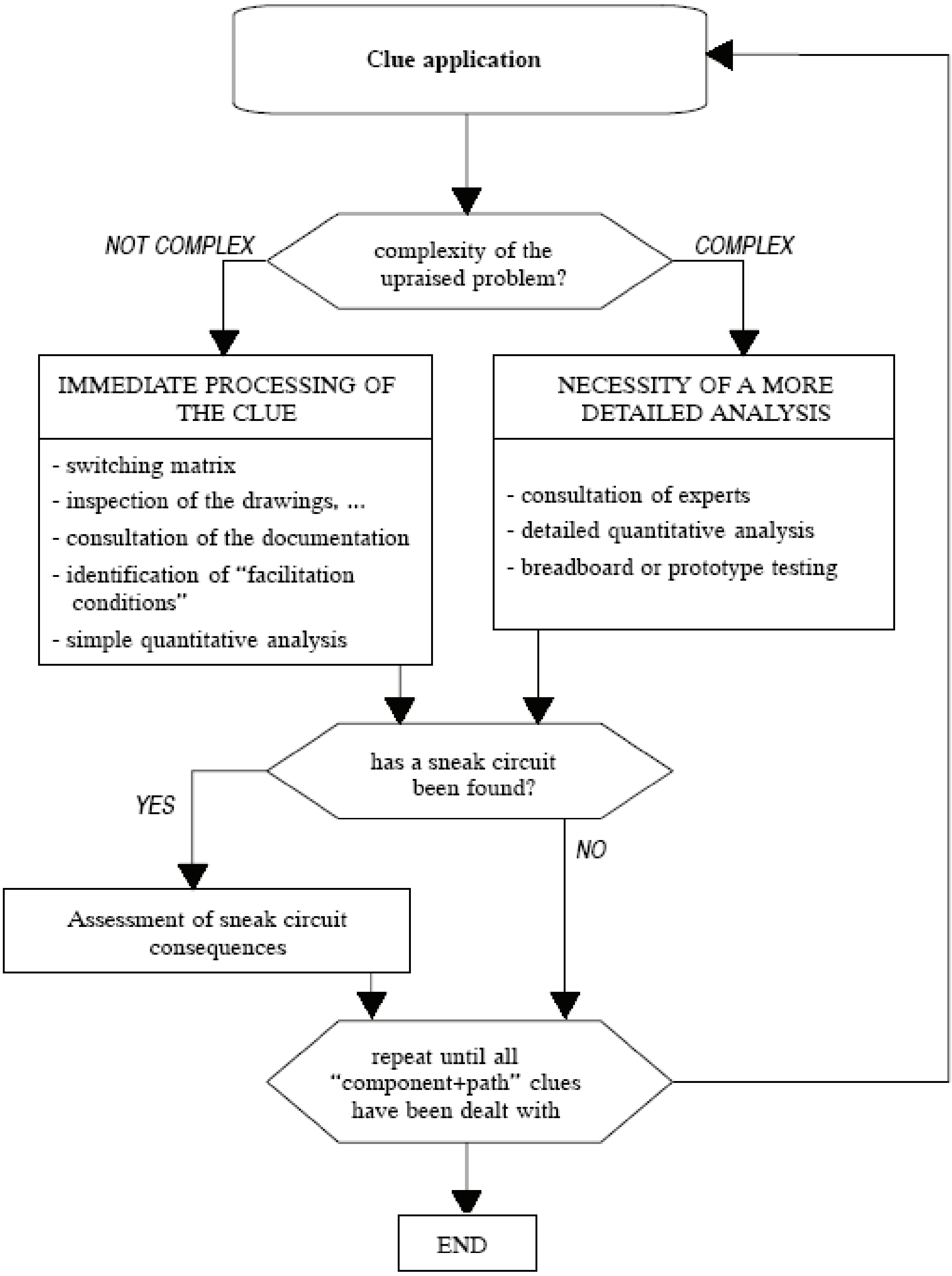

Clue application
complexity of the upraised problem?
NOT COMPLEX
COMPLEX
IMMEDIATE PROCESSING OF THE CLUE
- switching matrix
- inspection of the drawings, ...
- consultation of the documentation
- identification of "facilitation conditions"
- simple quantitative analysis
NECESSITY OF A MORE DETAILED 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 5-4: Illustration of answering process to "component+path" clues

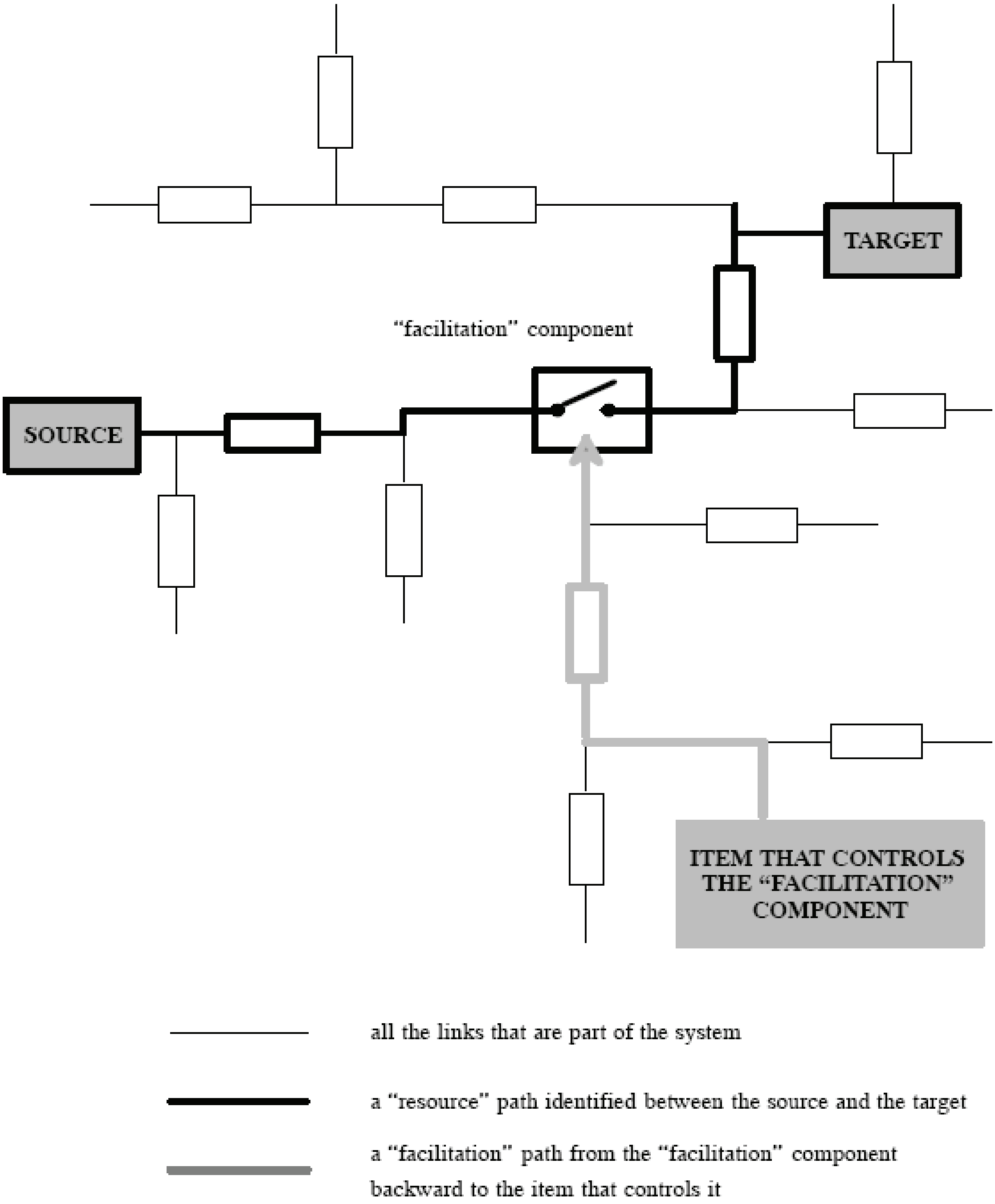

TARGET
"facilitation" component
SOURCE
ITEM THAT CONTROLS THE "FACILITATION" COMPONENT
all the links that are part of the system
a "resource" path identified between the source and the target
a "facilitation" path from the "facilitation" component backward to the item that controls it

Figure 5-5: Illustration of facilitation path search

5.6.2.6 Hardware / software path analysis

a. 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:

1. to inadvertent or untimely activation or inhibition of the item;

2. to improper command sequence to the item(s) or improper input data to the item(s).

b. The procedure specified in Annex C should be used to carry out the above analysis.

c. The sneak analyst shall be assessed whether the events found during step 1 of 5.6.2.5, when considered with the path under analysis can lead to a sneak circuit (i.e. an undesired causal relation between source and target).

d. 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").

5.6.2.7 Sneak timing

5.6.2.7.1 Objective

Through the process described in 5.6.2.6, sneak circuits (i.e. sneak paths, sneak timings, sneak indications) that are a manifestation of undesired causal relations can be identified.

5.6.2.7.2 Requirements

a. For clues pointing out to timing problems (e.g. races, timing discrepancies between interfacing hardware and software items), a timing analysis should be performed, using 5.6.1a.6 and 5.6.1a.7

b. Sneak timing analysis for digital items shall be supported by simulation tools

NOTE Sneak timing analysis needs in most cases the availability of relatively sophisticated simulation tools. Only in the simpler cases, the manual use of timeline diagrams is sufficient.

5.6.3 Outputs

a. The following information shall be derived:

1. list of targets;
2. list of sources;
3. intended relations between sources and targets;
4. undesired relations (path clues) between sources and targets;
5. sneak circuits.

5.7 Design concern analysis

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

5.7.1 Inputs

a. The following inputs should be used:

1. 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");
2. the documentation relevant for the items under a. (see task "data gathering");
3. the list of operational modes (see task "definition of analysis scope");
4. the hierarchical design decomposition (see associated task).

5.7.2 Contents

a. 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 5-6.
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.
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 are assessed (see task "assessment of sneak circuit consequences").

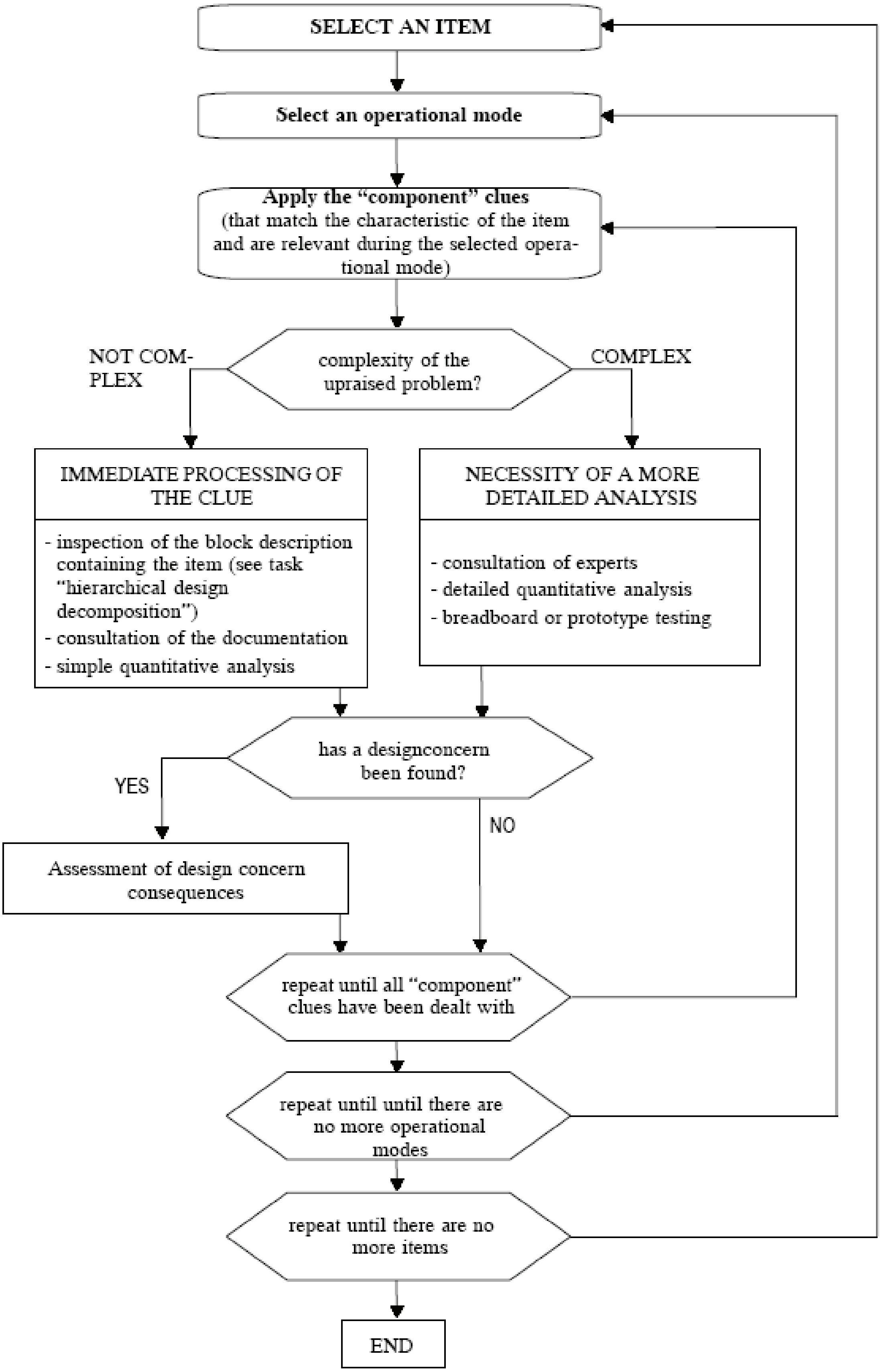

SELECT AN ITEM
Select an operational mode
Apply the "component" clues
(that match the characteristic of the item and are relevant during the selected operational mode)
NOT COMPLEX
complexity of the upraised problem?
COMPLEX
IMMEDIATE PROCESSING OF THE CLUE
- inspection of the block description containing 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 5-6: Illustration of design concern analysis and answering process to "component" clues

5.7.3 Outputs

a. The following outputs shall be produced:

1. design concerns;
2. missing information instances;.
3. sneak labels.

5.8 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.

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.

5.8.1 Inputs

a. The following inputs should be used:

1. the list of items subjected to the sneak analysis (see task "definition of the analysis scope");
2. the list of operational modes (see task "definition of the analysis scope");
3. the design blocks (see task "hierarchical design decomposition");
4. the list of sneak circuits (see tasks "sneak path analysis" and "design concern analysis");
5. the list of design concerns (see task "design concern analysis").

5.8.2 Contents

a. 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:
   (a) the operational mode;
   (b) 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 ensures consistency of the recommendations for the elimination of the sneak circuits or design concerns with the ones issued by the other RAMS analyses

5.8.3 Outputs

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

5.9 Reporting of findings

5.9.1 Inputs

a. The following inputs should be used:

1. the sneak circuits and associated consequences (see task "assessment of sneak circuit consequences");
2. the design concerns and associated consequences (see task "assessment of sneak circuit consequences");
3. the documentation identified during the task "data gathering";
4. the list of items subject to sneak analysis (see task "definition of analysis scope");
5. the mission phases to be considered (see task "definition of analysis scope");
6. the design blocks (see task "hierarchical design decomposition").

5.9.2 Contents

5.9.2.1 Sneak circuit reports

a. 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 in using information 5.9.1a.3 to 5.9.1a.6 for this purpose.

5.9.2.2 Grouping of sneak circuit reports

a. 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.

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

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

5.9.2.3 Issue of recommendations

a. In collaboration the 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.

5.9.3 Outputs

a. The sneak circuit reports shall be issued in conformance with Annex A.

Annex A (normative) Sneak analysis report - DRD

A.1 DRD identification

A.1.1 Requirement identification and source document

This DRD is called from ECSS-Q-TM-40-04, requirement 5.9.3a.

A.1.2 Purpose and objective

The purpose of the sneak analysis report is to:

a. list all the documents that have been used and/or issued during the previous tasks;

b. describe all the problems that have been identified during the analysis;

c. describe the recommendations that have been issued to solve these problems.

A.2 Expected response

A.2.1 Scope and content

<1> Identification

a. The sneak analysis report shall contain a document number, issue, revision and date of reference and applicable documents (including reference to the sources of clue list).

b. The sneak analysis report shall identify the parts (section, paragraph, page) of the documents (number, issue, revision, date) that are considered relevant as input information for the analysis.

<2> Introduction

a. The sneak analysis report shall contain an introduction recalling and justifying the analysis scope (items to be analysed, operational modes to be considered, depth of analysis).

<3> Summary of the requests for clarification

a. The sneak analysis report shall give a summary of the requests for clarification issued, their answers and status.

<4> Results of the hierarchical decomposition of the design

a. The sneak analysis report shall present the results of the hierarchical decomposition of the design (or a reference to the documents containing them.

<5> Input/Output matrices

a. The sneak analysis report shall include the input/output matrices (or a reference tot the document containing them or a reference to the documents containing them.

<6> Sources and targets

a. The sneak analysis report shall include the:

   1. list of targets

   2. list of sources

b. The sneak analysis report shall list the:

   1. intended relations between sources and targets

   2. undesired relations between sources and targets considered during the analysis

<7> Restrictions , gathering of other sneak analysis reports and other problems

a. The sneak analysis report shall provide the list of cases where lack of data was found in the input data.

b. If this sneak analysis report is the result of a gathering of several sneak analysis reports , it shall provide the list and references of all other sneak analysis reports.

c. The sneak analysis report shall provide a list and description of the others problems that although not classifiable as sneak circuits or design concerns can lead to an undesirable impact on system safety and/or reliability and are not identified by other RAMS analyses.

<8> Findings and recommendations

a. The sneak analysis report shall provide all findings, recommendations and status (coming possibly form other sneak analysis reports.

<9> Conclusion

a. The sneak analysis report shall provide the new clues synthesised during the analysis (if any).

b. The sneak analysis report shall provide a summary table of sneak circuits and design concerns and their status outlining the major problems found.

<10> Approval of the report

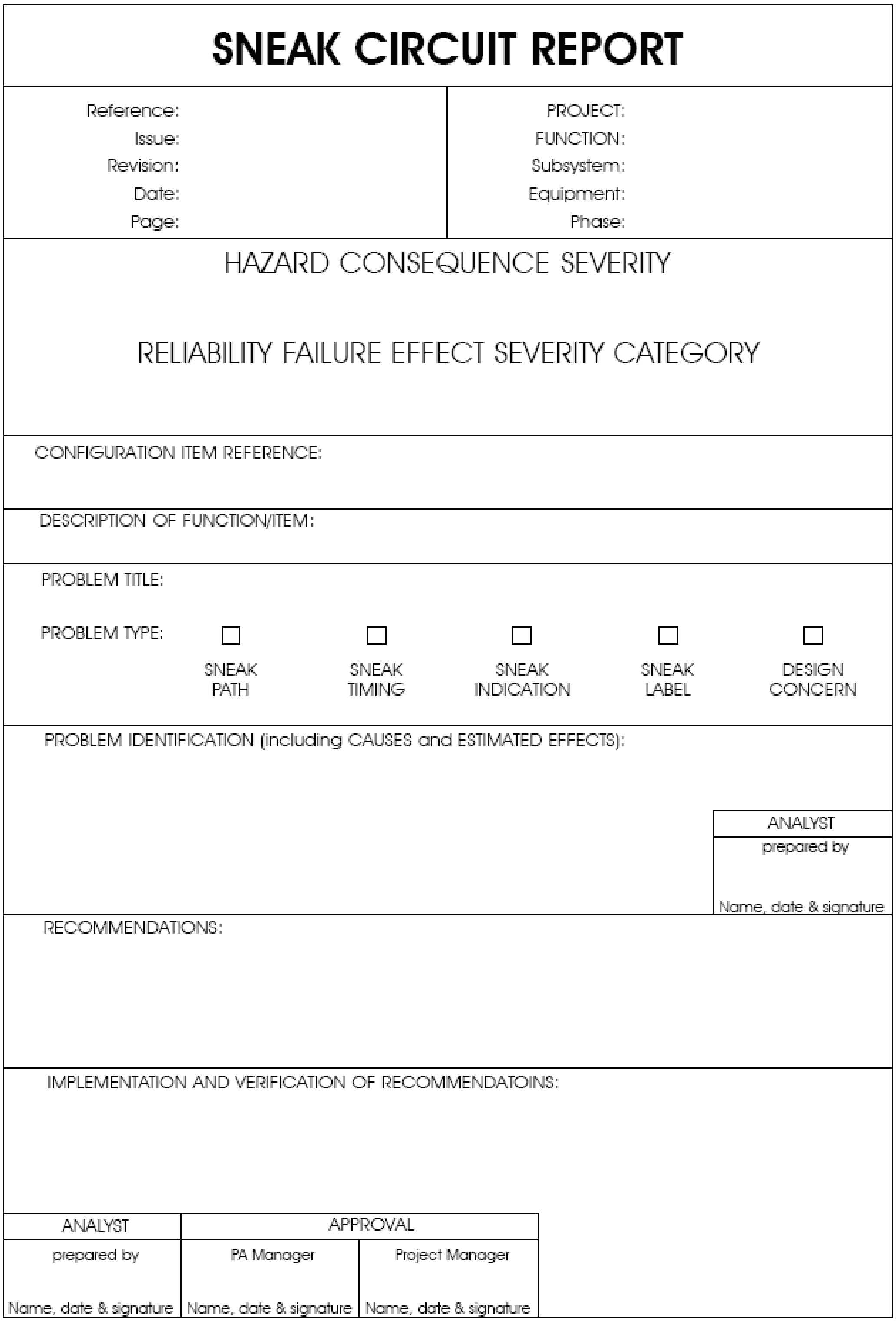
a. The sneak analysis report shall be:

1. signed by the analyst(s), and
2. approved by the supplier's product assurance manager and project manager.

A.2.2 Special remarks

a. 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 given in Bibliography. A short comment has been added after certain papers to explain the objective of the paper (if this is not clear from the title).

b. An example of a report form is given in Figure A-1.



SNEAK CIRCUIT REPORT

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

Figure A-1: Example of a Sneak Circuit Report form

Annex B (informative) Example of path clues identification

The following example shows how the path clues can be derived.

B.1 Assumptions and logic for generic clue

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?

NOTE If the relation is time-dependent (i.e. the targets are "on" or "off only during a certain time interval) then the following generic clues are 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?

B.2 Identification of the specific path clues

a. The following steps are followed in order to identify the (specific) path clues for a given number of binary targets and sources:

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".

3. in order to derive the specific clues, apply, for these identified states:

   (a) the generic clue as described in B.1.a

   (b) the generic clue as described in B.1.c if the relation is time dependent to derive the specific clues;

4. identify all the unwanted set states in which at least one of the targets is "off".

5. in order to derive the specific clues , apply, for these identified states:

(a) the generic clue as described in B.1.b

(b) The generic clue as described in B.1.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 "(as described in B.1.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 C (informative) Sneak analysis applied to computer software

C.1 Objective

The objective is to identify "facilitation conditions" related to software which can lead to the unwanted activation or deactivation of equipment

C.2 Inputs

a. The following inputs should be used:

1. the list of items that are included in the scope of the analysis (see task "definition of the analysis scope");
2. the design decomposition of the above items (see task "hierarchical design decomposition");
3. the relevant input/output matrices (see task "synthesis of input/output matrix);
4. the results of the "top level" RAMS analysis (e.g. preliminary hazards analysis, functional failure analysis);
5. the documentation relevant for the items under analysis (see task "data gathering").
6. 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.3 Contents

C.3.1 Preparation

a. By reviewing the results (see C.1a.6) of sneak path analysis at HW/SW level, the hardware items that are controlled by software are identified;

b. using the documentation (see C.1a.5) on the HW/SW interfaces and on the software, the control commands and data issued by the software to the hardware items are identified;

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

C.3.2 Sneak path analysis in computer software

a. 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 rules apply (as it generally the case nowadays):

   (a) the software is composed of modules;

   (b) each software have low cyclomatic complexity (e.g. less 10).

   NOTE If for the software under analysis the two above quoted rules are not applicable, then appropriate heuristic criteria 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.4 Outputs

a. The following outputs are derived:

1. list of intermediate targets;
2. facilitation conditions in the software that lead to unwanted or untimely activation (or deactivation) of the intermediate targets;
3. list of cases where improper sequence of commands or improper output data is produced by the software.

Bibliography

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

NOTE Sections 3, 4, 5 of this report contain guidance and examples about the application of the sneak path analysis (see section 5.5 of this document). The report is available in ESA/ESTEC

[2] Proceedings of the sneak analysis workshop, ESA/ESTEC WPP-033, Noordwijk, June 1992

NOTE 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 document. These proceedings are available at the ESA/ESTEC Technical documentation centre.

[3] Sneak circuit analysis. A means to verify design integrity, USA Department of the Navy, NAVSO P-3634, August 1987

NOTE 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 document. This report can be of use with respect to the tasks described in sections 5.1, 5.2, 5.4 and 5.6 of this document.

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

[5] 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

NOTE This paper can be of use with respect to the task described in Annex C of this document.

[6] De Mateo, G., Application of sneak analysis to hydraulic systems, ESA/ESTEC EWP-1801, Noordwijk, October 1994.

NOTE 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 sections 5.4 and 5.5 of this document. This report is available through the ESA/ESTEC Technical documentation centre.