ESA PSS-01-411 Issue 1
January 1994

Sneak analysis methods and procedures for ESA space programmes

Prepared by:
Product Assurance & Safety Department
European Space Research and Technology Centre
Noordwijk, The Netherlands

Approved by:
The Inspector General, ESA

european space agency / agence spatiale européenne
8-10, rue Mario-Nikis, 75738 PARIS CEDEX, France

Published by ESA Publications Division,
ESTEC, Noordwijk, The Netherlands

Printed in the Netherlands.

ESA Price Code: E1

ISSN 0379-4059

ABSTRACT

This document contains a procedure for performing Sneak Analysis and defines the output required from the Sneak Analysis activities that are carried out within an ESA programme.

DOCUMENT CHANGE RECORD

Issue number and date	Sections affected	Remarks
Issue 1	All	New document

TABLE OF CONTENTS

PAGE INTENTIONALLY LEFT BLANK

SECTION 1. SCOPE

This document contains an outline of the Sneak Analysis method (Section 3) and provides a procedure (Section 4) for Sneak Analysis performance. This procedure is intended as a guideline for the Contractor that is to perform the Sneak Analysis. Alternative approaches proposed by the Contractor may be accepted by the Agency provided that proper rationale is submitted. This document also defines (Section 5) the output required from the Sneak Analysis activities that are carried out within an Agency programme.

PAGE INTENTIONALLY LEFT BLANK

SECTION 2. GENERAL

2.1 INTRODUCTION

This document describes a procedure for performing Sneak Analysis when required by ESA PSS-01-40 "System Safety Requirements for ESA Space Systems".

Sneak Analysis is aimed at identifying 'sneak circuits', i.e. unexpected paths for a flow of mass, energy 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.

2.2 PURPOSE

The purpose of this document is:

- to present a procedure for the performance of Sneak Analysis;
- to provide the requirements on the presentation of the results of Sneak Analyses that are carried out during ESA programmes (in the following ESA is referred to as the Agency);
- to provide those that are not familiar with Sneak Analysis with an introduction to its basic concepts and its input and output data (see Sections 2, 3 and 5).

2.3 APPLICATION DOMAIN

Sneak Analysis makes use of check-lists containing questions (hereafter called "clues") indicating the way through which design errors associated to one or more system components can lead to system malfunction.

Therefore the application domain of Sneak Analysis corresponds to the one that is covered by the "clue list". A basic set of clues that can be used as a reference is provided in ESA STM-252 'Clue List for Sneak Analysis'. The list of clues relevant for a given application shall be identified by the Contractor and shall be submitted to the Agency.

2.4 REFERENCE DOCUMENTS

The following documents are quoted as reference in this specification:

- ESA PSS-01-40 System Safety Requirements for ESA Space Systems

- ESA PSS-05-0 ESA Software Engineering Standards
- ESA STM-252 Clue List for Sneak Analysis

2.5 RESPONSIBILITIES

2.5.1 Contractor's responsibility

It is the responsibility of the Contractor that is charged by the Agency to design and/or manufacture, and/or operate and/or refurbish a space system (or parts of it) to implement the requirements of this document that contain "the Contractor shall/should...", according to the provisions of the Contract between the Agency and the Contractor.

2.5.2 Agency's responsibility

It is the responsibility of the Agency Project Management to implement the requirements of this document that contain "the Agency Project Management shall/should ..." .

SECTION 3. SNEAK ANALYSIS BASIC PRINCIPLES AND APPLICATION GUIDANCE

3.1 DESIGN ERRORS AND SYSTEM FAILURES

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 "review" are hardly reproducible by a different group of reviewers, making the review a loose (and creative) process rather than an algorithmic one. In order to obviate this difficulty to some extent, administrative procedures for the performance of the various "review" phases have been implemented.

In parallel with this, analytical techniques (i.e. Sneak Analysis) aimed at identifying design errors have also been developed to improve the reproducibility of the "reviews" with regard to their effectiveness and the reliability of their results.

3.2 SNEAK ANALYSIS TERMINOLOGY

A **design error** is 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.

Design errors can be classified according to:
- the hierarchical level at which they take place;
- whether they are associated with more than one of the 'atomic items' (at the lower hierarchical level of the design that is of interest during the reliability and/or safety analysis associated to a given system).

A **design concern** is the result of a misapplication (or omission of application) of a design requirement or rule to one atomic item.

A **sneak circuit** is an unexpected path for a flow of mass, energy 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. A sneak circuit is therefore the manifestation of a design error that involves more than one atomic item.

Sneak circuits can be classified as:

- **sneak paths**, i.e. unexpected paths along which mass, energy, or a logical control sequence flows in an unintended direction;

- **sneak timing**, i.e. 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 or logical control flow along intended paths without respecting the intended dynamic behaviour of the system;

- **sneak indications**, i.e. ambiguous or false displays of system operating conditions that may cause the system or an operator to take undesired actions;

- **sneak labels**, i.e. incorrect or imprecise labelling of system functions (e.g. controls, displays) that may cause an operator to apply incorrect stimuli to the system.

Sneak Analysis is a generic term for 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.

A **target** is an item the unwanted activation or inhibition of which can trigger an undesired event. A **source** is any item which can contain or control mass, energy or logical sequence.

Design Concern Analysis is a Sneak Analysis technique that is based on the application of clues to atomic items.

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

The following three classes of clues are mentioned in this document:

- **"path"** clues, which are used during Sneak Path Analysis and depend only on the kind of causal relation (e.g. flow of energy between source and target allowed or inhibited) between sources and targets that is under investigation. An example is: "Can the target be "off" when the source is "on"?" This can also be worded, for electrical systems, 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, which are dependent on the type of system (electronic, pneumatic, hydraulic, software), are used during Sneak Path Analysis. These clues are derived from experience and are related to those kinds of behaviour of a system component that can affect the flow of mass, energy or logical sequence between sources and targets. An example is (for switches): "During change of state of switches, can transitory current paths exist?"

- **"component"** clues, which are used during Design Concern Analysis, are also derived from experience. They are dependent on the type of system. These clues are related to those kinds of behaviour of system components that do not significantly affect (at least in a first approximation) the flow of mass, energy or logical sequence between sources and targets. An example is (for an integrated circuit): "Have the maximum frequency signal levels been taken into account?"

3.3 SNEAK ANALYSIS BASIC STEPS

The detailed procedure for performing Sneak Analysis is described in Section 4. In the following, its basic steps are first outlined and then illustrated in Figure 1.

The **preparatory tasks** are aimed at:

- the definition of the analysis scope, that is the identification of 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;

- gathering the data for the subsequent steps of the analysis (see Section 3.4. for a list of typical input documents);

- the 'decomposition' of the design into '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 into parts that are easily understandable and manageable by the analyst; establishing a clear relation between functions and 'blocks' of the design;

- documentation in the "input/output switching matrix" of 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 that is aimed at identifying sneak paths, sneak timings and sneak indications through:
  * identification of targets;
  * identification of sources;
  * tracing of paths between sources and targets;
  * application of "path" clues (to the paths) and "component+path" clues to the components contained in the path.

- the Design Concern Analysis that is aimed at identifying 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 the descriptive data about the sneak circuits and design concerns is documented on "sneak circuit reports" together with the recommendations concerning ways to eliminate them, and a "Sneak Analysis Final Report" is produced that documents input data, interim results and conclusions.

3.4 INPUT DOCUMENTS FOR SNEAK ANALYSIS

Apart from the clue list, the following documents should be considered as inputs for Sneak Analysis:

(at system level)
- System Requirement Specification
- System Design Specification
- User Requirement Document (for software)
- Software Requirement Document (for software)
- Hardware/Software Interface Specification
- User Manual (including operation procedures)
- Preliminary RAMS Analyses (e.g. Functional Failure Analysis, Preliminary Hazard Analysis)
- Functional Analysis

(at lower level, for hardware)
- Subsystem Requirement Specification
- Subsystem Design Specification
- Equipment Requirement Specification
- Equipment Design Specification (including drawings)
- Component Specification (data sheets)
-(s) Worst Case Analysis
-(s) Part Stress Analysis
-(s) Development Testing Results

(at lower level, for software [see ESA PSS-05-0 for the content of the documents listed below])
- User Requirement Document (URD)
- Software Requirement Document (SRD)
- Architectural Design Document (ADD)
- Detailed Design Document (DDD)
- (s)Software Transfer Document

According to the scope of the analysis and its depth some of the above documents might not be relevant (e.g. in the case of an analysis to be performed at equipment level only on the Power Distribution Subsystem).

The documents marked with "(s)" generally contain support information that might help to avoid duplication with other analyses/activities.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" is generally contained in the System (Subsystem, Equipment) Design Specification and (for software) in the Source Code and DDD.

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);
- part list.

In a DDD the following data are of use for Sneak Analysis:
- software architecture describing the software decomposition with functions, their inter-relationships and sequencing;
- for each software 'item': function, subordinates, dependencies, interfaces, resources, processing, data;
- source code listing.

3.5 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. Application cost.
d. "Delta" analysis due to design changes.
e. Tailoring of the approach according to the application domain.

a. Concerning the **expertise**, it is important that the Analysis Team contain at least one design specialist in the domain (e.g. electrical, electronics) to which the system to be analysed belongs. 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 "non problems" raised during the analysis and to synthesise the recommendations for the design changes needed to eliminate the sneak circuits.

b. 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) helps 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 enables the required cost and time to be reduced.

c. On the **cost** of applying Sneak Analysis manually, it can be said that, as a first approximation, it is comparable to that of an FMECA performed at the same level of detail. Obviously, the availability of computerised tools to perform one or more Sneak Analysis tasks can lead to savings with respect to the manpower effort required in a purely manual Sneak Analysis.

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

e. The procedure described in Section 4 is fairly general and has been worded in such a way that application in several domains is in principle possible (if the clue list covers these domains, see Para. 2.4).
For practical application in specific domains, the following should be taken into account:
- when Sneak Path Analysis is applied to digital systems, the availability of "digital simulators" is recommended in order to enable the complexity problems to be tackled. The simulator 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. without HW/SW interface within them) is not recommended when inspections and static and dynamic analysis are already required;
- Sneak Analysis should be applied 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 so complicated (at a low detailed level) that it has to be represented in a "simplified way" (e.g. a system made of 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 (possibly supplemented by the application of the "component+path" clues).

Finally, it is noted that Sneak Analysis is particularly well suited for electrical systems and electronics system consisting of discrete and relatively few integrated circuits.

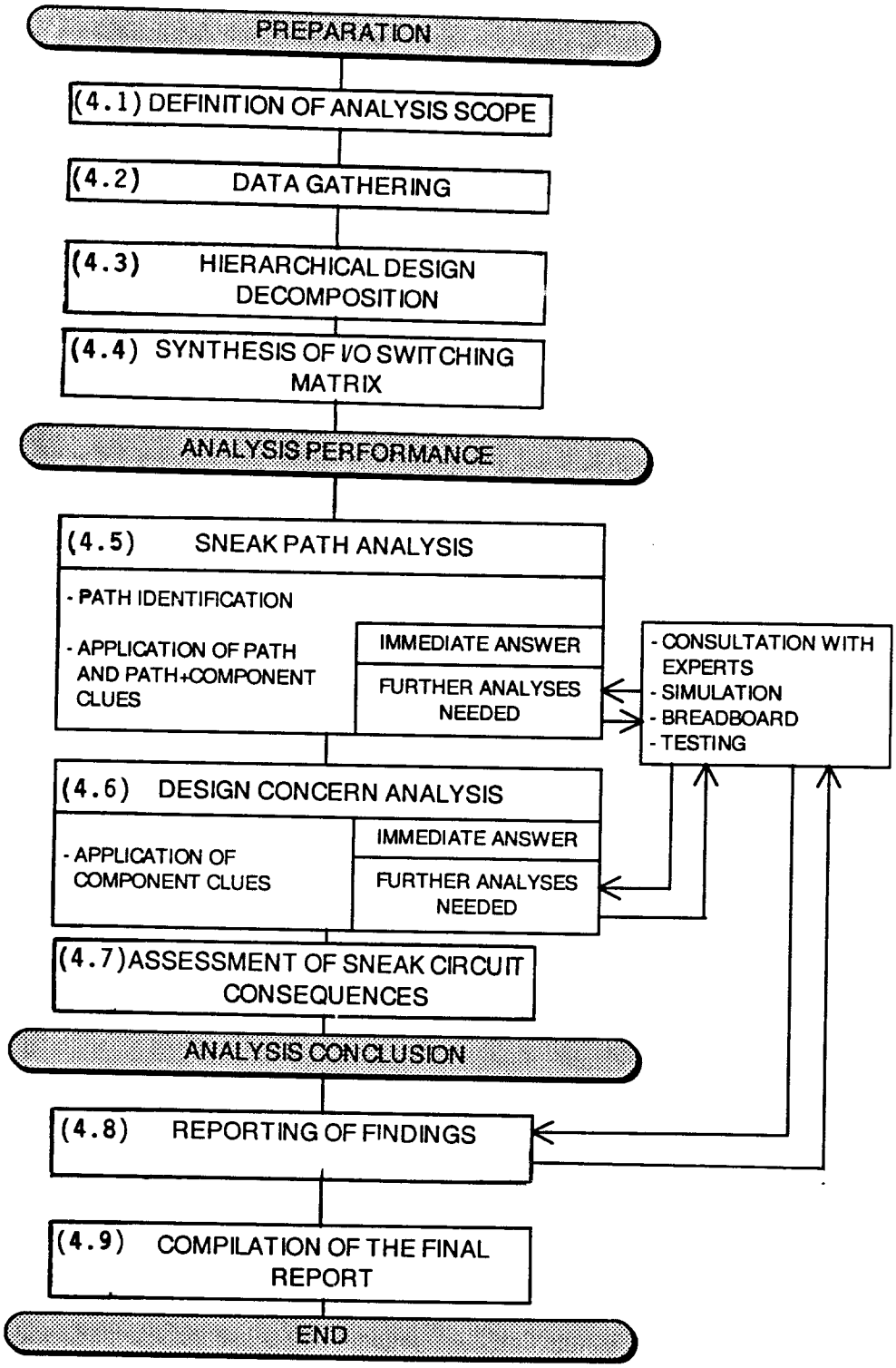


FIGURE 1 - OUTLINE OF SNEAK ANALYSIS PROCEDURE

SECTION 4. GUIDELINES FOR THE PERFORMANCE OF SNEAK ANALYSIS

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

- preparation;
- analysis;
- reporting and conclusions.

The relations between the main Sneak Analysis tasks are presented in Figure 1. The following pages describe the contents of the various tasks. Each task description, after a briefly stating the objective of the task, deals with:

- **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 described;
- **outputs**, where the information that is expected to be produced as output from the task is identified.

4.1 DEFINITION OF THE ANALYSIS SCOPE

This task is aimed at identifying the items of the system and phases of the mission that are to be analysed.

INPUTS:

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 (see Section 3.4).

c. The results of other RAMS analyses such as Preliminary Hazard Analysis and Functional Failure Analysis.

d. The criteria listed below in Table 1.

CONTENTS:

To determine the items subject to analysis, the following steps are to 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 shall be applied to Safety Critical Functions") and by means of the data contained in the documents quoted under b. and the results of the RAMS analyses quoted under c., define a list of functions and/or items that are to be analysed. Define also the phases of the mission that are to be considered. Go to step 4.

3. By means of the result of RAMS analyses quoted under c., identify the list of safety and reliability critical functions. Apply the "screening criteria" contained in Table 1 to the items contained in the safety or reliability critical functions, starting from the highest level of hierarchical decomposition that is of interest. Synthesise the results obtained through the application of the screening criteria.

4. To define the depth of the analysis, take into account the results of step 2. (or 3.) and, according to the available documentation, check whether the analysis can also be performed at lower levels (e.g. subsystem, assembly, equipment or component level). This can be done by applying the criteria of Table 1 at the relevant level.

OUTPUTS:

Upon completion of this task, the items that are to be analysed and the relevant mission phases within the system have been identified. The depth that is to be reached during the analysis has also been defined.

TABLE 1: SCREENING CRITERIA

Safety and Reliability Consequences

- Does the loss or inadvertent activation of the item lead to catastrophic or serious safety consequences?
- Does the loss or inadvertent activation of the item lead to loss of the mission?

Design aspects

- Is testing under all operating modes possible and/or 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 other items and/or functions?
- Has the item several modes of operation?

Programmatic aspects

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

4.2 DATA GATHERING

This task is aimed at collecting the input data necessary for the performance of a Sneak Analysis.

INPUTS:

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 (listed in Section 3.4 of this document) containing the design and operations data for the above quoted items.

CONTENTS:

1. SCREENING OF AVAILABLE DOCUMENTATION

Identify the parts of the documents quoted under c. that are relevant to the items a. at depth b.

2. HOMOGENEITY CONTROL

Check from a configuration management point of view the homogeneity of the documents that have been gathered under 1.

3. FAMILIARISATION WITH THE DOCUMENTATION

Become familiar with the documentation screened under 1. During this process, there might be the need to ask for clarification to the authors of the documents. If the analysts are not familiar with some key design or technology issues addressed in the documentation, a bibliographic research should be performed or "experts" on these issues might be consulted.

4. DOCUMENTATION OF THE FINDINGS

Document the outcome of the previous subtasks. In particular, make sure that:

- the points related to lack of documentation homogeneity (see 2.) or the requests for clarification are discussed with the PA and Engineering functions;
- the interfaces (between items or between elements of an item) are unambiguously described as pertains to:
  * identification;
  * 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.

Particular attention should be given to the interfaces between hardware and software items. Real time issues, software asynchronous behaviour and constraints on control flow sequence should be identified.

OUTPUTS:

- The parts of the documentation that are relevant for the analysis.
- The points where lack of homogeneity has been identified.
- The requests for additional information or clarification.
- The definition of the interfaces of the items that are to be analysed.

4.3 HIERARCHICAL DESIGN DECOMPOSITION

This task is aimed at producing a 'decomposition' of the design of the items that are subject to Sneak Analysis into 'blocks' associated with the items' functions. 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 Task 4.7).

This task is different from a pure Functional Analysis. In fact, the Functional Analysis deals with functions without considering solutions. However, once Phase C has begun, a detail design is chosen to fulfil these functions. It then becomes possible to define design blocks associated with these functions.

It is important to note that if a hierarchical decomposition compatible with the one described in the following is already available as a result of other PA or engineering activities (e.g. in the form of SADT, SART, Data flow diagrams), this task need not be performed.

INPUTS:

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 a. (see task "Data gathering").

d. The items' interfaces (see task "Data gathering").

CONTENTS:

This task is performed on the items within the scope of the analysis (see task "Definition of the analysis scope").

1. APPROACH TO DESIGN DECOMPOSITION

Perform successive decompositions down to the level that is above the lowest one that is of interest for Sneak Analysis (e.g. for

an electronic system if the analysis is to performed at 'component' level, the decomposition should arrive at board level). When this stepwise decomposition is performed, 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.

Identify precisely the kind and the origin of the 'causality' flows (e.g. data, electric current) associated with each block.

2. DESIGN DECOMPOSITION

Take an item and by looking at the documentation c., 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 'interactions' (e.g. in terms of data, electric current) between the various 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.

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.

To avoid confusion it is advisable that the names of the signals used in the documentation should also be used in the blocks.

3. DOCUMENTATION OF THE DESIGN DECOMPOSITION

For each block, the following information should be properly documented (if not already available):

- 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.

To this end, use the documents from input d.

At the end of this task, check that the diagrams produced during subtasks 2. and 3 match.

OUTPUTS:

A hierarchical decomposition into blocks of the items to be analysed, where each block is defined in terms of functions, inputs and outputs, and 'interactions' with the other blocks.

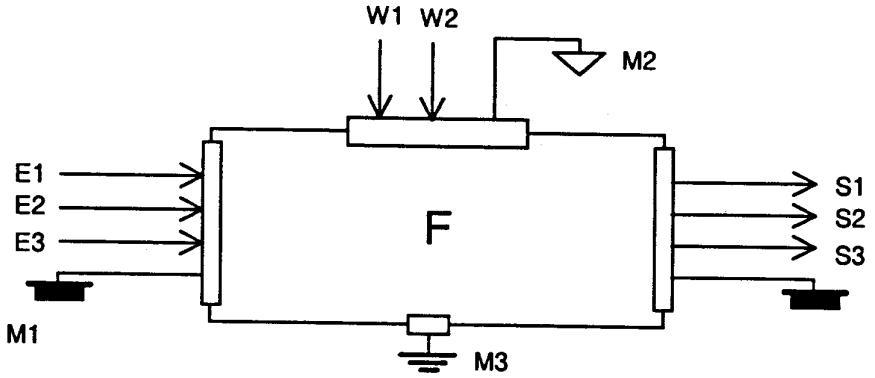
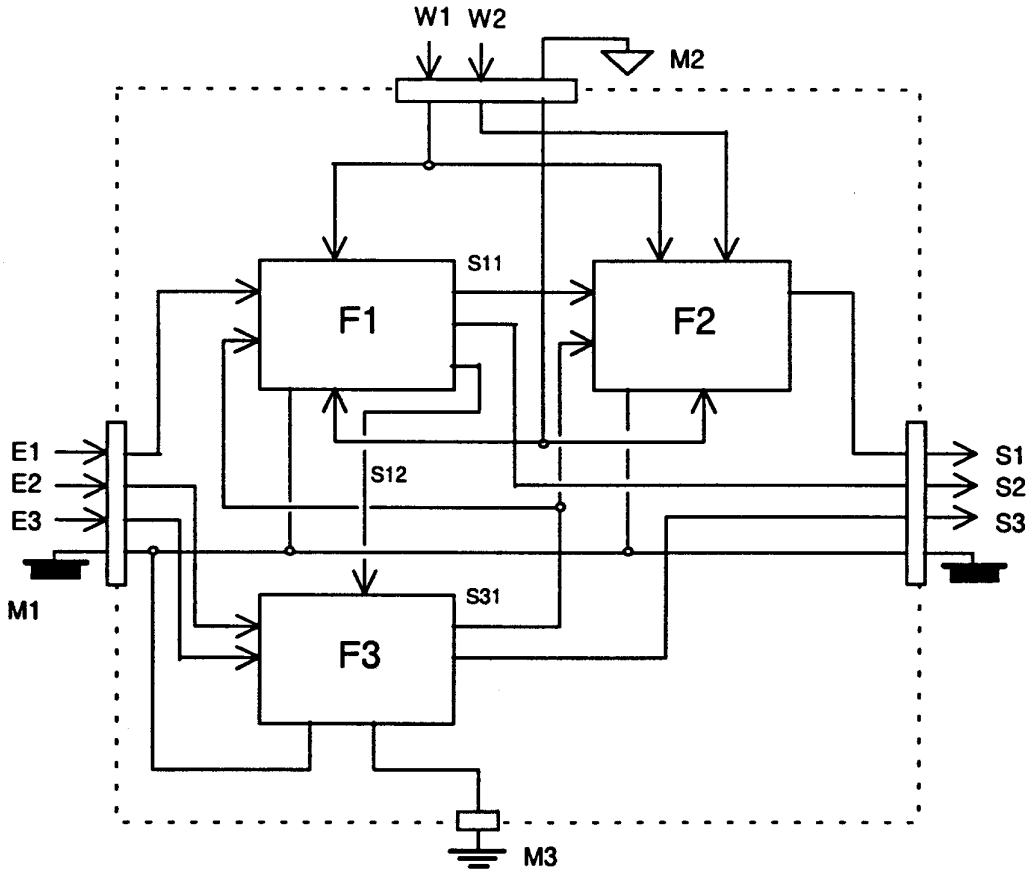


FIGURE 2A



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

FIGURE 2B

FIGURE 2 - ILLUSTRATION OF DESIGN DECOMPOSITION

4.4 DYNAMIC ASPECTS: SYNTHESIS OF INPUT/OUTPUT SWITCHING 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.

If the above information is already available as a result of PA or engineering activities in a format that is compatible with the one described in the following, this task need not be performed.

INPUTS:

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").

CONTENTS:

1. DOCUMENTATION OF OPERATIONAL MODES

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

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

2. IDENTIFICATION OF ELEMENTARY "SWITCHING" EVENTS

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

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

3. INPUT/OUTPUT STATES VERSUS OPERATIONAL MODES 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 is provided in Figure 3.

4. INPUT/OUTPUT STATES VERSUS OPERATIONAL MODES AT THE LOWER LEVELS OF DESIGN DECOMPOSITION

In some cases, it might be useful to built an input/output switching matrix for some of the lower level blocks. The selection of these blocks is to be done on a case-by-case basis. Criteria that should 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 input/output switching matrix can be built in a way similar to the one presented through the sub-tasks 1 to 3. The operational modes are the same as the ones identified there.

Discipline should be exercised when identifying input/output switching matrices at low level of design decomposition in order to avoid a 'combinatorial explosion' of the number of entries in these matrices.

OUTPUTS:

- Input/output switching matrix at first level of design decomposition.

- Identification of instances of simultaneous switching.

- Input/output switching matrix for some of the lower level blocks (and list of relevant blocks).

INPUT/OUTPUT SWITCHING 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

FIGURE 3 - EXAMPLE OF INPUT/OUTPUT SWITCHING MATRIX

4.5 SNEAK PATH ANALYSIS: PATH IDENTIFICATION AND CLUE APPLICATION

This task is aimed at identifying sneak circuits, mainly sneak paths, sneak timings and sneak indications.

INPUTS:

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 switching matrix (see task "Dynamic aspects-Synthesis of input/output switching matrix").

d. The results of "top-level" RAMS analyses (e.g. Preliminary Hazard Analysis, Functional Failure Analysis).

e. The documentation relevant for the items under a. (see task "Data gathering").

CONTENTS:

1. IDENTIFICATION OF THE TARGETS

Within the items that are to be analysed, "targets" for the Sneak Path Analysis are to be identified. This is done, for each planned operational mode, by identifying the safety or reliability critical outputs that are either required or to be inhibited. Obviously, in carrying out the above, use should be made of the results of b. and d. (if available).

2. IDENTIFICATION OF THE SOURCES

Identify the resources (e.g. electrical current) that are to be studied in connection with the targets. Then identify the "sources" (e.g. batteries) of these resources within the items under study. Note their dependence (if any) on the operational modes.

3. IDENTIFICATION OF THE INTENDED AND UNDESIRED CAUSAL RELATIONSHIPS ('PATH CLUES') BETWEEN SOURCES AND TARGETS

Identify, for each operational mode, the intended causal relations between the states of the sources and the ones of the targets. Use the data contained in inputs b. and c. for this purpose.

Then identify the unintended causal relations (i.e. by definition all the relations in the source target space that are different from the

intended ones). Pinpoint the undesired causal relations (i.e. the subset of the unintended ones that lead to the loss or the inadvertent activation of the targets). It is noted that these 'undesired causal relations' are the 'path clues' (see also Paragraph 3.2 and Annex A) that are relevant for the sources and target under examination.

Note also that in this way those unintended relations between sources and targets associated to "benign" effects are screened out.

4. IDENTIFICATION OF THE "RESOURCE" PATHS

Identify all the paths that link the sources to the targets through the system 'structure' (e.g. electrical drawings) for each operational mode. The analyst can thus "follow" the flow of the resource on the path.

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

The procedure for path identification is:

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 that can be found between the target and the source, and that are compatible with:
   - the input/output switching matrix;
   - the characteristics of the components between the source and the target (e.g. for electrical systems: a diode allows current to flow in only one direction);
   - the undesired relation ('path clue') 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:
   - 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);
   - if the paths allow the undesired relation to occur then a potential sneak circuit has been found. Consult the designers to check whether the problem is a real one. If this is the case, the sneak circuit consequences are to be assessed (see task "Assessment of sneak circuit consequences").
   - if it cannot be decided whether there is a sneak circuit or not, perform the subtask "Component+path clue application" (see 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. 'COMPONENT+PATH' CLUE APPLICATION

When it cannot be decided merely on the ground of the application of 'path clues' whether or not a sneak circuit is associated to a path, perform a more detailed study of the behaviour of the components on the path. To this end apply the "component+path" clues.

For each clue, apply the following sequence.

a. 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, software implementing a control algorithm) 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; or
   - performing a simple quantitative analysis (bounding calculation).

b. 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.

c. If the clue under examination does not lead to a sneak circuit, go to the next clue.

d. If a potential sneak circuit is detected, consult the designers to check whether the problem is a real one). If this is the case the sneak circuit consequences are to be assessed (see task "Assessment of sneak circuit consequences").

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 "timing problems" (e.g. races), in most cases, a "timing analysis" is necessary. This analysis, especially for digital items, will require the availability of relatively sophisticated simulation tools. Only in the simpler cases is the manual use of "time line" diagrams sufficient.

It should be noted that sneak labels are identified through the application of component clues in the "Design Concern Analysis" task.

OUTPUTS:

- List of targets.
- List of sources.
- Intended relations between sources and targets.
- Undesired relations (path clues) between sources and targets.
- Sneak circuits.

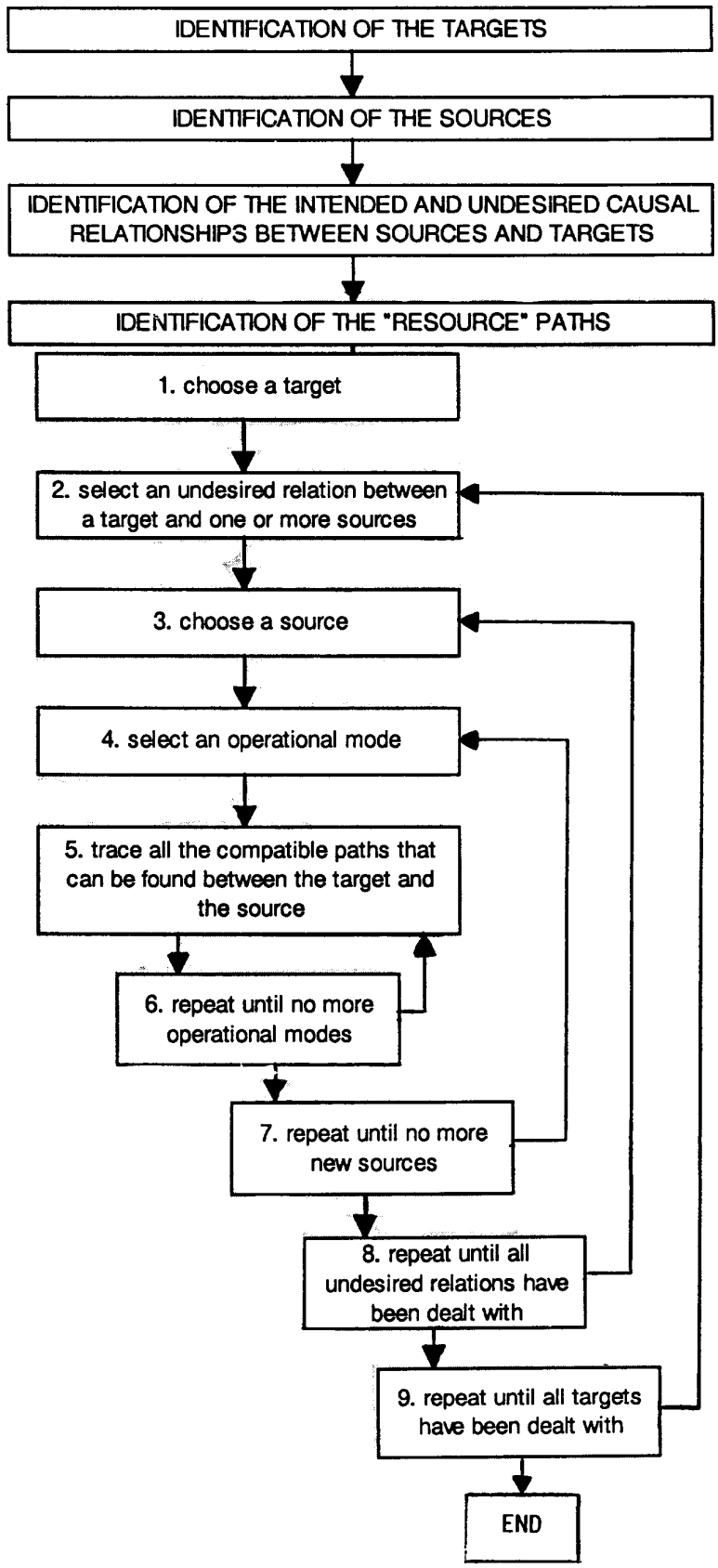
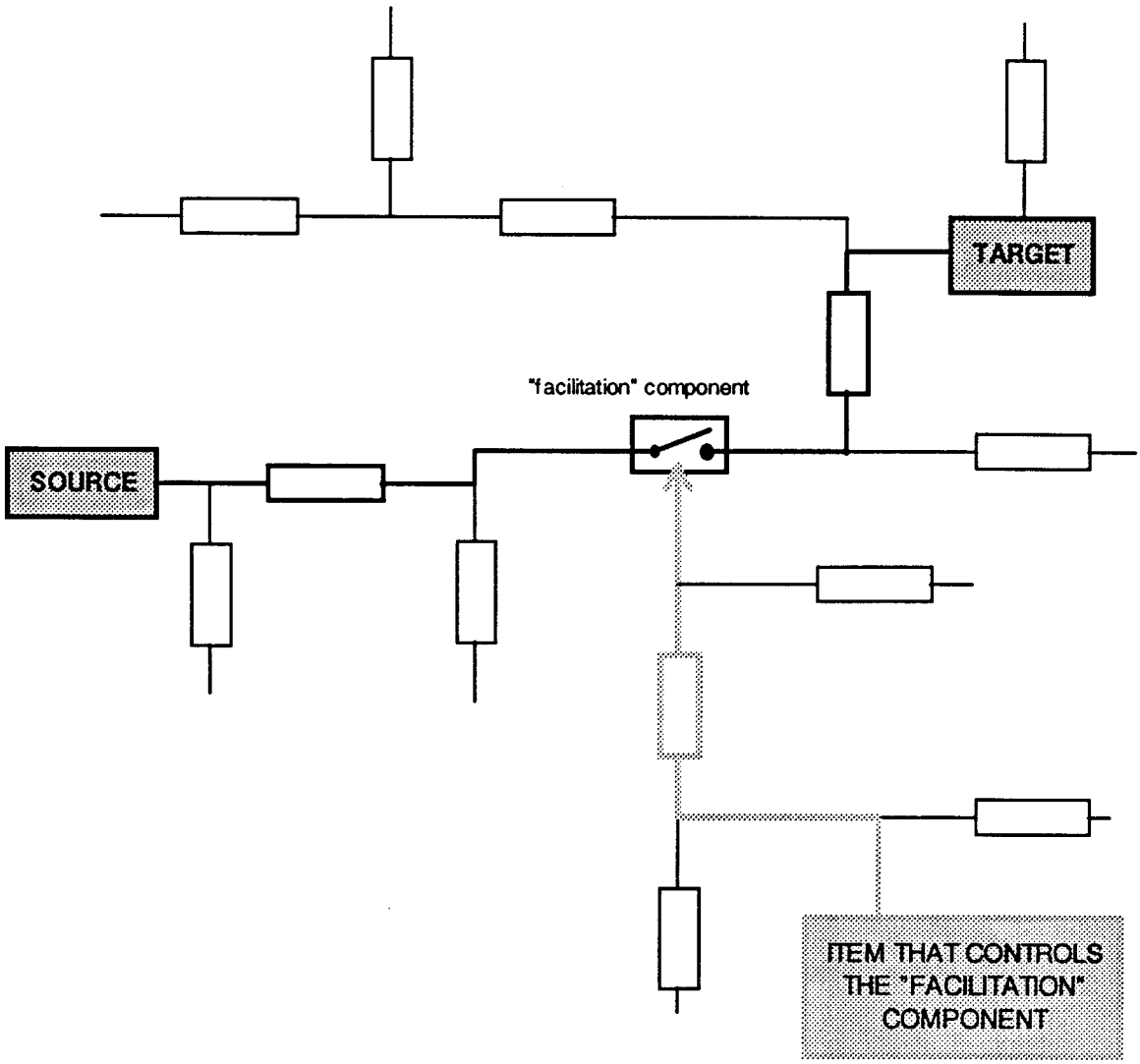


FIGURE 4 - PATH IDENTIFICATION PROCESS



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

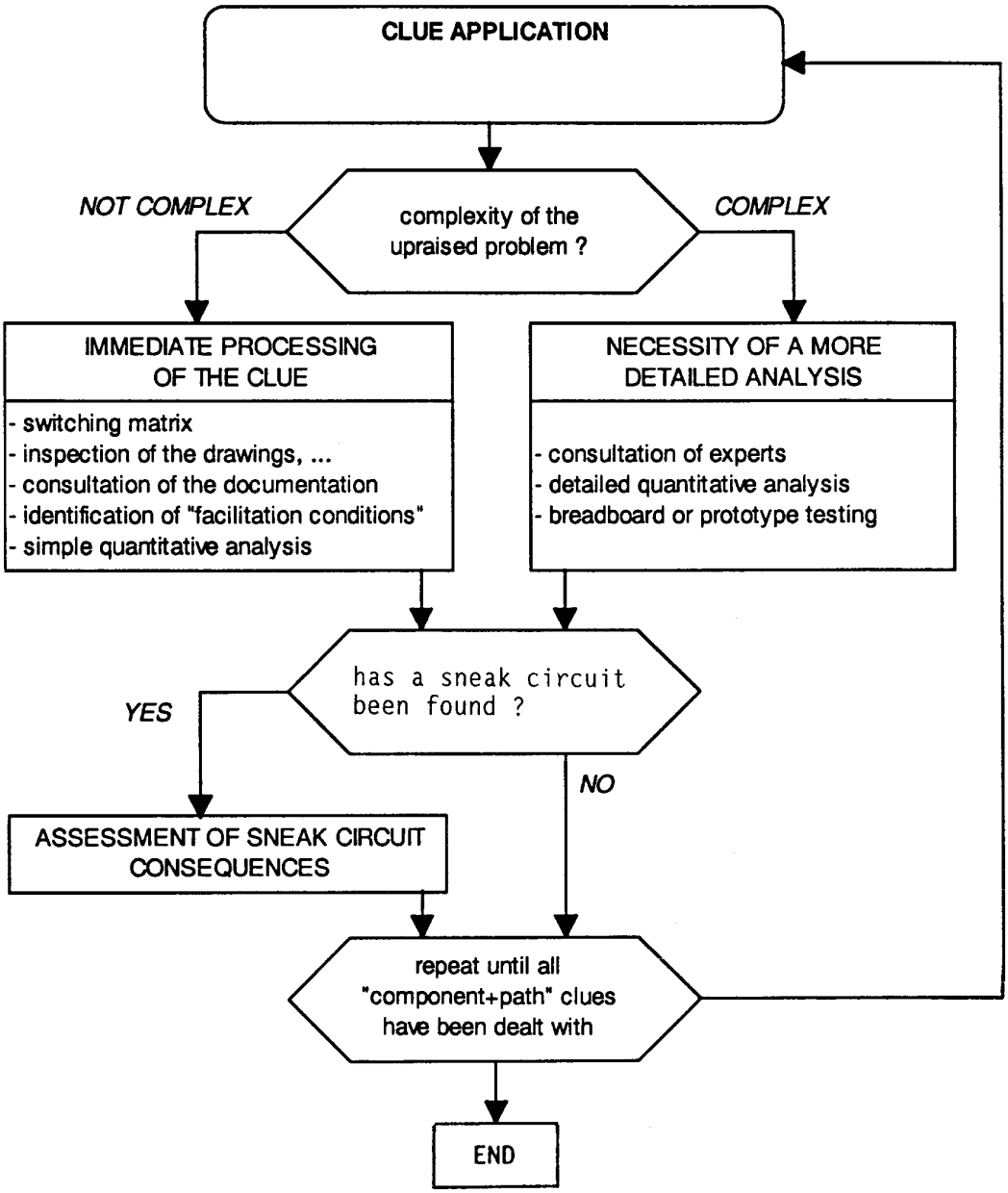


FIGURE 6 - ILLUSTRATION OF ANSWERING PROCESS TO "COMPONENT+PATH" CLUES

4.6 DESIGN CONCERN ANALYSIS

This task is aimed at identifying design concerns and sneak labels.

INPUTS:

a. The list of atomic items (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).

CONTENTS:

1. Select an item from the list under a.

2. Select an operational mode from the list under c.

3. Apply the "component" clues that match the characteristic of the item and are relevant during the selected operational mode.

4. Repeat steps 2-3 until there are no more operational modes

5. Repeat steps 1 to 4 until there are no more items.

For each clue, try to provide an answer by applying the sequence outlined in Figure 7. During the application of the component specific clues, drawing errors or missing information can be detected. Treat drawing errors as design concerns. Flag missing information instances to Engineering.

OUTPUTS:

- Design concerns.

- Missing information instances.

- Sneak labels.

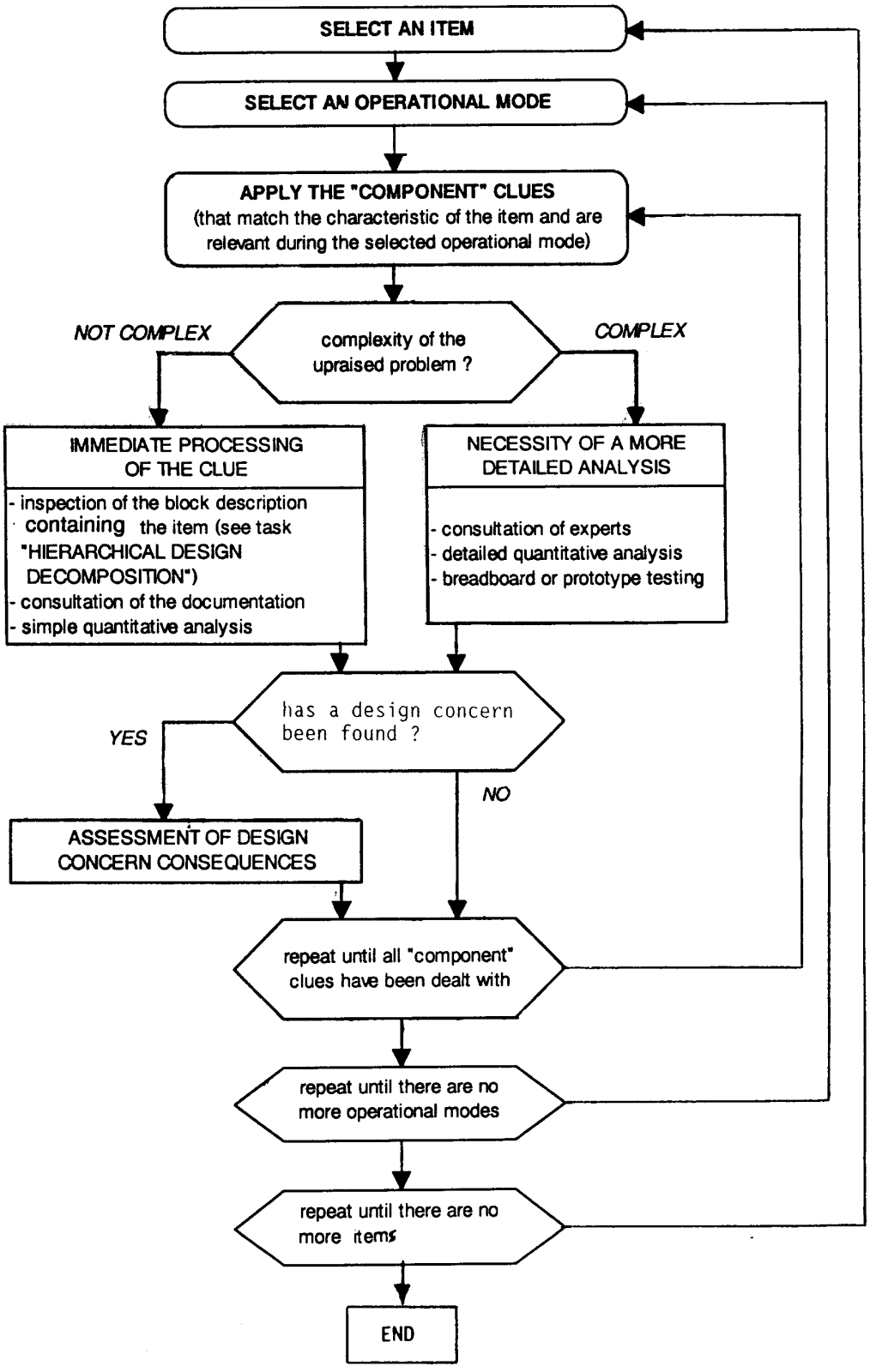


FIGURE 7 - ILLUSTRATION OF DESIGN CONCERN ANALYSIS AND ANSWERING PROCESS TO "COMPONENT" CLUES

4.7 ASSESSMENT OF SNEAK CIRCUIT CONSEQUENCES

This task is aimed at assessing the consequences of a sneak circuit or design concern up to the higher level of design decomposition that is of interest.

INPUTS:

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: Path identification and clue application" and "Design concern analysis").

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

CONTENTS:

1. Select a sneak circuit or design concern (and retrieve the operational mode(s) under which it 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 that 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 avoid duplications when issuing recommendations for the elimination of the sneak circuits or design concerns.

OUTPUTS:

- Consequences on safety and/or reliability of each sneak circuit and design concern.

4.8 REPORTING OF FINDINGS

This task is aimed at reporting about the identified sneak circuits and design concerns. This reporting covers also the recommendations for their elimination.

INPUTS:

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").

CONTENTS:

1. SNEAK CIRCUIT REPORTS

   Prepare, 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. Use information c. to f. for this purpose.

2. GROUPING OF SNEAK CIRCUIT REPORT

   Gather the reports related to the same items (system, subsystem, assembly, equipment, component). Check 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 this case, fill in a new Sneak Circuit Report form. Re-assess (if necessary) the safety and reliability consequences associated with the "new" sneak circuit.

3. ISSUE OF RECOMMENDATIONS

   In collaboration with Engineering, study the options for elimination of the identified problems. Complete the Sneak

Circuit Reports with the most appropriate recommendation for elimination.

Note that according to the results of this review there might be the need to: delete one or more reports; or modify one or more reports; or start some specific study on 'how' a sneak circuit or design concern is to be eliminated. Eventually, for each identified design change, it will also be necessary to obtain evidence that the change has been properly implemented (and does not introduce a new sneak circuit). Concurrence of the PA Manager and the Project Manager on both the recommendation for design changes and their implementation is obviously to be obtained.

OUTPUTS:

Sneak Circuit Reports including: recommendations for eliminating sneak circuits and design concerns; and concurrence by the appropriate project functions on these recommendations.

4.9 COMPILATION OF THE FINAL REPORT

The Sneak Analysis Final 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.

INPUTS:

All the outputs of the previous tasks.

CONTENTS:

The contractor will compile the Sneak Analysis Final Report, including as a minimum the information listed in Section 5.

OUTPUTS:

- Sneak Analysis Final Report.

SECTION 5. SNEAK ANALYSIS FINAL REPORT: DOCUMENT REQUIREMENT DESCRIPTION

The Contractor shall produce a Sneak Analysis Final Report containing:

- document number, Issue, Revision and Date of reference and applicable documents (including reference to the sources of clue list);
- an introduction recalling 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 switching matrices (or reference to the document containing them);
- the sneak path analysis interim results (list of targets, list of sources, intended relations between sources and targets, undesired relations between sources and targets);
- the missing information instances;
- the Sneak Circuit Reports, their associated recommendations and status: A Sneak Circuit Report shall contain all the descriptive entries mentioned in the form contained in Annex B. Continuation sheets may be used if more room is needed for some entries. The sneak circuit 'descriptive part' (i.e. up to the 'problem identification' entry) shall be signed by the analyst(s). In general, the 'recommendations for design changes' associated with a sneak circuit should be signed by the Analyst(s) and approved by the Contractor's Product Assurance Manager and Project Manager. In particular, when a sneak circuit leads to the identification of a non-compliance of the current baseline configuration (i.e. the last one submitted to the Agency) with the applicable requirements, the 'recommendations for design changes' shall be handled by the Contractor according to the 'Change Request procedure' applicable in the project concerned. The Contractor shall be responsible for the correct implementation of the "recommendations for design changes".
- 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.

The Sneak Analysis Final Report shall be signed by the Analyst(s) and shall be approved by the Contractor's Product Assurance Manager and Project Manager.

ANNEX A
DERIVATION OF PATH CLUES

In the following it is illustrated 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) ones:
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 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).

PAGE INTENTIONALLY LEFT BLANK

ANNEX B
SNEAK CIRCUIT REPORT FORM

SNEAK CIRCUIT REPORT

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

HAZARD CONSEQUENCE SEVERITY:

CATASTROPHIC SERIOUS MARGINAL

RELIABILITY FAILURE EFFECT SEVERITY CATEGORY

LOSS OF SYSTEM | LOSS OF MISSION | MISSION DEGRADATION | UNSCHEDULED MAINTENANCE

CONFIGURATION ITEM REFERENCE: *"an aggregation of hardware or software, or any of its discrete portions which serves an end use function and is designated for Configuration Management"*

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):

"a sketch of the problem is generally useful"

ANALYST
prepared by
Name, date & signature

RECOMMENDATION FOR ELIMINATION:

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

If a sneak circuit or design concern leads to a noncompliance of the current baseline configuration with the applicable requirements, this part of the form can be used for concurrence signatures according to the Change request procedure for the project concerned'

PAGE INTENTIONALLY LEFT BLANK

ANNEX C
SNEAK ANALYSIS APPLIED TO COMPUTER SOFTWARE

Many hardware systems also contain computer software for control purposes. The software performs switching functions and is important for Sneak analysis. Consequently, as a complement to the Sneak Path Analysis task outlined in Paragraph 4.5, there might be the need to apply Sneak Path Analysis to the software part of the items under analysis.

This task is aimed at identifying sneak circuits and/or facilitation conditions which can lead to the unwanted activation or de-activation of equipment. It is important in such an analysis to focus attention on the most critical parts of the software, that is those which activate equipment in the hardware system.

It is noted that in the following the term 'logical condition' is used to indicate the condition under which a certain path can be followed. For example, if a path traverses the statement "IF x>0 THEN ...", the logical condition for the "YES" branch is "x>0".

INPUTS:

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

b. The design blocks (see task "Hierarchical design decomposition").

c. The program code or the data flow diagram and functional specification for the software items in the scope of the analysis (see task "Data gathering").

d. The results of the "top level" RAMS analysis (e.g. Preliminary Hazards Analysis, Functional Failure Analysis).

e. The input/output switching matrix (see task "Dynamic aspects -Synthesis of input/output switching matrix").

CONTENTS:

1. PREPARATION

   Combine the hardware schematic and the flow chart or the data flow diagram for the software. Link the diagrams by drawing lines between output statements and the output register hardware symbols, and between the input statements and input registers.

2. IDENTIFICATION OF THE TARGETS

   Targets for the sneak analysis are identified in the hardware system in the way described in the task "Sneak Path Analysis: Path identification and clue application".

3. PATH TRACING

The procedure for path identification within computer software is:

1. choose a target in the hardware system;
2. trace a path in the hardware schematic between the target and the source as explained in Paragraph 3.5;
3. if along the above path a computer output register is found, trace a facilitation path in the software flow chart or data flow diagram, backwards from the computer output register to the software inputs (e.g. input registers, data initialisation instructions, operator commands, program start). Record the logical conditions necessary for the path to be followed;
4. check the path as it is built up by comparing with the 'input/output switching 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 clues to the software instructions (e.g. conditionals, loops, function calls, assignments) along the path (this might also lead to the identification of some design concerns);
7. assess whether the software can lead to the unwanted activation (or de-activation) of the target. In this case assess the consequences of the sneak circuit (see task "Assessment of sneak circuit consequences");
8. repeat steps 2 to 7 for all paths through the hardware and software;
9. repeat steps 2 to 8 for all targets.

OUTPUTS:

- List of targets.
- Sneak circuits.
- Facilitation conditions for activation for sneak circuits.
- Design concerns.

ANNEX D
GLOSSARY OF TERMS AND ACRONYMS

The definitions of terms contained in ESA PSS-01-40 are used. The Sneak Analysis terminology is introduced in Paragraph 3.2. The acronyms used in this document are as follows:

ADD	Architectural Design Document
CAD	Computer Aided Design
DDD	Detailed Design Document
ESA	European Space Agency
ESTEC	European Space Research and Technology Centre
FMECA	Failure Modes, Effects and Criticality Analysis
HA	Hazard Analysis
HW	Hardware
PA	Product Assurance
PHA	Preliminary Hazard Analysis
PSA	Part Stress Analysis
RID	Review Item Discrepancy
SADT	Structured Analysis and Design Technique
SART	Structured Analysis Real Time
SCA	Sneak Circuit Analysis
SPA	Sneak Path Analysis
STM	Scientific and Technical Memoranda
SW	Software
WCA	Worst Case Analysis