SLOVENSKI STANDARD
oSIST prEN 9721:2020

01-september-2020

Aeronavtika - Splošno priporočilo za arhitekturo BIT v integriranem sistemu

Aerospace series - General recommendation for the BIT Architecture in an integrated system

Luft- und Raumfahrt - Allgemeine Empfehlungen für die integrierte Prüfungs-(BIT)-Architektur in einem integrierten System

Série aerospatiale - Recommandations générales pour l'architecture des BIT dans un système intégré

Ta slovenski standard je istoveten z: prEN 9721

ICS:

49.020	Letala in vesoljska vozila na splošno	Aircraft and space vehicles in general

oSIST prEN 9721:2020 **en,fr,de**

EUROPEAN STANDARD

NORME EUROPÉENNE

EUROPÄISCHE NORM

DRAFT
prEN 9721

July 2020

ICS 49.020

English Version

Aerospace series - General recommendation for the BIT Architecture in an integrated system

Série aerospatiale - Recommandations générales pour l'architecture des BIT dans un système intégré

Luft- und Raumfahrt - Allgemeine Empfehlungen für die integrierte Prüfungs-(BIT)-Architektur in einem integrierten System

This draft European Standard is submitted to CEN members for enquiry. It has been drawn up by the Technical Committee ASD-STAN.

If this draft becomes a European Standard, CEN members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

This draft European Standard was established by CEN in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CEN member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CEN members are the national standards bodies of Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and United Kingdom.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Warning : This document is not a European Standard. It is distributed for review and comments. It is subject to change without notice and shall not be referred to as a European Standard.

EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

CEN-CENELEC Management Centre: Rue de la Science 23, B-1040 Brussels

Contents

European foreword

This document (prEN 9721:2020) has been prepared by the Aerospace and Defence Industries Association of Europe - Standardization (ASD-STAN).

After enquiries and votes carried out in accordance with the rules of this Association, this Standard has received the approval of the National Associations and the Official Services of the member countries of ASD-STAN, prior to its presentation to CEN.

This document is currently submitted to the CEN Enquiry.

Introduction

A Built-in-test (*BIT*) is a test carried out exclusively with the hardware and software resources specific to an item of equipment/system, in order to test it and/or its sub-assemblies, in view of detecting failures and isolating or even diagnosing them.

System designers are faced with the following questions:

- How do you define a strategy or method for a test built into a system?
- How do you assess the operational efficiency of a system's *BIT* architecture? (False alarms, non-reproducible alarms and false removals)
- How do you obtain a coherent *BIT* architecture between the various levels of a system? of a system of systems?
- How do you take into account the needs of the various users of the *BIT* function bearing in mind that the implementation, accesses, *BIT* reports, etc. are specific to the users?

1 Scope

The purpose of this document is to harmonise the dialogue between manufacturers, prime contractors, owners and the customer in view of making it easier to draw up specifications, share BIT architecture models and the *BIT* technical configuration of systems during the operational use phase.

This recommendation proposes adopting *BIT* operational efficiency and performance definitions, architecture design principles, and *BIT* specification or validation principles. It provides no recommendations regarding the numeric values for operational efficiency or performance. The diversity of situations, development of technological solutions and ever-changing operational requirements make it impossible to list general recommendations.

Clause 6 and Clause 9 set out the general context of use of the *BIT*.

Clause 7 lists the constraints to be taken into account to design a *BIT* architecture.

Clause 8 lists the various *BIT* types currently known and the definitions of performance and operational efficiency (metrics).

Clause 10 provides recommendations on the *BIT* architecture.

Clause 11 recommends a language for exchanging *BIT* architecture models for assembling the complete model of a system.

Clause 12 is an introduction to the prognosis.

This document is mainly intended for system designers.

Although it is based on examples of aeronautic systems, it is applicable to any type of system.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 5577, *Non-destructive testing — Ultrasonic testing — Vocabulary*

3 Terms, definitions and abbreviations

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 5577 and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at http://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

3.1.1
ambiguity group
associated to a signature and consists of a set of replaceable elements of the system of which at least one of the failures contributes to this signature but cannot be clearly indentified. Depending on the maintenance level considered, the replaceable elements may be *LRU*, *SRU*, components, etc.; the notion of ambiguity group refers to the requirement for isolating the failed element on the system tested with a given maintenance level

3.1.2
cut set
combination of failures, taken from the total list of possible failures (internal or external to the system), which lead to the loss of a service; it is said to be minimal if by removing any failure from the list, the service is no longer failing; the size (or degree) of the cut set is the number of elements on the list

EXAMPLE

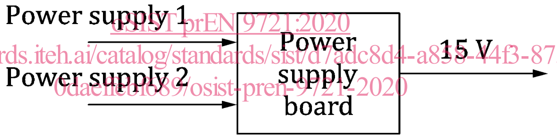

Figure 1 — Cut set example

In this example, it is presumed that Power supply 1 and 2 are operating as dual redundant parts. Therefore, the service: "Provide 15 V" is lost in the case both power supplies fail.

There are 2 separate minimal cuts sets that have the same service failure (15 V loss):

— cut set 1: Loss of "Power supply 1" AND Loss of "Power supply 2" (upstream output fails);

— cut set 2: "Power supply board" failure.

However, there are many non minimal cut sets. For example, the following cut set 3 is not minimal:

— cut set 3: (Loss of "Power supply 1" AND "Power supply board" failure) OR (No loss of "Power supply 1" AND "Power supply board" failure).

Cut set 2 is preferable over Cut set 3.

3.1.3
defect
non-compliance to a requirement, within the context of a specified or expected use

3.1.4
degradation or failure cause
circumstances related to the design, manufacture and use that resulted in the failure or incident

Note 1 to entry: In this document, it is assumed that there is no system design fault.

[SOURCE: adapted from NF X 60-500]

3.1.5
degradation
gradual and partial change in a system's ability to complete certain but not all required functions

3.1.6
degraded state
following a degradation (see the definition of "degradation" above), a system

3.1.7
detectability
system's failure detection capability is assessed for each of the failures that may occur: a failure is detectable or not

Note 1 to entry: Detectability is a metric that assesses the operational efficiency of an architecture. It takes into account the operational efficiency of the tests (or presumes total operational efficiency of the tests).

3.1.8
diagnostic
identification of the probable cause of the failure (or failures) using a logical reasoning based on a set of information coming from an inspection, a control or a test

3.1.9
disturbing test
test that is likely to modify the operational behaviour of the element tested

3.1.10
effect
result of a cause. This effect may be cascaded (domino effect) in the system; it is then a cause in relation to the effects propagated at the upper level

3.1.11
failure
stopping of a system's ability to complete the required function; it is observable through its effects (lack of behaviour) such as the deviation of a physical variable outside of a given tolerance range; it is noted f in this document

[SOURCE: adapted from NF X 60-500]

3.1.12
failure isolation

(troubleshooting) involves reducing the size of the ambiguity group through observations or additional tests; the failure isolation process (troubleshooting) is iterative; it ends when the diagnostic stops being ambiguous and when the troubleshooting is validated

3.1.13
failure sets

in this document, various failure sets (in the mathematical meaning of the term) will be used; they include

— E: set of all failures: $E = HF \cup SDF = \{f_i\}$ for i from 1 to Card(E),

— HF: set of failures caused by the hardware. This set excludes hardware design faults,

— DF: set of failures detectable by the test considered. DF is included in E,

— FE: set of failures that have an effect deemed "to be considered" (for example, with regard to criticality, a given usage scenario, etc.). FE is included in E. The scope of FE depends on the purpose of the analysis and therefore on the type of effects: operational, safety, commercial, etc.,

— DFE: set of detectable failures that have an effect deemed "to be considered". $DFE = DF \cap FE$ and

— SDF: set of software design failures (whether executed by a micro-processor or by a programmable component). Theoretically, this set should be empty. Consequently, these software failures are not considered in the $FMECA$ analyses or in the detection rate calculations. However, experience shows that they exist and that some of them can be detected by integrity tests.

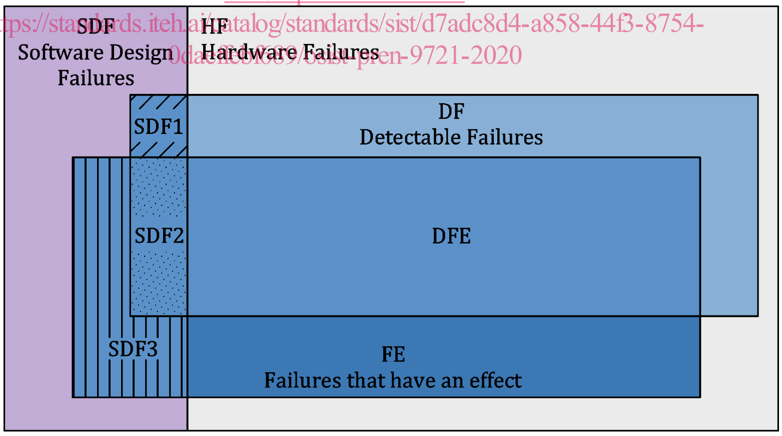

Figure 2 — Failure sets

— $SDF1$: Detectable software design faults and that do not have an effect.

— $SDF2$: Detectable software design faults and that have an effect.

— $SDF3$: Software design faults that have an effect but that are not detected by integrity tests.

Note 1 to entry: Software design faults will not be considered in the remainder of this document. This document will focus on the HF set. Consequently, and used somewhat imprecisely, all of the sets that will be mentioned in the remainder of this document will be understood to be restricted to the intersection with HF.

Note 2 to entry: Mathematical reminder: the cardinal of the set E noted “Card(E)” is the number of elements constituting this set.

3.1.14
failure signature

exhaustive combination (not minimal) of observable symptoms (OK or NOK results) resulting from the failure of a service; the signature consists of a core and periphery; the signature core is the combination of symptoms, always observable, that are sufficient to diagnose the failure; the periphery is the set of symptoms that may accompany the core (cascade failure phenomenon)

Note 1 to entry: **Recommendation 5:** With respect to failure signatures, it is important to state whether it is the signature core (design approach) that is adressed or the signature “core + periphery” set (maintenance approach).

Note 2 to entry: Several failures may have the same signature**.**

Note 3 to entry: The degree of the signature is n when this signature is associated to an ambiguity group of size n.

3.1.15
fault (Failed state)

internal cause that lead to a failure. In the case of fault, the item is in failed state

Note 1 to entry: The system can continue providing the service for example if its architecture has redundancies.

[SOURCE: adapted from NF X 60-500]

3.1.16
false alarm

result of a decision made between two possible choices (positive and negative), declared as positive, when it is in reality negative

3.1.17
list of system failures

The list of failures is the set of failures identified during the design stage and enhanced by return of experience. It may be formalised in a *FMECA* type form [2]; for each failure, it will also give as a minimum:

— its occurrence rate (except for the failures which causes are outside the scope of the system considered);

— its effect(s);

— the *LRU*/*SRU* or the resource/condition outside the scope of the system considered

Note 1 to entry: **Recommendation 4:** The design of the testability and the tests shall be based on the list of system failures.

Note 2 to entry: The level of detail of the list of failures shall be precise enough to guarantee the relevance of the values from metrics.

Note 3 to entry: For this, the failure modes of the functions provided by replaceable elements at the chosen maintenance level should be defined.

3.1.18
operational efficiency

<solution> measures either

— the level of result obtained with regard to the effect sought by unit of time or effort to be made by the operator or

— the time necessary or the degree of effort to be made by the operator to obtain the level of result expected with regard to the effect sought

Note 1 to entry: Operational efficiency is an operational metric.

Note 2 to entry: Operational efficiency is the result of the performance and context of use:

Operational efficiency = *function*(*Performance, Context*)

Note 3 to entry: This notion is illustrated by the example given in A.1.

3.1.19
performance

intrinsic quality of the solution irrespective of the usage contexts; it is a design metric

Note 1 to entry: This notion is illustrated by the example given in A.1.

3.1.20
symptom

physical manifestation of a failure; symptoms can be observed through inspection, through tests or come from the system's usage information

EXAMPLE

For example, the failure symptom (landing gear not down) has a *NOK* result to the test "Is the LG down?" when the "landing gear up" information is coded following a "lower landing gear" command. Respectively, the result will be OK if the "landing gear down" information is coded after sending the command.

Distinction between symptom and test result:

A test result is the coded expression of the result of observation of a symptom.

There are 2 types of test results:

— primary test results: those that result from the direct observation of a symptom;

— summary test results: those that result from an equation that combines other test results (primary or summary).

Distinction between symptom and coded information

In the landing gear example, the "landing gear not down" symptom originates from a combination of coded usage information: "lower landing gear" command and "landing gear up" information that do not come from the *BIT*.

3.1.21
system failure rate

$\boldsymbol{\lambda}_{\mathbf{f}}$

frequency of occurrence of the failure *f*, expressed in number of occurrences per hour

Note 1 to entry: In the remainder of this document, it is considered that failure rates are constant over time. (This hypothesis is commonly accepted for electronic systems).

Note 2 to entry: The failure rate λ of a system is assessed based on the failure rates of the set of failures identified for this system. The system failure rate is equal to the sum of the failure rates for the failures identified for this system (with the constant failure rate hypothesis).

Note 3 to entry: The system failure rate only applies to the intrinsic failures of the system considered.

3.1.22
technical efficiency

measurement of operational efficiency related to the technical resources necessary for the solution; it is a design metric

3.1.23
test result

image of the presence or absence of a symptom; the result may take the OK or NOK value

3.1.24
troubleshooting

failure isolation process

3.2 Abbreviations

A table of indexes is provided at the end of the document (Annex C). It is used to find the definitions of the main terms used in this document.

The abbreviations are explained in Table 1.

Table 1 — Explanation of abbreviations

Abbreviation	Explanation
BIT	built-in test
C	constraint
CAS	crew alert system
CBIT	continuous *BIT*
CO	correct operation
COR	correct operation rate
COTS	commercial off-the-shelf
DBIT	demanded *BIT*
DF	detectable failures
DFE	detectable failures with effect
DV	diagnostic value
E	set of failures
EBIT	external *BIT*
F	failure
FAR	false alarm rate
FCOR	false correct operation rate
FDC	failure detection capability
FDP	failure detection probability
FE	failures with effect
FIP	failure isolation probability
FM	failure modemode
FMECA	failure modes, effects and criticality analysis
FNOK	false NOK
FOK	false OK
FRP	failure resolution probability
HF	gardware failures
IBIT	initiated *BIT*
LRU	line replaceable unit
MBIT	maintenance *BIT*