

IEEE Standard for Verification of Vital Functions in Processor-Based Systems Used in Rail Transit Control

Rail Transit Vehicle Interface Committee
of the
IEEE Vehicular Technology Society

Approved 30 March 2000
Reaffirmed 5 December 2007

IEEE-SA Standards Board

Abstract: A set of standard verification tasks for processor-based equipment used in safety-critical applications on rail and transit systems is covered. This standard also covers processes that verify the level of safety achieved in the implementation of safety-critical functions that are required to be fail-safe. Quality assurance or validation processes that affect the overall level of system safety are not covered.

Keywords: rail, safety, safety critical, software, transit, verification, vital

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2000 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 5 April 2000. Printed in the United States of America.

Print: ISBN 0-7381-1966-0 SH94825
PDF: ISBN 0-7381-1967-9 SS94825

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

IEEE Standard for Verification of Vital Functions in Processor-Based Systems Used in Rail Transit Control

1. Overview

This standard defines a method for the identification and subsequent verification of vital functions implemented in processor-based equipment used in safety-critical applications on rail and transit systems. This standard requires the production of analyses and other supporting documentation necessary to demonstrate the achievement of the established safety goals.

This standard is limited to verifying fail-safe implementation of functions in processor-based equipment. This includes the identification of the functions that must be implemented fail-safely, the allocation of those functions to specific hardware and/or software components of the system, the identification of how those hardware and/or software components are expected to achieve the desired fail-safe operation (safety assurance concept), and the verification that the hardware and/or software component implements the functions in a fail-safe manner (see 3.1.2 for the definition of fail-safe used in this standard). This standard is intended to complement the execution of a total system safety program, and does not address all system safety issues. Detailed system, subsystem, and interface hazard analyses, hazard tracking, and risk assessment are considered part of the overall system safety program and, while elements of such hazard analyses may be required and/or referenced, generation of the analyses is not within the scope of this standard.

Throughout this standard the term system will be used. A system may be an entire transit or railroad control system, or it may be a subsystem or device, such as a track circuit or vehicle overspeed control unit. The application of this standard requires that the system or subsystem being verified is clearly defined, including its place within the overall system, and that all assumptions and interface requirements are clearly documented.

This standard is divided into five clauses. Clause 1 provides the scope and purpose of this standard. Clause 2 lists references to other standards that are useful in applying this standard. Clause 3 provides definitions that are either not found in other standards, or have been modified for use with this standard. Clause 4 describes the verification approach, dividing it into activities performed at three levels: functional, concept, and implementation. Clause 5 defines analyses and supporting documentation required to identify the imposed safety design requirements derived from each of the three levels, to allocate those safety requirements to the various components of the system, and, finally, to confirm that the safety goals have been attained.

The standard also contains three annexes. Annex A provides description and examples of techniques, analyses, and procedures that may be used in completing the concept-level tasks established in Clause 5. Annex B provides an example of the application of this standard. Annex C is a bibliography.

It is not intended that systems installed prior to the effective date of this standard comply with this standard.

1.1 Scope

This standard provides a set of standard verification tasks for processor-based equipment used in safety-critical applications on rail and transit systems. The scope of this standard shall encompass, and be limited to, processes that verify the level of safety achieved in the implementation of safety-critical functions that are required to be fail-safe. This standard does not address quality assurance or validation processes, which also affect the level of overall system safety achieved.

Figure 1 illustrates the elements of the safety verification process (within the dotted box) in the context of the overall system safety and design and development processes.

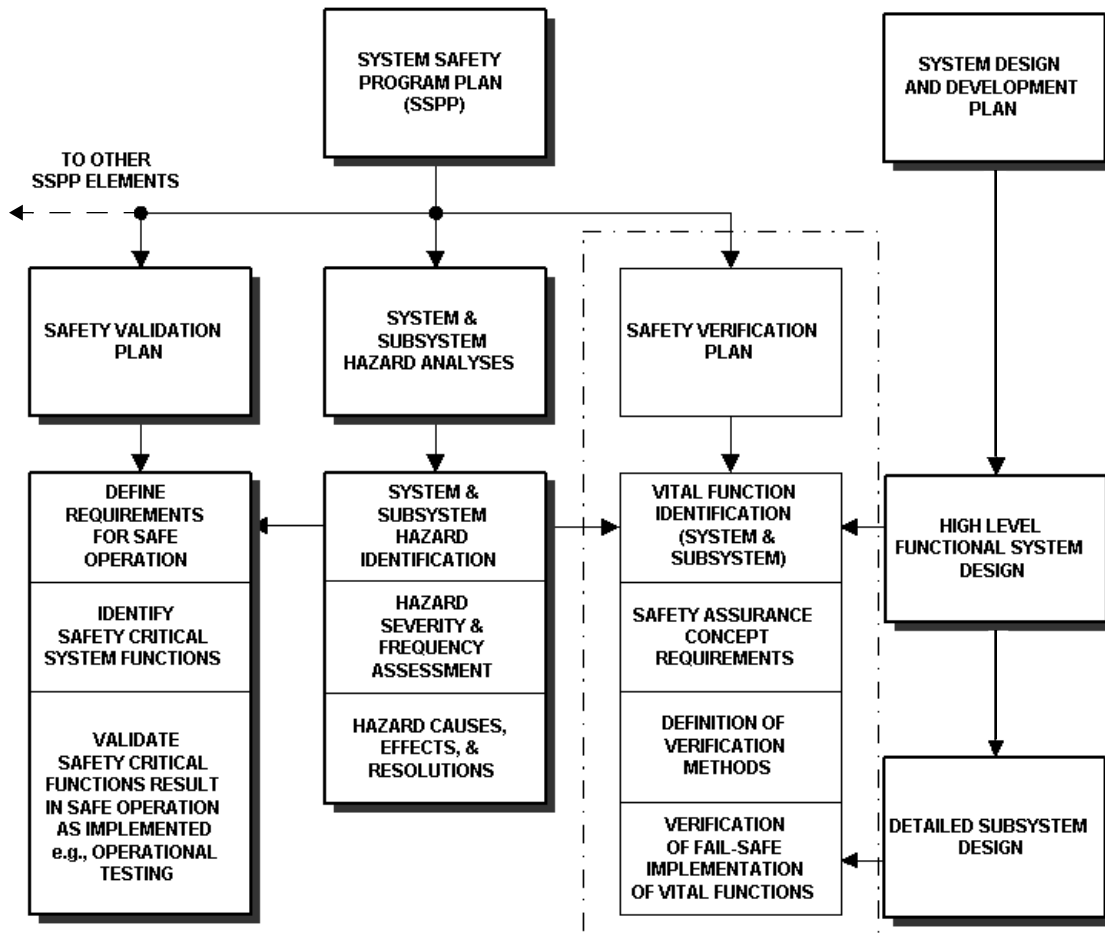


Figure 1 – Safety verification standard context diagram

1.2 Purpose

The purpose of this safety verification process standard is to provide a well-defined and well-structured set of analysis methods and documentation that

- Fulfills the primary purpose of the verification process.
- Is flexible enough to accommodate all viable design methods.
- Satisfies the safety requirements of the end user.

This standard

- Defines a necessary and sufficient set of analyses at the concept, functional, and implementation levels that comprehensively identify and verify all functions required to be implemented as fail-safe.
- Defines responsibilities for equipment suppliers and end users.

2. References

This standard shall be used in conjunction with the following publication. If the following publication is superseded by an approved revision, the revision shall apply. In case of a conflict between this standard and the referenced document, this standard shall take precedence. Those provisions of the referenced document that are not in conflict with this standard shall apply as referenced.

MIL-Std-882C:1996, System Safety Program Plan Requirements.¹

3. Abbreviations, acronyms, and definitions

3.1 Definitions

For the purposes of this standard, the following terms and definitions apply. IEEE 100-1996, The IEEE Standard Dictionary of Electrical and Electronic Terms [B13],² should be referenced for terms not defined in this clause.

3.1.1 concept level: The level of verification activities at which vital functions and vital implementation requirements, imposed on the system's design and implementation by the safety assurance concept selected, are determined and identified.

3.1.2 fail-safe: A design philosophy applied to safety-critical systems such that the result of hardware failure or the effect of software error shall either prohibit the system from assuming or maintaining an unsafe state, or shall cause the system to assume a state known to be safe.

3.1.3 fail-safely: The implementation of a function in a fail-safe manner.

3.1.4 fault tree analysis (FTA): A structured analysis method used to comprehensively identify faults and combinations of faults of software and hardware components as they relate to a hazard.

3.1.5 functional fault tree (FFT): A structured analysis method used to identify vital functions at the system functional level by comprehensively examining system functional faults that could precipitate hazards.

3.1.6 functional level: The level of verification activities at which vital system functions are identified from system functional and operational requirements.

3.1.7 hardware failure: A change in the characteristics of a system hardware element beyond its design tolerances.

3.1.8 hazard: An existing or potential condition that can result in a mishap.

¹MIL publications are available from Customer Service, Defense Printing Service, 700 Robbins Ave., Bldg. 4D, Philadelphia, PA 19111-5094.

²The numbers in brackets correspond to those of the bibliography in Annex C.

3.1.9 implementation level: The level of verification activities at which system components implementing vital functions are comprehensively identified and analyzed to verify that all functions identified as vital are implemented fail-safely.

3.1.10 mean time between hazardous events (MTBHE): The average time between occurrences of events, where hazardous events and the equipment that may precipitate them are defined at the system level. The hazardous events included in MTBHE are those whose consequences are of a given severity, as determined by the organization generating the safety goals.

3.1.11 mishap: An unplanned event or series of events resulting in death, injury, occupational illness, or damage to or loss of equipment or property, or damage to the environment; an accident.

3.1.12 safe: Having acceptable risk of the occurrence of a hazard.

3.1.13 safety assurance: A characteristic of the implementation of a system that assures a level of safe operation.

3.1.14 safety assurance concept: A design concept applied to processor-based systems that assures the fail-safe implementation of identified functions, including safe operation in the presence of hardware failures and/or software errors. Examples are: Checked Redundancy; Diversity and Self-Checking; Numerical Assurance; and N-Version Programming.

3.1.15 safety-critical: A term applied to a system or function, the correct performance of which is critical to safety of personnel and/or equipment; also a term applied to a system or function, the incorrect performance of which may result in an unacceptable risk of a hazard.

3.1.16 safety validation: A structured and managed set of activities that demonstrate that the system, as specified and implemented, performs the intended functions, and that those functions result in overall safe operation. Validation answers the question, “Did we build the right system?”

3.1.17 safety verification: A structured and managed set of activities that identify the vital functions required to be performed by the system, and demonstrate that the system, including its subsystems, interfaces and components, implements the vital functions fail-safely to a level that meets the allocated system safety goals. Verification answers the question, “Did we build the system right?”

3.1.18 self-revealing component failures: Component failures whose effects on system operation are immediately and clearly apparent to a properly trained person.

3.1.19 software error: An error in a software element which, when executed, results in unintended system operation.

3.1.20 system safety: The application of engineering and management principles, criteria, and techniques to optimize all aspects of safety within the constraints of operational effectiveness, time, and cost throughout all phases of the system life cycle.

3.1.21 system safety goals—quantitative: A quantitative limit of the probability and/or frequency with which any vital function fails to be implemented safely.

3.1.22 system safety program: The combined tasks and activities of system safety management and system safety engineering that enhance operational effectiveness by satisfying the system safety requirements in a timely, cost-effective manner throughout the system life cycle.

3.1.23 system safety program plan: A formal document that fully describes the planned safety tasks required to meet the system requirements, including organizational responsibilities, methods of

accomplishment, milestones, depth of effort, and integration with other program engineering and management functions.

3.1.24 unsafe: Having unacceptable risk of the occurrence of a hazard.

3.1.25 vital function: A function in a safety-critical system that is required to be implemented in a fail-safe manner.

NOTE—Vital functions are a subset of safety-critical functions.

3.2 Abbreviations and acronyms

ASCE	American Society of Civil Engineers
ATP	automatic train protection
CENELEC	European Committee for Electrotechnical Standardization
FFT	functional fault tree
FMEA	failure modes and effects analysis
FTA	fault tree analysis
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
MTBHE	mean time between hazardous events
PHA	preliminary hazard analysis
SSHA	subsystem hazard analysis
SSIHA	subsystem interface hazard analysis
SSn-FTA	subsystem fault tree analysis
SSPP	system safety program plan
SVP	safety verification plan

4. Safety verification approach

Comprehensive safety verification shall include the identification of vital functions and the verification that the identified vital functions have been implemented fail-safely to the degree required by the system safety goals.

The activities associated with safety verification shall be divided into the following three levels:

- Concept level
- Functional level
- Implementation level

Figure 2 illustrates the three levels and their relationships. These levels correspond to different areas of safety verification. Concept-level activities shall analyze the safety assurance concept employed, identify the concept-specific design requirements necessary for the fail-safe implementation of vital functions, and identify the concept-specific verification methods. Functional-level activities shall identify all user-specified system functions required to be implemented fail-safely. Implementation-level activities shall apply the verification methods determined at the concept level to the implemented system to ensure that the identified set of vital functions has been implemented fail-safely.

Software and hardware components implementing vital functions are thus identified from two sources; the user specified system functional requirements identified at the functional level, and concept-specific design

requirements imposed on the hardware and software design by the safety assurance concept identified at the concept level.

The results of the verification activities shall demonstrate achievement of the system safety goals.

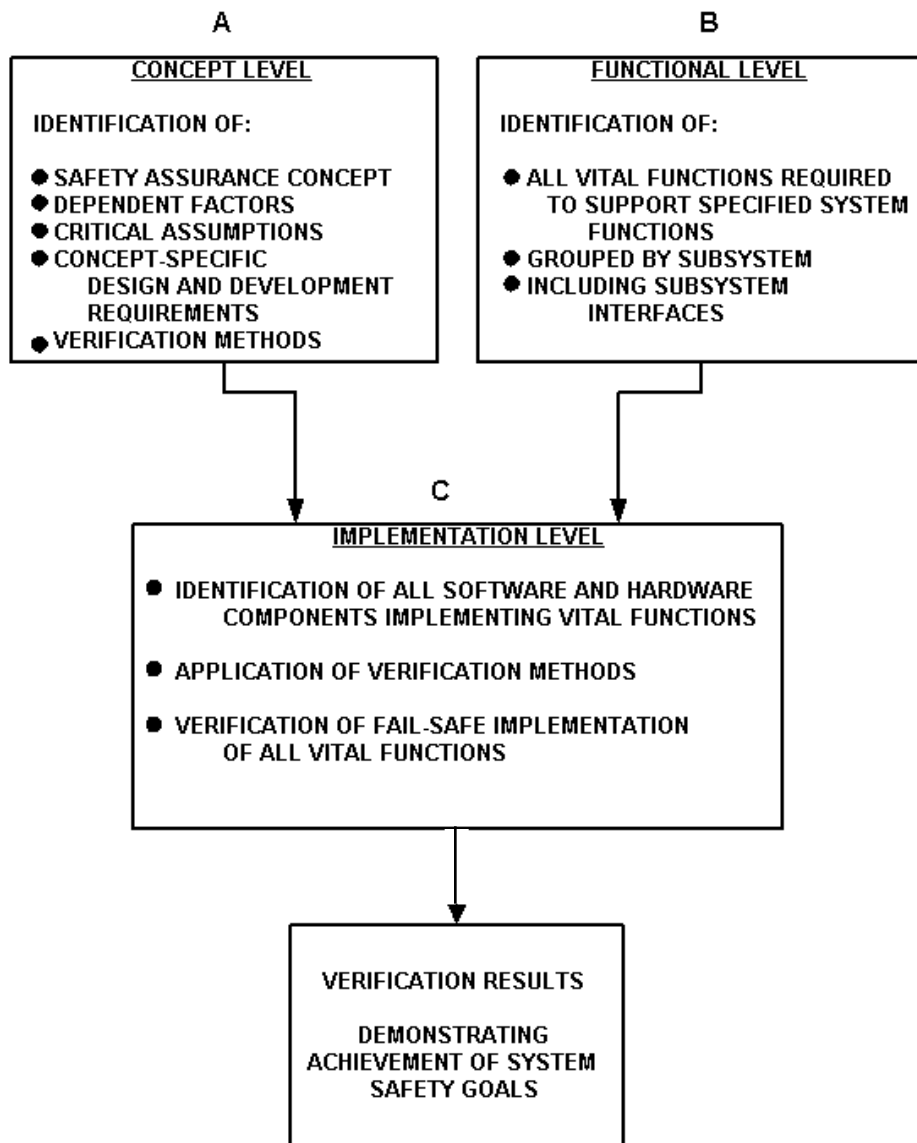


Figure 2—Approach to safety verification—Division of safety verification activities into levels

4.1 Concept-level verification [Figure 2(A)]

Activities at the concept level shall include

- Describing the “safety assurance concept(s)” employed to ensure fail-safe implementation of vital functions.
- Comprehensive identification of factors specific to the concept upon which the assurance of fail-safe implementation is dependent.
- Defining the concept-specific design requirements necessary for fail-safe implementation.
- Identifying verification methods to be applied at the implementation level that verify that the concept-specific design requirements have been met and that vital functions have been implemented fail-safely.

Safety assurance concepts used to design and implement vital functions in processor-based systems include, but are not limited to

- Checked Redundancy
- Numerical Assurance
- N-Version Programming
- Diversity
- Self-Checking

In addition, fail-safe discrete circuit design used to implement vital input/output functions in processor-based systems is also identified as a “safety assurance concept” (refer to Annex A).

NOTE—An example of a concept-specific design requirement imposed by the safety assurance concept Checked Redundancy would be that vital parameters and results of vital functions performed by the redundant systems must be periodically compared by hardware and/or software elements to detect discrepancies in the results and take action to assure safe operation. This concept-specific fail-safe requirement is not imposed by the user specified system functional requirements; however, it must be identified as a requirement and incorporated into the implementation for the assurance of safe operation.

Each safety assurance concept has an associated set of factors upon which the fail-safe implementation of vital functions is dependent, that is, “dependent factors,” and a set of critical assumptions. The dependent factors shall be identified and used to subsequently identify concept-specific design and development requirements. The critical assumptions shall be identified and included in the concept level documentation.

NOTE—Again using Checked Redundancy as an example, dependent factors include: the ability of the comparison mechanism to detect all discrepancies that affect safe operation; frequency of comparison; the extent to which failures in one redundant element are detectable at the points of comparison; executable software correctness; and others.

A Checked Redundancy concept-specific design requirement related to the dependent factor, frequency of comparison, would be that the interval between comparison of the outputs of each redundant unit be implemented fail-safely such that it is no greater than a specified time.

One example of a concept-specific development requirement associated with the dependent factor “software correctness” would be that the software is developed in accordance with a process that ensures a specified level of correctness.

Verification methods shall be identified at the concept level. Verification methods shall comprise analysis, demonstration, test, and/or inspection applied to products and processes, as required, to verify that the processor-based system fail-safely implements all vital functions, both those identified at the concept level and those identified at the functional level, to the degree required by the system safety goals.

4.2 Functional-level verification [Figure 2(B)]

Verification activities performed at the functional level shall include analyses that comprehensively identify the safety-critical functions associated with the performance of user-specified system functions implemented by the processor-based equipment. Safety-critical functions shall be identified using a functional fault tree (FFT). A subset of the fault events identified in the FFT shall be associated with functions that shall be implemented fail-safely to avoid the occurrence of those fault events or protect the system against the effects of those fault events when they occur. This set of associated functions shall comprise the vital functions.

The FFT shall be developed in three levels. The top level of the functional fault tree (FFT1) shall identify system fault events based on PHA identified hazards, prior to the allocation of system functions to individual subsystems. The second level of the functional fault tree (FFT2) shall continue the identification of faults grouped according to subsystem function allocation. The third level of the functional fault tree (FFT3) shall include the identification of faults associated with the interfaces between subsystems.

The terminal fault events of the FFT shall include, per subsystem, all faults for which functions shall be implemented in a fail-safe manner (vital functions) to prevent and/or protect against the occurrence of those faults.

4.3 Implementation-level verification [Figure 2(C)]

Verification activities performed at the implementation level shall include the identification of the hardware and software components involved in the implementation of the identified vital functions within each subsystem. In addition, these activities shall include the application of previously defined concept-specific verification methods (4.1) to verify that the identified vital functions have been implemented fail-safely to the degree required by the system safety goals.

Verification methods shall also be applied to those concept-specific development activities previously identified.

5. Safety verification tasks

This clause describes in detail the tasks that shall be performed at each verification level (concept, functional, implementation) and the associated documentation required for safety verification of the processor-based portion of a system. Figure 3 illustrates the overall safety verification approach used by this standard and its relationship to the overall system safety program plan (SSPP).

Graphical flow diagrams describing the safety verification tasks are shown in Figure 4 through Figure 6. Plain blocks indicate documents, analyses, or activities performed for this standard. Blocks with shadow indicate documents, analyses, or activities referenced external to this standard. Dotted lines indicate an external relationship to items in the standard. Solid lines indicate a firm relationship between items in the standard. The identification numbers associated with each block on Figure 4 through Figure 6 reference specific process activities that are described in this clause. The order in which process activities are numbered and described does not mandate the order in which they are performed. Process activities are performed in an order consistent with the overall system development and when required inputs are available. Figure 4 illustrates the concept-level verification process linked to Figure 6, which shows a standard implementation-level verification process. Figure 5 illustrates the functional-level verification process that is also linked to Figure 6.

These verification process activities shall culminate in a safety verification final report confirming that identified vital functions have been implemented fail-safely to the degree required by the system safety goals. This report shall provide references to all supporting documentation.