BS EN 62566:2014

BSI Standards Publication

Nuclear power plants — Instrumentation and control important to safety — Development of HDL-programmed integrated circuits for systems performing category A functions

bsi.

...making excellence a habit.™

National foreword

This British Standard is the UK implementation of EN 62566:2014. It is identical to IEC 62566:2012. It supersedes BS IEC 62566:2012, which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee NCE/8, Reactor instrumentation.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

Published by BSI Standards Limited 2014

ISBN 978 0 580 84483 6

ICS 27.120.20

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 January 2013.

Amendments/corrigenda issued since publication

Date	Text affected
31 October 2014	This corrigendum renumbers BS IEC 62566:2012 as BS EN 62566:2014. Annex ZA inserted

EUROPEAN STANDARD

NORME EUROPÉENNE

EUROPÄISCHE NORM

EN 62566

August 2014

ICS 27.120.20

English Version

Nuclear power plants - Instrumentation and control important to safety - Development of HDL-programmed integrated circuits for systems performing category A functions (IEC 62566:2012)

Centrales nucléaires de puissance - Instrumentation et contrôle-commande importants pour la sûreté - Développement des circuits intégrés programmés en HDL pour les systèmes réalisant des fonctions de catégorie A (CEI 62566:2012)

Kernkraftwerke - Leittechnik für Systeme mit sicherheitstechnischer Bedeutung - Entwicklung HDL-programmierter integrierter Schaltkreise für Systeme, die Funktionen der Kategorie A ausführen (IEC 62566:2012)

European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

CEN-CENELEC Management Centre: Avenue Marnix 17, B-1000 Brussels

Ref. No. EN 62566:2014 E

Foreword

This document (EN 62566:2014) consists of the text of IEC 62566:2012 prepared by SC 45A "Instrumentation, control and electrical systems of nuclear facilities" of IEC/TC 45 "Nuclear instrumentation".

The following dates are fixed:

– latest date by which this document has to be implemented at national level by publication of an identical national standard or by endorsement (dop) 2015-08-04

– latest date by which the national standards conflicting with this document have to be withdrawn (dow) 2017-08-04

As stated in the nuclear safety directive 2009/71/EURATOM, Chapter 1, Article 2, item 2, Member States are not prevented from taking more stringent safety measures in the subject-matter covered by the Directive, in compliance with Community law. In a similar manner, this European standard does not prevent Member States from taking more stringent nuclear safety measures in the subject-matter covered by this standard.

Endorsement notice

The text of the International Standard IEC 62566:2012 was approved by CENELEC as a European Standard without any modification.

Annex ZA
(normative)

Normative references to international publications with their corresponding European publications

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

NOTE 1 When an international publication has been modified by common modifications, indicated by (mod), the relevant EN/HD applies.

NOTE 2 Up-to-date information on the latest versions of the European Standards listed in this annex is available here: www.cenelec.eu.

Publication	Year	Title	EN/HD	Year
IEC 60671	-	Nuclear power plants - Instrumentation and control systems important to safety - Surveillance testing	EN 60671	-
IEC 60880	2006	Nuclear power plants - Instrumentation and control systems important to safety - Software aspects for computer-based systems performing category A functions	EN 60880	2009
IEC 60987	2007	Nuclear power plants - Instrumentation and control important to safety - Hardware design requirements for computer-based systems	EN 60987	2009
IEC 61513	2011	Nuclear power plants - Instrumentation and control important to safety - General requirement for systems	EN 61513	2013
IEC 62138	-	Nuclear power plants - Instrumentation and control important for safety - Software aspects for computer-based systems performing category B or C functions	EN 62138	-
IEC 62340	-	Nuclear power plants - Instrumentation and control systems important to safety - Requirements for coping with common cause failure (CCF)	EN 62340	-
IAEA guide NS-G-1.3	2002	Instrumentation and control systems important to safety in nuclear power plants	-	-

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

NUCLEAR POWER PLANTS – INSTRUMENTATION AND CONTROL IMPORTANT TO SAFETY – DEVELOPMENT OF HDL-PROGRAMMED INTEGRATED CIRCUITS FOR SYSTEMS PERFORMING CATEGORY A FUNCTIONS

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 62566 has been prepared by subcommittee 45A: Instrumentation and control of nuclear facilities, of IEC technical committee 45: Nuclear instrumentation.

The text of this Standard is based on the following documents:

FDIS	Report on voting
45A/859/FDIS	45A/865/RVD

Full information on the voting for the approval of this Standard can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC web site under "http://webstore.iec.ch" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

INTRODUCTION

a) Technical background, main issues and organisation of the Standard

The electronic systems of class 1 (according to IEC 61513) used in Nuclear Power Plants (NPP) which are required in emergency situations, need to be fully validated and qualified before being used in operation.

In traditional systems that are computer-based, a separation can be drawn between the hardware and software portions. The hardware is mainly designed with standardised components having pre-defined electronic functions such as microprocessors, timers or network controllers, whereas software is used to coordinate the different parts of the hardware and to implement the application functions.

Nowadays, I&C designers may build application functions directly in one integrated circuit using devices such as FPGAs or similar technologies. The function of such an integrated circuit is not defined by the supplier of the physical component or micro-electronic technology but by the I&C designer.

The specific integrated circuits addressed by this Standard are:

1) based on pre-developed micro-electronic resources,
2) developed within an I&C project,
3) developed with Hardware Description Languages (HDL) and related tools used to implement the requirements in a proper assembly of the pre-developed micro-electronic resources.

Therefore these circuits are named “HDL-Programmed Devices”, (HPD). The HDL statements which describe a HPD can include the instantiation of Pre-Developed Blocks (PDB) which are typically provided as libraries, macros, or Intellectual Property cores.

HPDs can be effective solutions to implement functions required by an I&C project. However, the verification and validation may be limited by issues such as high number of internal paths and limited observability, if the HPD has not been developed with verifiability in mind.

In order to achieve the reliability required for safety I&C systems, the development of HPDs shall comply with strict process and technical requirements such as those provided by this Standard, including the specification of requirements, the selection of blank integrated circuits and PDBs, the design and implementation, the verification, and the procedures for operation and maintenance.

It is intended that this Standard be used by hardware designers, operators of NPPs (utilities), and by regulators. Regulatory bodies will find guidance to assess important aspects such as design, implementation, verification and validation of HPDs.

b) Situation of the current Standard in the structure of the IEC SC 45A Standard series

IEC 61513 is a first level IEC SC 45A document and gives guidance applicable to I&C at system level. It is supplemented by guidance at hardware level (IEC 60987) and software level (IEC 60880 and IEC 62138). IEC 62340 gives requirements in order to reduce and overcome the possibility of common cause failure of category A functions.

IEC 62566 is a second level IEC SC 45A document which focuses on the activities when HPDs are developed. It complements IEC 60987 which deals with the generic issues of hardware design of computer based systems. It refers to IEC 60880 when issues identical to that of software development are addressed.

For more details on the structure of the IEC SC 45A Standard series, see item d) of this introduction.

c) Recommendations and limitations regarding the application of the Standard

It is important to note that this Standard establishes no additional functional requirements for safety systems.

Aspects for which special requirements and recommendations have been produced are:

1) an approach to specify the requirements of, to design, to implement and to verify “HDL-Programmed Devices” (HPD, 3.7), and to handle the corresponding aspects of system integration and validation;
2) an approach to analyse and select the blank integrated circuits, micro-electronic technologies and Pre-Developed Blocks (PDB, 3.11) used to develop HPDs;
3) procedures for the modification and configuration control of HPDs;
4) requirements for selection and use of software tools used to develop HPDs.

It is recognized that digital technology is continuing to develop at a rapid pace and that it is not possible for a Standard such as this one to include references to all modern design technologies and techniques.

To ensure that the Standard will continue to be relevant in future years the emphasis has been placed on issues of principle, rather than specific technologies. If new techniques are developed then it should be possible to assess the suitability of such techniques by applying the safety principles contained within this Standard.

d) Description of the structure of the IEC SC 45A Standard series and relationships with other IEC documents and other bodies documents (IAEA, ISO)

The top-level document of the IEC SC 45A Standard series is IEC 61513. It provides general requirements for I&C systems and equipment that are used to perform functions important to safety in NPPs. IEC 61513 structures the IEC SC 45A Standard series.

IEC 61513 refers directly to other IEC SC 45A Standards for general topics related to categorization of functions and classification of systems, qualification, separation of systems, defence against common cause failure, software aspects of computer-based systems, hardware aspects of computer-based systems, and control room design. The Standards referenced directly at this second level should be considered together with IEC 61513 as a consistent document set.

At a third level, IEC SC 45A Standards not directly referenced by IEC 61513 are Standards related to specific equipment, technical methods, or specific activities. Usually these documents, which make reference to second-level documents for general topics, can be used on their own.

A fourth level extending the IEC SC 45 Standard series, corresponds to the Technical Reports which are not normative.

IEC 61513 has adopted a presentation format similar to the basic safety publication IEC 61508 with an overall safety life-cycle framework and a system life-cycle framework and provides an interpretation of the general requirements of IEC 61508-1, IEC 61508-2 and IEC 61508-4, for the nuclear application sector. Compliance with IEC 61513 will facilitate consistency with the requirements of IEC 61508 as they have been interpreted for the nuclear industry. In this framework IEC 60880 and IEC 62138 correspond to IEC 61508-3 for the nuclear application sector.

IEC 61513 refers to ISO as well as to IAEA GS-R-3 and IAEA GS-G-3.1 for topics related to quality assurance.

The IEC SC 45A Standards series consistently implements and details the principles and basic safety aspects provided in the IAEA code on the safety of NPPs and in the IAEA safety series, in particular the Requirements NS-R-1, establishing safety requirements related to the design of Nuclear Power Plants, and the Safety Guide NS-G-1.3 dealing with instrumentation and control systems important to safety in Nuclear Power Plants. The terminology and definitions used by SC 45A Standards are consistent with those used by the IAEA.

NUCLEAR POWER PLANTS – INSTRUMENTATION AND CONTROL IMPORTANT TO SAFETY – DEVELOPMENT OF HDL-PROGRAMMED INTEGRATED CIRCUITS FOR SYSTEMS PERFORMING CATEGORY A FUNCTIONS

1 Scope and object

1.1 General

This International Standard provides requirements for achieving highly reliable “HDL-Programmed Devices” (HPD), for use in I&C systems of nuclear power plants performing functions of safety category A as defined by IEC 61226.

The programming of HPDs relies on Hardware Description Languages (HDL) and related software tools. They are typically based on blank FPGAs or similar micro-electronic technologies. General purpose integrated circuits such as microprocessors are not HPDs.

This Standard provides requirements on:

a) a dedicated development life-cycle addressing each phase of the development of HPDs, including specification of requirements, design, implementation, verification, integration and validation,

b) planning and complementary activities such as modification and production,

c) selection of pre-developed components. This includes micro-electronic resources (such as a blank FPGA or CPLD) and HDL statements representing Pre-Developed Blocks (PDB),

d) use of simplicity and deterministic principles, recognized to be of primary importance to achieve “fault free” implementation of category A functions,

e) tools used to design, implement and verify HPDs.

This Standard does not put requirements on the development of the micro-electronic resources, which are usually available as "commercial off-the-shelf" items and are not developed under nuclear quality assurance Standards. It addresses the developments made with these micro-electronic resources in an I&C project with HDLs and related tools.

This Standard provides guidance to avoid as far as possible latent faults remaining in HPDs, and to reduce the susceptibility to single failures as well as to potential Common Cause Failures (CCF). The requirements within this Standard for clear and comprehensive documentation should facilitate the effective application of IEC 62340.

Reliability aspects related to environmental qualification and failures due to ageing or physical degradation are not handled in this Standard. Other Standards, especially IEC 60987, IEC 60780 and IEC 62342, address these topics.

Subclause 5.7 of IEC 60880:2006 provides security requirements that apply to the development of HPDs as applicable.

1.2 Use of this Standard

This Standard provides guidance and requirements to produce verifiable designs and implementations where justification is necessary due for example to the function performed or to the importance to safety of its behaviour. Class 1 I&C systems may use HPDs for which full demonstration of compliance with the requirements of this Standard is not mandatory, e.g.

when they do not implement the logic of a safety function. However, deviations from this Standard should be justified.

This Standard describes the activities to develop HPDs, organized in the framework of a dedicated life-cycle. It also describes activities and guidelines to be used in addition to the requirements of IEC 61513 for system integration and validation when HPDs are included.

Those requirements of IEC 60987 that relate to programmable logic device development are applicable, in addition to those of this Standard, where HPDs are part of class 1 I&C systems.

NOTE In case of conflicting requirements, this Standard supersedes those in IEC 60987 about class 1 HPDs.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 60671, *Nuclear power plants – Instrumentation and control systems important to safety – Surveillance testing*

IEC 60880:2006, *Nuclear power plants – Instrumentation and control systems important to safety – Software aspects for computer-based systems performing category A functions*

IEC 60987:2007, *Nuclear power plants – Instrumentation and control important to safety – Hardware design requirements for computer-based systems*

IEC 61513:2011, *Nuclear power plants – Instrumentation and control important to safety – General requirements for systems*

IEC 62138, *Nuclear power plants – Instrumentation and control important for safety – Software aspects for computer-based systems performing category B or C functions*

IEC 62340, *Nuclear power plants – Instrumentation and control systems important to safety – Requirements for coping with common cause failure (CCF)*

IAEA guide NS-G-1.3:2002, *Instrumentation and Control Systems Important to Safety in Nuclear Power Plants*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1
Application Specific Integrated Circuit , ASIC
integrated circuit designed for specific applications

[IEC 60050-521:2002, 521-11-18]

NOTE Specialized integrated circuit designed for the purpose of one company. It embeds bespoke functions defined by this company.

3.2
block
one of the parts that make up a design; a block may be subdivided into other blocks

NOTE A block is either a Pre-Developed Block or a Native Block or a block developed during the considered project.

3.3
Common Cause Failure, CCF
failure of two or more structures, systems or components due to a single specific event or cause

[IAEA Safety Glossary 2007 Edition]

NOTE Common causes may be internal or external to an I&C system.

[IEC 61513]

3.4
Electronic System Level, ESL
high-level description of an electronic system, based on a set of processes representing functionalities of components such as microprocessors, memories, specialized computing units, or communication channels

NOTE This description allows the designer to partition the system into components, to assess its performance under different mapping of functions to the components, and to establish the requirements for the components.

It is typically performed with languages such as SystemC (IEEE 1666), SystemVerilog (IEEE 1800), or Matlab (R).

3.5
Field Programmable Gate Array, FPGA
integrated circuit that can be programmed in the field by the I&C producer. It includes programmable logic blocks (combinatorial and sequential), programmable interconnections between them and programmable blocks for input and/or outputs. The function is then defined by the I&C designer, not by the integrated circuit supplier.

NOTE While FPGAs are essentially digital devices, some of them may integrate analog input/outputs and analog to digital converters. FPGAs may include advanced digital functions such as hardware multipliers, dedicated memory and embedded processor cores.

3.6
Hardware Description Language, HDL
language used to formally describe the functions and/or the structure of an electronic component for documentation, simulation or synthesis

NOTE The most widely used HDLs are VHDL (IEEE 1076) and Verilog (IEEE 1364).

3.7
HDL-Programmed Device, HPD
integrated circuit configured (for NPP I&C systems), with Hardware Description Languages and related software tools

NOTE 1 HDLs and related tools (e.g. simulator, synthesizer) are used to implement the requirements in a proper assembly of pre-developed micro-electronic resources.

NOTE 2 The development of HPDs can use Pre-Developed Blocks.

NOTE 3 HPDs are typically based on blank FPGAs, PLDs or similar micro-electronic technologies.

3.8
module
one of the parts that make up a design; a module may be subdivided into other modules

NOTE Module" is a synonym of "Block"; "Block" is often used in the context of electronic design. "Module" is the term used by IEC 60880 and is needed in this Standard for references to IEC 60880.

3.9
native block
a Block which represents a pre-existing resource in the integrated circuit, e.g. an OR gate or a more complex block such as a multiplier or a serial transmission controller. By programming the HPD, the Native Blocks are configured and connected to provide the required function.

3.10
netlist
description of an electronic component in terms of interconnections between its terminal elements (e.g. Native Blocks)

3.11
Pre-Developed Block, PDB
pre-developed functional block usable in a HDL description

NOTE 1 PDBs are typically provided as libraries, macros, or Intellectual Property cores. They are used in the development of a HPD and incorporated in this HPD.

NOTE 2 A PDB may need significant work before incorporation in a HPD, e.g. synthesizing an electronic circuit from the HDL statements, mapping the notional components of this circuit on the hardware structures of the physical integrated circuit and routing the interconnections.

3.12
Pre-Developed Software, PDS
software part that already exists, is available as a commercial or proprietary product, and is being considered for use

[IEC 60880]

3.13
Programmable Logic Device, PLD
integrated circuit that consists of logic elements with an interconnection pattern, parts of which are user programmable.

[IEC 60050-521:2002, 521-11-01]

NOTE 1 Different kinds of PLD exist, e.g. Erasable PLD or Complex PLD (CPLD).

NOTE 2 The differences between “FPGA” and “PLD” are not well defined, but “PLD” usually refers to a simpler device than “FPGA”.

3.14
Register Transfer Level, RTL
synchronous parallel model of an electronic circuit, describing its behaviour by means of signals processed according to a combinatorial logic and transferred between registers on clock pulses. The RTL model is typically written in HDL or generated out of HDL source code.

4 Symbols and abbreviations

ASIC: Application Specific Integrated Circuit
CCF: Common Cause Failure
CPLD: Complex Programmable Logic Device
DRC: Design Rule Check
ESL: Electronic System Level
FPGA: Field Programmable Gate Array
HDL: Hardware Description Language
HPD: HDL-Programmed Device
IP: Intellectual Property

I&C: Instrumentation and Control
PAL: Programmable Array Logic
PDB: Pre-Developed Block
PDS: Pre-Developed Software
PLD: Programmable Logic Device
RAM: Random Access Memory
RTL: Register Transfer Level
SEU: Single Event Upset
SRAM: Static RAM
STA: Static Timing Analysis
VHDL: Very High Speed Integrated Circuit Hardware Description Language
V&V: Verification and Validation

5 General requirements for HPD projects

5.1 General

This clause first locates the HPD within the I&C system described by IEC 61513. Then it describes the HPD development life-cycle which structures the HPD project.

Finally it provides requirements for HPD projects, for quality assurance and for configuration management. As these issues are common with those of software development processes, the requirements are defined by reference to relevant sections of IEC 60880, supplemented by HPD specific requirements if needed.

With reference to Clause 1, the scope of this Standard excludes the development of micro-electronic technologies or blank integrated circuits. Therefore wordings such as “HPD development”, “HPD life-cycle”, “HPD design” or “HPD verification” refer to what is done within the I&C project, starting from these technologies or these blank integrated circuits, to produce the specific integrated circuit for use in the I&C system.

5.2 Life-cycle

The process of producing I&C systems for use in nuclear power plants is given in IEC 61513 that introduces the concept of system life-cycle. This is a vehicle by which the development process can be controlled and whose adoption should also result in the evidence necessary to justify the correct operation of safety systems. It includes and places requirements on, but does not dictate the project arrangements to be used for, production of systems (see Figure 1).

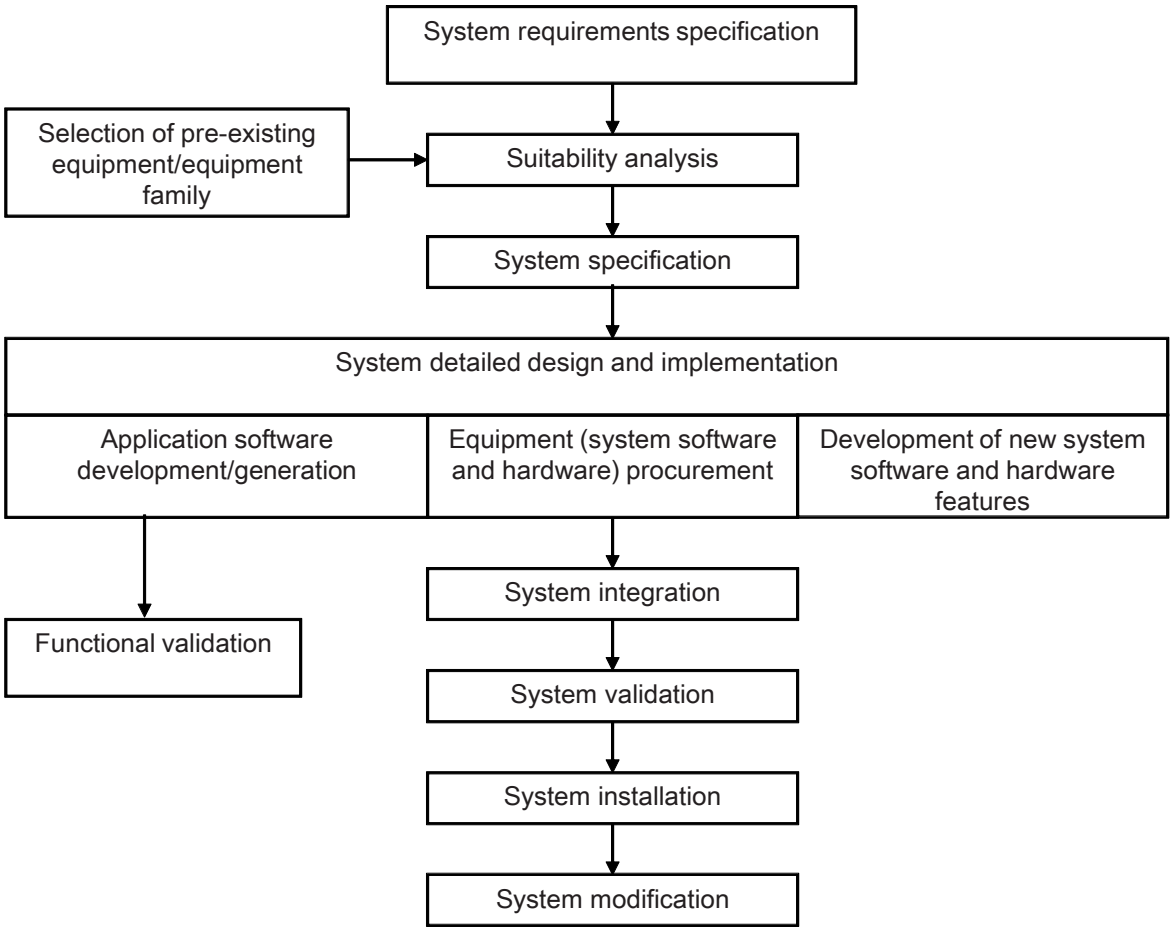

Figure 1 – System life-cycle (informative, as defined by IEC 61513)

The system life-cycle of IEC 61513 is complemented in IEC 60880 (for category A functions) and IEC 62138 (for category B and C functions) for software development and in IEC 60987 for hardware development of computer-based systems. The requirements of this Standard apply to the development of HPDs in class 1 systems, in addition to the requirements of IEC 60987.

NOTE In case of conflicting requirements, this Standard supersedes those in IEC 60987 about class 1 HPDs.

HPDs are developed by means of computer tools which tend to structure the development according to a cycle that includes activities dedicated to requirement capture, design and implementation, integration and validation, together with verification and test activities.

The system design and implementation phases of IEC 61513 shown in Figure 1 (particularly the “Equipment (system software and hardware) procurement” and “Development of new system software and hardware features”) are essential parts of the system life-cycle of IEC 61513. These phases are expanded in Figure 2 to illustrate in more detail the phases between the specification of requirements and the validation for the system components which are HPDs.

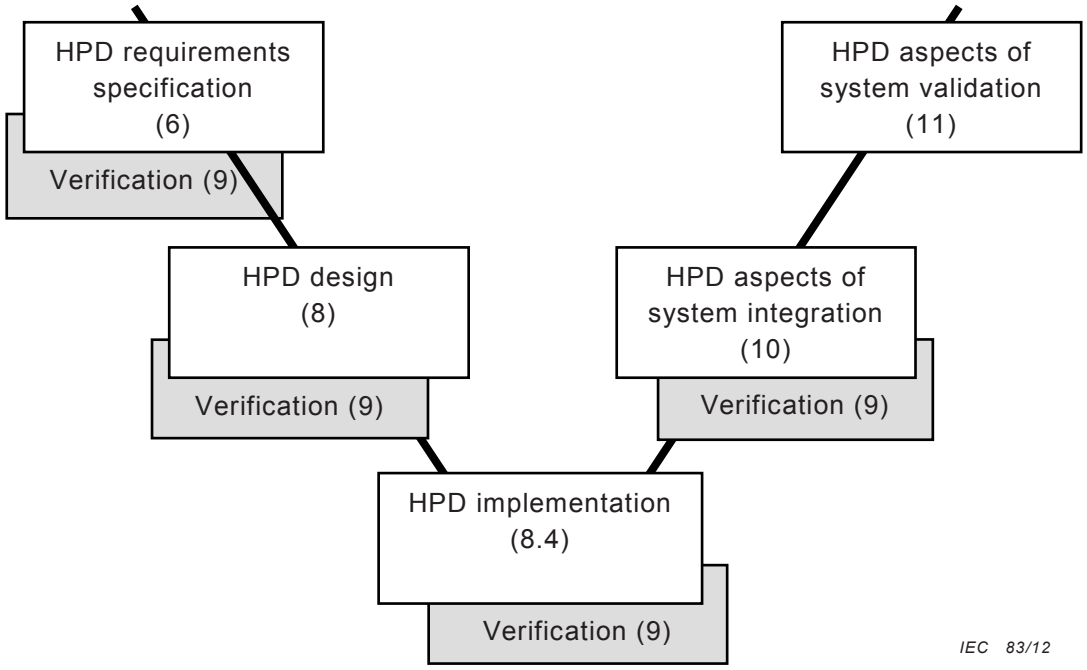

Figure 2 – Development life-cycle of HPD

Designers generally use pre-developed items such as programmable blank integrated circuits or Pre-Developed Blocks (PDB) to build integrated circuits specifically tailored to the needs of the project. The activities dedicated to the selection of these pre-developed items are addressed in Clause 7; they may be performed in parallel with the first phases of the life-cycle described in Figure 2, provided all dependencies are formally managed and documented.

The life-cycle described in Figure 2 shows the development life-cycle of one HPD that may be undertaken in parallel with the development of other components (software or hardware) of the system as shown in Figure 1, but coming together at the integration and validation phases of the system life-cycle.

The approach proposed to development is based on the traditional “V cycle” model as this approach has been reflected in other Standards and is also recommended in IAEA NS-G 1.3, but allowing necessary adjustments recognizing that some phases of the development can be done automatically by tools and that development may be iterative.

There is often no clear separation and well-identified boundary between the integration of a given component and the system integration. Therefore, in this Standard, the integration of a HPD is considered to be part of the system integration. Similarly, the validation of the HPD is considered to be part of the system validation.

Depending on the function achieved by the HPD, the system or subsystem to consider during integration may range:

1) from the I&C system when the HPD implements a safety function logic,
2) to an electronic board or cabinet when it implements a function (internal to the board or cabinet) that has been demonstrated, by suitable analysis, to be incapable of affecting the outputs of any safety function in the wider system.

The situation usually most critical from a safety standpoint is when the HPD directly implements the safety function logic.

The following activities support the development process of HPDs:

a) project management (5.3),

b) quality assurance and quality control (5.4),

c) configuration management (5.5),

d) verification (Clause 9).

There are also activities involving selection of tools to support the development (Clause 15), the production of documentation (Annex A) and modification (Clause 12).

5.3 HPD project management

5.3.1 General

5.3.1.1 Each HPD shall be developed within a dedicated HPD project.

5.3.1.2 The HPD project shall comply with the requirements of 5.4 of IEC 60880:2006 (by replacing “software” with “HPD”).

NOTE 1 A typical list of the documents required through the life-cycle is given in Annex A of this Standard.

NOTE 2 The documented inputs addressed by 5.4.6 of IEC 60880:2006 include parameters for the automated activities of the software tools (e.g.: optimize timing, optimize density, etc.).

5.3.1.3 The development process may be iterative; a phase may start before the activities of the preceding phase are complete; however, a phase shall only be terminated if the preceding phases have been completed and if its outputs are consistent with the inputs provided by these preceding activities.

5.3.1.4 The phases of the HPD project shall include the specification of requirements, the design and the implementation of the HPD.

5.3.2 Additional requirements

5.3.2.1 The selection of the pre-developed items used by the project shall be performed according to the requirements of Clause 7 of this Standard.

5.3.2.2 Transition criteria between phases shall be defined.

5.3.2.3 The criteria for phase termination shall have methodological and technical content, involving enough detail such that their evaluation requires an in-depth analysis of the phase outputs.

5.3.2.4 The documentation (5.4.11 of IEC 60880:2006) shall include the description of the functions performed by the HPD and its interface.

5.4 HPD quality assurance plan

A quality assurance plan for the HPD shall exist and shall comply with the requirements of 5.5 of IEC 60880:2006 (by replacing “software” with “HPD”).

NOTE In this context, “language” means “computer language”.

5.5 Configuration management

5.5.1 Configuration management of the HPD shall be performed according to the requirements of 5.6 of IEC 60880:2006 (by replacing “software” with “HPD”).

NOTE The segregation required by 5.6.6 of IEC 60880:2006 applies to the documentation and computer files used or produced by the HPD project.

5.5.2 The configuration management shall record the following items:

a) documentation of modules (blocks) developed within the project and of PDBs,

b) identification marking of integrated circuits,

c) computer files used for simulation, verification and production,

d) parameters used for the automated activities of the software tools (see Clause 15), such as “optimize timing, optimize density” for the Place and Route activity,

e) identification of the versions of all software tools (see Clause 15), including any “software patch” applied, as well as general purpose libraries and technology dependent libraries.

6 HPD requirements specification

6.1 General

6.1.1 A requirement specification shall document the requirements of the HPD, either in the document itself or by referencing sets of requirements stated at system or subsystem level (e.g. the functional behaviour to be implemented).

6.1.2 The requirement specification shall be understandable for all participants, including hardware engineers and people mentioned in 6.6.

6.1.3 The requirements specification shall be unequivocal, verifiable and achievable, including for temporal aspects.

6.1.4 When the HPD implements a safety function, its requirement specification shall be derived from the requirements of the I&C system implementing this safety function and shall be part of the specification of the subsystem which uses the HPD.

6.1.5 The requirement specification shall describe what is to be done and not how it is to be done.

6.1.6 A documented, formal and auditable process shall be defined and implemented for the establishment of the requirements specification.

6.1.7 The requirement specification shall be such that compliance with the requirement specification of the I&C system can be verified. If the HPD is used by a subsystem of the I&C system, it shall also be possible to verify the compliance with the system design specifications.

6.1.8 The requirement specification shall consider all plant operating conditions down to the HPD level for the functions that are impacted.

6.1.9 Interface requirements with other systems or components shall be addressed according to IEC 61513.

6.1.10 Interface requirements with other systems or components shall be documented.

6.1.11 When they are not part of the HPD requirements but result from HPD design decisions, the following interface requirements shall be documented:

a) electrical and temporal performance (e.g. input load, setup and hold time of inputs, operating frequency, fan-out, propagation time from any input to the associated outputs),

b) profiles of interfaced signal and power supplies,

c) power dissipation, operating temperature and cooling requirements.

6.2 Functional aspects of the requirement specification

This subclause describes the content of the requirements specification directly related to the functional needs. Subclauses 6.3 and 6.4 address additional aspects to be included in the Requirements Specification.

The requirement specification shall specify:

a) the functions to be provided by the HPD,

b) the HPD’s different modes, and the corresponding conditions of transition, including power-on and initialization,

c) the HPD's interfaces and interactions with its environment (operators and other I&C components), including the roles, protocols, types, data formats, bit numbering, ranges and constraints of inputs and outputs,

d) any HPD parameters which can be modified manually during operation, and their roles,

e) the HPD's performance, in particular response time,

f) what the HPD must not do or must avoid, when appropriate,

g) any assumptions regarding the HPD's environment (e.g. electrical and temporal characteristics of inputs-outputs, power supplies, specific profiles during power-on, cooling).

6.3 Deterministic design

The requirement specification shall specify that the function of the HPD is deterministic by design. This means that any given input sequence fulfilling the electrical and temporal specification always produces the same outputs.

NOTE Modern FPGA and other integrated circuits covered by this Standard can contain analogue functional blocks (e.g. analogue to digital converter) that are subject to electronic noise, digitisation error, etc. Variations in the response of these analogue functional blocks due to these causes, as well as their impacts on the response of the HPD, are not breaches in the deterministic design.

6.4 Fault detection and fault tolerance

The requirements of IEC 60987 subclauses 5.3 and 5.4 addressing the reliability with respect to random failures and the environmental withstand apply. This includes the faults due to SEU (single event upset) and neutron/alpha radiation when relevant.

Defensive design is typically based on a combination of techniques (e.g. redundancy, vote, parity and cyclic redundancy checks, watch-dog, range and plausibility checks).

6.4.1 The requirement specification shall specify requirements for defensive design to address fault detection and fault tolerance.

6.4.2 The benefit from defensive design measures should be balanced with their induced additional complexity. The overall objective is to take into account the testability of the HPD during design and implementation, using internal and external detection means to achieve high fault coverage.

6.4.3 The requirement specification shall describe the provisions to detect HPD malfunctions, taking into account the provisions already taken at subsystem or system level.

6.4.4 These provisions may need the HPD to provide additional outputs, either to be used by an external mechanism such as a watch-dog or to achieve the coverage of the supervision made by an external testing device.

6.4.5 The defensive design should allow the detection of erroneous behaviour (such as data corruption or deviation from specified processing algorithm, or deviation from specified operating conditions), erroneous data transmission between processing units, unintended modification of memories or configuration data.

6.4.6 The defensive design shall not have adverse influence on the I&C system functions, nor prevent the HPD from meeting its response time specification.

6.4.7 The requirement specification shall describe the expected logical and temporal behaviour (such as output values and specific information issued) when a fault is detected.

6.4.8 This behaviour shall comply with the system behaviour required by the system specification and with IEC 61513 system design requirements.

6.4.9 The requirement specification shall specify and justify the target coverage of the fault detection to be achieved by defensive design.

6.5 Requirements capture using Electronic System Level tools

6.5.1 General

This Standard does not prescribe a specific method to capture the HPD requirements. If they are captured using tools at Electronic System Level (ESL, see Clause B.1), then the requirements of 6.5.2 and 6.5.3 apply to these tools and to their use.

In the case of ESL, as the requirements specification language may be similar to implementation languages, it may be less practical to fulfil 6.1.5. (separation between what has to be done (the requirement) and how it is done (the design)). Provisions may be needed to fulfil it, e.g. comments to specify inputs, outputs and algorithms.

6.5.2 Requirements on the formalism of tools used at ESL level

6.5.2.1 When the HPD requirements are captured using an ESL tool:

a) this tool shall offer a formalism with a rigorous semantics and clarity (standardization of structure and presentation, modularity, sound comments);

b) the formalism used in the ESL tool shall be understandable for all participants;

c) if the tool offers flexible mechanisms to redefine functions and operators, then the actual characteristics of any given element should be clear to any participant, including hardware engineers and other personnel mentioned in 6.6.

6.5.2.2 The languages used at ESL level should allow taking due account of the system architecture, e.g. enable the assignment of functions to components, and support any fault tolerant design features.

6.5.3 Interface with design tools

The semantics of the languages used to express the requirement specification at ESL level may differ from the semantics of the HDL languages used during design. Examples where discrepancies may occur are in the interpretation of parallelism, the management of overflows, or the encoding of types and finite state machines.

a) If the semantics of the language used to express the requirement specification at ESL level differs from the semantics of the other languages used in the project, then discrepancies shall be identified for each involved item of the requirement specification;

b) each occurrence of a discrepancy within the requirement specification shall be documented. A generic list of discrepancies between the involved languages is a useful reference, but is not enough to clarify the Requirement Specification.

6.6 Requirements analysis and review

6.6.1 A critical analysis of the requirements shall be performed and documented, in order to find potential inconsistencies, omissions and ambiguities.

6.6.2 The scope of this analysis shall cover functional requirements and all other types of requirements, including those addressing abnormal behaviour such as unexpected input values or sequences.

6.6.3 The requirement specification shall be reviewed to check its completeness and its consistency.

6.6.4 For safety functions implemented in the HPD, process and I&C engineers shall participate in the review, as well as specialists of subsystems or components (including software) interfaced to the HPD.

7 Acceptance process for programmable integrated circuits, native blocks and pre-developed blocks

7.1 General

When developing the HPD, it is necessary to select and assess pre-developed items such as a blank integrated circuit (including their native blocks) or PDBs incorporated in the final HPD.

As these pre-developed items (or components) may include features not required for the HPD, the elaboration and the enforcement of specific “rules of use” may be recommended in order to restrict their use to what is needed and safe.

7.2 Component requirement specification

7.2.1 General

The requirements assigned to the pre-developed items (or components) result from the initial design activities of the HPD. For example the HPD requirements could include a specific pass-band filter, which the designer could implement using a PDB performing a Fast Fourier Transform.

Thus the component requirement specification (here for a Fast Fourier Transform PDB, defined by characteristics such as type of algorithm, radix size, decimation method, silicon area needed, etc.) differs from the HPD requirement specification (here for a Pass-band filter, defined by characteristics such as corner frequencies, gain, slopes, etc.).

7.2.1.1 A component requirement specification shall document the requirements applicable to each pre-developed item: blank integrated circuit, micro-electronic resources (seen as native blocks), associated tools when relevant or PDBs.

7.2.1.2 The component requirement specification shall state all the requirements, either in the document itself or by referencing sets of requirements stated at system or subsystem level (e.g. functional behaviour to be implemented).

7.2.1.3 Being the basis of the selection and use of the pre-developed item, the component requirement specification shall thus be understandable by all participants, including hardware and software designers when relevant, as well as verifiers, reviewers, and regulators.

7.2.1.4 The component requirement specification shall be unequivocal, verifiable and achievable, including for temporal aspects.

7.2.1.5 The component requirement specification shall be such that compliance with the requirements of IEC 61513 of the I&C system using this component can be demonstrated.

7.2.2 Requirements

The component requirement specification shall specify all characteristics required from the pre-developed item, in particular those of the list provided in 6.2.

NOTE The generic names of the characteristics (e.g. “function”) are identical to those of 6.2, but the contents differ in general as explained in 7.2.

7.2.3 Requirements analysis and review

7.2.3.1 A critical analysis of the component requirement specification shall be performed and documented, in order to find potential inconsistencies, lack of completeness or ambiguities.

7.2.3.2 The scope of this analysis shall cover functional requirements and all other types of requirements, including those addressing non-nominal behaviour such as unexpected input values or sequences.

7.2.3.3 The component requirement specification shall be formally reviewed by experts of all relevant domains to check its completeness and its consistency.

7.3 Rules of use

7.3.1 If the pre-developed item includes functions or operating modes that are not required to be implemented in the HPD, rules should be defined to prohibit the use of such functions and modes.

The use of functions or modes that are required to implement the HPD may be constrained by rules in order to improve design properties such as safety or testability.

7.3.2 If rules of use are established:

a) they shall be documented,

b) the quality plan shall give assurance that their fulfilment is verified during the project.

7.4 Selection

7.4.1 General

7.4.1.1 A documented analysis of each pre-developed item used in the HPD shall demonstrate that it fulfils the requirements of its component requirement specification, possibly with rules of use and modifications (see 7.6).

7.4.1.2 A user documentation for safety shall detail how designers are to use the pre-developed item consistently with its specification and design characteristics.

7.4.2 Documentation review

Documentation review is the primary method to demonstrate that the pre-developed item fulfils the component requirement specification.

7.4.2.1 This review should be based on the documentation of the pre-developed item including documentation of its design and its verification.

7.4.2.2 The documentation shall contain sufficient detail in order to demonstrate the fulfilment of the functional, electrical and temporal requirements of the pre-developed item.

7.4.2.3 The analysis of the documentation shall demonstrate that any functions and modes of the pre-developed item not used within the HPD do not impede the used ones.

7.4.3 Operating experience review

The operating experience of the pre-developed item may be invoked to compensate for some limited documentation weaknesses regarding its reliability or its design. If the operating experience is invoked then:

a) the analysis of the operating experience shall demonstrate that:
   1) its volume is commensurate to the reliability requirements,
   2) it has been collected in operating conditions equivalent to those in which the pre-developed item will be used,
   3) the actual use of the pre-developed item has been traced at the level of detail generally required by this Standard for the documentation;

b) the means and procedures used to collect the operating experience shall ensure that any pre-developed item failure that occurred in the analysed operation is recorded in such detail that a technical analysis can identify its cause as far as possible;

c) it shall be demonstrated by analysis of the failures recorded during operation that they do not impact the functions or the safety of the HPD;

d) the operating experience –and, if needed, complementary tests- shall demonstrate that the pre-developed item fulfils its requirements;

e) a documented technical analysis shall justify that all interactions of the pre-developed item with its environment are included within those covered by the operating experience;

f) the operating experience taken into consideration shall correspond to precisely identified versions of the pre-developed item and, when this item is specific to equipment, of the equipment in which it operates;

g) the operating experience should address the specific version of the pre-developed item or its sub-part used in the HPD; otherwise the differences between versions shall be analysed to demonstrate that the operating experience is relevant for the intended version.

7.4.4 Specific requirements related to the blank integrated circuits

7.4.4.1 The following aspect shall be addressed:

a) analysis of the adequacy of the programming mechanisms and circuitry;

b) demonstration that the programming process is fault-free or that any fault in this process is detected and correctly managed;

c) demonstration that the integrated circuit retains its programmed configuration for an adequate duration;

d) analysis of the potential for faults due to the additional internal and external mechanisms or power transients and justification according to the reliability requirements.

7.4.4.2 A detailed analysis shall demonstrate that:

a) the integrated circuit will be able to fulfil its component requirement specification,

b) the associated tools

– comply with Clause 15, and

– allow all verifications required by Clauses 8 and 9 (such as Static Timing Analysis).

7.4.4.3 The data needed to calculate the fault rate (in the sense of random physical faults) shall be available and based on sufficient operating experience.

7.4.4.4 The designers who design or implement the HPD shall have appropriate knowledge:

a) of the blank integrated circuit, including programming particularities, configuration and testing modes, protocols, pins and registers, and any electrical or logical specificity, and

b) of the associated tools, native blocks and PDBs. In particular, they should be able to predict, understand and (where necessary) control the choices made by the tools during synthesis, place and route.

7.5 Acceptance justification

7.5.1 A formal review shall examine the pre-developed item analysis, including the rules of use and the arrangements taken to ensure the compliance of each physical part used in production, to decide whether or not the pre-developed item is accepted for use in the HPD.

7.5.2 If the pre-developed item is accepted, any arrangement and rule of use taken into account in the analysis shall be applicable during the whole HPD life-cycle.

7.5.3 The review team shall include experts having skills relevant to the subjects (e.g. hardware technology, software) and engineers from the teams responsible for the components that are interfaced to the pre-developed item.

7.6 Modification for acceptance

7.6.1 If modifications of the pre-developed item are necessary to achieve acceptance, they shall be specified, designed, implemented and verified before the review.

7.6.2 These modifications shall be performed and documented in accordance with the requirements of this Standard regarding project structure and management, quality, specification of requirements, design, implementation and verification.

7.7 Modification after acceptance

The acceptance activities, including the review, shall be performed again after any modification of the pre-developed item involving its design or its micro-electronic aspects.

7.8 Acceptance documentation

The acceptance documentation of the pre-developed item shall be under configuration management.

7.8.1 The documentation shall include or shall make a reference to:

a) the requirement specification of the HPD,

b) all documents issued or invoked during the analysis of the pre-developed item,

c) all documents issued during modification of the pre-developed item,

d) the review report.

The documentation shall include any information necessary to use the pre-developed item correctly, taking into account constraints from its initial specification, from the rules of use, and from the modifications.

8 HPD design and implementation

8.1 General

This clause provides requirements and recommendations based on good practice for design and implementation, in order to meet appropriate safety features such as fault-free as possible and amenability to verification.

8.1.1 The development process shall define a design phase and an implementation phase.

8.2 Hardware Description Languages (HDL) and related tools

Even though the use of specific languages and tools cannot be required, the following may be considered as common basic rules for languages and tools used for the design and implementation of HPDs for class 1 systems.

8.2.1 Design and implementation should use Hardware Description Languages (HDL) and tools for simulation, synthesis, place and route.

NOTE When properly chosen and used, these tools improve essential aspects such as understandability of the descriptions, management of electrical and temporal constraints, verification, adequateness of coverage criteria, and documentation.

8.2.2 Even if 8.2.1 is not fulfilled, any documentation, analysis, or verification required by this Standard shall be provided.

8.2.3 The language in use:

a) shall follow strict (or well-defined) semantic and syntax rules;

b) shall have a syntax completely and clearly defined and documented;

c) should comply with a recognized Standard (e.g. IEEE 1076 for VHDL or IEEE 1364 for Verilog).

8.2.4 The use of the language should be restricted to a “safe” subset where appropriate, for example be restricted to features that are needed to implement the required functions and are synthesisable with standardized libraries (e.g. avoid the use of initial values, explicit delays or division).

8.2.5 The simulator in use shall produce results strictly compliant with the documented semantic of the language.

The simulator should comply with a recognized Standard (e.g. IEEE 1076 for VHDL or IEEE 1364 for Verilog).

8.2.6 Except as addressed in 8.2.7, only tools complying with the requirements of Clause 15 shall be used for analysis, simulation, synthesis, place and route. It is not necessary for users to repeat testing of the tools if this has already been performed and documented by the supplier.

8.2.7 If a tool partially compliant with the requirements of Clause 15 is used, additional verification of the results produced by this tool (e.g. netlist produced by a synthesis tool) shall provide evidence that the results are correct. Formal equivalence checking tools are valuable in achieving an error free design.

8.3 Design

8.3.1 General

Starting from the HPD requirement specification, the design initially aims at defining major choices such as the decomposition into modules (application-specific or pre-developed), the operation of the defensive design, as well as the identification of needed micro-electronic technologies (including their native blocks) and PDBs. Then an RTL description is built using HDLs. The following requirements aim at producing a clear and verifiable design.

8.3.1.1 The design phase shall produce a) a formalized description of the HPD, e.g. RTL and b) the associated documentation.

8.3.1.2 Communication links shall be designed in compliance with the requirements on data communication given in 5.4.2.4 of IEC 61513.

8.3.1.3 The design should allow easy verification.

8.3.1.4 Non-compliances with design rules should be justified.

8.3.2 Defensive design

8.3.2.1 When a selected native block or PDB (see Clause 7) is a processor core, it should support the requirements of IEC 60880 for self-supervision.

8.3.2.2 The design shall take into account the arrangements selected in the requirement specification to detect the faults and to elaborate the corresponding information within the HPD.

8.3.2.3 On fault detection, the HPD shall behave in accordance with the corresponding specified requirements.

8.3.3 Structure

8.3.3.1 A top down approach to design should be preferred to a bottom up approach.

NOTE Library items are the ultimate targets of the design. Therefore the use of libraries fulfilling the requirements of Clauses 7 and 15 is in line with the top-down approach and is recommended.

8.3.3.2 The structure of the design should be based on decomposition into modules. Related modules may be contained in a library.

8.3.3.3 Generic modules should be contained in libraries.

8.3.3.4 The structure should be simple and easy to understand, both in its overall design and in its details.

8.3.3.5 A conceptual model of the architecture should be generated at the beginning of the project.

8.3.4 Language and coding rules

8.3.4.1 In order to facilitate a stable and reliable design, proven design methodology and general good practice should be used.

8.3.4.2 In order to make the design more understandable and to reduce the potential for differences between the simulated and the synthesized behaviours:

a) a set of strict design rules which reflect the latest knowledge in terms of design safety and reliability shall be required by the quality plan and established;

b) the compliance with those design rules shall be enforced by appropriate means (e.g. review, tooling, etc.).

8.3.4.3 The list given below contains strongly recommended design considerations and techniques. However the list is not considered to be all-encompassing and parts may change with technology. Nevertheless any non-compliance with the rules in the following list shall be justified and taken into account in failure analysis:

a) only synthesizable features of the language should be used in the design of the HPD. The test and simulation environment (9.5) may use all language features. Any native blocks (see 3.9) which are already synthesized and routed in the pre-developed integrated circuit may be instantiated as they are, if they comply with Clause 7;

b) dedicated resources or design features (e.g. predefined clock trees and clock conditioning circuits, power rails, reset trees, etc.) should be used where appropriate;

c) coding rules should cover all relevant aspects, in particular naming of modules and signals, use of the structuring features (such as packages, functions, procedures, project libraries, instantiation), organization of the computations on critical paths, organization of processes, recommended constructs, forbidden constructs;

d) functions using side effects ("impure") should be forbidden in the design description. (Rationale: such a function can return different values when called several times with the same parameters. It is therefore very difficult to test and verify, as it breaks the basic concept of a function, and in fact of determinism);

NOTE 1 An impure function may also have side effects such as modifying objects out of their scope.

e) constructs that could lead to differences between simulated and synthesized behaviours should be forbidden. Depending on the language used, examples of such constructs may be the incomplete or conflicting assignments, use of "don't care" character in comparisons, comparisons (higher or lower) involving enumerated types (Rationale: simulation is an important verification method. If simulated and synthesized behaviours differ, then the verification chain is broken);

f) signals and variables should not be initialized at their declaration in the RTL description, but by an explicit mechanism such as reset (Rationale: initialization in HDL may lead to differences between simulated and synthesized behaviours);

g) use of explicit delays should be forbidden in the design description, as such delays lead to differences between simulated and synthesized behaviours;

NOTE 2 This does not forbid the existence of delays at system level or in the requirements of the HPD. It means that such delays cannot be implemented by a “delay” or “after” instruction in HDL, but e.g. by counters or shift registers.

h) creation of delays by means of combinatorial gates or by depending upon propagation delays along wires should be forbidden in the design description. If such design cannot be avoided, STA shall be done to justify the usage of such design (Rationale: such delays are not stable over parameters such as temperature, voltage, or from one part to another, or from one area of the die to another);

i) the types of the interface signals of the HPD should be defined in a clear and non-ambiguous way, preferably standardized, independently of any tool or micro-electronic technology;

j) HDL level definitions should not allow different interpretations, to avoid variations when compiled under different conditions. E.g. inputs / outputs of the HPD should be explicitly assigned to known pins.

NOTE 3 This subclause does not apply to the design of library components, which are build to be instantiated in different locations of future designs with different input/output allocations.

NOTE 4 To design HDL code that can be transferred between different technologies it is necessary for the pin allocation to be defined in a constraints file, not in the HDL code. Language features such as templates in VHDL-2008 may help doing this.

8.3.5 Synchronous vs asynchronous design

Synchronous design consists in enforcing the change in the state of the internal registers and of the outputs simultaneously only at times defined by a clock. It favours a modular and understandable design, it minimizes the potential for wrong behaviours due to glitches, and it favours the best use of synthesis and verification tools.

8.3.5.1 In order to facilitate stable, robust, and clearly structured designs:

a) a strictly synchronous architecture should be used;

b) non-compliances shall be justified.

8.3.5.2 The design shall ensure that signals at asynchronous interfaces are synchronized.

8.3.5.3 If an asynchronous architecture is used, a documented analysis of all paths shall demonstrate that the outputs comply with the Requirement Specification (Clause 6) and that there is no adverse glitch or metastability.

8.3.5.4 The HPD behaviour shall not be subject to the actual values of the internal propagation delays along wires and through gates

8.3.6 Power management

8.3.6.1 The internal electrical and temporal characteristics of the blank integrated circuit during power-up/start-up, power-down and sudden loss of power shall be known and taken into account in the design.

8.3.6.2 The behaviour of each pin (such as input or output type, impedance) during power-up/start-up, power-down and sudden loss of power shall be documented.

8.3.6.3 The use of HPDs based on programmable technology shall not rely on the assumption that they behave in accordance with their programmed behaviour (regarding e.g. functions, direction and impedance of each pin) during power-up/start-up, power-down and sudden loss of power, even in the case of one-time programmed devices.

8.3.6.4 The connection of input pins to a voltage source or to ground should follow the supplier’s application notes, in order to avoid potential current peaks during power-up/start-up, power-down and sudden loss of power.

8.3.6.5 When the power distribution is not predefined by the component supplier, particular care shall be taken in its design to avoid non-deterministic faults due to problems such as voltage transients due to current peaks on clock edges.

8.3.7 Initialization

8.3.7.1 The design shall have an input signal that puts all outputs, registers and finite state machines in a known and documented state.

8.3.7.2 The initialization signal, which is not always of purely digital nature, shall comply with the blank integrated circuit requirements such as rise time, fall time or monotonicity.

8.3.7.3 Asserting this signal shall produce the intended effect even when no clock activity is running.

8.3.7.4 De-asserting this signal shall be done in such a way that all outputs, registers and finite state machines are maintained in their initial known state until the clock activity is running.

8.3.8 Non-functional configurations

8.3.8.1 Special pins and registers that make the HPD switch to special configurations (such as test, diagnostic, debugging or programming) and which are not specified by the HPD Requirements Specification shall be analysed and configured in order to exclude any adverse impact on its functions.

8.3.8.2 The designers shall be familiar with the integrated circuit supplier documentation to know the characteristics given by the tools to the unused pins (input, output, high impedance, etc.).

8.3.8.3 The management of the configuration pins and registers of the HPD shall be documented.

8.3.9 Testability

8.3.9.1 Each function implemented in the HPD shall be testable (detection of failures), by means such as self-tests, periodic tests or observable contribution to a higher level function which is itself submitted to self-tests or periodic tests.

8.3.9.2 When self-test devices are used, their capability to perform their function shall be verified.

8.3.9.3 The actual coverage of fault detection (see 6.4.9) and periodic tests shall be determined and comply with the HPD Requirement Specification.

8.3.9.4 The consequences of faults shall be minimized, e.g. by detecting when states normally unreachable are reached and by taking a predefined action in such cases.

8.3.10 Design documentation

8.3.10.1 The end of the design phase shall be marked by production of the corresponding documentation.

8.3.10.2 The documentation shall describe and justify the adequacy of the design decisions in fulfilling the HPD requirement specification.

8.3.10.3 The design documentation shall be comprehensive enough so that implementation can proceed without further clarification.

8.3.10.4 The documentation shall describe the design decisions such as:

a) the organization in modules, as well as their interfaces and relations,

b) the control flows and data paths,

c) the protocols and algorithms,

d) the types, formats, and logic conventions of the signals,

e) the numbering of buses, the memory map,

f) the finite state machines definitions, encoding, and initializations,

g) the initialization value of all registers,

h) the test circuitry.

8.3.10.5 The design documentation shall define the variant actually used for each instantiation of each library component, to avoid ambiguities when variants with different speeds or electrical characteristics exist.

8.3.10.6 The design documentation shall include all parameters needed to unambiguously configure and use all native blocks and PDBs.

8.3.10.7 The design documentation shall include the estimated timings and electrical characteristics.

8.4 Implementation

8.4.1 General

Starting from the RTL description, the implementation synthesizes the gate-level description (netlist) of the HPD. Then place and route is performed and results in the physical description needed to produce the HPD, such as programming file or "bitstream".

8.4.2 Products

8.4.2.1 The implementation shall generate all information necessary to produce the HPD in a systematic way and to verify that each produced part complies with the design.

8.4.2.2 The implementation shall produce timing information to supplement the RTL description ("back-annotations") in order to precisely simulate the temporal behaviour taking into account all delays associated with gates and wires.

8.4.2.3 The back-annotated description shall be usable in the test-bench (9.5) and, when appropriate, in higher level tools such as board level simulation.

8.4.3 Files of parameters and constraints

The designer directs the synthesis, place and route operations with parameters and directives which specify constraints such as needed operating frequency, timing relations between signals, or fan-out. To fulfil these constraints (provided to the tools in “constraint files”) the tools may modify the placement to favour a given propagation path at the expense of other ones, duplicate one gate to reduce the load on each copy and thus increase their speed, etc.

Errors or omissions in parameters and constraints files may result in subtle non-deterministic faults, often not detectable during simulation and sensitive to normal variations of the micro-electronics process.

8.4.3.1 The files of parameters and constraints shall be built according to an auditable process.

8.4.3.2 The completeness and the correctness of the files of parameters and constraints shall be verified by the verification team (see Clause 9).

8.4.3.3 The files of parameters and constraints shall be documented and placed under configuration management.

8.4.4 Post-route analyses

8.4.4.1 A post-route analysis shall demonstrate the compliance of the design and implementation with the technology rules defined by the suppliers of the design and implementation tools and of the micro-electronic technology.

8.4.4.2 Post-route analyses or simulations (taking into account the post-route timing information, or back-annotations) shall confirm the cycle by cycle equivalence of the post-route description to the RTL description for fastest and slowest cases, including initialization, for example by using the two following steps:

a) demonstrating that the post-synthesis description is cycle by cycle equivalent to the RTL description,

b) demonstrating that the post-route description complies with the timing constraints.

8.4.4.3 Post-route simulations may use a subset of the test-bench cases used in the RTL simulations (see 9.5). It shall be justified that this subset covers the needs of equivalence demonstration. An alternative or complementary method to post-route simulation is to use a tool that will check that the RTL and the physical description level are mathematically equivalent. If this approach is adopted the tool used to perform this check shall be assessed for quality and suitability before use (see Clause 15).

8.4.4.4 Post-route timings shall be analysed.

8.4.4.5 The coverage of each function by self-supervision shall be analysed with respect to the required target (see 6.4.9) taking into account the effects the tools may have on the actual topology.

8.4.4.6 These analyses shall be detailed enough and documented, in order to allow further technical assessment by people not involved in the design and implementation.

8.4.4.7 Some of these analyses may be performed, optionally or automatically, by the tools. In that case it is not required to perform them again, but:

a) it shall be demonstrated that the analyses performed by the tools have appropriate coverage and correctness;

b) the analysis reports (including set-up and results) provided by the tools shall be included in the documentation.

8.4.4.8 If the analyses find non-compliances deemed acceptable then:

a) this acceptability shall be justified and documented;

b) all impacted documents shall be modified accordingly;

c) the quality plan shall ensure that any impact on other systems or components is documented and adequately taken into account by the people responsible for the impacted systems and components.

8.4.5 Redundancies introduced or removed by the tools

8.4.5.1 The replications of gates made by the tools to meet timing or technology constraints shall be analysed.

8.4.5.2 It shall be demonstrated that the additional states introduced by these replications are acceptable regarding the functional and safety requirements. It is recognized that gate replication is performed by many synthesis tools, but usually, this can be adequately controlled in the tool itself. However, care shall be taken as gate replication may cause problems if the same formal equivalency check is used to prove the RTL and the gate implementation.

8.4.5.3 As replication introduces new states they, shall be analysed to demonstrate that the safe behaviour of the design cannot be affected.

8.4.5.4 On the other hand, it shall be demonstrated that the logic optimization performed by the tools has not removed fault detection and tolerance mechanisms such as redundancies or processing of cases normally unreachable.

8.4.6 Finite state machines

8.4.6.1 The robustness of the final implementation of finite state machines shall be analysed.

8.4.6.2 In particular, finite state machines shall not have dead states other than those possibly specified in the HPD requirement specification.

NOTE A dead state is a state from which the finite state machine cannot reach any other state.

8.4.6.3 The potential additional states introduced by some encoding methods (such as "one-hot") shall be taken into account in failure analysis.

NOTE “one-hot” encoding uses one flip-flop per state to be represented; each particular state is represented by one specific flip-flop set to “true” and all others set to “false”. Thus, only combinations with exactly one flip-flop set to “true” are valid. In case of failure, several flip-flops could be simultaneously set to “true”, which would correspond to additional, undefined states.

8.4.7 Static timing analysis

8.4.7.1 A Static Timing Analysis (STA) shall be performed and documented for worst and best cases to calculate the margins, taking into account the timing information provided by the technology libraries and all relevant design and implementation tools.

8.4.7.2 If paths are excluded from STA (because seen as "false paths") or declared as multi-cycle paths, this decision shall be justified and documented.

8.4.7.3 STA shall demonstrate that the frequency of each clocked block is compatible with all non-excluded paths (see 8.4.7.2) with sufficient margin within the specified variability of the micro-electronic technology.

8.4.7.4 The effect of the clock skew on critical structures such as shift registers shall be analysed and documented.

NOTE The clock skew is the amount of time between the arrivals of the clock signal at different locations.

8.4.8 Implementation documentation

The end of the implementation phase shall be marked by production of the corresponding documentation including:

a) the gate-level description of the HPD, usable in the same test-bench as used at RTL level,
b) the technology specific description (e.g. “programming file”) necessary to program the HPD and to test each part (13.2),
c) the back-annotations that take into account all delays associated with gates and wires,
d) the timings (such as frequency, set-up and hold times, rise and fall times, propagation times) and electrical characteristics (such as voltage levels, input currents, fan-out, impedances, and power consumption) predicted by the tools unless they are already defined in the datasheet.

8.4.8.1 The implementation documentation shall:

a) give access (by inclusion or reference) to the implementation of each block, sub-block or module,
b) describe the choices made, in particular regarding testability, clock and power distribution, reset and critical paths implementation.

8.4.8.2 The implementation documentation shall describe and justify:

a) the constraints and parameters provided to the tools,

b) the analysis performed to guarantee the compliance of the HPD with its requirement specification, and any differences found,

c) any iterations made on design and implementation,

d) any redundancy added or removed during implementation.

8.4.8.3 The documentation shall be detailed enough to allow an engineer not involved in the project to run the synthesis, place and route tools and get the same results (HPD and verification output), as well as to verify the completeness and the correctness of the post-route analyses.

8.4.8.4 The documentation shall describe the tests to be performed periodically in operation, with due care to the structural modifications introduced by the tools.

8.4.8.5 When the commitment of the integrated circuit supplier on the design or implementation is required before production, this commitment shall be included in the documentation.

8.5 System level tools and automated code generation

The requirements of the different components of a system may be captured using ESL tools that provide a textual or graphical description.

This subclause provides guidance applicable when an ESL description of the HPD requirements is used in an automated way to generate part or all of the HPD design. This generation is sometimes called “high-level synthesis”.

8.5.1 If a requirement specification written in an ESL language is used to automatically generate an RTL description of the HPD or a part of it:

a) the generated description should be straightforward and avoid unnecessary complexity;

b) this description should allow the behaviour of the device to be easily understood so that errors and ambiguities can be identified promptly by the hardware design engineers.

8.5.2 The ESL language and the associated tools, in particular those used for code generation and analysis, should comply with the requirements of 8.2.

8.5.3 If 8.5.2 is not fulfilled:

a) the ESL description of the HPD shall be translated into an HDL description compliant with the requirements of 8.2, which will be the basis for the following activities of design, implementation and verification,

b) these following activities shall comply with the requirements of this Standard.

8.5.4 Any non-conformance of the generated descriptions (such as RTL, synthesized, routed) with the requirements for design and implementation (8.3 and 8.4) shall be identified and justified.

8.5.5 If some of the analyses, verifications and reviews defined by this Standard in Clauses 8, 9 and 10 are not performed, it shall be formally proven that the products which have not been analysed, verified or reviewed are necessarily correct.

8.5.6 The generated products shall not be modified by direct manual action on the products.

8.5.7 The products shall be regenerated if anything has to be modified, for example with respect to findings from verification or review activities.

8.6 Documentation

This subclause gives the general documentation requirements for design and implementation of the HPD. It supplements the requirements specific to particular activities addressed in 8.1 to 8.5.

8.6.1 The end of the design and implementation phases shall be marked by the production of the HPD design specification.

This document serves as the basis for the formal design and implementation review and the subsequent production.

8.6.2 Sufficient detail shall be included so that production can proceed without further clarification.

8.6.3 The document should be structured according to the phases of the development process. The design specification may be expressed as one document or as an integrated set of documents.

8.6.4 If multiple documents are used, each document shall have a defined relationship to the other documents and shall contain a well-bounded subject-matter.

8.6.5 Documentation formats should be selected according to the specific topics, including:

a) narrative description;

b) arithmetic and logic expressions;

c) graphical representations, diagrams and drawings.

8.7 Design and implementation review

8.7.1 The design and implementation phase shall be terminated by a formal review.

8.7.2 The design and implementation review shall examine the documentation, covering design, implementation, analyses and verifications.

8.7.3 The review shall examine the completeness and correctness of the files of parameters and constraints provided to the design and implementation tools.

8.7.4 The review shall examine the completeness and correctness of the Static Timing Analysis (STA) and post-route analyses, to check the correctness and the robustness of the design and implementation with due consideration to the potential adverse effects induced by the modifications made by the tools (such as logic simplification or gate duplication).

8.7.5 The review team shall include hardware experts and engineers from the teams responsible for the system or components that use the HPD or are interfaced to it (such as electronic board or software).

9 HPD verification

9.1 General

The verification activities undertaken as part of the HPD development are usually under the responsibility of the I&C producer and are undertaken by staff independent of those performing the HPD design and implementation. The most appropriate way is to engage a verification team.

Additional verification activities may be undertaken as part of a third party assessment of the HPD and of its development process in order to provide assurance that it will meet its targets. There are many ways by which this independent verification role can be resourced and implemented, this often being a matter of national regulatory preference.

9.1.1 The verification team shall be composed of individuals who are not engaged in the development and who have the necessary competencies and knowledge. The following requirements define explicitly the level of independence required.

9.1.2 The management of the verification team shall be separate and independent from the management of the development team.

9.1.3 Communication between the verification team and the development team, whether for clarification or fault reporting, shall be conducted formally in writing at a level of detail which may be audited.

9.1.4 Interactions between the two parties should aim at maintaining the independence of judgment of the verification team.

9.1.5 The verification team shall have clearly defined responsibilities and obligations.

9.1.6 The output of each development phase (Figure 2) shall be verified.

9.1.7 The verification activities shall confirm the adequacy of the HPD requirement specification in fulfilling the system or subsystem requirements assigned to the HPD by the system or subsystem specification.

9.1.8 The verification activities shall confirm the adequacy of the selection and rules of use of each blank integrated circuit, micro-electronic technology, native block and PDB in fulfilling its component requirement specification (see Clause 7).

9.1.9 The verification activities shall confirm the adequacy of the HPD design specification in fulfilling the HPD requirement specification.

9.1.10 The verification activities shall confirm the compliance of the HPD with the HPD design specification (see Clause 8).

NOTE Post-implementation verification is of major importance to detect the potentially adverse effects of logic simplifications and gate duplications that may be performed by the tools, as well as the potential faults resulting from the tools themselves or from their use.

9.1.11 Each production activity should be started on a basis of verified input data/documents.

9.1.12 Verification of the product of a phase should be performed before the start of the next phase. Otherwise, this verification shall be performed before the verification of the next phase.

Possible preparatory work for a subsequent phase may be done before the precedent phase has been verified.

9.1.13 If input data/documents for an activity have been modified, that activity and subsequent activities shall be repeated as necessary to address potential impact.

9.2 Verification plan

9.2.1 The verification plan shall be established prior to starting HPD verification activities.

9.2.2 The plan shall document all the criteria, the techniques and tools to be utilized in the verification process.

9.2.3 It shall describe the activities to be performed to evaluate each item of the HPD, each tool involved in the development process, and each phase to show whether the HPD requirements specification is met.

9.2.4 The level of detail shall be such that a verification team can execute the verification plan and reach an objective judgement on whether or not the HPD meets its requirement specification.

9.2.5 The verification plan shall be prepared by a verification team addressing:

a) selection and justification of verification strategies according to the nature of the requirements, to the design and implementation characteristics, and to the micro-electronic technology;

b) selection and utilisation of the verification tools;

c) execution of verification;

d) documentation of verification activities;

e) evaluation of verification results gained from verification tools directly and from tests, evaluation of whether the safety requirements are met or not.

9.2.6 The verification plan shall document each test, including its goal, its expected results, and the criteria to decide whether the result is correct or not.

9.2.7 The tests designed according to functional aspects should result in extensively exercising the HPD.

9.2.8 The verification plan shall identify any objective evidence required to confirm the extent of testing. For that purpose the test coverage criteria chosen according to the design and implementation shall be justified and documented.

9.2.9 There shall be adequate provision for the processing and resolution of all safety issues raised during the verification activities performed either during the development by the I&C producer or by a third-party assessment.

9.2.10 All safety issues shall be adequately resolved through appropriate corrective modifications or mitigating dispositions.

9.3 Verification of the use of the pre-developed items

The correct configuration and use of the pre-developed items such as blank integrated circuits, native blocks and PDBs, as well as their mutual compatibility, shall be verified against the rules specified by their suppliers and against those elaborated during the activities described by Clause 7.

9.4 Verification of the design and implementation

9.4.1 The verification shall include test and analyses to address:

a) the adequacy of the design specification for the HPD requirement specification with respect to consistency and completeness down to and including the lowest block and module level;

b) the decomposition of the design into a hierarchy of blocks and modules and the way they are specified with respect to:

   1) testability for further verification;

   2) understandability by the development and verification teams;

   3) modifiability to allow for further modification;

c) the correct implementation of safety requirements;

9.4.2 The result of the verification shall be documented.

9.4.3 The documentation shall include the conclusions and identify clearly issues that need actions, such as:

a) items which do not conform to the requirements;

b) items which do not conform to the design and implementation rules;

c) modules, data, structures and algorithms poorly adapted to the problem.

9.5 Test-benches

9.5.1 A simulation and test program (test-bench) shall be developed and documented. As needed, this test-bench may consist in several implementations, each with a different scope and objective, e.g. some test-benches may be dedicated to modules and one or more to top-level testing.

9.5.2 The test-bench (structure) may be developed by the design team for its own testing needs and used by the verification team. However, the test vectors (inputs and expected outputs) required by this Standard shall be developed by the verification team, so as to reduce the potential for error masking and to give additional confirmation of understandability and completeness of the design documentation.

9.5.3 The test-bench shall

a) exercise each module in its environment simulated with all needed logical details,

b) provide sufficient timing resolution when used after implementation for the temporal aspects.

9.5.4 The test-bench shall include test cases which exercise all the features mentioned in the HPD requirement specification and in the design specification such as functions, modes, finite state machines, algorithms, protocols.

9.5.5 The test-bench should include all required inputs, sequences and timings, and record all output sequences and timings produced during execution, to make the test execution fully automated.

9.5.6 The test-bench should include the expected output sequences and timings as well as an automated comparison with those actually produced during test execution (with respect to the adequate criteria, see 9.2), so as to provide a global "pass/fail" output in addition to the detailed test results.

9.5.7 If manual inputs, observations or comparisons are required:

a) the involved data and activities shall be documented with sufficient detail to allow a person not involved in the project to repeat the test. This may need definition of cycle by cycle steps and bit level values,

b) a documented justification shall be provided, because manual inputs, observations and comparisons are potentially error-prone.

9.5.8 The test-bench has to accurately report all failures and not falsely report passes. It shall therefore be built in accordance with 10.4.6 and 15.2.

9.6 Test coverage

9.6.1 Test coverage criteria shall be selected and documented.

9.6.2 A documented analysis of the test coverage criteria shall demonstrate that they are sufficient regarding the HPD requirement specification and design/implementation characteristics, and that the test-bench provides sufficient observability to take a pass/fail decision for each covered element.

9.6.3 Such criteria may be related, for example, to instructions, decisions, expressions, paths, finite state machines, or processes. If a coverage criterion target could not be achieved, e.g. due to RTL structure (100 % path coverage is particularly difficult to achieve), then a documented justification shall be produced.

NOTE A path is a particular sequence of branches taken when executing the code.

9.6.4 Each module developed within the project shall be specifically tested.

9.7 Test execution

9.7.1 Tests shall be performed using the test-benches following the design phase on the RTL description, in order to confirm its correctness.

9.7.2 Tests shall be performed after the implementation phase to confirm that the post route description complies with the timing constraints, taking account of the timing information provided by the tools and libraries (back-annotations).

9.7.3 The tests (using simulation) shall be performed for both “worst case” (maximum propagation delay) and “best case” (minimum propagation delay).

9.7.4 The test results (values, sequences, and timings) shall be documented.

9.7.5 A documented analysis of any discrepancy shall decide whether it is acceptable or not.

9.8 Static verification

9.8.1 The following verification activities should be performed:

a) type and syntax checking,

b) parameter checking in call or instantiation of modules, functions, procedures, native blocks and PDBs,

c) out of range checking,

d) completeness of the sensitivity list of processes (see note),

e) completeness of the cases explicitly programmed in instructions and constructs with multiple choices,

f) detection of dead states in finite state machines,

g) detection of side effects in functions or macros, detection of shared objects,

h) logical and physical DRC (Design Rule Check), which test the netlist and other generated files for physical and logic errors.

NOTE “Sensitivity list” is an element of VHDL.

9.8.2 Static verification methods such as STA (see 8.4.7) may be used for some aspects of the verification if their principles are mathematically sound. In this case, the tools used to implement these methods shall:

a) have maturity and standardization similar to those required by this Standard for the simulation tools,

b) comply with the requirements of Clause 15 applicable to verification tools.

10 HPD aspects of system integration

10.1 General

The process of system integration is the combining of verified hardware (and software when applicable) components into subsystems and finally into the complete system. This process consists of two kinds of activities:

a) system integration: assembling and interconnecting verified hardware components (and software components when applicable) in order to build the intermediate and final targets. The assembly sequence as well as the degree of integration of the successive targets depend on the project characteristics;

b) integrated system verification: verifying that the components comply with their design specification, are capable of operating together, and comply with their interface requirements.

This clause gives requirements for the system integration in supplement to 6.2.5 of IEC 61513, when HPDs are involved.

10.2 HPD aspects of the system integration plan

This subclause amplifies the requirements of IEC 61513, 6.3.4., which shall apply.

10.2.1 This plan shall be prepared and documented in the design and implementation phases and verified against the class 1 system requirements.

10.2.2 This plan shall be prepared sufficiently early in the development process to ensure that all integration requirements are included in the design of the HPD, of the system and of its components.

10.2.3 This plan shall specify the Standards and procedures to be followed in the system integration phase.

10.2.4 This plan shall document those provisions of the system quality assurance plan that are applicable to the system integration.

10.2.5 The integration plan shall specify:

a) the sequences and timings of input signals to the system or subsystem being tested,

b) the sequences and timings of expected outputs from the system or subsystem being tested,

c) the acceptance criteria.

10.2.6 The system integration plan shall take into account the requirements to be fulfilled by the HPD through the design of the system, the design of the hardware and the design of any software. The plan shall also include the requirements for procedures and control methods covering:

a) system configuration control (5.5);

b) system integration;

c) integrated system verification;

d) fault resolution.

10.2.7 The system integration plan shall define both the identification and control aspects of configuration management, according to the requirements of IEC 61513, 6.3.2.3.

10.2.8 In the process of verifying the interactions of the HPD with the other components of the system, certain aspects may be verified at the level of subsystems (computing units) or at the level of the complete system if more practical. When verification by testing is not feasible at these levels, then:

a) all requirements of the HPD shall be verified by other means (e.g. white box testing),

b) the corresponding verification strategy shall be documented in the integration plan.

10.2.9 All interdependencies between the verification of the HPD and the verification of the integrated system shall be documented in the system integration plan.

10.3 Specific aspects of system integration

The specific procedures for the system integration depend on the characteristics of the system architecture.

10.3.1 Procedures shall be established and referenced by the system integration plan to cover the following activities:

a) the acquisition of the correct components according to the system configuration management plan (6.3.2.3 of IEC 61513) and to the production procedures (Clause 13);

b) the integration of the HPD into the system (e.g. component positioning, configuration, interconnection wiring);

c) the preliminary functional test of the integrated system functions (see requirements below);

d) the documentation of the outcomes of the integration process and the system configuration subjected to the tests;

e) the formal release of the integrated system for validation testing.

10.3.2 If the resolution of a fault requires a modification to the verified HPD or the design specification, that fault shall be reported according to the procedures established in 10.5.

10.3.3 Any faults detected during the system integration which are only mistakes in the integration process itself, and which do not affect any HPD document, shall be corrected by updating the system integration plan.

10.4 Verification of the integrated system

The verification of the integrated system determines whether or not the verified components and the subsystems have been properly integrated into the system, that they are compatible and perform as specified.

10.4.1 The system shall be as complete as is practical for this verification.

10.4.2 The test cases selected for system verification shall:

a) exercise all HPD interfaces and all basic operations;

b) exercise all interface characteristics of the HPD described in the requirement specification and in the design specification such as protocols, sequences, timings and electrical features;

c) have sufficient coverage to demonstrate that the HPD performs as required for all cases reachable in the system.

10.4.3 The system integration plan shall identify the tests to be performed for each HPD interface requirement.

10.4.4 The integrated system test program shall be reviewed and the test results evaluated by a verification team with a good knowledge of the system specification.

10.4.5 Equipment used for system verification shall be calibrated appropriately.

10.4.6 The verification software tools used shall comply with the requirements of Clause 15 addressing verification tools.

10.4.7 The verification of the integrated system shall demonstrate that all system components have appropriate performance (e.g. processing units and communication devices).

10.5 Fault resolution procedures

10.5.1 The requirements of IEC 61513, 6.3.2.4 (Fault resolution procedures) shall apply.

10.5.2 The fault resolution procedures shall ensure that any required modification to the HPD fulfils the requirements of Clause 12.

10.6 HPD aspects of the integrated system test report

10.6.1 The requirements of 6.4.5.2 of IEC 61513 shall apply.

10.6.2 The test results shall be retained in a form that makes them verifiable by persons not directly engaged in the verification plan or in the actual performance of the tests.

11 HPD aspects of system validation

11.1 General

HPDs are typically validated within the system validation phase. System validation is covered by IEC 61513. This Standard provides additional requirements to validate the performance (functional, temporal and electrical) of HPDs.

a) Testing shall be performed to validate the system and the HPD in accordance with the class 1 systems requirements.

b) Validation tests shall be performed on the system in its final assembly configuration including the final version of the HPD.

11.2 HPD aspects of the system validation plan

11.2.1 The system validation shall be conducted in accordance with a formal system validation plan.

11.2.2 The plan shall identify static and dynamic test cases.

11.2.3 The system validation plan shall be developed and the result of the validation evaluated by individuals who did not participate in the design and implementation.

11.3 System validation

11.3.1 The system shall be exercised by static and dynamic input signals simulating normal operation, anticipated operational occurrences and accident conditions requiring action.

11.3.2 Each category A function of the system shall be exercised by a set of tests confirming each required output signal in a single or combined manner.

11.3.3 The tests shall:

a) cover all the functions of the HPD requirement specification, in all modes (6.2);

b) cover all ranges of signals and computed parameters as far as practical;

c) cover the voting, other single or combined logics in comprehensive manner;

d) be made for all trip or protective signals in the final assembly configuration;

e) cover the required response to specified failures;

f) cover all other functions which have an impact on reactor safety.

11.3.4 In addition, the values of input signals, the expected output signals and the acceptance criteria shall be stated in the system validation plan.

11.3.5 Equipment used for validation shall be calibrated and configured (hardware and software parameters) appropriately.

11.4 HPD aspects of the system validation report

11.4.1 The system validation report shall document the results related to the HPD included in the system.

11.4.2 The report shall identify the hardware, the software when applicable, the system configuration used, the tool configurations used and the test equipment used (including its calibration and the simulation models) in compliance with IEC 61513, 6.4.6.2.b.

11.4.3 This report shall also identify any discrepancies found during the test.

11.4.4 This report shall summarise the results of the system validation.

11.4.5 This report shall assess the system compliance with all requirements.

11.4.6 The results shall be retained in a form that makes them verifiable by persons not directly engaged in the validation.

11.4.7 Simulations of the plant and its systems used for the validation shall be documented.

11.5 Fault resolution procedures

The requirements of 10.5 shall also apply to the aspects of system validation related to the HPD.

12 Modification

12.1 Modification of the requirements, design or implementation

12.1.1 The modification process and documentation shall comply with the requirements of IEC 61513 (6.2.8 and 6.4.7), of IEC 60987:2007 (Clause 12) and of IEC 60880:2006 (Clause 11).

12.1.2 All impacted documents shall be verified according to the requirements of Clause 9, by individuals who are not engaged in the design or implementation of the modification.

12.2 Modification of the micro-electronic technology

The supplier may update the micro-electronic technology (e.g. new version of a blank FPGA to increase the speed or to reduce the die size). Even if the new part is claimed to be "compatible" this does not imply that any given design will perform identically on both devices.

12.2.1 The acceptance process (Clause 7) shall be performed again, followed if necessary by all impacted phases of the life-cycle depending on the differences found.

12.2.2 The relevant verification activities shall be performed again and duly documented to ensure that all functional, electrical, and timing requirements are met.

12.2.3 Even if the new and old parts have the same logic configuration and are pin-to-pin compatible, the need to re-generate the programming files (e.g. because of variations in timings or voltages of the programming pulses) shall be assessed and documented.

13 HPD production

13.1 General

The scope of this Standard excludes the design and manufacture of the pre-developed micro-electronic resources (e.g. a blank FPGA) used as inputs by the development process of the HPD. “Production” in this Standard designates the final steps which deliver the integrated circuit ready for use in the I&C system.

13.2 Production tests

13.2.1 Tests shall verify the functions of the HPD as well as its temporal performance (such as frequency, rise and fall times, propagation time, etc.) and its electrical characteristics (such as power consumption, capacitances, etc.).

13.2.2 It should be demonstrated that the tests performed by the supplier of the integrated circuit fulfil the needs (see 13.2.1). The I&C producer does not need to repeat the tests performed by the supplier of the integrated circuit nor to know the corresponding test vectors.

13.2.3 If 13.2.2 is not fulfilled, then additional tests (with documented inputs, expected outputs and acceptance criteria) shall be performed by the I&C producer to cover the needs (see 13.2.1).

13.2.4 Production tests performed at board level (after assembly of the HPD onto the printed circuit board – e.g. by soldering) shall verify that the interface of the part is operational (such as “I/O pin stuck at” fault test, global functional test).

13.2.5 Each produced part shall pass the production tests or shall be rejected.

13.2.6 The test results shall be stored together with identification information such as batch number in order to support the diagnosis of potential process problems.

13.3 Programming files and programming activities

13.3.1 The programming files shall include error detection codes, and the programming equipment shall verify them.

13.3.2 For each produced part:

a) the configuration after programming shall be verified and

b) relevant traceability information (such as batch number, programming log file, characteristics of programmable switches before and after programming) shall be stored.

13.3.3 All procedures and requirements given by the integrated circuit supplier shall be fulfilled (e.g. to prevent electrostatic discharge).

13.3.4 Only tools guaranteed and supported by the integrated circuit supplier shall be used.

14 HPD aspects of installation, commissioning and operation

a) The process and documentation for installation, commissioning and operation shall comply with the requirements of IEC 61513 (6.2.7 and 6.3.6), of IEC 60987:2007 (Clause 10 and Clause 13) and of IEC 60880:2006 (Clause 12).

b) According to IEC 60671 I&C systems and equipment performing category A functions are periodically tested to demonstrate proper function. To achieve the required test coverage for the HPD, appropriate test techniques shall be used to increase the testability, e. g. boundary scan.

15 Software tools for the development of HPDs

15.1 General

Clause 14 of IEC 60880:2006 shall apply to the software tools used for HPD development, with the exception of 14.3.4.3, 14.3.4.4 and 14.3.4.5.

NOTE 1 Tool vendor’s technical evaluation (not limited to quality assurance) is an acceptable method to fulfil the requirement 14.2.2 of IEC 60880:2006, provided that the corresponding documentation is available.

NOTE 2 The “reliability” attribute of software tools addressed in Clause 14 of IEC 60880:2006 means here “trustworthiness” or “correctness”.

NOTE 3 ISO/IEC 9126 is superseded by ISO/IEC 25000.

NOTE 4 Libraries integrated in tools may be evaluated in the context of the tool evaluation.

NOTE 5 The verification of tool outputs addressed by 14.3.2.4 of IEC 60880:2006 may be performed by different ways, e.g. simulation with a simulator diverse from the synthesizing toolset.

15.2 Additional requirements for design, implementation and simulation tools

15.2.1 Software tools shall give access to the parameters that control the logic synthesis and the implementation (e.g. through settings).

15.2.2 Software tools should not add structures not directly traceable to HDL source statements (e.g. gate duplication to match timing requirement) without warning.

15.2.3 The designers shall have previous knowledge of the software tools, in particular they shall know how they perform on the structures and constructs used in the project.

15.2.4 If a software tool requires command line arguments these shall be in a script file (placed under configuration management) to avoid manual invocation errors.

NOTE 1 This is useful not only for the consistency; it also helps in assessing the origin of a fault, which may lie in the source code, in the tool or in the tool parameters. It may also be necessary in the assessment of the potential for CCF due to design and implementation tools.

15.2.5 When moving to a new version of a software tool that is responsible for a transformation of design information (e.g. logic synthesis or place and route), all affected simulation, analysis and verification activities shall be performed again.

NOTE 2 It can be justified by documented analysis that a given modification of a tool cannot affect the abovementioned activities, e.g. correction of some inconsistent behaviour in the tool graphical user interface.

NOTE 3 Activities which have been completed before the tool change do not need to be repeated.

16 Design segmentation or partitioning

16.1 Background

It is possible on some HPD devices to design and implement circuits that are allocated using physically different areas of the integrated circuit, have minimal or no interconnections together, and use no common hardware resources. Some HPDs support such areas, sometimes called “lakes”, with unused/unusable space between them. Some of the advantages of design segmentation or partitioning may include the implementation of auxiliary or support functions (this is not to be a replacement for redundant channels/trains in a design at system level).

16.2 Auxiliary or support functions

16.2.1 General

In general, auxiliary or support functions implemented on a HPD, even if not performing Category A functions, have the potential to interfere with category A functions of that HPD. Thus, unless it can be shown that the requirements of 16.2.2 are met, auxiliary or support functions shall be developed, implemented and verified according to the requirements of this Standard (i.e., as Category A functions).

16.2.2 Partitioning of auxiliary or support functions of category other than A

This Standard recognizes that it may be possible with specific design measures and partitioning of the HPD to ensure that auxiliary or support functions are independent of those of Category A and cannot inappropriately interfere with them. In such cases, provided the following requirements are met, auxiliary or support functions may be implemented on a class 1 HPD without the same rigour as for Category A functions:

a) it shall be demonstrated by design, implementation, assessment and systematic verification that the operation or failure of such auxiliary or support functions cannot interfere directly or indirectly with any Category A function, whether the cause of the failure is internal or external to the HPD (e.g. induced by the power supplies, a short circuit on a connected line, etc.),

b) this demonstration shall address all potential causes of interference e.g. functional, electrical, electromagnetic, thermal, etc.,

c) in particular the areas of the integrated circuit used to implement such auxiliary or support functions shall be physically different from those used to implement the Category A functions,

d) in case of modification of the HPD, it shall be demonstrated that the requirements of 16.2.2 are still fulfilled,

e) the interface between circuits implementing Category A functions and auxiliary or support functions shall be simple and fully verifiable,

f) data received by Category A functions from auxiliary or support functions shall be limited to static parameter values (e.g. calibration constants, set-points),

g) Category A functions shall not have any time dependence on receipt of data from auxiliary or support functions,

h) appropriate safety measures (e.g. safe communications protocols) shall be implemented for any communication between Category A functions and auxiliary or support functions such that all data transfer errors will be detected and a suitable safe response taken, or correct receipt of data is acknowledged.

17 Defences against HPD Common Cause Failure

17.1 Background

Systematic faults may be introduced in any design and implementation process due to human error; and therefore such faults may be introduced during HPD design and implementation (either in the developed part or in an included pre-existing design). HPDs could therefore potentially be affected by latent systematic faults which could, under some triggering event, lead to the CCF of multiple instantiations of a HPD design.

The potential for CCF across multiple systems is in the scope of higher level SC 45A Standards, in particular IEC 61513 and IEC 62340. The potential for CCF due to multiple instantiations of a HPD design in a given system is addressed by this Standard. As explained in Clause 1, “Scope and object”, this Standard defines development and verification processes and requirements which minimise the potential for HPDs to have systematic faults and therefore –as such faults can cause CCF-, also minimise the potential for CCF due to HPDs.

Additional requirements to address protection against systematic faults which may lead to CCF due to HPDs are provided in the following subclause.

17.2 Requirements

17.2.1 Aspects of the HPD development processes which may lead to CCF of multiple instantiations of the HPD design (and which are not already addressed by clauses within this Standard) shall comply as applicable with the relevant requirements of IEC 60880:2006, 13.1 (by replacing “software” with “HPD”).

NOTE 1 These aspects are typically related to the development of HDL programs.

17.2.2 An analysis according to the relevant requirements of IEC 60880:2006, 13.3.1, 13.3.2, 13.3.3, 13.3.4, 13.3.7 and 13.3.8 shall be performed as applicable to address aspects of the HPD development processes which may lead to CCF of multiple instantiations of the HPD design (and which are not already addressed by clauses within this Standard).

NOTE 2 Some requirements of these subclauses address CCF across systems at I&C architecture level, although they would be better located in the high-level Standard dedicated to CCF, IEC 62340. In order to keep the structure of SC 45A Standard series, it is suggested to move these requirements to IEC 62340 at the next maintenance cycle.

Annex A
(informative)

Documentation

This annex identifies typical documentation for each of the main clauses of this Standard. The contents may be organized into a set of documents different from those suggested in this annex, provided that the sections are clearly identified.

A.1 Project

a) project management plan

b) quality assurance plan

c) configuration management plan

A.2 HPD requirement specification

a) requirement specification

b) requirements analysis report

c) review report

A.3 Acceptance of blank integrated circuits, native blocks and PDBs

a) component requirement specification

b) user documentation for safety

c) other documentation of the component, including any information such as specification, design, test, operating experience

d) analysis report

e) document containing the rules of use

f) acceptance review report

A.4 HPD design and Implementation

a) design specification including:

   1) description of: breakdown into main modules, defensive design choices, identification of the micro-electronic technology, tools, native blocks and PDBs

   2) description of detailed design including:

      - RTL description
      - organizational choices (modules, sub-modules, interfaces, protocols, etc.)
      - preliminary electrical characteristics and timings

   3) description of implementation including:

      - gate-level description ("netlist"), technology specific description for production, back-annotations
      - implementation of modules, critical signals and power distribution, tool options, auxiliary files used for implementation such as "constraints files"
      - post-route analyses report, STA report
      - testability analysis, test vectors for periodic testing
      - electrical characteristics and detailed timings

b) review reports

A.5 HPD verification

a) verification plan

b) document containing: description of test-bench, coverage criteria, test cases

c) document containing the analysis and justification of coverage criteria

d) report including: test results and analysis (RTL level, post-synthesis, post-route), analysis of the fulfilment of the rules of use

A.6 HPD aspects of system integration

a) integration plan including: integration strategy and procedures, configuration management interface, test cases

b) specific aspects of integrated system test report, including identification of components and tools, test results and analysis, faults found and resolution

c) integration review report

A.7 HPD aspects of system validation

a) validation plan including test cases

b) report including: identification of components and tools, test results, test analysis, faults found and resolution

A.8 Modification

IEC 60880, Annex F gives the typical documentation list regarding the modification process:

a) anomaly report

b) modification request

c) modification report

d) modification control history

In addition, the documents related to the development phases affected by the modification have to be updated.

A.9 HPD production

a) document containing the production tests

b) document containing the results of production tests, the part identification information and the part programming information

A.10 Software tools for the development of HPDs

a) tool selection report (analysis of tool support, evaluation, acceptance, limits of applicability)

b) document describing the strategy for modification, upgrade or replacement

Annex B
(informative)

Development of HPDs

The development activities addressed by this Standard are based on Hardware Description Languages and design tools running on workstations, according to a flow whose broad outline is presented here to ease the understanding of the corresponding clauses of this Standard.

B.1 Optional capture of requirements at Electronic System Level

Capture of requirements is sometimes done by means of a high level description of the system to which the HPD belongs: this description includes the other hardware and software components. Each component is represented by a behavioural model, and these models exchange information through communication channels to simulate the intended system.

This description level is called "Electronic System Level", or ESL, and uses system description languages such as SystemC or System Verilog.

This description is typically executed (simulation) with functional test cases to estimate the relevance of different system architectures, select the best one, and finally set-up the requirements of each component including the HPD in terms of behaviour and interface.

B.2 Design

Starting from the requirements, this activity initially aims at defining the main design principles, such as the partition in pre-developed or bespoke modules, the organization of the self-supervision and the identification of the micro-electronic technology (including its Native Blocks) and PDBs that could be used.

Then an RTL (Register Transfer Level) description is built and tested by simulation. HDLs such as VHDL or Verilog are used. This is mostly not dependant on the micro-electronic technology that will be used.

This high level description is a synchronous parallel model of the HPD, describing its behaviour by means of signals transformed by combinatorial functions and sequentially transferred between registers triggered by one or more clocks.

The RTL description has structural aspects, showing the logical relations between modules which can be designed specifically or taken from libraries. It also has behavioural aspects, making it possible to describe the function of a module by means of algorithmic descriptions. This description is carried out by means of a HDL (Hardware Description Language), typically VHDL (IEEE 1076) or Verilog (IEEE 1364).

The RTL description needs to be synthesizable, which means that it can be translated automatically into a set of interconnected electronic gates. To achieve this property, the designer uses only a subset of the HDL language, while the full language may be used for example to create simulation environments.

In parallel to the design, it is of use to develop a "test-bench" with the same language: the RTL description of the HPD is included in a broader HDL program, which sends it input sequences and reads its outputs in order to test it by simulation. The test-bench may use non-synthesizable language features to ease the design of the tests (e.g.: access to files, printing, explicit time management). The test-bench is then used to check the RTL description, and can be associated with various tools for test generation and coverage measurement.

Static analysis tools are being introduced to provide complementary verification approach. They typically make it possible to prove whether some expected properties hold or not on an HDL description. Examples of static analyses are: checking of properties, assertion based verification, checking of equivalence between different design levels (e.g. RTL and netlist), or Static Timing Analysis.

B.3 Implementation

Starting from the RTL description, an electronic description is produced that allows the actual implementation in the selected electronic technology. The main stages are the logic synthesis and the place and route.

The different families of components such as FPGAs, standard cells, and so on provide different pre-characterizations of the physical behaviour of the final product. Thus, while the activities described hereinafter are intrinsically necessary, they may or may not be handled automatically by the associated tools. The following description gives an overview of these activities for a design based on standard cells.

The logic synthesis transforms the RTL description into a network of logic cells of the micro-electronic technology, called "netlist". Depending on this micro-electronic technology, these cells may be only elementary gates (such as AND, OR) or may include larger functions (such as counters).

Although tools similar to software compilers are used to perform the synthesis, the designer directs the process by providing information on the expected performance (such as clock frequency, delay between two signals, power consumption) and on how critical signals such as clocks are to be handled. This information is typically stored in "constraint files" which can be very large. Their elaboration can thus be difficult, and an error or omission may result in generating a circuit suffering from subtle non-reproducible faults, almost impossible to detect by simulation. The verification of the constraint files is thus an essential activity.

The place and route stage defines the physical location of the cells on the silicon die, and inter-connects them taking into account the technological constraints (existence and capacity of predefined routing channels) as well as the application constraints (such as maximum propagation delay between two given nodes).

As the number of gates increases, more and more of them are inter-connected. So, more and more inter-connections have to be routed across the die. Additionally the requirements for speed usually impose to keep some paths short. This last constraint may lead to modifying the placement of some gates, which in turn reacts on the whole routing scheme. Finding the "best" solution is a very hard problem (in the sense of computability), so only approximations may be found by the tools, which need to use advanced and evolutionary algorithms.

The description after place and route is produced in a format which depends on the micro-electronic technology. As the physical layout is known at this stage, the propagation times may be refined by taking into account the resistance and capacitance of each path. This information is typically used to back-annotate again the description, in order to simulate it in the test-bench with realistic propagation times for cells and wiring.

Moreover, the supplier of the micro-electronic technology provides the propagation times of the cells included in its library, using formats such as VHDL-VITAL (IEEE 1076.4). This timing information is included as "back-annotation" of the netlist description, and is taken into account in the “post implementation” simulation.

In addition to the verification by "post implementation simulation”, tools for static analysis make it possible to check the propagation times (STA: Static Timing Analysis) or the equivalence between different description levels.

B.4 Types of specific integrated circuits

B.4.1 General

The evolving technology offers many variants of specific integrated circuits, so this Standard provides requirements based on principles and not on specific details of each variant.

This clause provides an overview of the main available variants (note: their names are not always used consistently in the industry).

From a theoretical point of view, any computable function can be implemented with only one type of well chosen elementary gate such as “NAND” (“*A nand B*” is “*not (A and B)*”). Therefore, the range of functions which can be implemented within a given circuit depends essentially on its size (number of gates) and on its internal connectivity which allows a more or less efficient use of the gates.

B.4.2 PAL (Programmable Array Logic)

PALs are low-size devices typically organized in OR/AND array in order to implement logic equations having the form of sum of products such as: *output = (A and B and not C) or (not B and not C) or (D).*

PALs are made specific by configuring connections, typically by blowing fuses or in some cases by configuring reprogrammable switches.

The AND structure is programmable, i.e. the product expression before programming is: (*A and not A and B and not B and C and not C, etc.*), where each term corresponds to one configurable connection. According to the functional requirement, the unneeded terms are removed to produce e.g. (*A and not C*).

The OR structure is fixed: the inputs of the “OR” are a fixed number of such programmable products, e.g. *(A and not C) or (A and not B) or (D).*

Low-level languages such as PALASM are typically used to configure PALs: the designer inputs the logic equations to be implemented and the tool translates them into a map of connections. No behavioural description such as in VHDL or Verilog is possible with such languages.

PALs typically provide a few inputs and outputs (e.g. 10 inputs, 8 outputs) and they are equivalent to a few hundreds gates at most. Due to this limited size they are not in the scope of this Standard.

B.4.3 PLD, CPLD (Programmable Logic Device, Complex PLD)

PLDs and CPLDs are essentially large arrays of PALs interconnected together, but new families may offer additional features.

Like PALs, they are based on sum of products with fixed structure, thus the signal routing from input to output is fixed and propagation delay times are quite constant. Of course, when additional features such as feedback paths or specialized logic are offered this property may be lost.

The size of CPLDs reaches the equivalent of tens of thousands gates.

B.4.4 FPGA

FPGAs are organized as a large number of programmable logic blocks including provisions for combinatorial logic and storage. These blocks are interconnected by a hierarchy of

programmable interconnects, and programmable input/output pads are provided (direction, impedance, voltage, and memorization are typically programmable). Specific paths are usually provided for critical signals such as clocks. FPGAs may in addition include specialized logic blocks such as memory, processor core, standardized interfaces, etc.

The gate equivalence of FPGAs is not really relevant because their complex and different structures make it difficult to predict how many blocks are needed for a given function. Some FPGAs include hundreds of thousands of programmable blocks, hundreds of inputs/outputs, and are made of billions of transistors.

FPGAs may retain their function (“configuration”) by using means such as:

a) Static RAM (the configuration is volatile, copied at start-up from an external memory),
b) Flash memory (the configuration is stored in non-volatile but reprogrammable internal memory elements),
c) Anti-fuse (the configuration is permanent; such devices are “One Time Programmable”).

The susceptibility of the configuration to SEU (Single Event Upset) and neutron/alpha radiation is high for Static RAM, low for Flash, and very low for anti-fuse parts.

B.4.5 Gate array, or pre-diffused integrated circuit

The integrated circuit supplier prepares in advance standard integrated circuits in which all transistors are already made but are not interconnected. The specific function to be implemented is synthesized into a specific interconnection of the transistors.

This approach involves non-recurring costs associated with the production of the specific masks for the metal layers (interconnection), but may offer a lower part cost compared to FPGAs because no silicon is used to implement the programmable circuitry. However, this technology seems to be increasingly replaced by FPGAs.

B.4.6 Standard cells

The supplier offers a micro-electronic technology and designs with it a range of standard cells such as elementary combinatorial gates, flip-flops, adders, counters, etc. These cells have known characteristics such as area, input current, capacitance and propagation delay. They are designed in such a way that they have the same height and different width, so they can be placed on the integrated circuit in rows in order to ease routing and power supplying.

The functional and physical characteristics of the cells are described in the technology library, which is provided to the I&C designer. This library is used during logic synthesis (see Clause B.3) which transforms the RTL description into a netlist of these cells, which are then placed on the integrated circuit and interconnected. After completion of functional and technology related verifications, the masks needed to produce the integrated circuits are fabricated and production may begin.

This approach involves higher non-recurring costs compared to gate-arrays because all masks are specific, but offers a lower part cost because the size of the integrated circuit is exactly what is needed. The availability of different cells for each type, optimizing different aspects such as speed, area, or power consumption, allows a better optimization of each area of the design, still under control of the I&C designer with only HDL related tools.

B.4.7 Full custom ASIC, or raw ASIC

This technology involves a specific design of all aspects of the integrated circuit, down to the transistor level, with specific tools. This implies very high non-recurring costs which need large volumes to be economically justified. These circuits are not in the scope of this Standard.

Bibliography

IEC 60780, *Nuclear power plants – Electrical equipment of the safety system – Qualification*

IEC 61226, *Nuclear power plants – Instrumentation and control important to safety – Classification of instrumentation and control functions*

IEC 62342, *Nuclear power plants – Instrumentation and control systems important to safety – Management of ageing*

ISO 9001, *Quality management systems – Requirements*

ISO/IEC 25000, *Software engineering – Software product Quality Requirements and Evaluation (SQuaRE)*

_____

This page deliberately left blank

British Standards Institution (BSI)

BSI is the national body responsible for preparing British Standards and other standards-related publications, information and services.

BSI is incorporated by Royal Charter. British Standards and other standardization products are published by BSI Standards Limited.

About us

We bring together business, industry, government, consumers, innovators and others to shape their combined experience and expertise into standards -based solutions.

The knowledge embodied in our standards has been carefully assembled in a dependable format and refined through our open consultation process. Organizations of all sizes and across all sectors choose standards to help them achieve their goals.

Information on standards

We can provide you with the knowledge that your organization needs to succeed. Find out more about British Standards by visiting our website at bsigroup.com/standards or contacting our Customer Services team or Knowledge Centre.

Buying standards

You can buy and download PDF versions of BSI publications, including British and adopted European and international standards, through our website at bsigroup.com/shop, where hard copies can also be purchased.

If you need international and foreign standards from other Standards Development Organizations, hard copies can be ordered from our Customer Services team.

Subscriptions

Our range of subscription services are designed to make using standards easier for you. For further information on our subscription products go to bsigroup.com/subscriptions.

With **British Standards Online (BSOL)** you'll have instant access to over 55,000 British and adopted European and international standards from your desktop. It's available 24/7 and is refreshed daily so you'll always be up to date.

You can keep in touch with standards developments and receive substantial discounts on the purchase price of standards, both in single copy and subscription format, by becoming a **BSI Subscribing Member**.

PLUS is an updating service exclusive to BSI Subscribing Members. You will automatically receive the latest hard copy of your standards when they're revised or replaced.

To find out more about becoming a BSI Subscribing Member and the benefits of membership, please visit bsigroup.com/shop.

With a **Multi-User Network Licence (MUNL)** you are able to host standards publications on your intranet. Licences can cover as few or as many users as you wish. With updates supplied as soon as they're available, you can be sure your documentation is current. For further information, email bsmusales@bsigroup.com.

BSI Group Headquarters

389 Chiswick High Road London W4 4AL UK

Revisions

Our British Standards and other publications are updated by amendment or revision.

We continually improve the quality of our products and services to benefit your business. If you find an inaccuracy or ambiguity within a British Standard or other BSI publication please inform the Knowledge Centre.

Copyright

Useful Contacts:

Customer Services
Tel: +44 845 086 9001
Email (orders): orders@bsigroup.com
Email (enquiries): cservices@bsigroup.com

Subscriptions
Tel: +44 845 086 9001
Email: subscriptions@bsigroup.com

Knowledge Centre
Tel: +44 20 8996 7004
Email: knowledgecentre@bsigroup.com

Copyright & Licensing
Tel: +44 20 8996 7070
Email: copyright@bsigroup.com

bsi.

...making excellence a habit.™