2026-06-16

Rumfartsteknik – Konstruktion af ASIC-kredse, FPGA-kredse og IP-kerner

Space engineering – ASIC, FPGA and IP Core engineering

DANSK STANDARD
Danish Standards Association

Göteborg Plads 1
DK-2150 Nordhavn

Tel: +45 39 96 61 01
dansk.standard@ds.dk
www.ds.dk

DS/EN 16603-20-40:2023

København
DS projekt: M372586
ICS: 49.140

Første del af denne publikations betegnelse er:
DS/EN, hvilket betyder, at det er en europæisk standard, der har status som dansk standard.

Denne publikations overensstemmelse er:
IDT med: EN 16603-20-40:2023

I tilfælde af redaktionelle fejl i DS-publikationen kan der skrives til:
editorial-mistakes@ds.dk

ADVARSEL: DS-publikationer revideres over tid. Derudover kan sådanne publikationer ændres ved rettelsesblade og/eller tillæg. Der kan også udgives rettelsesblade, der udelukkende angår oversættelsen af en publikation. Det er derfor vigtigt at sikre sig, at man benytter en gældende udgave, medmindre fx lovgivning kræver andet. Den enkelte publikations status fremgår af https://webshop.ds.dk/. Her kan man desuden tilmelde sig en gratis notifikationsservice og følge en udgivet DS-publikations udvikling ved at klikke på "Følg".

En oversigt over forskellige DS-publikationstyper og -betegnelser findes her:
https://www.ds.dk/publikationstyper

In case of editorial errors in the DS publication, please write to:
editorial-mistakes@ds.dk

NOTICE: DS publications are revised over time. In addition, such publications may be altered by corrigenda and/or amendments. Also, corrigenda may be published solely in relation to the translation of a publication. It is therefore important to ensure that the edition used is valid, unless, for example, legislation requires otherwise. The status of each publication can be found at https://webshop.ds.dk/. Here you can also sign up for a free notification service and follow the progress of a published DS publication by clicking on "Follow".

An overview of different DS publication types and designations can be found here:
https://www.ds.dk/publikationstyper

EUROPEAN STANDARD EN 16603-20-40

EUROPÄISCHE NORM

December 2023

ICS 49.140

English version

Space engineering - ASIC, FPGA and IP Core engineering

Ingénierie spatiale - Ingénierie des ASIC, FPGA et noyaux de PI

Raumfahrttechnik - Entwicklung von ASICs, FPGAs und IP-Kernen

This European Standard was approved by CEN on 3 December 2023.

CEN and CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration. Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CEN and CENELEC member.

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

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

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

Table of contents

Figures

Tables

European Foreword

This document (EN 16603-20-40:2023) has been prepared by Technical Committee CEN-CENELEC/TC 5 “Space”, the secretariat of which is held by DIN.

This standard (EN 16603-20-40:2023) originates from ECSS-E-ST-20-40C.

This European Standard shall be given the status of a national standard, either by publication of an identical text or by endorsement, at the latest by June 2024, and conflicting national standards shall be withdrawn at the latest by June 2024.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CEN [and/or CENELEC] shall not be held responsible for identifying any or all such patent rights.

This document has been prepared under a standardization request given to CEN by the European Commission and the European Free Trade Association.

This document has been developed to cover specifically space systems and has therefore precedence over any EN covering the same scope but with a wider domain of applicability (e.g. aerospace).

According to the CEN-CENELEC Internal Regulations, the national standards organizations of the following countries are bound to implement this European Standard: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.

Introduction

Developing custom designed monolithic integrated circuits such as ASICs or FPGAs, and developing IP Cores, as off-the-shelf Building Blocks for these complex ICs, make certain engineering and technical management activities crucial to the success of these developments.

ECSS-E-ST-20-40 was written in parallel and in co-ordination with the writing ECSS-Q-ST-60-03, by the same ECSS Working Group. These two new and complementary standards cover respectively the engineering and the product assurance requirements to be applied when developing ASICs, FPGAs and IP Cores, and these two new standards together supersede ECSS-Q-ST-60-02C.

The DEVICE qualification status is assessed based on the requirements from both ECSS-E-ST-20-40 and ECSS-Q-ST-60-03. In order for a DEVICE to be qualified and accepted according to ECSS standards, the DEVICE development reviews specified in ECSS-E-ST-20-40 have to be declared successful by the customer engineering and PA responsible persons and project management who monitored the DEVICE development.

1 Scope

This standard specifies a comprehensive set of engineering requirements for the successful development of digital, analogue and mixed analogue-digital signal custom designed integrated circuits, such as application specific integrated circuits (ASICs), field programmable gate arrays (FPGAs) and Intellectual Property Cores (IP Cores), from now on referred to with the single and generic term DEVICEs.

Microelectronics systems created by more than one DEVICE die but that are interconnected and packaged together as a single DEVICE are not considered single monolithic DEVICEs. However ECSS-ST-20-40 is to be applied to (a) the development of each individual monolithic die, (b) also for their integration onto a multi-die single DEVICE considering those dice as IP Cores.

This standard may be tailored for the specific characteristic and constraints of a space project in conformance with ECSS-S-ST-00. A pre-tailoring based on the actual DEVICE type and criticality category of the DEVICE is addressed in clause 5.1.2.
This standard does not cover requirements for the selection, control, procurement or usage of DEVICEs for space projects nor DEVICE ESCC qualification requirements, as those requirements are covered by ECSS-Q-ST-60C EEE components standard and the ESCC generic specification No. 9000 respectively. Nevertheless, this standard contemplates the possibility for the DEVICE to undergo ESCC qualification after the DEVICE customer acceptance as an ECSS qualified DEVICE, and thus a DEVICE ESCC Detail Specification and DEVICE Radiation Test Plan and Report are optional expected outputs.

2 Normative references

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

EN reference	Reference in text	Title
EN 16601-00-01	ECSS-S-ST-00-01	*ECSS system – Glossary of terms*
EN 16602-30	ECSS-Q-ST-30	*Space product assurance – Dependability*
EN 16602-40	ECSS-Q-ST-40	*Space product assurance – Safety*
-	ECSS-Q-ST-60-03	*Space product assurance – ASIC, FPGA and IP Core product assurance*

Terms, definitions and abbreviated terms

3.1 Terms from other standards

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

1. acceptance;
2. component;
3. customer;
4. engineering model;
5. flight model;
6. informative;
7. maintenance;
8. model;
9. normative;
10. performance;
11. project;
12. requirement;
13. risk;
14. supplier.

NOTE The old term *firmware* is defined in ECSS-S-ST-00-01C, and it is not used in the context of ECSS-E-ST-20--40 because with the emergence of new technologies it is now ambiguous and unnecessary. It is important not to confuse terms like *FPGA*, *FPGA programming file* or *FPGA programming bit stream* with *firmware*.

3.2 Terms specific to the present standard

3.2.1 Application Specific Integrated Circuit

full custom or semi custom designed monolithic integrated circuit

NOTE ASICs can be digital, analogue or a mixed function.

3.2.2 block diagram

abstract graphical presentation of interconnected named boxes or blocks representing an architectural or functional drawing

reusable IC design element that implements a self-standing function or group of functions for which ownership rights exist and that has been developed in the context of a specific IC project or technology, without the intention to be shared with third parties for its reuse in other IC projects

NOTE For example, an HDL model such as synthesizable VHDL code, or gate-level netlist, or an analogue function.

3.2.4 cell

specific circuit function including digital or analogue basic blocks

3.2.5 cell library

collection of all mutually compatible cells which conforms to a set of common constraints and standardized interfaces designed and characterized for a specified technology

3.2.6 code

string of words, numbers, letters and symbols that is used to model a DEVICE or its verification and validation environment

NOTE For example, Hardware Description Languages like VHDL, Verilog or SystemC are used to code DEVICE behavioral or synthesisable models, and code in other languages like C, Python or Tool Command Languages (Tcl) can be used in the DEVICE verification and validation.

3.2.7 data sheet

detailed functional, operational and parametric description of a DEVICE

NOTE A data sheet can include, for instance, a block diagram, truth table, pin and signal description, environmental, electrical and performance parameters, tolerances, timing information, and package description.

3.2.8 design for test

technique used to allow a complex integrated circuit to be tested with respect to potential manufacturing faults or to accelerate otherwise too slow validation tests

NOTE 1 For example, any dedicated circuits aimed to provide better observability or commandability of internal nodes of the DEVICE not accessible through primary inputs and outputs.

NOTE 2 Other examples of DFT are test busses, boundary scan as in JTAG, see IEEE 1149.1-2013, built-in self-test, and test modes for functional tests performed at DEVICE Validation, Qualification and Acceptance Phase.

design changes that occur in any single phase or between two consecutive phases as defined in the DEVICE Development Plan, before the final DEVICE is released

3.2.10 DEVICE

integrated circuit or an IP Core

NOTE 1 A DEVICE can be a digital, analogue or mixed-signal ASIC, a programmed FPGA, a blank FPGA, a microprocessor, and a model of an IC function that is conceived for reuse as an IP Core.

NOTE 2 A DEVICE can also be a group of dice or chiplets interconnected and integrated inside a single package, such as a system-in-package or a multi-chip-module.

3.2.11 DEVICE Database

set of all digital files that are needed for the development of a DEVICE

NOTE 1 Examples of files integrating this database are behavioral and HDL models of the DEVICE, layout description files, models of the DEVICE system environment used to verify by simulation the DEVICE functionality, configuration files and SW programs used for the automation of the verification and validation of the DEVICE, input and output files used and generated by the different CAD tools used, for example files describing the resources, area, timing and power constraints, stimuli and expected output values files, or FPGA bit stream binary files.

NOTE 2 This database of files is incrementally updated throughout the DEVICE development phases, and all necessary elements that enable support, maintenance and a new development of the same or a modified version of the DEVICE can be found in the DEVICE database at the end of the DEVICE Validation, Qualification and Acceptance Phase.

3.2.12 DEVICE development flow

selection and sequence of engineering methods and tools applied during the definition, design, verification, implementation and validation of the DEVICE

3.2.13 DEVICE model

textual or graphical representation of a DEVICE, or a part of it, which defines one or several DEVICE characteristics

NOTE 1 For example, digital or analogue functional behavior, timing performance, power

physical internal structures, input and output external interfaces, environmental effects due to temperature, radiation or aging.

NOTE 2 DEVICE models can be created graphically as graphical design drawings or with textual Hardware Description Languages like VHDL, Verilog, SystemC or EDIF.

NOTE 3 DEVICE models can implement several levels of abstraction such as behavioral, Register-Transfer-Level, gate-level netlist or transistor-level, and accuracy such as untimed, loosely timed or approximately timed.

NOTE 4 DEVICE models can be generated manually, assisted by CAD specialized IC design tools or a mix of both.

3.2.14 DEVICE technology

totality of all elements needed for the design, physical implementation and test of either ASIC or FPGA components

NOTE 1 For example, design tools and their description, cell libraries, procedures, design rules, manufacturing process, programming tools, test equipment.

NOTE 2 Sometimes the terms ASIC technology or FPGA technology are used when referring to the technology of only that type of DEVICE.

3.2.15 fault coverage

measure expressed as a percentage of the proportion of actually detectable faults versus all possible faults in a digital circuit, for a given set of test patterns and with respect to a specific fault model

3.2.16 Field Programmable Gate Array

standard semiconductor device that becomes customized when programmed by the user with specific software and hardware tools

3.2.17 FPGA Programming Test

test performed to validate the successful programming of an FPGA

NOTE For example, to detect inaccurate programming procedures, incorrect programming files, poor calibration of FPGA programmer, or material defects in the FPGA.

abstracted, scaled layout drawing of the die, outlining the form, size and position of the major functional blocks and the pads including power and ground lines, clock distribution and interconnect channels

3.2.19 HDL model

textual description of an integrated circuit based on a hardware description language suitable for the behavioural or structural description and simulation

3.2.20 IP Core

integrated circuit design element that implements a self-standing function or group of functions for which ownership rights exist and is developed for reuse and released with comprehensive verification, validation and documentation

NOTE 1 IP core can be acquired by a customer, for a given price and under an owner-defined license agreement specifying the customer's acquired rights.

NOTE 2 IP core can be supplied as an HDL model, as a synthesizable VHDL code or gate-level netlist, and with the essential complementary documentation that allows the customer to successfully integrate and use it in a system, for example User's Manual and verification files. From this point of view, it can be seen as a semi-finished product.

NOTE 3 IP Cores can be analogue functions provided by DEVICE technology providers as macrocells.

NOTE 4 In contrast with Building Blocks, IP Cores have gone through comprehensive verification, validation and documentation intended for reuse of third parties.

NOTE 5 IP Cores sometimes are referred as hard IPs if they are already placed and routed for one specific IC technology. For example, a macrocell already pre-diffused inside an FPGA or an already layouted function that is included in an ASIC, or a netlist that cannot be modified and is treated as a black-box.

NOTE 6 An IP Core can be composed by combination of different IP Cores.

3.2.21 macrocell

module that contains complex functions in a given technology's cell library built up out of hard-wired primitive cells

NOTE Macrocells can be provided as a library component for ASIC design, for example RAM

such as DSP blocks or microprocessor cores existing inside FPGAs.

3.2.22 netlist

formatted list of cells, basic circuits, and their interconnections

NOTE 1 The term pre-layout netlist is used for DEVICE netlists that do not contain yet the DEVICE layout information. For example, detailed timing behavior of the cells, timing impact of the interconnections.

NOTE 2 For FPGA, pre-layout netlist means usually the netlist obtained through synthesis tools.

3.2.23 phase

set of interrelated DEVICE development activities

NOTE 1 A DEVICE generic development flow usually consists of seven phases as depicted in Figure 5-1. See Annex J for more development flow examples.

NOTE 2 The concept of phase is equivalent to the concept of process defined in ECSS-S-ST-00-01.

3.2.24 processing unit

function which is defined to execute software

NOTE 1 The term covers hardware functions such as general purpose processing cores, and more specialized Graphical Processing Unit (GPU), Vision Processing Unit (VPU), Tensor Processing Unit (TPU), Neural Processing Unit (NPU), Physics Processing Unit (PPU), Digital Signal Processor (DSP), or Image Signal Processor (ISP).

NOTE 2 In the context of SW engineering, it also covers software processing units such as interpreters, emulators and virtual machines.

3.2.25 production test

test performed on the manufactured DEVICEs to detect functional problems resulting from faults or random material defects during wafer manufacturing, die assembly and packaging

NOTE 1 Production Tests are normally performed with test vectors generated by the DEVICE designer and Automatic Test Equipment (ATE) by the DEVICE manufacturer.

NOTE 2 Production Tests include for example stuck-at faults scan tests, timing delay faults, I/O parametric tests for digital ASICs.

reference voltage or current tests, AD/DA converters tests or tests to catch manufacturing defects affecting any analogue function.

NOTE 4 Production Tests are not DEVICE Validation tests, which are tests performed on the final DEVICE, ASIC or FPGA, to validate that all requirements in DRS are met. Validation tests are normally performed by the DEVICE development team (the supplier), and not by the DEVICE manufacturer (the foundry), and they include functional, environmental and mechanical tests.

3.2.26 prototype

fabricated ASIC or programmed FPGA used to verify preliminary implementations of the DEVICE against the DEVICE Requirements Specification

3.2.27 redesign

design changes which were not planned in the DEVICE Development Plan and which involve reopening an already closed phase

NOTE These design changes can be motivated by unexpected changes in the DRS, wrong interpretation of the DRS or design mistakes.

3.2.28 software

set of instructions and data executed on a processing unit

NOTE 1 A processing unit can be hardware, for example a processor chip, or software, for example a virtual machine or an interpreter.

NOTE 2 Some processing units only require data, for example configuration of state machines or configuration data of a neural network.

NOTE 3 Files using Hardware Description Languages (VHDL, Verilog, SystemC) used to model ASICs or bit stream files used to program FPGAs are not software.

3.2.29 stimuli

input data set for simulation or test to show a specific functionality or performance of a DEVICE

3.2.30 synthesis tool

tool that automatically converts a high-level textual representation of an integrated circuit, usually coded in Hardware Description Language at register–transfer-level, into an optimized gate-level netlist representation of the integrated circuit

functional, electrical, environmental, mechanical, test or quality requirement of the system wherein the DEVICE is used

3.2.32 Validation

<CONTEXT: ASIC, FPGA, IP Core engineering>

process which demonstrates through the provision of objective evidence that the DEVICE, in its final manufactured (ASIC) or programmed (FPGA) form, is able to perform and behave as expected in the intended system, operational environment and application scenarios

NOTE 1 DEVICE verification is normally performed before DEVICE validation.

NOTE 2 Typical validation methods are hardware tests of the DEVICE integrated in its intended or a representative system, operational environment and application scenarios.

NOTE 3 This term is defined in the present standard with a different meaning than in ECSS-S-ST-00-01. The term with the meaning defined herein is applicable only to the present standard.

3.2.33 Verification

<CONTEXT: ASIC, FPGA, IP Core engineering>

process which demonstrates through the provision of objective evidence that the DEVICE is free of design mistakes and is designed according to its DEVICE Requirements Specification, target technology and manufacturability requirements, as well as any agreed deviations and waivers

NOTE 1 A waiver can arise as an output of the verification process.

NOTE 2 DEVICE verification is normally performed before DEVICE validation.

NOTE 3 Verification can be accomplished by one or more of the following methods: analysis (including similarity), simulations of DEVICE models, test of preliminary prototypes, review of design and inspection.

NOTE 4 This term is defined in the present standard with a different meaning than in ECSS-S-ST-00-01. The term with the meaning defined herein is applicable only to the present standard.

3.3 Abbreviated terms

For the purpose of this Standard, the abbreviated terms from ECSS-S-ST-00-01 and the following apply:

Abbreviation	Meaning
AC	alternating current
ADR	DEVICE Architecture Definition Phase Review
ASIC	Application Specific Integrated Circuit
CAD	Computer Aided Design
DADR	DEVICE Architecture Definition Report
DC	direct current
DDP	DEVICE Development Plan
DDR	DEVICE Detailed Design Phase Review
DDS	DEVICE Data Sheet
DFT	design for test
DPR	DEVICE Definition Phase Review
DRC	design rule check
DRD	document requirements definition
DRS	DEVICE Requirements Specification
DSMP	DEVICE Support and Maintenance Plan
DVaP	DEVICE Validation Plan
DVeP	DEVICE Verification Plan
DVR	DEVICE Design and Verification Phase Review
EDA	electronic design automation
EDIF	electronic design interchange format
EM	engineering model
ERC	electrical rule check
ESCC	European Space Components Coordination
ESD	electrostatic discharge
FM	flight model
FPGA	field-programmable gate array
DFRAR	DEVICE Feasibility and Risk Assessment Report
GDSII	graphic design system two (industry standard graphics entry database file format for ASICs)

HDL	hardware description language
HW	hardware
I/O	input-output
IC	integrated circuit
IEEE	Institute of Electrical and Electronics Engineers
IP	intellectual property
IPR	DEVICE Implementation Phase Review
JTAG	joint test action group
LPR	DEVICE Layout Phase Review
LUT	look up table
MoM	minutes of meeting
P&R	place and route
RTL	register-transfer level
SDF	standard delay format
SEU	single event upset
SPEF	standard parasitic exchange format
SW	software
VQAR	DEVICE Validation, Qualification and Acceptance Phase Review
VCD	Verification Control Document
VHDL	Very High Speed Integrated Circuit Hardware Description Language

3.4 Conventions

3.4.1 Names of DEVICE development phases and reviews

The names of DEVICE development phases and reviews specified in ECSS-E-ST-20-40 do not follow the naming conventions specified in ECSS-M-ST-10 in an effort to use widely established ASIC, FPGA and IP Core engineering terminology which is self-explanatory and commonly used by IC engineers. In order to facilitate collaboration between IC and PA engineers ECSS-E-ST-20-40

phases and key names in Annex L.

3.4.2 Companies involved in the DEVICE development

In addition to *supplier* and *customer* terms used in ECSS standards, the following other terms for other companies that can be part of the DEVICE development are used in ECSS-E-ST-20-40:

a. FPGA technology provider;

b. ASIC technology provider;

c. ASIC manufacturer;

d. Company developing DEVICE layout.

The term *technology vendor*, even if widely used in the ASIC, FPGA and IP Core community, is avoided in ECSS-E-ST-20-40 in favour of *technology provider* for the sake of simplicity.

3.4.3 Types of DEVICEs and requirements tailoring tag notation

Find below the basic notation for the different types of DEVICEs in the scope of ECSS-E-ST-20-40 marked in blue color:

D-ASIC	applicable to fully digital ASICs, or the digital part of mixed-signal ASICs
A-ASIC	applicable to fully analogue ASICs, or the analogue part of mixed-signal ASICs
FPGA	applicable to FPGAs, which can also be mixed-signal DEVICEs
IP	applicable to digital or analogue IP Cores

Every requirement in ECSS-E-ST-20-40 contains a "tailoring tag" at the end of the last sentence of the requirement that indicates to which type of DEVICE the requirement is applicable. The tailoring tag can contain one, two or three of the following elements, always in the same order:

[D-ASIC, A-ASIC, FPGA, IP]

If applicable to all four DEVICE types indistinctly, the tag is a simpler

[ALL]

Whenever one of more DEVICEs types shall not be concerned by the requirement, the expression

"--"

is found in the relevant position. For example:

- **[D-ASIC, A-ASIC, FPGA, --]** requirement applicable to all DEVICE types, except for IP Cores;

the analogue part of mixed-signal ASICs;

- **[D-ASIC, --, FPGA, IP]** requirement applicable to all DEVICE types, except for analogue ASICs.

In the context of ECSS-E-ST-20-40, the development of general purpose complex ICs such as microprocessors, microcontrollers, blank FPGAs or other (re)programmable or extensively configurable DEVICEs commonly known as DSP, GPU, VPU are treated as ASIC developments.

3.5 Nomenclature

The following nomenclature applies throughout this document:

a. The word “shall” is used in this Standard to express requirements. All the requirements are expressed with the word “shall”.

b. The word “should” is used in this Standard to express recommendations. All the recommendations are expressed with the word “should”.

NOTE It is expected that, during tailoring, recommendations in this document are either converted into requirements or tailored out.

c. The words “may” and “need not” are used in this Standard to express positive and negative permissions, respectively. All the positive permissions are expressed with the word “may”. All the negative permissions are expressed with the words “need not”.

d. The word “can” is used in this Standard to express capabilities or possibilities, and therefore, if not accompanied by one of the previous words, it implies descriptive text.

NOTE In ECSS “may” and “can” have completely different meanings: “may” is normative (permission), and “can” is descriptive.

e. The present and past tenses are used in this Standard to express statements of fact, and therefore they imply descriptive text.