

DIN EN 16603-20-40

ICS 49.140

**Raumfahrttechnik –
ASIC, FPGA und IP-Kern Entwicklung;
Englische Fassung EN 16603-20-40:2023**

Space engineering –
ASIC, FPGA and IP Core engineering;
English version EN 16603-20-40:2023

Ingénierie spatiale –
Ingénierie des ASIC, FPGA et noyaux de PI;
Version anglaise EN 16603-20-40:2023

Gesamtumfang 151 Seiten

DIN-Normenausschuss Luft- und Raumfahrt (NL)

Nationales Vorwort

Dieses Dokument (EN 16603-20-40:2023) wurde vom Technischen Komitee CEN/CLC/JTC 5 „Raumfahrt“ erarbeitet, dessen Sekretariat von DIN (Deutschland) gehalten wird.

Das zuständige deutsche Normungsgremium ist der Arbeitsausschuss NA 131-06-01 AA „Raumfahrt“ im DIN-Normenausschuss Luft- und Raumfahrt (NL).

Dieses Dokument enthält unter Berücksichtigung des Präsidialbeschlusses 1/2004 nur die englische Originalfassung von EN 16603-20-40:2023.

Dieses Dokument wurde speziell zur Behandlung von Raumfahrtsystemen erarbeitet und hat daher Vorrang vor jeder anderen EN-Norm, die denselben Anwendungsbereich abdeckt, aber über einen größeren Geltungsbereich (z. B. Luft- und Raumfahrt) verfügt.

Aktuelle Informationen zu diesem Dokument können über die Internetseiten von DIN (www.din.de) durch eine Suche nach der Dokumentennummer aufgerufen werden.

Nationaler Anhang NA (informativ)

Begriffe

Reihenfolge und Inhalt der folgenden Begriffe sind identisch mit denen im Abschnitt Begriffe der englischen Fassung.

Für die Anwendung dieses Dokuments gelten die folgenden Begriffe.

ISO und IEC stellen terminologische Datenbanken für die Verwendung in der Normung unter den folgenden Adressen bereit:

- ISO Online Browsing Platform: verfügbar unter <https://www.iso.org/obp>
- IEC Electropedia: verfügbar unter <https://www.electropedia.org/>

3.1 Begriffe aus anderen Normen

- a) Für die Anwendung dieser Norm gelten die Begriffe nach ECSS-S-ST-00-01, besonders die folgenden Begriffe:
- 1) Akzeptanz
 - 2) Komponente
 - 3) Kunde
 - 4) technisches Modell
 - 5) Flugmodell
 - 6) informativ
 - 7) Instandhaltung
 - 8) Bauform
 - 9) normativ
 - 10) Leistung
 - 11) Projekt
 - 12) Anforderung
 - 13) Risiko
 - 14) Lieferant

ANMERKUNG Der alte Begriff *Firmware* ist in ECSS-S-ST-00-01C definiert und wird im Zusammenhang mit ECSS-E-ST-20-40 nicht mehr angewendet, da er mit dem Aufkommen neuer Technologien mehrdeutig und unnötig geworden ist. Es ist wichtig, Begriffe wie *FPGA*, *FPGA-Programmierdatei* oder *FPGA-Programmier-Bitstrom* nicht mit *Firmware* zu verwechseln.

3.2 Für diese Norm spezifische Begriffe

3.2.1

anwendungsspezifischer integrierter Schaltkreis

vollständig oder teilweise nach Kundenwünschen gestalteter monolithischer integrierter Schaltkreis

ANMERKUNG ASICs können digitale, analoge oder gemischte Funktionen haben.

3.2.2

Blockschaltbild

abstrakte grafische Darstellung von miteinander verbundenen und mit Namen versehenen Rechtecken oder Blöcken, die eine architektonische oder funktionsbezogene Zeichnung darstellen

3.2.3

Baustein

wiederverwendbares IC-Entwurfselement, das eine eigenständige Funktion oder eine Gruppe von Funktionen umsetzt, für die Eigentumsrechte bestehen und die im Kontext eines bestimmten IC-Projekts oder einer bestimmten Technologie entwickelt wurde, ohne die Absicht zu verfolgen, sie mit Dritten zu deren Wiederverwendung in anderen IC-Projekten zu teilen

ANMERKUNG Beispielsweise ein HDL-Modell wie ein synthetisierbarer VHDL-Code oder eine Gatternetzliste oder eine analoge Funktion.

3.2.4

Zelle

spezielle Schaltkreisfunktion einschließlich digitaler oder analoger Basisblöcke

3.2.5

Zellenbibliothek

Zusammenstellung von miteinander kompatiblen Zellen, die einer Reihe von allgemeinen Einschränkungen und genormten Schnittstellen entspricht, die für eine festgelegte Technologie ausgelegt und beschrieben sind

3.2.6

Code

Zeichenfolge aus Wörtern, Zahlen, Buchstaben und Symbolen, die zur Modellierung eines GERÄTS oder seiner Verifizierung und Validierungsumgebung angewendet wird

ANMERKUNG Zum Beispiel werden Hardwarebeschreibungssprachen wie VHDL, Verilog oder SystemC angewendet, um Verhaltens- oder Synthesemodelle des GERÄTS zu codieren; für die Verifizierung und Validierung eines GERÄTS kann ein Code in anderen Sprachen wie C, Python oder Tool Command Languages (Tcl) angewendet werden.

3.2.7

Datenblatt

detaillierte Beschreibung eines GERÄTS in Bezug auf seine Funktionen, Betrieb und Parameter

ANMERKUNG Ein Datenblatt kann z.B. Blockschaltbilder, Wahrheitstabellen, Anschlussstift- und Signalbeschreibungen, umweltbezogene, elektrische und leistungsbezogene Parameter, zulässige Abweichungen, Terminierungsinformationen und Verpackungsbeschreibungen enthalten.

3.2.8

prüffreundliches Entwickeln

Technik, die es ermöglicht, einen komplexen integrierten Schaltkreis auf potenzielle Herstellungsfehler zu prüfen oder ansonsten zu langsame Validierungsprüfungen zu beschleunigen

ANMERKUNG 1 Zum Beispiel jegliche spezielle Schaltkreise, die eine bessere Beobachtbarkeit oder Steuerbarkeit interner Knotenpunkte des GERÄTS ermöglichen, die nicht über die primären Ein- und Ausgänge zugänglich sind.

ANMERKUNG 2 Andere Beispiele für prüffreundliches Entwickeln sind Prüfbusse, Boundary Scan (Grenzpfadabtastung) wie bei JTAG, siehe IEEE 1149.1-2013, Built-in self-test und Prüfmodi für Funktionstests, die in der Validierungs-, Qualifikations- und Abnahmephase des GERÄTS durchgeführt werden.

3.2.9

Entwurfswiederholungsschritt

Entwurfsänderungen, die innerhalb einer oder mehreren Phasen oder zwischen zwei aufeinander folgenden Phasen auftreten, wie im GERÄTE-Entwicklungsplan festgelegt, bevor das endgültige GERÄT freigegeben wird

3.2.10

GERÄT

integrierter Schaltkreis oder IP-Kern

ANMERKUNG 1 Ein GERÄT kann ein digitaler, analoger oder Mixed-Signal-ASIC, ein programmiertes FPGA, ein leeres FPGA, ein Mikroprozessor und ein Modell einer IC-Funktion sein, das für die Wiederverwendung als IP-Kern konzipiert ist.

ANMERKUNG 2 Ein GERÄT kann auch eine Gruppe von Dice oder Chiplets sein, die miteinander verbunden und in ein einziges Gehäuse integriert sind, wie z. B. ein System-in-Package oder ein Multi-Chip-Modul.

3.2.11

GERÄTE-Datenbank

Satz aller digitalen Dateien, die für die Entwicklung eines GERÄTS erforderlich sind

ANMERKUNG 1 Beispiele für Dateien, die in diese Datenbank eingefügt werden, sind Verhaltens- und HDL-Modelle des GERÄTS, Layout-Beschreibungsdateien, Modelle der GERÄTE-Systemumgebung, die zur Verifizierung der GERÄTE-Funktionalität durch Simulation verwendet werden, Konfigurationsdateien und SW-Programme, die für die Automatisierung der Verifizierung und Validierung des GERÄTS angewendet werden, Eingabe- und Ausgabedateien, die von den verschiedenen verwendeten CAD-Tools verwendet und erzeugt werden, z. B. Dateien, die die Ressourcen-, Flächen-, Zeit- und Leistungsbeschränkungen beschreiben, Reize und Dateien mit erwarteten Ausgabewerten oder FPGA-Bitstrom-Binärdateien.

ANMERKUNG 2 Diese Dateidatenbank wird während der GERÄTE-Entwicklungsphasen schrittweise aktualisiert; alle erforderlichen Elemente, die eine Unterstützung, Instandhaltung und Neuentwicklung der gleichen oder einer abgeänderten Version des GERÄTS ermöglichen, sind am Ende der GERÄTE-Datenbank der GERÄTE-Validierung-, Qualifizierungs- und Akzeptanzphase enthalten.

3.2.12

Entwicklungsfluss des GERÄTS

Auswahl und Abfolge der technischen Verfahren und Werkzeuge, die während der Definition, dem Entwurf, der Verifizierung, der Umsetzung und der Validierung des GERÄTS angewendet werden

3.2.13

GERÄTEmodell

textliche oder grafische Darstellung eines GERÄTS oder eines Teils davon, die eine oder mehrere GERÄTEigenschaften festlegt

ANMERKUNG 1 Zum Beispiel digitales oder analoges Funktionsverhalten, Zeitverhalten, Stromverbrauch, Größe und Verbindungen von physikalischen internen Strukturen, externen Eingangs- und Ausgangsschnittstellen, Umwelteinflüsse durch Temperatur, Strahlung oder Alterung.

ANMERKUNG 2 GERÄTEmodelle können graphisch als graphische Entwurfszeichnungen oder mit textuellen Hardwarebeschreibungssprachen wie VHDL, Verilog, SystemC oder EDIF erstellt werden.

ANMERKUNG 3 GERÄTEmodelle können verschiedene Abstraktionsebenen umsetzen, z. B. Verhaltensebene, Registertransferebene, Gatternetzliste oder Transistorebene, und eine Genauigkeit aufweisen, z. B. ohne Zeitangabe, grobe Zeitangabe oder ungefähre Zeitangabe.

ANMERKUNG 4 GERÄTEmodelle können manuell, mit Hilfe von CAD-spezialisierten IC-Entwurfswerkzeugen oder einer Mischung aus beidem erstellt werden.

3.2.14

GERÄTEtechnik

Gesamtheit aller Elemente, die für den Entwurf, die physikalische Umsetzung und die Prüfung von ASIC- oder FPGA-Komponenten erforderlich sind

ANMERKUNG 1 Zum Beispiel Entwurfswerkzeuge und deren Beschreibung, Zellenbibliotheken, Verfahren, Entwurfsregeln, Fertigungsprozess, Programmierwerkzeuge, Prüfgeräte.

ANMERKUNG 2 Manchmal werden die Begriffe ASIC-Technologie oder FPGA-Technologie angewendet, wenn sie sich nur auf die Technologie des jeweiligen GERÄTEtyps beziehen.

3.2.15

Fehlzustandserkennungsgrad erkennbarer Fehleranteil

als Prozentsatz angegebenes Maß des Anteils der Fehlzustände eines digitalen Schaltkreises an dessen möglichen Fehlerzuständen, die im Falle eines gegebenen Satzes von Prüfmustern und in Bezug auf ein bestimmtes Fehlerzustandsmodell tatsächlich erkannt werden können

3.2.16

feldprogrammierbares Gatterfeld

Standard-Halbleitergerät, das im Falle seiner Programmierung durch den Anwender mit Hilfe spezifischen Hard- und Softwaretools zu einem kundenspezifischen Gerät wird

3.2.17

FPGA-Programmierprüfung

Prüfung, die durchgeführt wird, um die erfolgreiche Programmierung eines FPGA zu validieren

ANMERKUNG Zum Beispiel, um ungenaue Programmierverfahren, falsche Programmierdateien, eine schlechte Kalibrierung des FPGA-Programmierers oder Materialfehler im FPGA zu erkennen.

3.2.18

Bauplan

abstrahierte, maßstäbliche Zeichnung der Schaltungsanordnung des Chips, die die Form, Größe und Position der wesentlichen Funktionsblöcke und die Kontaktflächen einschließlich der Stromleitungen und Grundlinien, Zeittaktverteiler und Verbindungskanäle zeigt

3.2.19

HDL-Modell

textuelle Beschreibung eines integrierten Schaltkreises, die auf einer Hardwarebeschreibungssprache basiert und sich zur Beschreibung von Verhalten oder Struktur und Simulation eignet

3.2.20

IP-Kern

schutz-/urheberrechtlich geschützter Kern

Entwurfselement eines integrierten Schaltkreises, das eine eigenständige eigentumsrechtlich geschützte Funktion oder Funktionsgruppe implementiert und das für die Wiederverwendung entwickelt und mit umfassender Verifizierung, Validierung und Dokumentation freigegeben wird

ANMERKUNG 1 Ein IP-Kern kann von einem Kunden gegen Zahlung eines angegebenen Preises und unter einer vom Eigentümer festgelegten Lizenzvereinbarung, in der die vom Kunden erworbenen Rechte festgelegt sind, erworben werden.

ANMERKUNG 2 Ein IP-Kern kann als HDL-Modell, als ein synthesesfähiger VHDL-Code oder Gatternetzliste und mit der unerlässlichen Komplementärdokumentation geliefert werden, die es dem Kunden ermöglicht, diesen Kern erfolg-

reich in einem System zu implementieren und zu nutzen, z. B. Benutzerhandbuch und Verifizierungsdateien. Unter diesem Gesichtspunkt kann es als Halbfertigprodukt betrachtet werden.

ANMERKUNG 3 IP-Kerne können analoge Funktionen sein, die von GERÄTE-Technikvertreibern als Makrozellen bereitgestellt werden.

ANMERKUNG 4 Im Gegensatz zu Bausteinen haben IP-Kerne, die für die Wiederverwendung durch Dritte vorgesehen sind, eine umfassende Verifizierung, Validierung und Dokumentation durchlaufen.

ANMERKUNG 5 IP-Kerne werden manchmal als Hard IPs bezeichnet, wenn sie bereits für eine bestimmte IC-Technik geortet und geroutet sind. Zum Beispiel eine Makrozelle, die bereits in einem FPGA vorverdrahtet ist, oder eine bereits gelayoutete Funktion, die in einem ASIC enthalten ist, oder eine Netzliste, die nicht verändert werden kann und als Black-box behandelt wird.

ANMERKUNG 6 Ein IP-Kern kann durch Kombination verschiedener IP-Kerne zusammengestellt werden.

3.2.21

Makrozelle

Modul, das komplexe Funktionen in einer Zellenbibliothek einer bestimmten Technologie enthält, die aus festverdrahteten primitiven Zellen aufgebaut ist

ANMERKUNG Makrozellen können als Bibliothekskomponente für den ASIC-Entwurf zur Verfügung gestellt werden, z. B. RAM-Blöcke, oder sie befinden sich in bereits diffundierten Bausteinen wie DSP-Blöcken oder in FPGA enthaltenen Mikroprozessorkernen.

3.2.22

Netzliste

formatierte Liste von Zellen, Grundsaltungen und deren Verbindungen

ANMERKUNG 1 Der Begriff Pre-Layout-Netzliste wird für GERÄTE-Netzlisten verwendet, die noch keine GERÄTE-Layout-Angaben enthalten. Zum Beispiel das genaue Timingverhalten der Zellen oder die zeitlichen Auswirkungen der Verbindungen.

ANMERKUNG 2 Bei FPGAs ist mit Pre-Layout-Netzliste üblicherweise die Netzliste gemeint, die mit Synthesewerkzeugen erhalten wurde.

3.2.23

Phase

Reihe von miteinander verbundenen Entwicklungsaktivitäten für GERÄTE

ANMERKUNG 1 Ein allgemeiner Entwicklungsablauf für GERÄTE besteht üblicherweise aus sieben Phasen, wie in Bild 5-1 dargestellt. Siehe Anhang J für weitere Beispiele für Entwicklungsabläufe.

ANMERKUNG 2 Das Konzept der Phase entspricht dem Konzept des Prozesses, das in ECSS-S-ST-00-01 festgelegt ist.

3.2.24

Verarbeitungseinheit

Funktion, die festgelegt ist, um Software auszuführen

ANMERKUNG 1 Der Begriff umfasst Hardware-Funktionen wie Allzweck-Verarbeitungskerne und einen spezielleren Grafikprozessor (GPU), Bildverarbeitungseinheiten (VPU), Tensor-Verarbeitungseinheit (TPU), neuromorpher Prozessor (NPU), Physikbeschleuniger (PPU), digitale Signalprozessor (DSP) oder Bildsignalprozessor (ISP).

ANMERKUNG 2 Im Kontext des SW-Engineering umfasst er auch Software-Verarbeitungseinheiten wie Interpreter, Emulatoren und virtuelle Maschinen.

3.2.25

fertigungsbegleitende Prüfung

Prüfung, die an den hergestellten GERÄTEN durchgeführt wird, um funktionale Probleme zu erkennen, die durch Fehler oder zufällige Materialfehler während der Waferherstellung, der Chipmontage und des Verpackens entstehen

ANMERKUNG 1 Fertigungsbegleitende Prüfungen werden normalerweise mit Prüfvektoren durchgeführt, die vom Entwickler des GERÄTS und von den automatischen Prüfgeräten (en: automatic test equipment, ATE) des GERÄTEherstellers erstellt werden.

ANMERKUNG 2 Zu den fertigungsbegleitenden Prüfungen gehören z. B. Haftfehler-Abtastprüfungen, Zeitverzögerungsfehler und parametrische I/O-Prüfungen für digitale ASICs.

ANMERKUNG 3 Zu den fertigungsbegleitenden Prüfungen für analoge ASICs gehören z. B. Referenzspannungs- oder -stromprüfungen, AD/DA-Wandlerprüfungen oder Prüfungen zum Entdecken von Herstellungsfehlern, die jegliche analogen Funktionen betreffen.

ANMERKUNG 4 Fertigungsbegleitende Prüfungen sind keine Validierungsprüfungen für GERÄTE, d. h. Prüfungen, die am fertigen GERÄT, ASIC oder FPGA durchgeführt werden, um zu validieren, ob alle Anforderungen der DRS erfüllt sind. Validierungsprüfungen werden normalerweise vom GERÄTE-Entwicklungsteam (dem Lieferanten) und nicht vom GERÄTEhersteller (der Gießerei) durchgeführt und sie umfassen funktionale, umweltbezogene und mechanische Prüfungen.

3.2.26

Prototyp

gefertigter ASIC oder programmiertes FPGA, der bzw. das zur Verifizierung von vorläufigen Implementierungen des GERÄTS mithilfe der GERÄTE-Anforderungsspezifikation angewendet wird

3.2.27

Neukonstruktion

Entwurfsänderungen, die nicht im GERÄTE-Entwicklungsplan geplant waren und die die Wiederaufnahme einer bereits abgeschlossenen Phase umfassen

ANMERKUNG Diese Entwurfsänderungen können durch unerwartete Änderungen in der DRS, eine falsche Interpretation der DRS oder Entwurfsfehler begründet sein.

3.2.28

Software

Satz von Anleitungen und Daten, die auf einer Verarbeitungseinheit ausgeführt werden

ANMERKUNG 1 Eine Verarbeitungseinheit kann aus Hardware, z. B. einem Prozessorchip, oder aus Software, z. B. einer virtuellen Maschine oder einem Interpreter bestehen.

ANMERKUNG 2 Einige Verarbeitungseinheiten benötigen nur Daten, z. B. die Konfiguration von Zustandsautomaten oder Konfigurationsdaten eines neuronalen Netzes.

ANMERKUNG 3 Dateien mit Hardwarebeschreibungssprachen (VHDL, Verilog, SystemC), die zur Modellierung von ASICs verwendet werden, oder Bitstromdateien zur Programmierung von FPGAs sind keine Software.

3.2.29

Reize

Eingangsdatensatz für Simulations- oder Prüfzwecke, um eine bestimmte Funktionalität oder Leistung eines GERÄTS nachzuweisen

3.2.30**Synthesewerkzeug**

Werkzeug, das eine auf hohem Niveau befindliche textuelle Darstellung eines integrierten Schaltkreises, der üblicherweise in der Hardwarebeschreibungssprache auf Registertransferebene codiert ist, automatisch in eine optimierte Gatternetzlistendarstellung des integrierten Schaltkreises umwandelt

3.2.31**Systemanforderung**

funktionale, elektrische, umweltbezogene, mechanische, Prüf- oder Qualitätsanforderungen des Systems, in dem das GERÄT eingesetzt wird

3.2.32**Validierung**

<KONTEXT: ASIC, FPGA, IP-Kern Entwicklung>

Prozess, der durch die Erbringung eines objektiven Nachweises zeigt, dass das GERÄT in seiner endgültigen gefertigten (ASIC) oder programmierten (FPGA) Form in der Lage ist, im beabsichtigten System, in der Betriebsumgebung und in den Anwendungsszenarien wie erwartet zu funktionieren und sich wie erwartet zu verhalten

ANMERKUNG 1 Eine GERÄTE-Verifizierung wird normalerweise vor der GERÄTE-Validierung durchgeführt.

ANMERKUNG 2 Typische Validierungsverfahren sind Hardwareprüfungen des GERÄTS in seinem vorgesehenen oder einem repräsentativen System, seiner Betriebsumgebung und seinen Anwendungsszenarien.

ANMERKUNG 3 Dieser Begriff wird in dieser Norm mit einer anderen Bedeutung als in ECSS-S-ST-00-01 festgelegt. Der Begriff mit der in diesem Dokument festgelegten Bedeutung gilt nur für diese Norm.

3.2.33**Verifizierung**

<KONTEXT: ASIC, FPGA, IP-Kern Entwicklung>

Prozess, der durch die Erbringung eines objektiven Nachweises zeigt, dass das GERÄT frei von Entwurfsfehlern ist und entsprechend der GERÄTE-Anforderungsspezifikation, der Zieltechnologie und den Herstellbarkeitsanforderungen sowie jeglichen vereinbarten Abweichungen und Ausnahmen entwickelt wurde

ANMERKUNG 1 Eine Ausnahme kann als Ergebnis des Verifizierungsprozesses entstehen.

ANMERKUNG 2 Eine GERÄTE-Verifizierung wird normalerweise vor der GERÄTE-Validierung durchgeführt.

ANMERKUNG 3 Die Verifizierung kann mit einem oder mehreren der folgenden Verfahren durchgeführt werden: Analyse (einschließlich Ähnlichkeit), Simulationen von GERÄTE-Modellen, Prüfung von vorläufigen Prototypen, Überprüfung des Entwurfs und Überprüfung.

ANMERKUNG 4 Dieser Begriff wird in dieser Norm mit einer anderen Bedeutung als in ECSS-S-ST-00-01 festgelegt. Der Begriff mit der in diesem Dokument festgelegten Bedeutung gilt nur für diese Norm.

Nationaler Anhang NB (informativ)

Abkürzungen

Für die Anwendung dieser Norm gelten die Abkürzungen nach ECSS-S-ST-00-01 und die folgenden Abkürzungen:

Abkürzung	Bedeutung
AC	Wechselstrom (en: alternating current)
ADR	Überprüfung der Definitionsphase der GERÄTE-Architektur (en: DEVICE architecture definition phase review)
ASIC	anwendungsspezifischer integrierter Schaltkreis (en: application specific integrated circuit)
CAD	rechnergestützte Konstruktion (en: computer aided design)
DADR	Definitionsbericht der GERÄTE-Architektur (en: DEVICE architecture definition report)
DC	Gleichstrom (en: direct current)
DDP	GERÄTE-Definitionsphase oder GERÄTE-Entwicklungsplan (en: DEVICE definition phase or DEVICE development plan)
DDR	ausführliche Überprüfung der Entwurfsphase des GERÄTS (en: DEVICE detailed design phase review)
DDS	GERÄTE-Datenblatt (en: DEVICE Data Sheet)
DFT	prüffreundliches Entwickeln (en: design for test)
DPR	Überprüfung der Definitionsphase des GERÄTS (en: DEVICE definition phase review)
DRC	Prüfung der Designregeln (en: design rule check)
DRD	Definition der Anforderungen an Dokumente (en: document requirements definition)
DRS	GERÄTE-Anforderungsspezifikation (en: DEVICE requirements specification)
DSMP	Plan zur Unterstützung und Instandhaltung des GERÄTS (en: DEVICE support and maintenance plan)
DVaP	GERÄTE-Validierungsplan (en: DEVICE validation plan)
DVeP	GERÄTE-Verifizierungsplan (en: DEVICE verification plan)
DVR	Überprüfung der Entwurfs- und Verifizierungsphase des GERÄTS (en: DEVICE design and verification phase review)
EDA	elektronische Entwurfsautomatisierung (en: electronic design automation)
EDIF	elektronisches Entwurfsaustauschformat (en: electronic design interchange format)
EM	technisches Modell, Engineering-Modell (en: engineering model)
ERC	Prüfung der elektrischen Regeln (en: electrical rule check)
ESCC	European Space Components Coordination
ESD	elektrostatische Entladung (en: electrostatic discharge)
FM	Flugmodell (en: flight model)
FPGA	feldprogrammierbares Gatterfeld (en: field-programmable gate array)
DFRAR	Umsetzbarkeits- und Risikoanalysebericht des GERÄTS (en: DEVICE Feasibility and risk analysis report)

GDSII	Grafikdesignsystem 2 (Industriestandard-Grafikeingabewerkzeug für ASICs) (en: graphic design system two (industry standard graphics entry database file format for ASICs))
HDL	Hardwarebeschreibungssprache (en: hardware description language)
HW	Hardware
I/O	Eingabe-Ausgabe (en: input-output)
IC	integrierter Schaltkreis (en: integrated circuit)
IEEE	Institute of Electrical and Electronics Engineers
IP	Schutz-/Urheberrecht (en: intellectual property)
IPR	Überprüfung der Implementierungsphase des GERÄTS (en: DEVICE implementation phase review)
JTAG	Gemeinsame Prüfkaktionsgruppe (en: joint test action group)
LPR	Überprüfung der GERÄTE-Layoutphase (en: DEVICE layout phase review)
LUT	Nachschlagetabelle (en: look up table)
MoM	Sitzungsprotokoll (en: minutes of meeting)
P&R	Ort und Route (en: place and route)
RTL	Registertransferebene (en: register-transfer level)
SDF	Standard Verzögerungsformat (en: standard delay format)
SEU	Einzelereignisstörung (en: single event upset)
SPEF	Standardformat für den Austausch von Parasiten (en: standard parasitic exchange format)
SW	Software
VQAR	Überprüfung der Validierungs-, Qualifizierungs- und Akzeptanzphase des GERÄTS (en: DEVICE Validation, Qualification and Acceptance Phase Review)
VCD	Dokument zur Kontrolle der Verifizierung (en: Verification Control Document)
VHDL	VHSIC-Hardwarebeschreibungssprache (en: Very High Speed Integrated Circuit Hardware Description Language)

— Leerseite —

English version

Space engineering — ASIC, FPGA and IP Core engineering

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

Raumfahrttechnik —
ASIC, FPGA und IP-Kern Entwicklung

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

European Foreword.....6

Introduction.....7

1 Scope.....8

2 Normative references9

3 Terms, definitions and abbreviated terms.....10

 3.1 Terms from other standards.....10

 3.2 Terms specific to the present standard10

 3.3 Abbreviated terms.....17

 3.4 Conventions.....19

 3.4.1 Names of DEVICE development phases and reviews19

 3.4.2 Companies involved in the DEVICE development20

 3.4.3 Types of DEVICES and requirements tailoring tag notation20

 3.5 Nomenclature21

4 Principles22

 4.1 DEVICE development.....22

 4.2 Verification methods22

5 DEVICE engineering23

 5.1 General requirements23

 5.1.1 Overview23

 5.1.2 Tailoring according to DEVICE type and DEVICE criticality.....23

 5.1.3 DEVICE engineering development flow.....23

 5.1.4 Phase Reviews25

 5.1.5 DEVICE Verification Control Document.....25

 5.2 DEVICE Definition Phase27

 5.2.1 Overview27

 5.2.2 DEVICE Requirements Specification.....27

 5.2.3 DEVICE Development Plan.....27

 5.2.4 Preliminary Verification and Validation Plans28

 5.2.5 Preliminary DEVICE Support and Maintenance Plan28

5.2.6	Feasibility and Risk Assessment	28
5.2.7	DEVICE Definition Phase Review	29
5.3	DEVICE Architecture Definition Phase.....	29
5.3.1	Overview	29
5.3.2	Architecture Definition	29
5.3.3	Updated DEVICE Verification and Validation Plans.....	30
5.3.4	DEVICE Architecture Definition Phase Review.....	30
5.4	DEVICE Design and Verification Phase	30
5.4.1	Overview	30
5.4.2	DEVICE Verification Plan	31
5.4.3	DEVICE Design and Verification	31
5.4.4	DEVICE Database	32
5.4.5	Preliminary DEVICE Data Sheet	33
5.4.6	DEVICE Design and Verification Phase Review.....	33
5.5	DEVICE Detailed Design Phase	34
5.5.1	Overview	34
5.5.2	Netlist Generation	34
5.5.3	Netlist verification	36
5.5.4	DEVICE Data Sheet update	36
5.5.5	DEVICE Database update.....	36
5.5.6	DEVICE Detailed Design Phase Review	37
5.6	DEVICE Layout Phase.....	37
5.6.1	Overview	37
5.6.2	Layout generation	37
5.6.3	Layout verification	39
5.6.4	DEVICE Validation Plan	39
5.6.5	DEVICE Database update.....	39
5.6.6	DEVICE Data Sheet update	39
5.6.7	Preliminary ESCC Detail Specification	39
5.6.8	DEVICE Layout Phase Review	40
5.7	DEVICE Implementation Phase	40
5.7.1	Overview	40
5.7.2	Production and test	41
5.7.3	DEVICE Database update.....	41
5.7.4	DEVICE Validation Plan completion	42
5.7.5	DEVICE Implementation Phase Review	42
5.8	DEVICE Validation, Qualification and Acceptance Phase	42

5.8.1 Overview42

5.8.2 DEVICE validation.....43

5.8.3 DEVICE Support and Maintenance43

5.8.4 Experience Summary Report43

5.8.5 Final versions of application and procurement documents44

5.8.6 DEVICE Validation, Qualification and Acceptance Phase Review44

6 Pre-tailoring according to DEVICE criticality and type46

6.1 DEVICE criticality categories46

6.2 Pre-tailoring Matrix.....49

Annex A (normative) DEVICE Requirements Specification (DRS) - DRD93

Annex B (normative) DEVICE Development Plan (DDP) - DRD.....98

Annex C (normative) DEVICE Verification Plan (DVeP) - DRD.....101

Annex D (normative) DEVICE Validation Plan (DVaP) - DRD106

**Annex E (normative) DEVICE Support and Maintenance Plan (DSMP) -
DRD.....108**

**Annex F (normative) DEVICE Feasibility and Risk Assessment Report
(DFRAR) - DRD.....110**

Annex G (normative) DEVICE Architecture Definition Report (DADR) - DRD...114

Annex H (normative) DEVICE Data Sheet (DDS) - DRD117

Annex I (normative) Experience Summary Report - DRD119

Annex J (informative) Generic Development Flow Variations120

Annex K (informative) DEVICE Development Expected Outputs126

**Annex L (informative) Equivalence of phase and milestone terminology of
ECSS-M-ST-10 and ECSS-E-ST-20-40.....134**

Bibliography.....139

Figures

Figure 5-1: DEVICE development flow (generic case)26

Figure J-1 : Example of DEVICE development flow with intermediate additional reviews ...121

Figure J-2 : Example of DEVICE development flow variation with two DEVICE modules
developed and reviewed in parallel122

Figure J-3 : Example of DEVICE development flow variation where three phases have
been merged124

Figure J-4 : Example of DEVICE development flow where three phases are iterated125

Tables

Table 6-1: DEVICE criticality categories47

Table 6-2: Pre-tailoring Matrix50

Table K-1 : Summary of expected outputs of engineering flow126

Table K-2 : ECSS-E-ST-20-40 and ECSS-Q-ST-60-03 list of expected document
outputs128

Table L-1 : Equivalence of phase and milestone terminology of ECSS-M-ST-10 and
ECSS-E-ST-20-40.....135

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	<i>ECSS system – Glossary of terms</i>
EN 16602-30	ECSS-Q-ST-30	<i>Space product assurance – Dependability</i>
EN 16602-40	ECSS-Q-ST-40	<i>Space product assurance – Safety</i>
-	ECSS-Q-ST-60-03	<i>Space product assurance – ASIC, FPGA and IP Core product assurance</i>

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

3.2.3 Building Block

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.

3.2.9 design iteration

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

consumption, sizes and interconnections of 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.

3.2.18 floorplan

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

blocks, or be present inside pre-diffused devices 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.

NOTE 3 Analogue ASICs Production Tests examples are 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

3.2.31 system requirement

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)

Abbreviation	Meaning
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

provides a comparison between the ECSS-E-ST-20-40 phases and ECSS-M-ST-10 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 or 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;

- **[--, A-ASIC, --, --]** requirement only applicable to analogue ASICs or to 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.

4

Principles

4.1 DEVICE development

The end-to-end DEVICE development involves several phases. The DEVICE supplier derives from the system requirements WHAT are the functional, performance, environmental and physical requirements that the DEVICE has to meet. The detailed process of HOW the DEVICE will be developed, which includes detailed verification and validation plans, is incrementally specified and implemented by the supplier and reviewed with the customer applying the requirements of ECSS-E-ST-20-40.

The supplier derives requirements and plans of actions for the development of the DEVICE within the boundaries of ECSS-E-ST-20-40. The DEVICE requirements are based on the requirements of the system for which it is intended and take into consideration the operational and environmental requirements of the space project or programme where the DEVICE will be used.

Starting with a global development plan, the verification and validation plans are specified and executed incrementally by the supplier as the DEVICE design progresses through several development phases and until fully verified and validated DEVICES are accepted by the customer.

4.2 Verification methods

Verification methods allow to provide objective evidence that the DEVICE or parts of it are designed according to its requirements specifications and being free of design errors. Such methods can be grouped in five major categories:

1. ANALYSIS of documents and DEVICE models, often using specialised IC design CAD tools for the analysis of the DEVICE timing, power consumption, resources occupation, parasitic effects, etc.;
2. SIMULATIONS of DEVICE models using IC design CAD tools to perform behavioural and time-accurate simulations of DEVICE internal and output signals obtained when stimuli are applied at DEVICE inputs or internal DEVICE circuit nodes;
3. TEST of DEVICE hardware prototypes prior to final DEVICE manufacturing or programming;
4. REVIEW of DEVICE models or circuit schematics performed visually or aided by ad-hoc computer programs. This category includes what is known as review-of-design (ROD) in ECSS-E-ST-10-02;
5. INSPECTION of the DEVICE hardware. For example, visual inspection of the prototypes used for verification tests in order to check correct mounting on the board or the use of the correct part.

DEVICE engineering

5.1 General requirements

5.1.1 Overview

The development of a new DEVICE involves going through a flow of several phases which encompass several engineering steps. It is important to agree with the customer what is the final set of requirements in compliance with ECSS-E-ST-20-40 that are applied and thus specify the engineering steps to go through, taking into consideration the type of DEVICE, its criticality category and the specific constraints of the project. In addition, several general requirements apply when defining the DEVICE development flow, when implementing certain recurrent steps in every phase, including the review at the end of each phase which, if successful, implies customer authorisation to start of the next phase.

5.1.2 Tailoring according to DEVICE type and DEVICE criticality

- a. Criticality category of the DEVICE under development shall be discussed and agreed with the customer. [\[ALL\]](#)
- b. Customer and supplier shall specify the final tailoring of the requirements of ECSS-E-ST-20-40 according to the type of DEVICE and according to its criticality category in compliance with clause 6. [\[ALL\]](#)

5.1.3 DEVICE engineering development flow

- a. The DEVICE development phases and milestones shall be in conformance with the generic engineering flow presented in Figure 5-1. [\[ALL\]](#)

NOTE 1 The equivalence of ECSS-M-ST-10 phases and milestone names with respect to ECSS-E-ST-20-40 is presented in Annex L.

NOTE 2 The most complex expected document outputs are specified in DRD annexes, while the content of other expected documents outputs is specified in the main chapters of the standard, and this information is summarized in Table K-2.

- b. If the DEVICE Development Plan includes any variations with respect to the generic flow shown in Figure 5-1, all the outputs expected at every new phase and associated review shall reflect these changes too. [\[ALL\]](#)

NOTE 1 For example, in case of having phase iterations or additional sub-phases and reviews, the

Verification and Validation Plans include what requirements are covered in each new phase.

NOTE 2 Annex J describes the main types of DEVICE development flow variations.

- c. Each planned additional intermediate step, parallel step or not planned design iteration leading to a new DEVICE database release shall undergo its own dedicated review. [\[ALL\]](#)
- d. If the DEVICE development is planned to undergo a first Engineering Model development followed by a Flight Model development, customer and supplier shall agree whether EM and FM are treated as two independent DEVICE developments or as a single DEVICE development where the EM is mainly a prototype verification step for the development of the final FM DEVICE. [\[ALL\]](#)

NOTE 1 Treating EM and FM as independent developments can be adequate whenever their respective implementation technologies are significantly different, and therefore the models of EM and FM DEVICES are significantly different too.

NOTE 2 Treating EM and FM as independent developments can also be adequate if EM DEVICE is planned to be used as a stand-alone product and not just as an intermediate verification step for the FM DEVICE.

- e. All inputs and tools used to reproduce DEVICE development steps in every phase shall follow the documentation and configuration management requirements in compliance with clause 8 of ECSS-Q-ST-60-03. [\[ALL\]](#)

NOTE 1 Examples of such inputs are simulation test patterns, schematics, VHDL source codes and synthesis scripts.

NOTE 2 Examples of DEVICE development steps are netlist generation and netlist verification.

NOTE 3 Tools are any design, verification and validation software and hardware tools used during the DEVICE development.

- f. The supplier shall ensure automatic repeatability of all development steps that make use of CAD tools that support automation in order to facilitate iterations in the flow. [\[ALL\]](#)

NOTE Examples of planned iterations include repetitions of self-checking simulations as the design is maturing. But iterations in the development flow can also be needed due to unexpected changes to the requirements, the technology or any unforeseen modifications to the design.

- g. In order to reopen a phase that was successfully closed already, supplier and customer shall discuss and reach an agreement on it. [\[ALL\]](#)
- h. Every output shall be updated at the end of every phase with any relevant new information gathered during the phase. [\[ALL\]](#)

5.1.4 Phase Reviews

- a. The outputs generated within each phase shall be reviewed by the customer with the support of the supplier. [\[ALL\]](#)
- b. The reviewers shall analyse the preventive measures and contingency plans for all identified open issues and risk items identified in the DEVICE Feasibility and Risk Assessment to define and agree with the customer the risks that are taken for starting the next phase. [\[ALL\]](#)
- c. The reviewers shall check that all expected outputs contain all the relevant information and with a level of detail that avoids ambiguity. [\[ALL\]](#)
- d. Any missing information and open points and their impact on following phases shall be assessed, registered and agreed with the customer in the MoM of the phase review with an indication of the expected time of completion of the open points. [\[ALL\]](#)
- e. The criteria for a successful phase review shall be defined in advance. [\[ALL\]](#)
- f. Phase reviews shall be conducted in accordance with ECSS-M-ST-10 and ECSS-M-ST-10-01. [\[ALL\]](#)

5.1.5 DEVICE Verification Control Document

- a. At DEVICE Definition Phase review the supplier shall provide for customer approval an initial DEVICE VCD, in compliance with ECSS-E-ST-10-02 Annex B VDC DRD, including: [\[ALL\]](#)
 - 1. the DEVICE requirements with the combination of the selected verification and validation methods for the different verification and validation levels at the applicable development phases and the applicable verification stages;
 - 2. the traceability between DEVICE requirements and System requirements in compliance with ECSS-E-ST-10-06 clause 8.2.3.

NOTE 1 Verification stages can be qualification, acceptance, prelaunch, in-orbit (including commissioning) and postlanding, as described in ECSS-E-ST-10-02 clause 4.2.4.

NOTE 2 Verification methods are described in ECSS-E-ST-20-40 clause 4.2.

NOTE 3 Some verification and validation information can be found in ECSS-E-ST-20-40 reports and used to populate parts of the VCD. These reports are Design Verification Report, Netlist Verification Report, Layout Verification Report,

ASIC Production Test Report, FPGA Programming Test Report, DEVICE Validation Report and Radiation Test Report.

NOTE 4 References to relevant sections of the DVerP, DValP, or reports, waivers, RFD, NCR, NRB or customer closeout reports can be added to VCD to demonstrate compliance with the requirements.

NOTE 5 The DEVICE VCD can be included in the upper level VCD, for example, equipment or unit VCD.

- b. The supplier shall provide for customer approval an updated DEVICE VCD with the verification and validation close out status for the DEVICE at each phase review in compliance with ECSS-E-ST-10-02 Annex B VCD DRD. [\[ALL\]](#)

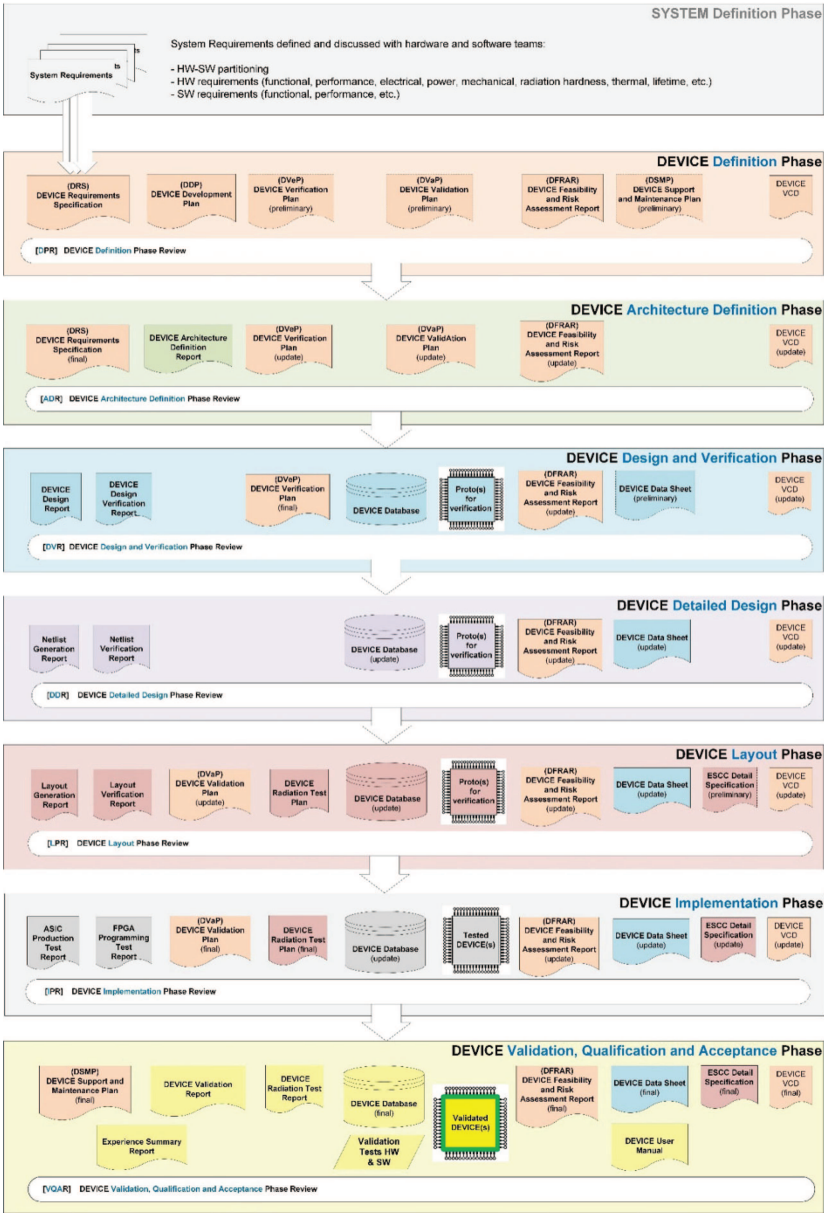


Figure 5-1: DEVICE development flow (generic case)

5.2 DEVICE Definition Phase

5.2.1 Overview

The aim of this initial DEVICE Definition Phase is to establish a DEVICE Development Plan that specifies the development flow, indicating the resources and schedules, a DEVICE Requirements Specification that results from carefully flowing down the requirements of the system(s) where the DEVICE is used and ensuring good traceability between the DEVICE requirements and the System Requirements. In addition, a Feasibility and Risk Assessment of the DEVICE development is an important part of this initial DEVICE Definition Phase and is revised in the following phases in order to minimise risks.

5.2.2 DEVICE Requirements Specification

- a. The supplier shall specify the complete set of DEVICE requirements in the DEVICE Requirements Specification in compliance with DRD from Annex A. [\[ALL\]](#)
- b. The supplier shall ensure that all System Requirements that are relevant to the DEVICE are flowed down to the DRS allowing back traceability. [\[ALL\]](#)
- c. If the DEVICE contains one or more processing unit, the supplier shall flow the HW-SW partitioning requirements in the System Requirements down into the DEVICE Requirements Specification. [\[D-ASIC, --, FPGA, IP\]](#)

NOTE HW-SW partitioning is usually done by the System team, often the customer, in collaboration with HW and SW development teams (see requirement 5.2.2.4.a of ECSS-E-ST-40).

5.2.3 DEVICE Development Plan

- a. The supplier shall provide the customer with a complete description of the DEVICE development global strategy in the DEVICE Development Plan which in compliance with DRD from Annex B. [\[ALL\]](#)
- b. The development methodology of any subcontractors in charge of developing any of the Building Blocks for the DEVICE shall be ascertained by the supplier and agreed with the customer. [\[ALL\]](#)

NOTE For example, whether deviations or tailoring to ECSS-E-ST-20-40 are applied.

- c. Design and verification responsibilities shall be assigned to different people in order to guarantee independent verification. [\[ALL\]](#)

NOTE For example, the FPGA Responsible Engineer can take over the role of the FPGA Designer or of the FPGA Verification Engineer, but not both.

- d. If the DEVICE contains one or more processing units, the supplier shall plan and document how to synchronise the DEVICE development flow

key milestones with the System and SW development flows key milestones in order to manage dependencies between DEVICE, SW and System developments. [\[D-ASIC, --, FPGA, IP\]](#)

- e. If the DEVICE contains both analogue and digital blocks, the supplier shall plan and document how to synchronise the DEVICE development flow key milestones with the analogue and digital development flows key milestones in order to manage dependencies. [\[ALL\]](#)
- f. Modifications to IP Cores shall be justified and agreed between supplier and customer, documented in the DDP, DVeP and DVaP, specified in DRDs of Annex B, Annex C, and Annex D, and updated accordingly whenever these modifications are agreed. [\[ALL\]](#)

5.2.4 Preliminary Verification and Validation Plans

- a. The supplier shall define the general approach to verifying and validating the DEVICE in preliminary Verification and Validation Plans in compliance with DRDs from Annex C and Annex D. [\[ALL\]](#)
- b. The supplier shall include the different types of verification and validation methods and test procedures, and a preliminary matrix for the traceability. [\[ALL\]](#)

NOTE For example, which parts of the DEVICE are verified and validated separately, whether or not preliminary HW prototypes are used for DEVICE verification, specific Design-for-Test and test modes needed to accelerate otherwise too lengthy verification or validation steps, verification and validation approach when processing units are part of the DEVICE.

- c. Modified IP Cores shall be treated as Building Blocks that undergo full verification and validation. [\[ALL\]](#)

5.2.5 Preliminary DEVICE Support and Maintenance Plan

- a. If agreed between supplier and customer a preliminary DEVICE Support and Maintenance Plan shall be produced by the supplier in compliance with DRD from Annex E. [\[ALL\]](#)

5.2.6 Feasibility and Risk Assessment

- a. The supplier shall perform a feasibility and risk assessment of the development of the DEVICE and document it in the DEVICE Feasibility and Risk Assessment Report in compliance with DRD in Annex F. [\[ALL\]](#)

5.2.7 DEVICE Definition Phase Review

- a. The DEVICE Definition Phase shall be concluded by a DEVICE Definition Phase Review in compliance with requirements from clause 5.1.4. [\[ALL\]](#)
- b. The reviewers shall check that the DEVICE development activity as specified in the DDP as in DRD in Annex B is feasible within the limits imposed by the project requirements, resources, schedule and budgetary constraints. [\[ALL\]](#)
- c. The following expected outputs shall be reviewed during DEVICE Definition Phase Review: [\[ALL\]](#)
 1. DEVICE Requirements Specification (DRS) as in DRD in Annex A;
 2. DEVICE Development Plan (DDP) as in DRD in Annex B;
 3. DEVICE Verification Plan (preliminary) as in DRD in Annex C;
 4. DEVICE Validation Plan (preliminary) as in DRD in Annex D;
 5. DEVICE Support and Maintenance Plan (preliminary) as in DRD in Annex E;
 6. DEVICE Feasibility and Risk Assessment Report (DFRAR) as in DRD in Annex F;
 7. DEVICE VCD (preliminary).

5.3 DEVICE Architecture Definition Phase

5.3.1 Overview

The aim of this development phase is to define the architecture of the DEVICE design in terms of its main functional blocks, hierarchies and dependencies of these blocks, their interfaces and how they interconnect. The objective is to facilitate the modular and detailed design of all the blocks and their integration in the following phases.

5.3.2 Architecture Definition

- a. The supplier shall define the DEVICE architecture and document it in the DEVICE Architecture Definition Report in compliance with DRD in Annex G. [\[ALL\]](#)
- b. Any pertinent additions and modifications to the DRS of DRD in Annex A due to the DEVICE architecture definitions shall be included in a final DRS. [\[ALL\]](#)

NOTE This is done in order to guide the DEVICE designers' work in the lower-level DEVICE Design and Verification Phase and later in the DEVICE Detailed Design Phase.

5.3.3 Updated DEVICE Verification and Validation Plans

- a. The supplier shall update and maintain the preliminary DEVICE Verification and Validation Plans in accordance with Annex C and Annex D, defining verification methods and validation test procedures for requirements impacting the DEVICE architecture definition. [\[ALL\]](#)

NOTE For example, test procedures to validate radiation effects mitigation techniques introduced at DEVICE architectural level or HW-SW interfaces and functional dependencies.

5.3.4 DEVICE Architecture Definition Phase Review

- a. The DEVICE Architecture Definition Phase shall be concluded by a DEVICE Architecture Definition Phase Review in compliance with requirements from clause 5.1.4. [\[ALL\]](#)
- b. The reviewers shall check that the selected architectural trade-offs meet the requirements fixed during the DEVICE Definition Phase. [\[ALL\]](#)
- c. The following expected outputs shall be reviewed during DEVICE Architecture Definition Phase Review: [\[ALL\]](#)
 - 1. DEVICE Architecture Definition Report as in DRD in Annex G;
 - 2. DEVICE Requirements Specification (final) as in DRD in Annex A;
 - 3. DEVICE Verification plan (update) (DVeP) as in DRD in Annex C;
 - 4. DEVICE Validation Plan (update) (DVaP) as in DRD in Annex D;
 - 5. DEVICE Feasibility and Risk Assessment Report (update) (DFRAR) as in DRD in Annex F;
 - 6. DEVICE VCD (update).

5.4 DEVICE Design and Verification Phase

5.4.1 Overview

The aim of the DEVICE Design and Verification Phase is to further develop the DEVICE creating the first DEVICE models and their verification environment. These models are part of the DEVICE Database. The DEVICE model creation and its verification is documented in the DEVICE Design Report and the DEVICE Design Verification Report respectively, a preliminary DEVICE Data Sheet is prepared, and the phase concludes with the DEVICE Design and Verification Phase Review securing that everything is ready for the DEVICE Detailed Design Phase.

5.4.2 DEVICE Verification Plan

- a. A DEVICE Verification Plan in compliance with DRD in Annex C shall be completed and ready at DEVICE Design and Verification Phase Review. [\[ALL\]](#)

5.4.3 DEVICE Design and Verification

- a. Tasks specified in requirements 5.4.3b to 5.4.3j shall be performed and documented by the supplier in a DEVICE Design Report. [\[ALL\]](#)
- b. The supplier shall generate the DEVICE models needed as input to the subsequent DEVICE Detailed Design Phase. [\[ALL\]](#)

NOTE 1 For example, synthesizable RTL models for digital circuits, or behavioral models for analogue circuits.

NOTE 2 Simulations of DEVICE behavioral models of critical functions and algorithms can be very useful during the DEVICE Architecture Definition Phase and Feasibility and Risk Assessment, as they can be valuable tools for further verification tasks.

- c. The supplier shall model analogue block interfaces, drivers and loads including their estimated parasitics. [\[--, A-ASIC, --, --\]](#)

NOTE For example, parasitics introduced by the package, bonding or wiring tracks.

- d. The supplier shall perform an analysis of key analogue parameters and their sensitivity to manufacturing and environmental variations in order to consequently protect the DEVICE analogue blocks against those variations. [\[--, A-ASIC, --, --\]](#)

NOTE For example, parameters such as circuit dimensions or electrical parameters which can be affected by process manufacturing and external T, V or I variations.

- e. The supplier shall specify and document the global approach to integration of new and existing Building Blocks and IP Cores up until the entire DEVICE is integrated at top level. [\[ALL\]](#)
- f. The supplier shall generate the necessary DEVICE verification files, verify the DEVICE models following the DEVICE Verification Plan of DRD in Annex C and document results in the first iteration of the DEVICE Design Verification Report. [\[ALL\]](#)

NOTE Verification files examples are HDL simulation testbenches, external components models, and data files defining stimuli and expected outputs.

- g. The supplier shall perform a preliminary floorplan and a preliminary technology mapping to ensure a successful place-and-route and timing

closure in cases where this early assessment can be of added value. [\[D-ASIC, A-ASIC, FPGA, --\]](#)

NOTE Technology mapping for digital circuits can be achieved for example by synthesis tools that transform RTL models into gate-level netlists.

- h. The supplier shall re-assess the feasibility and risks, including trade-offs for any conflicting requirements, update the DFRAR and implement the best design choices accordingly. [\[ALL\]](#)

NOTE For example, power consumption versus speed and performance, pin count versus package size and complexity versus die area, update the DFRAR and implement the best design choices accordingly.

- i. The supplier shall specify the configuration applied to IP Cores used in the DEVICE. [\[ALL\]](#)
- j. The supplier shall specify any modifications done to IP Cores as justified in the DFRAR and agreed between supplier and customer. [\[ALL\]](#)
- k. The radiation hardening approach shall be implemented and verified at this development level. [\[ALL\]](#)

NOTE For example, TMR, safe state machines or error detection and correction at RTL HDL level.

5.4.4 DEVICE Database

- a. The supplier shall create and maintain a DEVICE Database with all the files needed as inputs for the DEVICE Detailed Design Phase and phase reiterations, including items specified in 5.4.4b to 5.4.4d. [\[ALL\]](#)
- b. The DEVICE Database shall include the DEVICE models and other electronic files needed for the complete DEVICE netlist generation. [\[ALL\]](#)

NOTE For example, high-level simulation, behavioral and RTL models and analogue design models.

- c. The DEVICE Database shall include the executable and script files used for verification of the DEVICE models. [\[ALL\]](#)

NOTE For example, testbenches, simulation control or constraint files, external component models, stimuli input files and expected output files, ad-hoc executable files to process verification results and results log files.

- d. The DEVICE Database shall include the description of what the DEVICE Database contains, including the files' structure, naming conventions and version control labels. [\[ALL\]](#)

5.4.5 Preliminary DEVICE Data Sheet

- a. If agreed with the customer, the supplier shall generate a preliminary DEVICE Data Sheet in compliance with DRD from Annex H. [\[ALL\]](#)

NOTE 1 Data Sheets are in general intended for users of the DEVICE who do not have access to the DEVICE development documents such as the DEVICE Requirements Specification.

NOTE 2 For example, if the DEVICE is marketed as an off-the-shelf product.

NOTE 3 Some of the DEVICE technical parameters are inherent to the technology used to implement the DEVICE and therefore can be provided by the technology provider for example in the form of blank FPGA Data Sheets or ASIC manufacturer and standard cell library documents.

- b. The preliminary DEVICE Data Sheet shall contain all parts of the final DEVICE Data Sheet, with the same level of detail, using clearly identified estimated values for those parameter values that are not yet confirmed. [\[ALL\]](#)

NOTE For example, preliminary values can be obtained by simulation or measurements on a prototype, while final parameter values are confirmed with the validated DEVICE.

- c. The preliminary DEVICE Data Sheet shall contain at least the main functionalities and interfaces of the DEVICE. [\[ALL\]](#)

NOTE this information can be provided in the form of Interface Control Document (ICD) as specified in ECSS-E-ST-40 Annex E.

5.4.6 DEVICE Design and Verification Phase Review

- a. The DEVICE Design and Verification Phase shall be concluded by the DEVICE Design and Verification Phase Review in compliance with requirements from clause 5.1.4. [\[ALL\]](#)

- b. The following expected outputs shall be reviewed during DEVICE Design and Verification Phase Review: [\[ALL\]](#)

1. DEVICE Verification Plan (final) (DVeP) as in DRD in Annex C;
2. DEVICE Design Report;
3. DEVICE Design Verification Report;
4. DEVICE Data Sheet (preliminary) as in DRD in Annex H;
5. DEVICE database;
6. DEVICE Feasibility and Risk Assessment Report (update) (DFRAR) as in DRD in Annex F;

7. DEVICE VCD (update).

5.5 DEVICE Detailed Design Phase

5.5.1 Overview

During the DEVICE Detailed Design Phase, the DEVICE design is translated into a structural description at the level of elementary cells of the selected technology and cell library: the pre-layout netlist.

Additional information is generated for the subsequent development phases, such as layout constraints, floorplanning, Production Tests programs and a detailed pin description.

For digital designs, the above-mentioned design description is the technology specific gate-level pre-layout netlist normally obtained by synthesis tools.

For analogue designs, it is a verified sized transistor-level netlist. However, in many analogue designs, there is no separation between circuit netlist design and layout.

Meeting DEVICE timing, occupancy and power targets sometimes can need performing iterations between netlist generation and layout generation.

5.5.2 Netlist Generation

- a. Tasks specified in requirements 5.5.2b to 5.5.2u shall be performed and documented by the supplier in a Netlist Generation Report. [\[ALL\]](#)
- b. The supplier shall confirm use of design tools version as specified in the DEVICE Development Plan of DRD in Annex B, including the tools maintenance status, known bugs and existing patches. [\[ALL\]](#)
- c. The supplier shall choose and implement the pre-layout netlist generation with the selected tool options and constraints. [\[ALL\]](#)

NOTE For example, script variables and commands to control the timing, area, power, types of library cells used or different synthesis modes for complex DEVICES.

- d. Clock, reset or other signals needing specific generation, buffering and delay-controlled distribution with high fan-out shall be implemented using available netlist generation resources. [\[D-ASIC, A-ASIC, FPGA, --\]](#)

NOTE These buffers and signal distribution trees can be implemented during the Layout Phase also, depending on what is the target implementation technology.

- e. DFT and Production Tests requirements shall be implemented. [\[ALL\]](#)
- f. The radiation hardening approach shall be implemented and verified at netlist level. [\[ALL\]](#)