

AerosystemsLab

# VHDL Design and Coding Standard

## for Safety-Critical FPGAs

Airborne: RTCA DO-254 / EUROCAE ED-80, EASA AMC 20-152A, DAL A to D  
Space : ECSS-E-ST-20-40C / ECSS-Q-ST-60-03C, Class 1 to 3

|              |                     |            |  |            |  |
|--------------|---------------------|------------|--|------------|--|
| Written by : | Eren ALTUN          | Checking : |  | Approval : |  |
| Function :   | HW & FPGA Tech Lead | Function : |  | Function : |  |
| Date :       | 14/10/2026          | Date :     |  | Date :     |  |
| Visa :       | E.A                 | Visa :     |  | Visa :     |  |

|                    |                                          |
|--------------------|------------------------------------------|
| Document reference | Hardware Development Coding Rules (VHDL) |
| Issue / Revision   | A                                        |
| Date               | 14/10/2026                               |
| Classification     | Confidential & Proprietary               |
| Language           | English                                  |

## Table of contents

|                                                                              |           |
|------------------------------------------------------------------------------|-----------|
| <b>PART A: GENERAL</b>                                                       | <b>11</b> |
| <b>1. Scope</b>                                                              | <b>11</b> |
| 1.1. Purpose                                                                 | 11        |
| 1.2. Field of application                                                    | 11        |
| 1.3. Targeted assurance levels                                               | 11        |
| 1.4. Structure of this document                                              | 12        |
| 1.5. Deviation policy (summary)                                              | 12        |
| <b>2. Applicable and reference documents</b>                                 | <b>13</b> |
| 2.1. Applicable and reference documents                                      | 13        |
| 2.2. AMC 20-152A and HDL coding rules                                        | 14        |
| <b>3. Terminology</b>                                                        | <b>16</b> |
| 3.1. Verbal forms                                                            | 16        |
| 3.2. Definitions                                                             | 16        |
| 3.3. Project policy reminder                                                 | 17        |
| <b>4. List of acronyms</b>                                                   | <b>18</b> |
| <b>5. Classification and applicability of the rules</b>                      | <b>21</b> |
| 5.1. Severity levels                                                         | 21        |
| 5.2. Assurance levels                                                        | 21        |
| 5.2.1. Aeronautics: Design Assurance Levels                                  | 21        |
| 5.2.2. Space: criticality categories and EEE classes                         | 21        |
| 5.2.3. Equivalence between aeronautical and space levels                     | 22        |
| 5.3. Reading the applicability matrix                                        | 22        |
| 5.4. Format of a rule card                                                   | 22        |
| 5.5. Rule families                                                           | 23        |
| 5.6. Deviation and waiver management                                         | 24        |
| 5.7. Project tailoring                                                       | 24        |
| <b>6. Design philosophy for critical FPGA designs</b>                        | <b>25</b> |
| 6.1. The ten principles                                                      | 25        |
| 6.2. General rules                                                           | 25        |
| [GEN.01] Fully synchronous design                                            | 25        |
| [GEN.02] Deterministic behavior                                              | 26        |
| [GEN.03] Functional hierarchical decomposition                               | 27        |
| [GEN.04] Requirements traceability of the code                               | 27        |
| [GEN.05] No dead code, documented deactivated code                           | 28        |
| [GEN.06] Defensive design and robustness                                     | 28        |
| [GEN.07] Complexity limits                                                   | 29        |
| [GEN.08] Mandatory reuse of the reference library                            | 30        |
| [GEN.09] Defined state at power-up and during reset                          | 30        |
| [GEN.10] Observability of status and errors                                  | 31        |
| <b>PART B: CODING RULES</b>                                                  | <b>32</b> |
| <b>7. Language subset and libraries</b>                                      | <b>32</b> |
| 7.1. Language versions                                                       | 32        |
| [LNG.01] VHDL-93 for RTL, VHDL-2008 only where necessary                     | 32        |
| [LNG.02] Tool and language configuration recorded                            | 33        |
| [LNG.03] Full VHDL-2008 allowed in testbenches                               | 33        |
| 7.2. Libraries and packages                                                  | 33        |
| [LNG.04] Only IEEE standard packages                                         | 33        |
| [LNG.05] No user functions, procedures, variables or shared variables in RTL | 34        |
| 7.3. Forbidden and restricted constructs                                     | 35        |
| [LNG.06] Forbidden constructs in RTL                                         | 35        |
| [LNG.07] Assertions in RTL limited to elaboration checks                     | 36        |
| <b>8. File organization</b>                                                  | <b>37</b> |
| 8.1. Files and design units                                                  | 37        |
| [FIL.01] File extension                                                      | 37        |
| [FIL.02] One design unit pair per file, file named after the unit            | 37        |
| [FIL.03] Project directory structure                                         | 37        |

|          |                                                                   |           |
|----------|-------------------------------------------------------------------|-----------|
| [FIL.04] | Architecture names .....                                          | 38        |
| [FIL.05] | Order of the declarations in a file .....                         | 38        |
| 8.2.     | Layout .....                                                      | 38        |
| [FIL.06] | One statement per line .....                                      | 38        |
| [FIL.07] | One port, generic or association per line .....                   | 39        |
| [FIL.08] | Indentation with spaces .....                                     | 39        |
| [FIL.09] | Vertical alignment .....                                          | 39        |
| [FIL.10] | English, ASCII, lower-case keywords .....                         | 39        |
| [FIL.11] | Vendor-generated files untouched and isolated .....               | 40        |
| 9.       | <b>File header and traceability .....</b>                         | <b>41</b> |
| [HDR.01] | Mandatory file header .....                                       | 41        |
| [HDR.02] | Revision history consistent with configuration management .....   | 42        |
| [HDR.03] | Requirement traceability tags .....                               | 42        |
| [HDR.04] | Header of generated and third-party files .....                   | 42        |
| 10.      | <b>Naming conventions .....</b>                                   | <b>43</b> |
| 10.1.    | Prefixes .....                                                    | 43        |
| [NAM.01] | Mandatory prefixes .....                                          | 43        |
| [NAM.02] | Suffixes .....                                                    | 43        |
| [NAM.03] | Case and word separation .....                                    | 44        |
| [NAM.04] | Clock names .....                                                 | 44        |
| [NAM.05] | Reset names .....                                                 | 45        |
| [NAM.06] | Meaningful names .....                                            | 45        |
| [NAM.07] | Name preserved across hierarchy .....                             | 45        |
| [NAM.08] | Reserved words of VHDL, Verilog and SystemVerilog .....           | 45        |
| [NAM.09] | FSM naming .....                                                  | 46        |
| [NAM.10] | Top-level pad names .....                                         | 46        |
| [NAM.11] | Instance labels .....                                             | 46        |
| [NAM.12] | Positive logic inside the design .....                            | 46        |
| [NAM.13] | Labels of processes and generate statements .....                 | 47        |
| 11.      | <b>Comments and documentation .....</b>                           | <b>48</b> |
| [COM.01] | Comments explain the intent .....                                 | 48        |
| [COM.02] | Comment block before each process, instance and generate .....    | 48        |
| [COM.03] | Comment on each declaration .....                                 | 48        |
| [COM.04] | Comments are accurate and maintained .....                        | 49        |
| [COM.05] | Standard comment markers .....                                    | 49        |
| [COM.06] | Units, latency and timing assumptions documented .....            | 49        |
| 12.      | <b>Types, vectors and arithmetic .....</b>                        | <b>50</b> |
| [TYP.01] | Allowed types .....                                               | 50        |
| [TYP.02] | Top-level port types .....                                        | 50        |
| [TYP.03] | Integer types constrained .....                                   | 50        |
| [TYP.04] | Vector direction and bit order .....                              | 51        |
| [TYP.05] | Array index direction .....                                       | 51        |
| [TYP.06] | Arithmetic on numeric_std types with explicit widths .....        | 51        |
| [TYP.07] | Comparison operands of identical type and width .....             | 52        |
| [TYP.08] | No meta-values in RTL .....                                       | 52        |
| [TYP.09] | Named constants instead of literals .....                         | 52        |
| [TYP.10] | Widths derived from generics and constants .....                  | 53        |
| [TYP.11] | Records for internal grouping only .....                          | 53        |
| [TYP.12] | Restricted arithmetic operators .....                             | 53        |
| [TYP.13] | Fixed-point representation documented .....                       | 53        |
| 13.      | <b>Signals, constants, generics and initialization .....</b>      | <b>55</b> |
| [SIG.01] | Scope of signals .....                                            | 55        |
| [SIG.02] | Single driver .....                                               | 55        |
| [SIG.03] | Initial value at declaration equal to the reset value .....       | 55        |
| [SIG.04] | Constants .....                                                   | 56        |
| [SIG.05] | Generics .....                                                    | 56        |
| [SIG.06] | No implicit default values on unconnected inputs .....            | 57        |
| 14.      | <b>Concurrent statements and combinational logic .....</b>        | <b>58</b> |
| 14.1.    | Why no combinational process? .....                               | 58        |
| [CMB.01] | No combinational process .....                                    | 58        |
| [CMB.02] | Registered outputs of entities .....                              | 59        |
| [CMB.03] | Conditional assignment ends with an unconditional else .....      | 59        |
| [CMB.04] | Selected assignment with complete choices and "when others" ..... | 60        |

|            |                                                                          |           |
|------------|--------------------------------------------------------------------------|-----------|
| [CMB.05]   | No combinational loop .....                                              | 60        |
| [CMB.06]   | No latch .....                                                           | 60        |
| [CMB.07]   | Combinational signals never drive clocks, resets or crossings .....      | 61        |
| [CMB.08]   | Readable and bounded logic cones .....                                   | 61        |
| [CMB.09]   | Explicit operator precedence .....                                       | 62        |
| [CMB.10]   | Multiplexer and priority coding style .....                              | 62        |
| <b>15.</b> | <b>Synchronous processes .....</b>                                       | <b>63</b> |
| 15.1.      | The reference process template .....                                     | 63        |
| [SEQ.01]   | Clocked process template .....                                           | 63        |
| [SEQ.02]   | Every register of a process is reset in that process .....               | 64        |
| [SEQ.03]   | One clock domain and one function per process .....                      | 64        |
| [SEQ.04]   | Explicit default values for pulse signals .....                          | 65        |
| [SEQ.05]   | Case and if statements .....                                             | 65        |
| [SEQ.06]   | Synchronous edge detection .....                                         | 65        |
| [SEQ.07]   | Pipelines and delay lines .....                                          | 66        |
| <b>16.</b> | <b>Finite state machines .....</b>                                       | <b>68</b> |
| 16.1.      | FSM architectures .....                                                  | 68        |
| [FSM.01]   | States encoded with an enumerated type .....                             | 68        |
| [FSM.02]   | Single-process FSM .....                                                 | 68        |
| [FSM.03]   | Registered FSM outputs .....                                             | 69        |
| [FSM.04]   | Every state explicitly handled .....                                     | 69        |
| [FSM.05]   | Defined reset state .....                                                | 69        |
| [FSM.06]   | Reachability, no dead-lock, bounded waits .....                          | 69        |
| [FSM.07]   | Safe FSM: recovery from illegal states .....                             | 70        |
| [FSM.08]   | FSM synthesis attributes and report review .....                         | 72        |
| [FSM.09]   | Synchronous FSM inputs .....                                             | 72        |
| [FSM.10]   | FSM size and decomposition .....                                         | 73        |
| 16.2.      | SEU-hardened state machines .....                                        | 73        |
| [FSM.11]   | Single-error-correcting state encoding for critical FSMs .....           | 73        |
| <b>17.</b> | <b>Counters, timers and comparators .....</b>                            | <b>76</b> |
| [CNT.01]   | Counter type and width .....                                             | 76        |
| [CNT.02]   | Non-strict end-of-count comparison .....                                 | 76        |
| [CNT.03]   | Wrap-around or saturation specified .....                                | 77        |
| [CNT.04]   | Clock-enable ticks instead of divided clocks .....                       | 77        |
| [CNT.05]   | Time-out on every wait for an external event .....                       | 78        |
| [CNT.06]   | Time constants derived from the clock frequency .....                    | 78        |
| [CNT.07]   | Watchdogs .....                                                          | 78        |
| <b>18.</b> | <b>Reset architecture .....</b>                                          | <b>80</b> |
| 18.1.      | Reset strategy .....                                                     | 80        |
| 18.1.1.    | Comparison of reset styles .....                                         | 80        |
| 18.1.2.    | Reliability of the reset mechanisms per target technology .....          | 81        |
| 18.2.      | Reset rules .....                                                        | 82        |
| [RST.01]   | Single reset manager at the top level .....                              | 82        |
| [RST.02]   | Qualified and registered reset sources .....                             | 83        |
| [RST.03]   | Single active-low polarity, no inversion outside the reset manager ..... | 85        |
| [RST.04]   | Asynchronous assertion, synchronous de-assertion per clock domain .....  | 85        |
| [RST.05]   | Every register is reset .....                                            | 87        |
| [RST.06]   | Synchronous reset in user logic .....                                    | 87        |
| [RST.07]   | Reset adaptation for vendor macros .....                                 | 87        |
| [RST.08]   | Minimum reset duration .....                                             | 88        |
| [RST.09]   | Reset sequencing between domains .....                                   | 88        |
| [RST.10]   | Reset domain crossings .....                                             | 88        |
| [RST.11]   | Functional clear distinct from reset .....                               | 89        |
| [RST.12]   | Reset net implementation and timing .....                                | 89        |
| [RST.13]   | Reset tree hardening against SET (space) .....                           | 89        |
| <b>19.</b> | <b>Clock architecture .....</b>                                          | <b>90</b> |
| [CLK.01]   | Single clock management entity .....                                     | 90        |
| [CLK.02]   | Minimum number of clock domains .....                                    | 91        |
| [CLK.03]   | Clocks only from the clock manager .....                                 | 91        |
| [CLK.04]   | Single active edge .....                                                 | 91        |
| [CLK.05]   | No gated clock, use clock enables .....                                  | 92        |
| [CLK.06]   | No clock re-assignment .....                                             | 92        |
| [CLK.07]   | Clock never used as data .....                                           | 92        |
| [CLK.08]   | One clock per entity, except CDC blocks .....                            | 93        |

|            |                                                                               |            |
|------------|-------------------------------------------------------------------------------|------------|
|            | [CLK.09] Complete timing constraints .....                                    | 93         |
|            | [CLK.10] Clock and PLL monitoring .....                                       | 93         |
| <b>20.</b> | <b>Clock domain crossing .....</b>                                            | <b>95</b>  |
| 20.1.      | Metastability and synchronizers .....                                         | 95         |
| 20.2.      | Selection of the crossing mechanism .....                                     | 96         |
| 20.3.      | General CDC rules .....                                                       | 96         |
|            | [CDC.01] Every crossing through an approved mechanism .....                   | 96         |
|            | [CDC.02] Synchronizer structure integrity .....                               | 97         |
|            | [CDC.03] No independent synchronization of related bits .....                 | 97         |
|            | [CDC.04] No reconvergence of separately synchronized signals .....            | 97         |
|            | [CDC.05] Synchronizer depth and MTBF .....                                    | 97         |
| 20.4.      | Single-bit crossings .....                                                    | 98         |
|            | [CDC.10] Level synchronizer .....                                             | 98         |
|            | [CDC.11] Asynchronous top-level inputs .....                                  | 99         |
|            | [CDC.12] Pulse crossing with toggle synchronizer .....                        | 99         |
|            | [CDC.13] Loss-free event crossing with handshake .....                        | 100        |
| 20.5.      | Data crossings through block-RAM FIFOs .....                                  | 103        |
|            | [CDC.20] Data crossings through block-RAM dual-clock FIFOs .....              | 103        |
|            | [CDC.21] FIFO sizing analysis .....                                           | 106        |
|            | [CDC.22] Full / empty respected .....                                         | 106        |
|            | [CDC.23] Overflow / underflow detection and reporting .....                   | 107        |
|            | [CDC.24] FIFO reset in both domains .....                                     | 107        |
|            | [CDC.25] Gray coding restricted to the library .....                          | 107        |
|            | [CDC.26] CDC timing constraints .....                                         | 107        |
|            | [CDC.27] CDC verification .....                                               | 108        |
| <b>21.</b> | <b>Embedded memories .....</b>                                                | <b>109</b> |
|            | [MEM.01] Block RAM through technology wrappers .....                          | 109        |
|            | [MEM.02] Defined memory content .....                                         | 109        |
|            | [MEM.03] Read-during-write and collision behavior .....                       | 109        |
|            | [MEM.04] Memory latency and output registers .....                            | 110        |
|            | [MEM.05] Address range checks .....                                           | 110        |
|            | [MEM.06] No distributed (LUT) RAM for critical storage in space designs ..... | 110        |
| <b>22.</b> | <b>Top level and I/O .....</b>                                                | <b>111</b> |
|            | [TOP.01] Structural top level .....                                           | 111        |
|            | [TOP.02] Pin assignment and I/O standards reviewed .....                      | 114        |
|            | [TOP.03] I/O primitives in technology wrappers .....                          | 115        |
|            | [TOP.04] Input registers .....                                                | 115        |
|            | [TOP.05] Registered top-level outputs .....                                   | 115        |
|            | [TOP.06] Unused pins defined .....                                            | 116        |
|            | [TOP.07] Defined output enables .....                                         | 116        |
|            | [TOP.08] No unused ports .....                                                | 116        |
|            | [TOP.09] Tri-states only at the pads .....                                    | 116        |
|            | [TOP.10] Debug and test logic .....                                           | 117        |
|            | [TOP.11] Device configuration options under control .....                     | 117        |
| <b>23.</b> | <b>Hierarchy, instantiation and packages .....</b>                            | <b>118</b> |
|            | [HIE.01] Direct entity instantiation .....                                    | 118        |
|            | [HIE.02] Named association .....                                              | 118        |
|            | [HIE.03] Port ordering .....                                                  | 118        |
|            | [HIE.04] Port modes .....                                                     | 119        |
|            | [HIE.05] Unused outputs open, unused inputs tied .....                        | 119        |
|            | [HIE.06] No logic in port maps .....                                          | 119        |
|            | [HIE.07] Output ports never read internally .....                             | 119        |
|            | [HIE.08] Generate statements .....                                            | 120        |
|            | [PKG.01] Package content .....                                                | 120        |
|            | [PKG.02] Package organization .....                                           | 121        |
|            | [PKG.03] Design identification constants .....                                | 121        |
| <b>24.</b> | <b>Technology wrappers, attributes and synthesis .....</b>                    | <b>122</b> |
|            | [TEC.01] Vendor-specific elements isolated in wrappers .....                  | 122        |
|            | [TEC.02] One wrapper file per technology, same entity .....                   | 122        |
|            | [TEC.03] Technology-independent wrapper interface with protections .....      | 122        |
|            | [TEC.04] Approved synthesis attributes .....                                  | 122        |
|            | [TEC.05] Optimizations restricted for critical designs .....                  | 123        |
|            | [TEC.06] Tool options under configuration management .....                    | 123        |
|            | [TEC.07] Netlist versus RTL consistency .....                                 | 123        |

|                                                                      |                                                         |            |
|----------------------------------------------------------------------|---------------------------------------------------------|------------|
| [TEC.08]                                                             | Warnings analyzed .....                                 | 124        |
| [TEC.09]                                                             | Synthesis report review .....                           | 124        |
| [TEC.10]                                                             | Bitstream identification and integrity .....            | 124        |
| <b>PART C: AERONAUTICS (DO-254 / ED-80, AMC 20-152A, AC 20-152A)</b> |                                                         | <b>125</b> |
| <b>25. Regulatory framework</b>                                      |                                                         | <b>125</b> |
| 25.1.                                                                | Certification context .....                             | 125        |
| 25.2.                                                                | Simple and complex custom devices .....                 | 125        |
| 25.3.                                                                | Hardware design life cycle .....                        | 126        |
| 25.4.                                                                | Stages of involvement .....                             | 126        |
| <b>26. Objectives per DAL and their impact on HDL design</b>         |                                                         | <b>128</b> |
| 26.1.                                                                | Aeronautical process rules .....                        | 128        |
| [AER.01]                                                             | This standard referenced by the plans .....             | 128        |
| [AER.02]                                                             | Derived requirements captured and fed back .....        | 129        |
| [AER.03]                                                             | Detailed design review against this standard .....      | 129        |
| [AER.04]                                                             | Implementation review .....                             | 129        |
| [AER.05]                                                             | Robustness requirements and verification .....          | 129        |
| [AER.06]                                                             | Independence of verification .....                      | 130        |
| [AER.07]                                                             | Elemental analysis through HDL code coverage .....      | 130        |
| [AER.08]                                                             | COTS IP integration .....                               | 130        |
| [AER.09]                                                             | Tool assessment .....                                   | 130        |
| [AER.10]                                                             | Reuse of previously developed hardware .....            | 131        |
| [AER.11]                                                             | Configuration management of HDL data .....              | 131        |
| [AER.12]                                                             | Errata of the device and of the IP .....                | 131        |
| [AER.13]                                                             | Single event effects in avionics .....                  | 131        |
| <b>27. Elemental analysis and code coverage criteria per DAL</b>     |                                                         | <b>133</b> |
| 27.1.                                                                | Coverage metrics .....                                  | 133        |
| 27.2.                                                                | Company criteria .....                                  | 133        |
| 27.3.                                                                | Accepted justifications for uncovered items .....       | 134        |
| <b>28. COTS IP, tools and reuse</b>                                  |                                                         | <b>135</b> |
| 28.1.                                                                | COTS IP assessment .....                                | 135        |
| 28.2.                                                                | Tool assessment strategy .....                          | 135        |
| <b>29. DAL-specific summary</b>                                      |                                                         | <b>136</b> |
| <b>PART D: SPACE (ECSS-E-ST-20-40C, ECSS-Q-ST-60-03C, radiation)</b> |                                                         | <b>137</b> |
| <b>30. ECSS framework for FPGA development</b>                       |                                                         | <b>137</b> |
| 30.1.                                                                | Applicable ECSS standards .....                         | 137        |
| 30.2.                                                                | Development phases and reviews .....                    | 137        |
| 30.3.                                                                | Criticality-based tailoring .....                       | 138        |
| 30.4.                                                                | Space process rules .....                               | 138        |
| [SPC.01]                                                             | This standard referenced by the development plan .....  | 138        |
| [SPC.02]                                                             | Design rule compliance report at DVR .....              | 138        |
| [SPC.03]                                                             | Independent verification for Class 1 .....              | 138        |
| [SPC.04]                                                             | SEE analysis of the design .....                        | 139        |
| [SPC.05]                                                             | Deactivated and unreachable functions .....             | 139        |
| [SPC.06]                                                             | Reuse files for reused HDL and IP cores .....           | 139        |
| [SPC.07]                                                             | Coverage criteria per class .....                       | 139        |
| <b>31. Radiation effects on FPGAs</b>                                |                                                         | <b>141</b> |
| 31.1.                                                                | Effects .....                                           | 141        |
| 31.2.                                                                | Sensitivity per FPGA technology .....                   | 141        |
| <b>32. Radiation mitigation rules</b>                                |                                                         | <b>142</b> |
| 32.1.                                                                | Strategy .....                                          | 142        |
| [RAD.01]                                                             | Mitigation strategy derived from the SEE analysis ..... | 142        |
| [RAD.02]                                                             | Approved mitigation mechanisms only .....               | 142        |
| 32.2.                                                                | Triple modular redundancy .....                         | 142        |
| [RAD.10]                                                             | TMR scheme selected per technology and class .....      | 143        |
| [RAD.11]                                                             | Self-correcting TMR registers .....                     | 143        |
| [RAD.12]                                                             | Redundancy preserved by synthesis .....                 | 145        |
| [RAD.13]                                                             | No single point of failure in the TMR boundary .....    | 145        |
| [RAD.14]                                                             | Physical separation of redundant copies .....           | 146        |
| [RAD.15]                                                             | Automatic TMR insertion tools .....                     | 146        |

|                                                                     |                                                              |            |
|---------------------------------------------------------------------|--------------------------------------------------------------|------------|
| 32.3.                                                               | Memories: EDAC and scrubbing .....                           | 146        |
| [RAD.20]                                                            | EDAC on memories holding critical data .....                 | 146        |
| [RAD.21]                                                            | Vendor block-RAM ECC where available .....                   | 148        |
| [RAD.22]                                                            | Memory scrubbing .....                                       | 149        |
| 32.4.                                                               | Configuration memory and SEFI .....                          | 149        |
| [RAD.30]                                                            | Configuration scrubbing of SRAM-based FPGAs .....            | 149        |
| [RAD.31]                                                            | SEFI detection and recovery .....                            | 149        |
| 32.5.                                                               | Transients, inputs and timing .....                          | 149        |
| [RAD.32]                                                            | Filtering of critical discrete inputs .....                  | 149        |
| [RAD.33]                                                            | Hardened state machines and counters .....                   | 151        |
| [RAD.34]                                                            | Timing margin for radiation and aging .....                  | 151        |
| [RAD.35]                                                            | Verification of the mitigation .....                         | 151        |
| 33.                                                                 | Space class summary .....                                    | 152        |
| <b>PART E: VERIFICATION .....</b>                                   |                                                              | <b>153</b> |
| 34.                                                                 | Testbench rules .....                                        | 153        |
| [VER.01]                                                            | Self-checking testbenches .....                              | 153        |
| [VER.02]                                                            | Requirements-based test cases with traceability .....        | 153        |
| [VER.03]                                                            | Testbench architecture .....                                 | 155        |
| [VER.04]                                                            | Automatic end of simulation .....                            | 155        |
| [VER.05]                                                            | Realistic clocks and resets .....                            | 156        |
| [VER.06]                                                            | Design under test unmodified .....                           | 156        |
| [VER.07]                                                            | Assertion and reporting conventions .....                    | 156        |
| [VER.08]                                                            | Reproducible regressions .....                               | 156        |
| [VER.09]                                                            | Review of test cases and procedures .....                    | 157        |
| [VER.10]                                                            | Robustness test cases .....                                  | 157        |
| [VER.11]                                                            | Simulation of clock domain crossings .....                   | 157        |
| [VER.12]                                                            | Fault injection for unreachable robustness code .....        | 159        |
| [VER.13]                                                            | Post-layout timing simulation .....                          | 160        |
| [VER.14]                                                            | Hardware tests on the target .....                           | 160        |
| 35.                                                                 | Code coverage closure .....                                  | 161        |
| [COV.01]                                                            | Coverage collected on the full regression .....              | 161        |
| [COV.02]                                                            | Exclusions only through a reviewed justification table ..... | 161        |
| [COV.03]                                                            | Coverage on the final baseline .....                         | 161        |
| 36.                                                                 | Static analyses .....                                        | 163        |
| 36.1.                                                               | Linting .....                                                | 163        |
| [VER.20]                                                            | Linters with a rule set derived from this standard .....     | 163        |
| [VER.21]                                                            | Linters results analyzed .....                               | 163        |
| [VER.22]                                                            | Quality metrics .....                                        | 163        |
| 36.2.                                                               | Timing analysis, CDC and RDC .....                           | 163        |
| [VER.23]                                                            | Static timing analysis review .....                          | 163        |
| [VER.24]                                                            | Formal property verification for critical blocks .....       | 164        |
| 37.                                                                 | Code review process .....                                    | 165        |
| [VER.30]                                                            | Review entry criteria .....                                  | 165        |
| [VER.31]                                                            | Review records and finding tracking .....                    | 165        |
| [VER.32]                                                            | Re-review after modification .....                           | 165        |
| <b>Annex A: Summary of the rules and applicability matrix .....</b> |                                                              | <b>166</b> |
| <b>Annex B: Reference library .....</b>                             |                                                              | <b>173</b> |
| B.1                                                                 | Listings not reproduced in the body of the document .....    | 174        |
| B.2                                                                 | Testbenches .....                                            | 183        |
| <b>Annex C: Review checklists .....</b>                             |                                                              | <b>190</b> |
| C.1                                                                 | HDL code review checklist .....                              | 190        |
| C.2                                                                 | Top level and board interface checklist .....                | 191        |
| C.3                                                                 | Synthesis and implementation report checklist .....          | 191        |
| C.4                                                                 | Verification review checklist .....                          | 192        |
| <b>Annex D: Compliance matrix template .....</b>                    |                                                              | <b>193</b> |
| <b>Annex E: File templates .....</b>                                |                                                              | <b>201</b> |
| <b>Annex F: Linter rule set .....</b>                               |                                                              | <b>203</b> |
| F.1                                                                 | Principles .....                                             | 203        |

|                                                             |            |
|-------------------------------------------------------------|------------|
| F.2 Tool chain.....                                         | 203        |
| F.3 Severity, assurance-level profiles and acceptance ..... | 204        |
| F.4 Vendor checks to disable .....                          | 205        |
| F.5 Custom checks ASL-CK.....                               | 205        |
| F.6 Mapping of the rules to the checks .....                | 207        |
| F.7 Check families of the commercial linters .....          | 209        |
| F.8 Resolution of native identifiers and validation .....   | 211        |
| F.9 Limits .....                                            | 212        |
| <b>Annex G: Evolution of this standard .....</b>            | <b>213</b> |
| <b>Bibliography .....</b>                                   | <b>214</b> |
| Aeronautical regulations and standards .....                | 214        |
| Space standards .....                                       | 214        |
| Language standards.....                                     | 214        |
| Design guidelines .....                                     | 215        |
| Vendor documentation .....                                  | 215        |
| Technical papers and books .....                            | 215        |

## List of figures

|                                                                       |     |
|-----------------------------------------------------------------------|-----|
| Figure 1: Reset architecture (rst_mgr).....                           | 81  |
| Figure 2: Clock architecture (clk_mgr).....                           | 90  |
| Figure 3: Two-stage synchronizer timing.....                          | 95  |
| Figure 4: Selection of the CDC mechanism .....                        | 96  |
| Figure 5: cdc_event_hs timing .....                                   | 101 |
| Figure 6: Dual-clock FIFO principle.....                              | 103 |
| Figure 7: DO-254 hardware design life cycle .....                     | 126 |
| Figure 8: ECSS-E-ST-20-40C DEVICE development phases and reviews..... | 137 |
| Figure 9: Local and distributed TMR .....                             | 142 |
| Figure 10: Coverage closure process .....                             | 161 |

## List of tables

|                                                                                                         |     |
|---------------------------------------------------------------------------------------------------------|-----|
| Table 1: Organization of the standard .....                                                             | 12  |
| Table 2: Applicable and reference documents .....                                                       | 14  |
| Table 3: AMC 20-152A provisions related to HDL design .....                                             | 15  |
| Table 4: Verbal forms .....                                                                             | 16  |
| Table 5: Definitions .....                                                                              | 17  |
| Table 6: List of acronyms .....                                                                         | 20  |
| Table 7: Severity levels .....                                                                          | 21  |
| Table 8: Aeronautical Design Assurance Levels .....                                                     | 21  |
| Table 9: Space classes used in this standard .....                                                      | 22  |
| Table 10: Engineering equivalence of levels .....                                                       | 22  |
| Table 11: Structure of a rule card .....                                                                | 23  |
| Table 12: Rule families .....                                                                           | 24  |
| Table 13: Deviation approval per level .....                                                            | 24  |
| Table 14: Design principles .....                                                                       | 25  |
| Table 15: Complexity limits .....                                                                       | 29  |
| Table 16: Reference library blocks .....                                                                | 30  |
| Table 17: VHDL-2008 features: allowed usage.....                                                        | 33  |
| Table 18: Constructs forbidden in RTL.....                                                              | 36  |
| Table 19: Recommended directory structure .....                                                         | 38  |
| Table 20: Fields of the file header .....                                                               | 41  |
| Table 21: Prefixes .....                                                                                | 43  |
| Table 22: Suffixes .....                                                                                | 44  |
| Table 23: Effect of declaration initial values per technology.....                                      | 56  |
| Table 24: Defects induced by combinational processes .....                                              | 58  |
| Table 25: FSM architectures .....                                                                       | 68  |
| Table 26: FSM synthesis attributes .....                                                                | 72  |
| Table 27: Behavior of FSM encodings under single bit-flip .....                                         | 73  |
| Table 28: Reset styles.....                                                                             | 80  |
| Table 29: Reliable reset mechanisms per target .....                                                    | 81  |
| Table 30: Synchronizer MTBF parameters.....                                                             | 95  |
| Table 31: Approved CDC mechanisms .....                                                                 | 96  |
| Table 32: Approved synthesis attributes .....                                                           | 123 |
| Table 33: Aeronautical certification documents.....                                                     | 125 |
| Table 34: DO-254 processes and the HDL design.....                                                      | 126 |
| Table 35: Stages of involvement and HDL evidence.....                                                   | 127 |
| Table 36: HDL activities per DAL .....                                                                  | 128 |
| Table 37: HDL code coverage metrics .....                                                               | 133 |
| Table 38: Coverage criteria per DAL ("100 %" = covered or individually justified; R = recommended)..... | 133 |
| Table 39: Accepted justifications of coverage holes.....                                                | 134 |
| Table 40: COTS IP objectives and their implementation.....                                              | 135 |
| Table 41: Tool assessment strategy .....                                                                | 135 |
| Table 42: DAL quick reference .....                                                                     | 136 |
| Table 43: ECSS reviews and expected HDL status .....                                                    | 138 |
| Table 44: Coverage criteria per space class .....                                                       | 140 |
| Table 45: Radiation effects.....                                                                        | 141 |

|                                                                                  |     |
|----------------------------------------------------------------------------------|-----|
| Table 46: Radiation sensitivity per FPGA technology .....                        | 141 |
| Table 47: TMR schemes .....                                                      | 143 |
| Table 48: Space class quick reference.....                                       | 152 |
| Table 49: Example of coverage justification table .....                          | 161 |
| Table 50: Code review steps .....                                                | 165 |
| Table 51: Summary of the rules and applicability matrix .....                    | 172 |
| Table 52: Reference library content and verification status .....                | 173 |
| Table 53: HDL code review checklist .....                                        | 191 |
| Table 54: Top level checklist .....                                              | 191 |
| Table 55: Synthesis and implementation checklist.....                            | 192 |
| Table 56: Verification review checklist .....                                    | 192 |
| Table 57: Compliance matrix template .....                                       | 200 |
| Table 58: Tool chain of the linter rule set .....                                | 203 |
| Table 59: Severity mapping.....                                                  | 204 |
| Table 60: Severity of the 96 automatically checkable rules in each profile ..... | 204 |
| Table 61: Profile files of the rule set .....                                    | 204 |
| Table 62: Vendor checks conflicting with this standard .....                     | 205 |
| Table 63: Custom checks ASL-CK.....                                              | 206 |
| Table 64: Mapping of the automatically checkable rules .....                     | 209 |
| Table 65: Check families and native checks .....                                 | 211 |
| Table 66: Bibliography (aeronautics).....                                        | 214 |
| Table 67: Bibliography (space).....                                              | 214 |
| Table 68: Bibliography (language standards .....                                 | 215 |
| Table 69: Bibliography (design guidelines .....                                  | 215 |
| Table 70: Bibliography (vendor documentation .....                               | 215 |
| Table 71: Bibliography (technical papers and books .....                         | 216 |

# PART A: GENERAL

*Scope, documents, terminology, acronyms and the rule classification system used throughout this standard.*

## 1. Scope

### 1.1. Purpose

This VHDL Design and Coding Standard for Safety-Critical FPGAs defines the rules, methods and reference design patterns to be applied when developing VHDL code for FPGA devices used in safety-critical airborne and space equipment. It is the reference ("the bible") of every FPGA engineer of the company, from the first line of RTL to the delivery of the bitstream and of the associated life-cycle data.

The objectives of the standard are:

- to produce designs that are robust, deterministic and fully verifiable, with no behavior that cannot be explained by the specification ("no suspicious behavior");
- to make the code reviewable by an independent engineer in a reasonable time, which is a pre-requisite of DO-254 design reviews and of ECSS design reviews;
- to make structural code coverage (statement, branch, condition, expression, FSM, toggle) achievable at 100 %, or justifiable item by item;
- to remove the classes of defects that escape functional simulation: clock domain crossing (CDC) errors, reset domain crossing (RDC) errors, latches, combinational loops, simulation / synthesis mismatches, unprotected state machines, radiation sensitivity;
- to standardize practices between projects, sites and subcontractors (naming, file structure, headers, traceability tags), which eases reuse, configuration management and certification;
- to provide a reference library of approved, verified building blocks (synchronizers, reset manager, CDC FIFO wrapper, safe FSM, TMR register, EDAC) that shall be reused instead of being redesigned.

The standard is a Hardware Design Standard in the sense of RTCA DO-254 / EUROCAE ED-80 section 10.2.2 and Appendix A. EASA AMC 20-152A §5.10 (and FAA AC 20-152A) explicitly states that "HDL Coding Standards are part of the Hardware Design Standards"; objective CD-4 (§5.5.2.2) requires that DAL A and B designs be reviewed against them and that DAL C designs be shown to satisfy them (see 2.2). For space projects it is the design-rule document requested by the device development plan of ECSS-E-ST-20-40C and ECSS-Q-ST-60-03C.

### 1.2. Field of application

This standard applies to:

- all synthesizable VHDL (RTL) code developed in-house or by subcontractors for FPGA devices (SRAM-based, flash-based and antifuse-based), including code reused from previous projects;
- all VHDL testbenches and simulation models used to produce verification credit (chapter 34);
- the technology wrappers encapsulating vendor primitives and IP cores (the content generated by vendor tools is not hand-coded, but its wrapper is);
- the associated design constraints (timing, CDC, placement) where explicitly stated.

This standard does not apply to: Verilog / SystemVerilog RTL (separate standard), High-Level Synthesis sources, embedded software running on soft or hard processors (DO-178C / ECSS-E-ST-40C and ECSS-Q-ST-80C apply), and the internal content of third-party IP cores delivered as netlists (their integration is covered by chapters 24 and 28).

### 1.3. Targeted assurance levels

The rules are written for the most demanding levels, DO-254 / AMC 20-152A Design Assurance Level (DAL) A and space Class 1 (see definition in 5.2.2), and are relaxed for lower levels through an applicability matrix

carried by each rule. A design shall apply the column of its own level; a lower-level design may voluntarily apply the higher-level requirements (recommended for reusable modules, whose future DAL is unknown).

#### DAL / CLASS GUIDANCE

A reusable module (reference library, IP developed for several products) shall always be developed to the highest level of its foreseeable uses. In practice, all modules of the AerosystemsLab reference library are developed to DAL A / Class 1.

## 1.4. Structure of this document

| Part            | Chapters | Content                                                                                                                                                                                | Readers                                    |
|-----------------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|
| A: General      | 1 to 6   | Scope, documents, terminology, acronyms, classification of rules, design philosophy                                                                                                    | Everybody                                  |
| B: Coding rules | 7 to 24  | Language subset, files, headers, naming, types, combinational and sequential logic, FSM, counters, resets, clocks, CDC, memories, top level, hierarchy, technology wrappers, synthesis | Designers, reviewers                       |
| C: Aeronautics  | 25 to 29 | DO-254 / ED-80, AMC 20-152A, AC 20-152A: life cycle, objectives per DAL, elemental analysis, COTS IP, tools                                                                            | Designers, verification, certification     |
| D: Space        | 30 to 33 | ECSS-E-ST-20-40C, ECSS-Q-ST-60-03C, radiation effects and mitigation rules (TMR, EDAC, safe FSM, scrubbing)                                                                            | Designers, verification, product assurance |
| E: Verification | 34 to 37 | Testbench rules, coverage closure, static analyses (lint, CDC, RDC, STA), code review process                                                                                          | Verification, reviewers                    |
| Annexes         | A to G   | Rule summary and applicability matrix, reference library, review checklists, compliance matrix template, file templates, linter rule set, evolution of the standard                    | Everybody                                  |
| Bibliography    | -        | Standards, guidelines, vendor documentation and technical papers                                                                                                                       | Everybody                                  |

Table 1: Organization of the standard

## 1.5. Deviation policy (summary)

Any deviation from a Mandatory rule shall be justified, recorded and independently approved before the code is baselined. The deviation process, which depends on the DAL / Class, is defined in section 5.6. A deviation is never implicit: code that violates a rule without an approved deviation is a non-conformance, whatever its functional correctness.

## 2. Applicable and reference documents

### 2.1. Applicable and reference documents

The documents below are either applicable (AD: their requirements apply to the extent specified in this standard) or reference documents (RD: used as guidance or background). In case of conflict between this standard and an applicable document required by the project (PHAC, device development plan, customer specification), the applicable document prevails and the conflict shall be reported to the quality department. The complete bibliography, including vendor documentation and technical papers, is given at the end of this document.

| ID   | Status     | Reference                     | Issue                                                  | Title                                                                                                           | Use in this standard                                                                                                              |
|------|------------|-------------------------------|--------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| AD1  | Applicable | RTCA DO-254 / EUROCAE ED-80   | April 2000                                             | Design Assurance Guidance for Airborne Electronic Hardware                                                      | Life cycle, objectives, design standard (§10.2.2): chapters 25 to 29                                                              |
| AD2  | Applicable | EASA AMC 20-152A              | Initial issue, AMC-20 Amdt 19 (ED Decision 2020/010/R) | Development Assurance for Airborne Electronic Hardware (AEH)                                                    | HDL coding standards (§5.10), design reviews (§5.5.2), code coverage (§5.7, App. B.2.1.5), tools (§5.8), COTS IP (§5.11): see 2.2 |
| AD3  | Applicable | FAA AC 20-152A                | 07/10/2022                                             | Development Assurance for Airborne Electronic Hardware                                                          | FAA counterpart of AD2 (same objectives CD-n, IP-n, COTS-n)                                                                       |
| AD4  | Applicable | SAE ARP4754B / EUROCAE ED-79B | 2023                                                   | Guidelines for Development of Civil Aircraft and Systems                                                        | DAL allocation (5.2.1)                                                                                                            |
| AD5  | Applicable | ECSS-E-ST-20-40C              | 11 October 2023                                        | Space engineering: ASIC, FPGA and IP Core engineering                                                           | Chapter 30                                                                                                                        |
| AD6  | Applicable | ECSS-Q-ST-60-03C              | 11 October 2023                                        | Space product assurance: ASIC, FPGA and IP Core product assurance                                               | Chapter 30                                                                                                                        |
| AD7  | Applicable | IEEE Std 1076-1993            | 1994                                                   | IEEE Standard VHDL Language Reference Manual (VHDL-93)                                                          | Chapter 7                                                                                                                         |
| AD8  | Applicable | IEEE Std 1076-2008            | 2009                                                   | IEEE Standard VHDL Language Reference Manual (VHDL-2008)                                                        | Chapter 7                                                                                                                         |
| AD9  | Applicable | IEEE Std 1164-1993            | 1993                                                   | Multivalued Logic System for VHDL Model Interoperability (std_logic_1164)                                       | Chapter 12                                                                                                                        |
| AD10 | Applicable | IEEE Std 1076.3-1997          | 1997                                                   | Standard VHDL Synthesis Packages (numeric_std)                                                                  | Chapter 12                                                                                                                        |
| RD1  | Reference  | ECSS-Q-ST-60-02C              | 31 July 2008                                           | ASIC and FPGA development: superseded by AD5 and AD6, still applicable to legacy contracts                      | Chapter 30                                                                                                                        |
| RD2  | Reference  | ECSS-E-HB-20-40A              | September 2016                                         | Engineering techniques for radiation effects mitigation in ASICs and FPGAs handbook (formerly ECSS-Q-HB-60-02A) | Chapters 31, 32                                                                                                                   |
| RD3  | Reference  | ECSS-Q-ST-30C                 | latest                                                 | Space product assurance: Dependability (criticality categories)                                                 | 5.2.2                                                                                                                             |
| RD4  | Reference  | ECSS-Q-ST-60C Rev.4           | 30 April 2025                                          | Space product assurance: EEE components (component classes 1, 2, 3)                                             | 5.2.2                                                                                                                             |
| RD5  | Reference  | SAE ARP4761A / ED-135         | 2023                                                   | Guidelines for Conducting the Safety Assessment Process on Civil Aircraft, Systems, and Equipment               | Chapter 25                                                                                                                        |

| ID    | Status    | Reference            | Issue                | Title                                                                                                        | Use in this standard |
|-------|-----------|----------------------|----------------------|--------------------------------------------------------------------------------------------------------------|----------------------|
| RD6   | Reference | RTCA DO-330 / ED-215 | 2011                 | Software Tool Qualification Considerations (referenced by AD2 / AD3 for tool qualification)                  | Chapter 28           |
| RD7   | Reference | FAA AC 00-72         | 2017                 | Best Practices for Airborne Electronic Hardware Design Assurance Using EUROCAE ED-80 and RTCA DO-254 (DAL D) | Chapter 26           |
| RD8   | Reference | EASA CM-AS-004       | Issue 01, 08/01/2018 | Single Event Effects (SEE) on aircraft systems caused by cosmic rays (referenced by AMC 20-152A §1.4)        | [AER.13]             |
| RD9   | Reference | ECSS-E-ST-10-12C     | latest               | Space engineering: Methods for the calculation of radiation received and its effects                         | Chapter 31           |
| +RD10 | Reference | ECSS-Q-ST-60-15C     | latest               | Space product assurance: Radiation hardness assurance, EEE components                                        | Chapter 31           |

Table 2: Applicable and reference documents

**NOTE**

The issue of ECSS documents marked "latest" shall be taken from the project applicability list. When a contract still refers to ECSS-Q-ST-60-02C, its requirements apply in place of AD5 / AD6; the coding rules of this standard are unchanged.

## 2.2. AMC 20-152A and HDL coding rules

EASA AMC 20-152A (and its FAA counterpart AC 20-152A, developed jointly and harmonized) does not contain HDL coding rules itself. It does, however, make the existence and the application of an HDL coding standard an explicit part of the development assurance of custom devices (FPGA, ASIC), and it gives guidance on the reviews, the code coverage and the tools that this standard implements. The table below lists the provisions of AMC 20-152A that concern HDL design and where they are implemented in this document.

| AMC 20-152A provision                              | Content (summary)                                                                                                                                                                                                                           | Implementation in this standard                                                                                                         |
|----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| §5.10: Clarification of ED-80 / DO-254 Table A-1   | "HDL Coding Standards are part of the Hardware Design Standards." The row 10.2.2 "Hardware Design Standard" also applies to Level C (control category HC2).                                                                                 | This document is the hardware design standard for HDL: [AER.01] (referenced in the plans), [AER.11] (configuration control), chapter 26 |
| Objective CD-4: §5.5.2.2 Detailed design review    | DAL A/B: review of the detailed design with respect to the design standards and to the requirements (traceability). DAL C: demonstration that the detailed design satisfies the hardware design standards.                                  | [AER.03], checklist C.1 (Annex C), traceability rules [GEN.04], [HDR.03], linter [VER.20]                                               |
| Objective CD-5: §5.5.2.3 Implementation review     | Review of the outputs of the implementation tools (e.g. synthesis and place-and-route reports).                                                                                                                                             | [AER.04], [TEC.08], [TEC.09], checklist C.3                                                                                             |
| §5.2 / Objective CD-1: Simple or complex device    | Classification based on the design content (functions, interfaces, synchronous / asynchronous design, clocks, state machines); a device that cannot be shown to be comprehensively verifiable is complex.                                   | Chapter 25; the synchronous, few-clock, safe-FSM design style of Part B reduces verification complexity                                 |
| §5.7 / Objective CD-9: HDL code coverage (DAL A/B) | HDL code coverage is recognized as a method to perform elemental analysis; coverage criteria defined in the plans; any non-covered element analyzed and justified; coverage does not assess the completeness of requirements-based testing. | [AER.07], chapter 27, chapter 35 ([COV.01] to [COV.03])                                                                                 |

| AMC 20-152A provision                           | Content (summary)                                                                                                                                                                                              | Implementation in this standard                                            |
|-------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|
| Appendix B.2.1.5: Coverage criteria             | Suggests criteria such as statement, branch, condition and FSM state / transition coverage.                                                                                                                    | Coverage criteria per DAL (chapter 27), per space class ([SPC.07])         |
| §5.8 / Objectives CD-10, CD-11: Tool assessment | Independent assessment of tool outputs covering the potential tool errors; tool history credible and relevant, never a stand-alone means for design tools.                                                     | [AER.09], [TEC.07], [LNG.02], chapter 28                                   |
| §5.11 / Objectives IP-1 to IP-7: COTS IP        | Selection, assessment, complementary assurance, verification of the IP, its implementation and integration, requirements for used and unused functions; code coverage may not be possible on the COTS IP part. | [AER.08], [AER.12], technology wrappers ([TEC.01] to [TEC.03]), chapter 28 |
| Objectives CD-6, CD-7, CD-8                     | Review of verification cases and procedures; timing verification across temperature, voltage and process; robustness requirements (DAL A/B).                                                                   | [VER.09], [CLK.09], [VER.23], [AER.05], [GEN.06], [VER.10]                 |
| §1.4: Single event effects                      | SEE are not addressed by AMC 20-152A; they are usually addressed through a certification review item, with guidance in EASA CM-AS-004.                                                                         | [AER.13], Part D mitigation techniques (chapter 32)                        |

Table 3: AMC 20-152A provisions related to HDL design

**DAL / CLASS GUIDANCE**

Consequence for the designer: compliance with this coding standard is not optional for DAL A, B and C airborne FPGAs. For DAL A and B it is verified by an independent design review (CD-4); for DAL C it shall be demonstrated (review or automated checks). The design rules of Part B are therefore part of the certification evidence presented at SOI #2.

## 3. Terminology

### 3.1. Verbal forms

| Verbal form                       | Meaning in this standard                                                                                |
|-----------------------------------|---------------------------------------------------------------------------------------------------------|
| <b>shall</b>                      | Requirement. Its violation is a non-conformance unless an approved deviation exists.                    |
| <b>should</b>                     | Recommendation. Its violation shall be justified in the code (comment) and accepted during code review. |
| <b>may</b>                        | Permission. No justification needed.                                                                    |
| <b>can</b>                        | Possibility or capability, not a requirement.                                                           |
| is / are (in rationale and notes) | Statement of fact, informative only.                                                                    |

Table 4: Verbal forms

### 3.2. Definitions

| Term                        | Definition                                                                                                                                                                                                       |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Asynchronous signal         | Signal that is not generated by a flip-flop clocked by the clock of the domain that samples it: device pins, signals coming from another clock domain, signals generated without clock.                          |
| Clock domain                | Set of flip-flops clocked by the same clock net (same source, same edge). Two clocks derived from the same PLL with a fixed phase relationship are "related"; otherwise they are asynchronous.                   |
| Clock domain crossing (CDC) | Any path from a flip-flop of one clock domain to a flip-flop of another clock domain.                                                                                                                            |
| Reset domain crossing (RDC) | Any path from a flip-flop reset by one reset to a flip-flop reset by another reset (or not reset), which may be corrupted when only one of them is asserted.                                                     |
| Combinational logic         | Logic whose outputs depend only on the current value of its inputs. In this standard it is described exclusively by concurrent signal assignments (chapter 14).                                                  |
| Deterministic design        | Design whose outputs, for a given sequence of inputs applied on given clock edges, are uniquely defined by the specification, independently of routing delays, power-up conditions, simulator or synthesis tool. |
| Determinism (power-up)      | Property of every flip-flop of the design being in a known state when the first functional clock edge after reset release occurs.                                                                                |
| Dead code                   | Code that cannot be executed and does not correspond to any requirement. Forbidden.                                                                                                                              |
| Deactivated code            | Code that implements a requirement but is not used in a given configuration (e.g. disabled by a generic). Allowed if documented and verified to be inactive (DO-254 / ECSS-Q-ST-60-03C).                         |
| Elemental analysis          | DO-254 Appendix B method by which the verification coverage of each design element (here: each HDL statement, branch, condition, FSM state/transition) is analyzed. Usually supported by code coverage.          |
| Glitch                      | Short, unintended pulse on a combinational signal caused by unequal path delays. Harmless on a synchronous data path, dangerous on a clock, reset, enable of a latch or asynchronous input.                      |
| Metastability               | State of a flip-flop whose setup or hold time has been violated: its output can remain at an intermediate level or resolve late to either value.                                                                 |
| MTBF (synchronizer)         | Mean time between failures of a synchronizer, i.e. between two metastable events not resolved within the allowed time (chapter 20).                                                                              |
| Registered output           | Output driven directly by a flip-flop, without any logic between the flip-flop and the port.                                                                                                                     |

| Acronym | Meaning                                                                   |
|---------|---------------------------------------------------------------------------|
| TB      | Testbench                                                                 |
| TID     | Total Ionizing Dose                                                       |
| TMR     | Triple Modular Redundancy                                                 |
| TSO     | Technical Standard Order                                                  |
| VHDL    | VHSIC Hardware Description Language                                       |
| VHSIC   | Very High Speed Integrated Circuit                                        |
| VQAR    | DEVICE Validation, Qualification and Acceptance Review (ECSS-E-ST-20-40C) |
| XPM     | Xilinx Parameterized Macro (AMD)                                          |

Table 6: List of acronyms

| Family    | Domain                                       | Chapter  |
|-----------|----------------------------------------------|----------|
| RAD / SPC | Radiation mitigation and space process rules | 30 to 33 |
| VER / COV | Verification, testbenches and coverage       | 34 to 37 |

Table 12: Rule families

## 5.6. Deviation and waiver management

A deviation is a documented, approved decision not to apply a rule at a given location of the code. The process below is mandatory.

- 1) Identification: the designer identifies the rule and the exact location (file, line, signal or process label).
- 2) Justification: the designer writes the technical justification: why the rule cannot or should not be applied, what the residual risk is, and which compensating measure covers it (additional verification, analysis, constraint, assertion).
- 3) In-code marking: the code carries a standard comment immediately above the deviating construct: -- DEVIATION [RULE.nn] DEV-<project>-<number> : <one-line summary>. Linter waivers reference the same identifier.
- 4) Approval: Mandatory rules: approval by an engineer independent of the author (DAL A/B, Class 1/2) and by the project quality representative. Recommended rules: acceptance during peer review.
- 5) Recording: the deviation is recorded in the project deviation log (HDD annex for aeronautics, design justification file / NCR-RFW system for space). The log is under configuration control (HC1 / CC1).
- 6) Verification: the compensating measure is verified and the evidence is referenced in the deviation log.

| Level       | Mandatory rule                                                                                                           | Recommended rule       | Record                                |
|-------------|--------------------------------------------------------------------------------------------------------------------------|------------------------|---------------------------------------|
| DAL A / B   | Independent reviewer + quality; reported in the HAS when it affects compliance with the design standard                  | Peer review acceptance | Deviation log (HDD), HC1              |
| DAL C       | Reviewer + quality                                                                                                       | Peer review            | Deviation log, HC2                    |
| DAL D       | Technical lead                                                                                                           | None                   | Code comment                          |
| Class 1 / 2 | Independent reviewer + PA; customer agreement (RFD / RFW per ECSS-Q-ST-10-09) when the rule is a contractual requirement | Peer review acceptance | Design justification file, NCR system |
| Class 3     | Technical lead                                                                                                           | None                   | Code comment                          |

Table 13: Deviation approval per level

### COMPLIANT Deviation marking in the code

```
-- DEVIATION [LNG.06] DEV-PRJX-0012 : for loop on a static range (barrel shifter)
-- Justification : 32 x 5-bit shift stages; a generate-based description would
-- need 160 named intermediate signals and be less reviewable.
-- Unrolled structure : s_Shift(k) <= s_Shift(k-1) rotated by 2**k when i_Amount(k) = '1'.
-- Compensation : bounds are generics checked by assertion; exhaustive test of
-- i_Amount (TC-SHF-001), equivalence check of the netlist.
proc_Shift : process(i_Clk)
```

## 5.7. Project tailoring

At the start of each project, the applicable column (DAL or class) is selected and recorded in the PHAC / device development plan. The project may only strengthen the rules (e.g. make a Recommended rule Mandatory). Any relaxation of a Mandatory rule for the whole project is a project-level deviation following 5.6 and shall be agreed with the certification authority (through the PHAC) or the customer (through the development plan). The compliance matrix template of [Annex D](#) is used to record the tailoring.

## 6. Design philosophy for critical FPGA designs

Before any coding rule, this chapter states the design principles from which all the rules derive. A rule can always be traced back to one of these principles; when a situation is not covered by a rule, the principles decide.

### 6.1. The ten principles

| #  | Principle                   | Consequence on the code                                                                                                                                                                   |
|----|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | Synchronous by construction | One active clock edge per domain, all state held in flip-flops, no latch, no combinational loop, no gated or derived clock, asynchronous inputs synchronized at the boundary.             |
| 2  | Deterministic               | The behavior does not depend on routing delays, on the order of evaluation of the simulator, on the power-up state of uninitialized flip-flops, nor on synthesis optimizations.           |
| 3  | Simple and explicit         | Each process does one thing, each signal has one meaning, each construct is the most explicit one (no implicit latch, no implicit priority, no implicit conversion, no implicit default). |
| 4  | Defensive                   | Every register can recover from any value (illegal FSM state, counter beyond its limit, invalid command); abnormal inputs are detected and reported; failures are never silent.           |
| 5  | Verifiable                  | Every line can be reached and observed; unreachable robustness code is identified and verified by fault injection; the code allows 100 % structural coverage.                             |
| 6  | Reviewable                  | An independent engineer can understand a 450-line architecture in less than one hour: naming conventions, headers, comments, limited complexity, no "clever" code.                        |
| 7  | Traceable                   | Every function implements a requirement (or a derived requirement fed back to the safety process); every requirement is implemented and verified.                                         |
| 8  | Technology independent      | Vendor primitives are isolated in wrappers; the functional RTL can be retargeted from one FPGA family to another without modification.                                                    |
| 9  | Reuse of proven blocks      | CDC, reset, EDAC and TMR structures come from the verified reference library, never re-invented.                                                                                          |
| 10 | Radiation-aware (space)     | The design assumes that any flip-flop or memory bit can flip at any time and defines how each upset is masked, corrected or detected.                                                     |

Table 14: Design principles

### 6.2. General rules

#### [GEN.01] Fully synchronous design

|                     |                                                                                    |
|---------------------|------------------------------------------------------------------------------------|
| <b>M: Mandatory</b> | DAL A: M B: M C: M D: M   Space Class 1: M 2: M 3: M                               |
| <b>Basis</b>        | Company rule                                                                       |
| <b>Verification</b> | Linters, synthesis report (no latch, no combinational loop), CDC tool, peer review |

**Rule:** The design shall be fully synchronous: every storage element is an edge-triggered flip-flop or an embedded memory clocked by a clock net coming from the clock manager; all flip-flops of a clock domain use the rising edge; no signal other than a clock drives a clock pin.

#### Rationale

Static timing analysis (STA) can only prove the timing of synchronous paths. Any asynchronous structure (latch, combinational feedback, pulse generator based on gate delays, signal used as clock) creates paths that STA does not analyze or analyses wrongly. Their behavior then depends on temperature, voltage, process and routing, i.e. it is not deterministic and cannot be verified by simulation.

AMC 20-152A §5.2 / AC 20-152A objective CD-1 lists "synchronous vs asynchronous design" among the criteria of complexity; a fully synchronous design is a pre-requisite to claim that its timing has been comprehensively verified (CD-7).

### Detailed requirements

- Only the `rising_edge()` function is used to detect the active clock edge in RTL (no `'event`, no `falling_edge()` except in an approved DDR I/O wrapper).
- No combinational path from a top-level input to a top-level output (see [TOP.05]).
- The only asynchronous constructs allowed are those of the reference library: reset manager (`rst_mgr`), synchronizers (`sync_bit`, `sync_pulse_toggle`, `cdc_event_hs`) and dual-clock FIFOs (`fifo_async_wrp`).

#### NON-COMPLIANT *asynchronous structures*

```
-- One-shot built from gate delays : width depends on routing, temperature, voltage
s_Pulse <= i_Trig and not s_Trig_Delayed;
s_Trig_Delayed <= not (not (not i_Trig));      -- "delay line" : removed by synthesis
-- Signal used as a clock
proc_Count : process(i_Trig)
begin
    if rising_edge(i_Trig) then                -- i_Trig is NOT a clock
        s_Count <= s_Count + 1;
    end if;
end process proc_Count;
```

#### COMPLIANT *synchronous equivalent*

```
-- Same functions, synchronous : i_Trig sampled, edge detected with flip-flops
inst_Sync_Trig : entity work.sync_bit
generic map (
    G_STAGES => 2,
    G_RESET_VAL => '0'
)
port map (
    i_Clk => i_Clk,
    i_Rst_n => i_Rst_n,
    i_Async => i_Trig,
    o_Sync => s_Trig_Sync
);

inst_Edge_Trig : entity work.edge_detect
port map (
    i_Clk => i_Clk,
    i_Rst_n => i_Rst_n,
    i_Sig => s_Trig_Sync,
    o_Re => s_Trig_Re,
    o_Fe => open
);

proc_Count : process(i_Clk)
begin
    if rising_edge(i_Clk) then
        if i_Rst_n = '0' then
            s_Count <= (others => '0');
        elsif s_Trig_Re = '1' then
            s_Count <= s_Count + 1;
        end if;
    end if;
end process proc_Count;
```

## [GEN.02] Deterministic behavior

|                     |                                                                                                       |
|---------------------|-------------------------------------------------------------------------------------------------------|
| <b>M: Mandatory</b> | DAL A: <b>M</b> B: <b>M</b> C: <b>M</b> D: <b>M</b>   Space Class 1: <b>M</b> 2: <b>M</b> 3: <b>M</b> |
| <b>Basis</b>        | Company policy                                                                                        |
| <b>Verification</b> | Peer review, gate-level / post-layout simulation, coverage analysis                                   |

**Rule:** The behavior of the design shall be fully defined by the RTL and its requirements: it shall not depend on propagation delays, on simulation event ordering (delta cycles), on the power-up content of flip-flops or memories, on optimizations performed by the synthesis tool, nor on the relative frequency or phase of asynchronous clocks beyond what is specified.

```

    end loop;
    s_Parity <= v_Par;
  end if;
end process proc_Parity;

```

#### COMPLIANT *concurrent reduction + register*

```

-- Explicit reduction : every gate visible, every term covered
s_Parity_Comb <= i_Data(0) xor i_Data(1) xor i_Data(2) xor i_Data(3) xor
                i_Data(4) xor i_Data(5) xor i_Data(6) xor i_Data(7);

proc_Parity : process(i_Clk)
begin
  if rising_edge(i_Clk) then
    if i_Rst_n = '0' then
      s_Parity <= '0';
    else
      s_Parity <= s_Parity_Comb;
    end if;
  end if;
end process proc_Parity;

```

#### COMPLIANT *generic reduction with generate*

```

-- Generic width : xor-chain described with a generate statement
s_Par_Chain(0) <= i_Data(0);
gen_Parity : for idx_Bit in 1 to G_WIDTH-1 generate
  s_Par_Chain(idx_Bit) <= s_Par_Chain(idx_Bit-1) xor i_Data(idx_Bit);
end generate gen_Parity;
s_Parity_Comb <= s_Par_Chain(G_WIDTH-1);

```

## 7.3. Forbidden and restricted constructs

### [LNG.06] Forbidden constructs in RTL

| M: Mandatory | DAL A: M B: M C: M D: M   Space Class 1: M 2: M 3: M |
|--------------|------------------------------------------------------|
| Basis        | Company rule                                         |
| Verification | Linters, peer review                                 |

**Rule:** The constructs of the table below shall not be used in synthesizable code.

#### Rationale

These constructs are either not synthesizable (and therefore create simulation / synthesis mismatches), or are synthesizable with tool-dependent semantics, or describe structures forbidden by other rules (tri-states, latches, asynchronous logic).

#### Detailed requirements

- A for loop in a clocked process may be accepted by deviation for regular shift structures when its bounds are static (constants or generics) and when the unrolled structure is described in a comment. generate remains the preferred solution.

| Construct                                                                 | Reason                                                                     | Replacement                                      |
|---------------------------------------------------------------------------|----------------------------------------------------------------------------|--------------------------------------------------|
| wait statement, after clause                                              | Not synthesizable; ignored or rejected by synthesis                        | Clocked processes, counters                      |
| 'event, 'stable, 'last_event, 'last_value, 'delayed, 'quiet, 'transaction | Simulation semantics only (except 'event which is replaced by rising_edge) | rising_edge(clk)                                 |
| falling_edge() in RTL                                                     | Second clock edge ([CLK.04])                                               | Rising edge only; DDR I/O primitives in wrappers |
| buffer and linkage port modes                                             | Tool dependent, forbids reuse                                              | Internal signal copied to an out port            |
| inout ports below the top level, 'Z' assignment                           | Internal tri-states do not exist in FPGA fabric                            | Multiplexers ([TOP.09])                          |

| Construct                                                             | Reason                                                        | Replacement                                                                     |
|-----------------------------------------------------------------------|---------------------------------------------------------------|---------------------------------------------------------------------------------|
| Guarded blocks, bus, register, disconnect                             | Obsolete, poorly supported                                    | Processes                                                                       |
| while loops, loop without iteration scheme, exit, next                | Data-dependent iteration count, infinite loop risk            | generate, counters                                                              |
| for loops in processes                                                | Unrolled hardware not visible, coverage aggregated            | generate, vector operations (deviation possible, see note)                      |
| null statement in a clocked process branch                            | Hides an implicit "hold"; usually a forgotten case            | Explicit assignment of the current value or no branch                           |
| Signal initial value different from its reset value                   | Power-up and post-reset behaviors differ                      | [SIG.03]                                                                        |
| Multiple architectures for one entity, configurations                 | Ambiguous binding                                             | One architecture per entity ([FIL.02]); technology selected by the compile list |
| alias of a signal                                                     | Two names for one object, confusing in waveforms and coverage | Use the original name                                                           |
| Physical types, real, time, access, file types                        | Not synthesizable                                             | Integer constants in cycles                                                     |
| translate_off / translate_on, synthesis_off pragmas                   | Simulation / synthesis mismatch by construction               | Allowed only around assert/report with an approved deviation                    |
| Implicit latch (incomplete assignment in a combinational description) | Asynchronous storage                                          | [CMB.01] makes it impossible                                                    |

Table 18: Constructs forbidden in RTL

**[LNG.07] Assertions in RTL limited to elaboration checks**

|                       |                                                                                                       |
|-----------------------|-------------------------------------------------------------------------------------------------------|
| <b>R: Recommended</b> | DAL A: <b>R</b> B: <b>R</b> C: <b>R</b> D: <b>O</b>   Space Class 1: <b>R</b> 2: <b>R</b> 3: <b>O</b> |
| <b>Basis</b>          | Company rule                                                                                          |
| <b>Verification</b>   | Peer review                                                                                           |

**Rule:** Concurrent assert statements may be used in RTL to check generics and constant relationships at elaboration (severity failure). Assertions that check dynamic behavior should be placed in testbenches or bound verification units, not in the RTL.

**Rationale**

Elaboration checks are evaluated by simulators and by most synthesizers and prevent silent misuse of a generic block (e.g. G\_STAGES = 1 for a synchronizer). Dynamic assertions in RTL are ignored by synthesis, may create coverage holes and clutter the design.

**COMPLIANT** *sync\_bit.vhd (excerpt)* [analyzed with ModelSim]

```
assert G_STAGES >= 2
report "sync_bit: G_STAGES shall be >= 2"
severity failure;
```

## 8. File organization

### 8.1. Files and design units

#### [FIL.01] File extension

|                     |                                                                                                       |
|---------------------|-------------------------------------------------------------------------------------------------------|
| <b>M: Mandatory</b> | DAL A: <b>M</b> B: <b>M</b> C: <b>M</b> D: <b>R</b>   Space Class 1: <b>M</b> 2: <b>M</b> 3: <b>R</b> |
| <b>Basis</b>        | Company rule                                                                                          |
| <b>Verification</b> | Build scripts                                                                                         |

**Rule:** VHDL source files shall have the extension `.vhd`. The extensions `.vhd1` and `.vho` (except for vendor-generated netlists) are not used.

#### Rationale

A single extension simplifies scripts, linting and configuration management.

#### [FIL.02] One design unit pair per file, file named after the unit

|                     |                                                                                                       |
|---------------------|-------------------------------------------------------------------------------------------------------|
| <b>M: Mandatory</b> | DAL A: <b>M</b> B: <b>M</b> C: <b>M</b> D: <b>M</b>   Space Class 1: <b>M</b> 2: <b>M</b> 3: <b>M</b> |
| <b>Basis</b>        | Company rule                                                                                          |
| <b>Verification</b> | Linters, build scripts                                                                                |

**Rule:** Each file shall contain exactly one entity and its unique architecture, or exactly one package (declaration and body), or one testbench. The file name shall be the name of the design unit in lower case: `<entity>.vhd`, `pkg_<name>.vhd`, `tb_<entity>.vhd`.

#### Rationale

One unit per file removes compilation-order ambiguities, makes the configuration of each unit traceable individually, and allows reviewers and tools to find a unit from its name. A single architecture per entity removes any ambiguity of binding between simulation and synthesis.

#### Detailed requirements

- Package names start with `pkg_` (prefix rule [NAM.01]).
- Testbench names start with `tb_`.
- Technology wrappers have one file per technology, in separate directories, with the same entity name; the build script selects the directory ([TEC.02]).

#### [FIL.03] Project directory structure

|                       |                                                                                                       |
|-----------------------|-------------------------------------------------------------------------------------------------------|
| <b>R: Recommended</b> | DAL A: <b>R</b> B: <b>R</b> C: <b>R</b> D: <b>O</b>   Space Class 1: <b>R</b> 2: <b>R</b> 3: <b>O</b> |
| <b>Basis</b>          | Company rule                                                                                          |
| <b>Verification</b>   | Inspection                                                                                            |

**Rule:** The design database should follow the structure of the table below.

#### Rationale

A fixed structure lets any engineer, reviewer or auditor find the configuration items immediately and allows generic build and lint scripts.

| Directory                                                                             | Content                                                                                                                     |
|---------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| <code>rtl/</code>                                                                     | Technology-independent synthesizable VHDL (one sub-directory per functional block if needed)                                |
| <code>rtl/pkg/</code>                                                                 | Project packages <code>pkg_*.vhd</code>                                                                                     |
| <code>tech_xilinx/</code> , <code>tech_microchip/</code> , <code>tech_lattice/</code> | Technology wrappers (same entity names in each directory) and generated IP (in <code>ip/</code> sub-directories, untouched) |

## 12. Types, vectors and arithmetic

### [TYP.01] Allowed types

|                     |                                                                                                       |
|---------------------|-------------------------------------------------------------------------------------------------------|
| <b>M: Mandatory</b> | DAL A: <b>M</b> B: <b>M</b> C: <b>M</b> D: <b>M</b>   Space Class 1: <b>M</b> 2: <b>M</b> 3: <b>M</b> |
| <b>Basis</b>        | Company rule                                                                                          |
| <b>Verification</b> | Linters                                                                                               |

**Rule:** Synthesizable code shall only use the following types: `std_logic`, `std_logic_vector`, `unsigned`, `signed`, `integer` / `natural` / `positive` with range (indices, generics, constants), `boolean` (generics and constants only), enumerated types (FSM states only), and records or arrays (maximum two dimensions) of these types.

#### Rationale

These types have an unambiguous hardware mapping. `bit`, `bit_vector`, `std_ulogic_vector` mixed with `std_logic_vector`, `real`, `time`, `physical` and access types are either simulation-only or create conversion noise. Boolean signals are not reset-friendly and do not show meta-values in simulation.

#### Detailed requirements

- `std_logic` (resolved) is used rather than `std_ulogic` for consistency with vendor libraries and IP cores.
- `integer` types are not used for ports of synthesizable entities (vectors are), except for generics.

### [TYP.02] Top-level port types

|                     |                                                                                                       |
|---------------------|-------------------------------------------------------------------------------------------------------|
| <b>M: Mandatory</b> | DAL A: <b>M</b> B: <b>M</b> C: <b>M</b> D: <b>M</b>   Space Class 1: <b>M</b> 2: <b>M</b> 3: <b>M</b> |
| <b>Basis</b>        | Company rule                                                                                          |
| <b>Verification</b> | Linters                                                                                               |

**Rule:** Ports of the top-level entity shall be of type `std_logic` or `std_logic_vector` only. Records, integers, enumerated types and user-defined arrays are forbidden on the top level.

#### Rationale

Top-level ports map to physical pins: the pin constraint file, the post-layout netlist and the board-level simulation models must see the same scalar names. Records and integers are flattened differently by each tool.

### [TYP.03] Integer types constrained

|                     |                                                                                                       |
|---------------------|-------------------------------------------------------------------------------------------------------|
| <b>M: Mandatory</b> | DAL A: <b>M</b> B: <b>M</b> C: <b>M</b> D: <b>M</b>   Space Class 1: <b>M</b> 2: <b>M</b> 3: <b>M</b> |
| <b>Basis</b>        | Company rule                                                                                          |
| <b>Verification</b> | Linters                                                                                               |

**Rule:** Every integer signal, constant or generic shall be constrained with a range (or be of subtype `natural` / `positive` when the full range is really intended for generics). Arithmetic signals should use `unsigned` / `signed` rather than integers.

#### Rationale

An unconstrained integer is synthesized on 32 bits; its overflow is an error in simulation (bound check failure) but a silent wrap-around in hardware, a simulation / hardware mismatch. `unsigned` / `signed` have the same wrap-around behavior in both.

#### NON-COMPLIANT

```
signal s_Count : integer;           -- 32 bits, simulation stops at overflow
```

#### COMPLIANT

```
signal s_Index : integer range 0 to c_Nb_Channels-1 := 0;    -- channel index
signal s_Count : unsigned(7 downto 0) := (others => '0');    -- frame counter, wraps at 255
```