EUROPEAN AVIATION SAFETY AGENCY
AGENCE EUROPÉENNE DE LA SÉCURITÉ AÉRIENNE
EUROPÄISCHE AGENTUR FÜR FLUGSICHERHEIT

Research Project EASA.2011/3

SHARDELD - Safety Implications from the use of hardware design tools for programmable Airborne Electronic Hardware Items

easa.europa.eu

Disclaimer

This study has been carried out for the European Aviation Safety Agency by an external organization and expresses the opinion of the organization undertaking the study. It is provided for information purposes only and the views expressed in the study have not been adopted, endorsed or in any way approved by the European Aviation Safety Agency. Consequently it should not be relied upon as a statement, as any form of warranty, representation, undertaking, contractual, or other commitment binding in law upon the European Aviation Safety Agency.

SHARDELD

Safety implications from the use of HARdware Development tools for programmable airborne ELectronic harDware

Final Study Report

SUMMARY:

This final study report describes the activities and conclusions of EASA's SHARDELD study to identify the most relevant commercial CAE tools which are used for the development of programmable AEH, with the purpose of recording their advantages and limitations, safety benefits and risks, and elaborating a comprehensive list of best practices, recommendations and guidelines in order to reduce the safety risks while keeping their effectiveness.

Table of Contents

Index of Tables

Index of Figures

1. Acknowledgements

IOxOS Technologies wishes to express its gratitude and appreciation to EASA for its trust and support throughout this study. EASA's contribution was essential for the consultations that took place with aircraft and equipment manufacturers and which are one of the core activities of the SHARDELD study.

We also want to thank PLD vendors and CAE tool providers for their support and continuous efforts to attain excellence, which is reflected in the comprehensive documentation, proactive technical discussion forums and above all, the permanent evolution of all their products and solutions.

IOxOS Technologies would also like to extend its appreciation to all the aircraft and equipment manufacturers who dedicated their precious time to answer a questionnaire which was longer than expected. Their answers were the cornerstone of the SHARDELD study.

2. Introduction

2.1 Purpose

This final study report describes the activities and conclusions of EASA's SHARDELD study to identify the most relevant commercial CAE tools which are used for the development of programmable AEH, with the purpose of recording their advantages and limitations, safety benefits and risks, and elaborating a comprehensive list of best practices, recommendations and guidelines in order to reduce the safety risks while keeping their effectiveness.

2.2 Scope

This final study report aims to:

- Give a brief introduction of the SHARDELD study, its context and objectives
- Describe the activities performed by IOxOS Technologies to accomplish the following tasks indicated in the study:
  - ✔ "Selection of types of tools needing assessment"
  - ✔ "Selection of commercial software tools and their generic assessment", including a consultation of key aircraft and equipment manufacturers developing programmable AEH
  - ✔ "Design tools usage assessment"
- Summarize the conclusions of these activities

2.3 Acronyms and Abbreviations

Acronyms & Abbreviations	
Acronym	***Definition***
ABV	Assertion-Based Verification
AEH	Airborne Electronic Hardware
ASIC	Application-Specific Integrated Circuit
BFM	Bus Functional Model
CAE	Computer-Assisted Engineering
CCPP	Common Clock Path Pessimism
CDC	Clock Domain Crossing
CLB	Configurable Logic Block
CPLD	Complex Programmable Gate Array
CRC	Cyclic Redundancy code

Acronyms & Abbreviations	
CVS	Concurrent Versions System
DAL	Design Assurance Level
DCM	Digital Clock Manager
DSP	Digital Signal Processing
DUV	Design Under Verification
EASA	European Aviation Safety Agency
ECC	Error Correction Coding
ECO	Engineering Change Order
EDA	Electronic Design Automation
FAE	Field Application Engineer
FF	Flip Flop
FIT	Failures In Time
FPGA	Field Programmable Gate Array
FSM	Finite state Machine
GUI	Graphical User Interface
HDL	Hardware Description Language
HVP	Hardware Verification Plan
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
IP	Intellectual Property
JTAG	Joint Test Action Group
LAB	Logic Array Block
LEC	Logic Equivalence Checking
LUT	LookUp Table
MBD	Model-Based Design
MTBF	Mean Time Between Failures
OTP	One-Time Programmable
OVA	Open Vera Assertions
OVL	Open Verification Library
PAR	Place and Route
PDC	Physical Design Constraints
PHAC	Plan for Hardware Aspects of Certification
PLD	Programmable Logic Device
PLL	Phase-Locked Loop
PSL	Property Specification Language

Acronyms & Abbreviations	
PVT	Process-Voltage-Temperature
RTL	Register Transfer Level
SDC	Synopsys Design Constraints
SEU	Single Event Upset
SSN	Simultaneous Switching Noise
STA	Static Timing Analysis
SVA	System Verilog Assertions
SVN	Subversion (Version Control System)
Tcl	Tool Command Language
TMR	Triple Modular Redundancy
UCF	User Constraint File
VCD	Value Change Dump
XCF	XST Constraint File

Table 2.1. Acronyms and Abbreviations

2.4 Applicable Documents

2.4.1 External Documents

External Documents		
Reference	***Document***	***Issue/Rev.***
[EXT-01]	Specifications attached to the Invitation to Negotiate EASA.2011.NP.33	
[EXT-02]	EASA CM-SWCEH-001, Development Assurance of Airborne Electronic Hardware	Issue 01 August 11, 2011
[EXT-03]	EUROCAE ED-80, Design Assurance Guidance for Airborne Electronic Hardware	April 19, 2000
[EXT-04]	VHDL Synthesis for High-Reliability Systems 2004 MAPLD International Conference	September 7, 2004
[EXT-05]	Altera's Quartus II Handbook Volume 2: Design Implementation and Optimization	Version 12.0 June 2012
[EXT-06]	Actel Application Note AC23	April 2004

Table 2.2. Applicable External Documents

2.4.2 Internal Documents

Internal Documents		
Reference	***Document***	***Issue***
[INT-01]	IOxOS Technologies Hardware Design Standards	Feb 18, 2010 v.0.1.0

Table 2.3. Applicable Internal Documents

3. SHARDELD Objectives and Tasks

3.1 SHARDELD Object and Scope

Custom micro-coded components such as FPGAs, CPLDs and Structured ASICs are gaining acceptance in today's aeronautical industry due to their performance, embedded features and flexibility.

The development of programmable Airborne Electronic Hardware (AEH) using these components for safety critical applications requires a design flow involving commercial Computer-Assisted Engineering (CAE) tools with a high degree of complexity.

The way these tools are used for both design and verification activities may have an important impact in terms of safety.

The object of the SHARDELD study is to identify the most relevant commercial CAE tools which are used for the development of programmable AEH, with the purpose of recording their advantages and limitations, safety benefits and risks, and elaborating a comprehensive list of best practices, recommendations and guidelines in order to reduce the safety risks while keeping their effectiveness.

This study is not intended to replace good design techniques; it focuses on how tools should be used to increase reliability. No development tool can be as efficient as proper design style.

Within the framework of the SHARDELD study, programmable AEH applies to the following custom micro-coded components or PLD (Programmable Logic Devices): CPLDs, FPGAs and structured ASICs.

3.2 SHARDELD Tasks

The following sections summarize the activities planned by IOxOS Technologies in order to accomplish the tasks indicated in the study, which are described in the Specifications attached to the Invitation to Negotiate EASA.2011.NP.33 [EXT-01].

3.2.1 Task 1: Selection of Types of Tools Needing Assessment

The first task aims to identify the types of tools used for the development of programmable AEH, as well as to determine which types of tools should be covered by the study on the basis of complexity and/or configuration criteria.

To accomplish this task the following activities are planned:

1. Identification of the types of tools used for the development of programmable AEH: Two main types are identified as key tools within any design flow:
   - Design tools
   - Verification tools
2. Evaluation of alternative and/or complementary types of tools such as:
   - HDL Rule Checkers (also known as linters)

- Clock Domain Crossing (CDC) analysers
- Logical Equivalence Checking (LEC) tools
- Assertion-Based Verification (ABV) tools

3. Selection of the types of tools to be reviewed in the framework of the study after evaluating their complexity in terms of functional features and configuration. IOxOS Technologies experience with most of these tools, together with the feedback from PLD tool vendors and FAEs, is essential to perform this activity, which will be the backbone of the study

IOxOS Technologies Hardware Design Standards already includes guidelines for the design tools identified in the first activity -synthesis and place and route tools- which reinforces the fact that these type of tools should be considered for further assessment.

3.2.2 Task 2: Survey Available Commercial Tools

The purpose of this task is to identify the tools available on the market, within the types of tools selected in the previous task, which are relevant for the study. It also aims to assess these tools in order to determine their limitations, benefits, and the different methods followed to fulfil ED-80/DO-254 objectives in terms of tool assessment and qualification.

The following activities will be carried out:

1. Identification of the custom micro-coded components (types and vendors) which are used as programmable AEH: This activity helps to be more selective when considering tools provided by PLD vendors
2. Commercial tools survey: Based on IOxOS Technologies own experience and feedback from PLD tool vendors and FAEs
3. Aircraft and equipment manufacturers consultation: IOxOS Technologies close contact with aircraft and equipment manufacturers together with the support of EASA, will be useful when carrying out this consultation
4. Assessment of the selected commercial tools, in order to provide a comparison table which contains detailed information on the following aspects:
   - Tools limitations due to functional features and configuration options
   - Technical features and benefits
   - Tool complexity
   - Known technical issues and their impact on safety
   - Integration within design life cycles requiring compliance with ED-80/DO-254 guidance:
     - ✔ Data availability from tool vendors (problem reports)
     - ✔ Relevant service experience
     - ✔ Tool assessment and qualification

   This activity is based in IOxOS Technologies experience with design tools, as well as on additional research tasks and feedback from PLD tool vendors and FAEs

3.2.3 Task 3: Design Tools Usage Assessment

This task aims to assess the effects of different configuration options and functionalities of the tools selected in the previous task.

The following activities are planned:

1. Assessment of different configuration options for each selected commercial CAE tool: Benefits and risks in terms of safety
2. Assessment of embedded functionalities for each selected commercial CAE tool: Benefits and risks in terms of safety
3. Elaboration of a comprehensive list of best practices, recommendations and guidelines in order to maximize the tools effectiveness while reducing the safety risks
4. Develop recommendations to amend the EASA Certification Memorandum "Development Assurance of Airborne Electronic Hardware" CM-SWCEH-001 [EXT-02]

4. Task 1: Selection of Type of Tools Needing Assessment

The first task of the SHARDELD study aims to identify the types of tools used for the development of programmable AEH. It also aims to ascertain which types should be subject to a more detailed assessment on the basis of complexity and/or configuration criteria.

The following sections describe the activities performed in order to carry out this task.

4.1 Identification of the types of tools used for the development of programmable AEH:

Figure 4.1 illustrates the design life cycle for the development of AEH, including the mapping of the main processes with the corresponding sections described in ED-80/DO-254 and the usual tool flow. The purpose of the SHARDELD study is to cover the tools which can be used in the following development processes:

- Conceptual design: As stated in ED-80/DO-254 §5.2, *"The conceptual design process produces a high-level design concept that may be assessed to determine the potential for the resulting design implementation to meet the requirements. This may be accomplished using such items as functional block diagrams, design and architecture descriptions, circuit card assembly outlines, and chassis sketches"*
- Detailed design: Mentioned in ED-80/DO-254 §5.3, *"The detailed design process produces detailed design data using the hardware item specification and conceptual design as the basis for the detailed design"*
- Validation and Verification: Defined in ED-80/DO-254 §6 as follows, *"The validation process provides assurance that the hardware item derived requirements are correct and complete with respect to system requirements allocated to the hardware item. The verification process provides assurance that the hardware item implementation meets all of the hardware requirements, including derived requirements"*

Consequently, the tools used for the development of programmable AEH can be divided into two main groups:

- Design tools: CAE tools used to perform the tasks concerning both conceptual and detailed design processes. A brief description of the main tools is provided in § 4.1.1 of this interim report
- Verification tools: These tools may be used for the purpose of validation and verification. This group can also be divided into two categories, depending on the verification approach:
  - ✔ Dynamic verification tools: This approach is based on the generation of test vectors to exercise the design under verification (DUV). HDL simulators are key tools to carry out this process (refer to § 4.1.2.1)
  - ✔ Static verification tools: Static verification makes a comprehensive analysis of the DUV behaviour using logic and discrete mathematics, with the aim of specifying and designing accurate simulation models for analysis purposes (refer to § 4.1.2.2)

§ 5.2 identifies and describes the verification tools considered alternative and/or complementary, including formal verification methods. The latter are identified in ED-80/DO-254 Appendix B, "Design assurance considerations for level A and B functions" §3.3, as advanced verification methods.

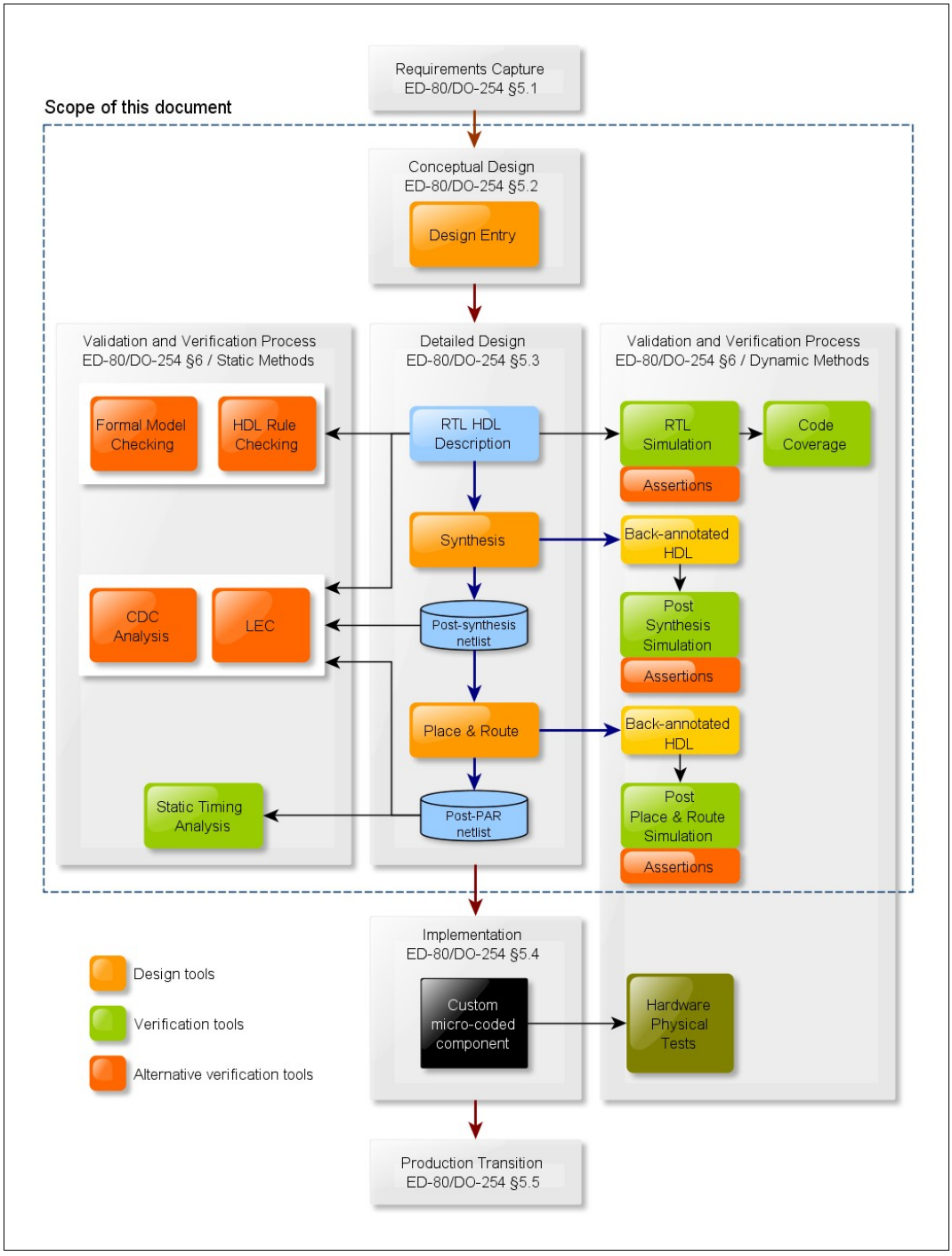

Figure 4.1. AEH Design Life Cycle and Typical Development Tool Flow

4.1.1 Design Tools

Design tools generate the detailed design data which produces the hardware item. In the framework of programmable AEH, design entry editors, synthesis and place and route tools, are identified in any design flow. Detailed design data usually corresponds to an RTL (Register Transfer Level) HDL (Hardware Description Language) description, user-defined timing and physical constraints files, and scripts for controlling the automated transformations.

4.1.1.1 Design Entry Tools

4.1.1.1.1 Introduction to Design Entry

Design entry tools which may be used during the conceptual design process range from text editors -for handwritten HDL code- to graphical editors or modelling languages to automatically generate the HDL high-level description of the design.

4.1.1.1.2 Design Entry Tools

Standard text editors may be used to produce handwritten HDL code. HDL templates are usually available for most of text editors.

Graphical design editors combines different methods of design entry, such as block diagrams, flow charts, state diagrams for Finite State Machine (FSM) description, truth tables and HDL code.

Design entry can also be done following a Model-Based Design (MBD) approach. This methodology uses high-level modelling languages -such as Matlab/Simulink- to automatically generate HDL code.

The output of this type of tool is the RTL HDL description, directly handwritten or automatically generated from a high-level graphical description or modelling language, using the hardware requirements as input.

4.1.1.1.3 Tool Assessment and Qualification

As stated in ED-80/DO-254 §11.4 *"When design tools are used to generate the hardware item or the hardware design, an error in the tool could introduce an error in the hardware item […] Prior to the use of a tool, a tool assessment should be performed. The results of this assessment and, if necessary, tool qualification should be recorded and maintained"*.

Tool assessment is mandatory for design tools used in the development of hardware items with Design Assurance Level (DAL) A, B and C. Additionally, ED-80/DO-254 §11.4.1 provides a flow chart indicating tool assessment considerations and activities, together with guidance to determine when tool qualification is needed. The first recommended approach is an independent assessment of the tool output.

In addition, CM-SWCEH-001 §8.4.2.1 item (i) establishes that *"If a Hardware Description Language (HDL) is used, an HDL code review against the conceptual design and requirements should be performed"*.

Consequently, the output of any design entry tool -which is the RTL HDL description- should be independently reviewed against the hardware requirements and also with the purpose of establishing that the coding rules specified in the Hardware Design Standards have been followed. These reviews make no further tool assessment necessary.

4.1.1.1.4 Configuration Management

As set out in ED-80/DO-254 §7, *"The configuration management process is intended to provide the ability to consistently replicate the configuration item, regenerate the information if necessary and modify the configuration item in a controlled fashion if modification is necessary"*.

In order to fulfill the configuration management objectives, all the files used by the tool to produce its output -the RTL HDL description- should be subject to revision control, using software tools such as CVS or SVN among others, as part of the configuration management process. The design entry tool version also needs to be tracked, complying with ED-80/DO-254 §5 statement: *"The design representation should allow the hardware item to be consistently replicated"*.

4.1.1.2 Synthesis Tools

4.1.1.2.1 Introduction to Synthesis Process

Synthesis is the process which translates the RTL HDL description into a network of logic elements. This process can also be considered as a logic synthesis to distinguish between this function and the physical synthesis, which is a complementary advanced optimization method for timing closure purposes.

Synthesis is a mandatory process for the development of programmable AEH.

4.1.1.2.2 Synthesis Tools

The CAE tool that performs the logic synthesis may be provided by third party EDA (Electronic Design Automation) tool manufacturers or by the PLD vendor as part of an Integrated Development Environment (IDE).

The synthesis tool takes as inputs the high-level HDL description together with the user-defined timing constraint file. The RTL HDL description may contain dedicated attributes which are interpreted by the synthesis tool as directives to improve area and/or timing performance, and keep control of the automated process.

The outputs of the synthesis tool are a gate level netlist together with a comprehensive synthesis report.

The tool can also generate a back-annotated HDL model for post-synthesis simulation.

4.1.1.2.3 Additional Functions of Synthesis Tools

Some synthesis tools offer additional functions to enhance the synthesis process, such as:

- Graphical edition for FSM inspection and edition

- Triple Modular Redundancy (TMR) to mitigate Single Event Upset (SEU) phenomena. This method automatically implements each register in triplicate, together with a vote-based mechanism to determine the register's true state. This additional logic is generally very difficult to verify
- FSM compilation for automated FSM extraction and optimization
- Incremental synthesis: It is part of the incremental design methodology which aims to reduce the synthesis time on high-density designs preserving the synthesis results of unchanged logic. This feature also targets modular design allowing multiple designers to work independently on parts of the same design
- Optimization methods to improve area and/or timing performance: Resource sharing, retiming, pipelining, and register duplication among others

4.1.1.2.4 Tool Assessment and Qualification

Regarding tool assessment and qualification, synthesis tools are considered design tools. The independent assessment of the tool output can be done by combining different methods:

- Post-synthesis simulation of the back-annotated HDL model to be matched with the results of the RTL functional simulation
- Visual inspection of the synthesis output focusing on critical design items, such as reset logic, additional mitigation logic, and FSM implementation among others (the graphic viewer provided by the tool may lighten this task)
- Use of Logic Equivalence Checking (LEC) tools to formally prove that the netlist produced by the synthesis tool and the RTL HDL description have the same behaviour (§ 4.2.3 provides a more detailed description of LEC)

These methods aim to assess that no tool option will change the behaviour of the design, being in line with CM-SWCEH-001 §8.4.2.2 item (e) statement: *"An analysis of the process used to perform the synthesis, place and route should confirm that the verification of the device requirements demonstrates the behaviour of the implementation of the device"*.

4.1.1.2.5 Configuration Management

Most synthesis tools support script languages such as Tcl (Tool Command Language) for flow automation and tool configuration. The use of scripts also increases the traceability and reproducibility of the design process. Scripts can be subject to revision control (CVS, and SVN among others) as part of the configuration management process.

The further assessment of the selected synthesis tools will clearly identify the type of files to be subject to revision control.

Synthesis tool version should also be tracked together with the design technology libraries used for synthesis, complying with ED-80/DO-254 §5 statement: *"The design representation should allow the hardware item to be consistently replicated"*.

4.1.1.3 Place and Route Tools

4.1.1.3.1 Introduction to Place and Route Process

These tools include several processes where the elements of the netlist generated by the synthesis tool are mapped in the PLD physical resources (place) and connected together (route) to implement the specified function. The output of this process is a fully routed gate level netlist, which is used to generate the binary file to be downloaded in the programmable AEH.

Place and route are mandatory processes for the development of programmable AEH.

4.1.1.3.2 Place and Route Tools

Place and route tools are provided by the PLD vendor, frequently integrated into an IDE, which is a comprehensive set of tools covering both design and verification activities.

This type of tool uses the gate level netlist produced by the synthesis tool together with a user-defined file recording timing and placement constraints.

The outputs of the place and route tool are a fully routed gate level netlist, and several reports covering the different processes (translation, mapping, place and route, pad location and asynchronous delay reports among others).

Place and route tools include the function to generate the device programming file.

The tool can also generate a back-annotated HDL model with timing information (including internal routing delays) for post-place and route simulation.

4.1.1.3.3 Additional Functions of Place and Route Tools

Usually, place and route tools are provided by the PLD vendor as part of an IDE. These development environments provide additional functions to enhance the place and route process, which may include:

- Physical synthesis: This function takes as input the netlist generated by the logic synthesis tool and creates a new optimized netlist based on detailed placement and timing information, which is obtained after performing place and route and Static Timing Analysis (STA), in order to fulfill timing, area and routability requirements. This process may require several iterations before meeting its goals
- Incremental compilation: Same principle as incremental synthesis, reduce the compilation time on high-density designs preserving the compilation results of unchanged logic
- Floorplanning: Graphical tool which allows the designer to manually map selected parts of the design onto the target device. This function is recommended for high-density and/or timing critical designs

4.1.1.3.4 Tool Assessment and Qualification

In terms of tool assessment and qualification, place and route tools are considered design tools. The independent assessment of the tool output can be done by combining different

methods:

- Post-place and route simulation of the back-annotated HDL model to be matched with the results of the RTL functional simulation
- Visual inspection of the place and route output focusing on critical design items as with synthesis tools (the graphic viewer provided by the tool may lighten this task)
- Physical tests on the programmed device
- Use of LEC tools to formally prove that the netlist produced by the place and route tool and the RTL HDL description have the same behaviour (§ 4.2.3 provides a more detailed description of LEC)

These methods aim to assess that no tool option will change the behaviour of the design, being in line with CM-SWCEH-001 §8.4.2.2 item (e) statement: *"An analysis of the process used to perform the synthesis, place and route should confirm that the verification of the device requirements demonstrates the behaviour of the implementation of the device"*.

4.1.1.3.5 Configuration Management

As with synthesis tools, most place and route tools support script languages for flow automation and tool configuration. These scripts can be subject to revision control (CVS, and SVN among others) as part of the configuration management process.

The further assessment of the selected place and route tools will clearly identify the type of files to be subject to revision control.

Place and route tool version also needs to be tracked together with the design technology libraries used to carry out the place and route processes and STA, complying with ED-80/DO-254 §5 statement: *"The design representation should allow the hardware item to be consistently replicated"*.

4.1.2 Verification Tools

CAE tools used for the purpose of validation and verification. This group can be divided into two categories, depending on the verification approach: Dynamic and Static verification tools.

4.1.2.1 Dynamic Verification Approach

4.1.2.1.1 Introduction to Dynamic Verification

This approach is based on the generation of test vectors to exercise the DUV, which is instantiated in a testbench. The outputs recorded in log files and displayed in waveform viewers are matched against the expected results to verify the correctness of the design.

Figure 4.2 depicts a standard dynamic simulation environment. Bus Functional Models (BFM) are used within the testbench to emulate peripheral devices.

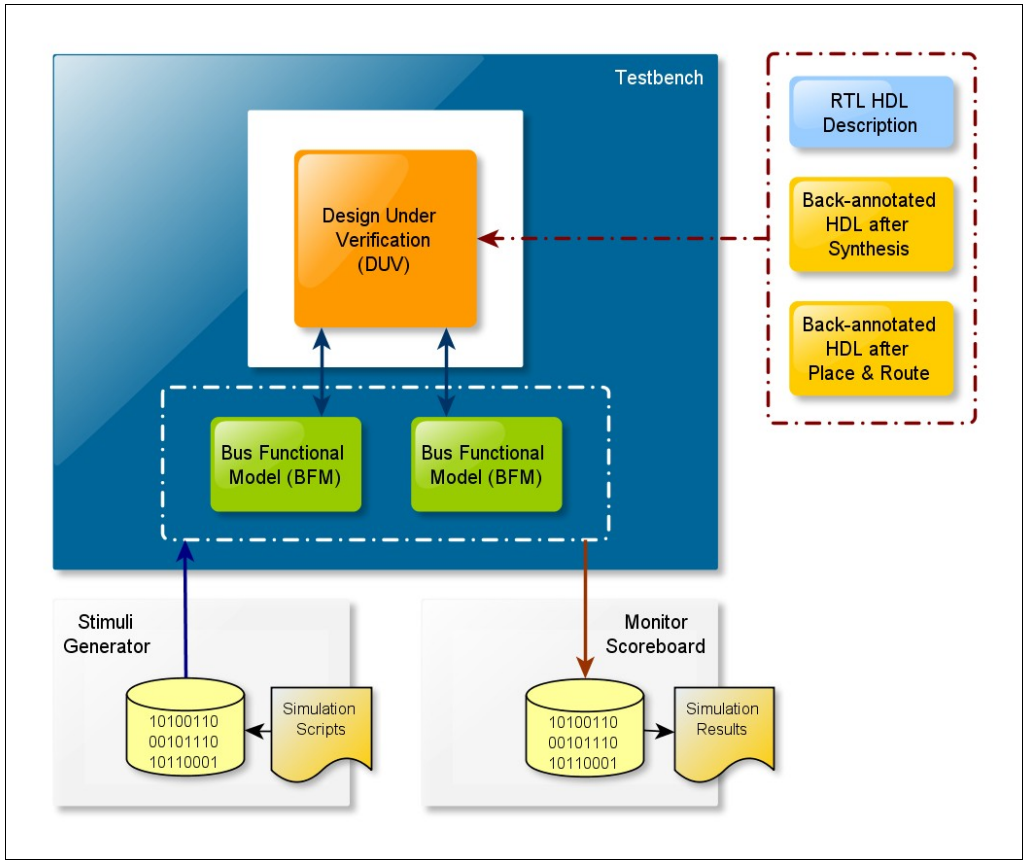

Figure 4.2. Dynamic Simulation Environment

In order to fulfill the verification objectives defined by ED-80/DO-254 §6.2.1, the test vectors should be produced by requirement-based test cases described in the test plan with clearly defined acceptance test criteria.

Functional simulation can be performed at different levels within the design flow depending on the format of the instantiated DUV while keeping almost the same simulation environment (with the exception of simulation libraries). This capability provides a verification mean to each level of the implementation, showing compliance with CM-SWCEH-001 §8.4.2.2 item (b) statement: *"The PHAC (or HVP) should define and justify for each level of implementation (Register Transfer Level – RTL, post layout, physical device, board level) the type of planned verification activity (test, simulation, analysis, inspection...)."*

The different simulations are:

- RTL simulation: The RTL HDL description is used as DUV. The whole design can be simulated (top level) or some functions can be isolated for unitary simulation (block or cluster level). Code coverage is performed by the HDL simulator at this level
- Post-Synthesis simulation: The DUV is the back-annotated HDL model produced by the synthesis tool. This simulation contributes to verify the consistency of the results obtained in the functional simulation while providing an independent assessment of the synthesis tool output
- Post-Place and route simulation: The back-annotated HDL model with additional routing delay timing information is used as DUV. This simulation level, which also contributes to assess the place and route tool output, is a time-consuming method, but it can be useful to verify the following design features:
  - ✔ Power up and reset operation
  - ✔ Critical asynchronous paths not covered by STA
  - ✔ User-defined timing constraints also used in STA
  - ✔ Design behaviour under temperature and voltage variations
  - ✔ Accuracy of models used in RTL simulation to emulate the vendor-specific macro functions
  - ✔ Operation of external interfaces, by using testbenches featuring accurate BFMs to emulate peripherals and physical parameters such as board signal delays

Functional simulation offers high visibility and accessibility to the DUV; It can provide an independent assessment of the design tools and it can also perform elemental analysis (code coverage after RTL simulation). The high degree of accessibility allows the designer to force internal signals for fault injection purposes, allowing the simulation of the DUV under abnormal conditions.

Functional simulation is well complemented with STA, which provides a comprehensive analysis of all paths within the design, highlighting critical paths and providing an accurate estimation of the design operating frequency. The combination of these two verification means is extensively used in programmable AEH design flows to prove the correctness of functionality and timing.

The use of assertions can increase the effectiveness of the functional simulation. § 4.2.4 describes the use of assertions in simulation, its advantages and its limitations.

4.1.2.1.2 Dynamic Verification Tools

HDL simulators are the type of tool used to perform this verification methodology. Most of them support Dynamic Assertion Based Verification (ABV) by using advanced HDL languages, such as System Verilog and specific assertion languages such as PSL (Property Specification Language), SVA (System Verilog Assertions) and OVA (Open Vera Assertions). § 4.2.4 provides an overview of ABV methodologies.

4.1.2.1.3 Elemental Analysis: Code Coverage

ED-80/DO-254 Appendix B, "Design assurance considerations for level A and B functions" §3.3, specifies several methods in order to add design assurance for most critical levels. Elemental analysis is an accepted method when developing programmable AEH. When HDL is used for the detailed design description, then code coverage is an acceptable means of elemental analysis.

In addition, CM-SWCEH-001 §8.4.2.1 item (g) states that: *"If a Hardware Description Language (HDL), as defined in ED-80/DO-254, is used, an HDL code coverage measurement is an acceptable means to assess the way the HDL code has been exercised during device functional verification by simulation. The HDL code coverage at sub-function level may alleviate the HDL code coverage measurement at device level"*.

Code coverage can be used to validate the completion of verification activities when it is performed as a result of requirement-based simulation. Acceptance criteria in terms of coverage rate should be established early on the hardware design life cycle, ideally in the PHAC or HVP.

Code coverage is performed by the HDL simulator during the RTL simulation of the whole design. However, the aim of attaining full coverage may be very difficult to achieve using requirement-based test vectors only. Block unitary simulation may be performed to assess that non-covered code elements (statements, conditions, expressions) are successfully exercised at a lower level.

Most HDL simulators are able to provide the HDL code coverage metrics required for DAL A and B designs, which include:

- Statement coverage: It checks that every executable statement has been reached. A statement is a line of code that ends in a semicolon (;)
- Branch coverage: It checks if all the possible branch directions were exercised. This metric is also known as decision coverage
- Condition coverage: It checks that all conditions for taking a branch were exercised
- Expression coverage: It checks that every input to an expression has taken both the true and false values under conditions that allow the input control the output

CM-SWCEH-001 §8.4.2.1 item (g) states that: "*The non-covered areas should be analysed and justified with the objective of reaching those coverage criteria*". Consequently, the non-covered code elements needs to be analysed and one of the following actions should be taken:

- Add a test case if there is a requirement associated to the non-covered code

- Add a requirement together with a test case to verify it if the non-covered code is necessary and there is no requirement associated
- Remove the non-covered code if it is not necessary
- Perform a safety analysis if the non-covered code is necessary but very difficult to verify

4.1.2.1.4 Tool Assessment and Qualification

As stated in ED-80/DO-254 §11.4 *"When verification tools are used to verify the hardware item, an error in the tool may cause the tool to fail to detect an error in the hardware item or hardware design. Prior to the use of a tool, a tool assessment should be performed. The results of this assessment and, if necessary, tool qualification should be recorded and maintained"*.

Tool assessment is mandatory for verification tools used in the development of hardware items with DAL A and B.

The independent assessment of the HDL simulator output may be performed by running two HDL simulators in parallel under the same verification environment. The simulation results of both tools can be compared through inspection of waveform viewers and output log files.

The verification processes supported by the HDL simulator should be identified and documented. The limitations and aspects not verified by the tool and the use of other verification methods to cover these aspects, such as STA to verify timing performance under worst-case conditions, should also be documented.

Concerning the code coverage function, as stated in ED-80/DO-254 §11.4.1 item (4), no further assessment is necessary if code coverage metrics are used to assess the completion of the verification process.

4.1.2.1.5 Configuration Management

HDL simulators support script languages for tool operation and configuration. These scripts can be subject to revision control (CVS, and SVN among others). Verification means used within the simulation environment such as testbenches and BFM needs to be also subject to revision control as part of the configuration management process.

The further assessment of the HDL simulators will clearly identify the type of files to be subject to revision control.

HDL simulation tool version should also be tracked together with the design technology libraries used for simulation. These libraries are linked with the ones used by the place and route tool.

4.1.2.2 Static Verification Approach

4.1.2.2.1 Introduction to Static Verification

Static verification makes a comprehensive analysis of the DUV behaviour using logic and discrete mathematics, aiming at specifying and designing accurate simulation models for analysis purposes.

Among the static verification methods, Static Timing Analysis (STA) is a highly recognized method which is broadly used in programmable AEH design flows.

STA is a comprehensive technique of analysis used to validate the timing performance of a design by checking all possible paths (real or potential false paths) for timing violations under worst-case conditions. This analysis considers the worst possible delay through each logic element, but not the logical operation of the circuit.

The user-defined timing constraints file is used as an input for STA, which performs its analysis in four different stages:

- Identification of timing paths (critical, false, multi-cycle, single, worst and best paths among others) in the DUV
- Calculation of propagation delay along each path
- Checks for timing constraints violations, applying Process-Voltage-Temperature (PVT) variations to assess the timing performance of the DUV under a range of operating conditions
- Reports timing results expressed as a positive slack or a negative slack. A negative slack indicates timing violation

The analysis of timing paths may also include:

- Clock Domain Crossing (CDC) analysis, since paths crossing multiple clock domains are also taken into account
- Analysis of combinational loop structures
- Designs with DCM (Digital Control Manager) or PLL (Phase-Locked Loop) macro-functions, including advanced timing features such as clock multiplication, division, or phase shifting

§ 4.1.4 mentions other static verification techniques which are considered as advanced verification methods.

4.1.2.2.2 Static Timing Analysis (STA) Tools

STA tools may be provided by third party EDA tool manufacturers or by the PLD vendor as part of a design environment (IDE). Nevertheless, for the development of programmable AEH, it is expected that STA tools from PLD vendors obtain more accurate results, since the analysis is performed on the design after placement and routing and, therefore, on the basis of detailed physical information which may not be available to the third party EDA tool.

STA is closely linked to place and route processes, allowing the designer to identify

failing paths -not meeting timing- to correct them by defining new constraints and/or modifying the RTL HDL description.

4.1.2.2.3 Tool Assessment and Qualification

In terms of tool assessment and qualification, STA tools are considered verification tools. The independent assessment of the tool output can be done by combining different methods:

- Post-place and route simulation of the back-annotated HDL model to check that no timing violation is triggered by the test vectors
- Visual inspection of the STA output report focusing on critical design items (critical, false and multi-cycle paths, CDC, and advanced timing features introduced by macro-functions such as DCM and PLL)
- Physical tests on the programmed device

These methods aim to assess that no tool option will change the behaviour of the design, being in line with CM-SWCEH-001 §8.4.2.2 item (e) statement: *"An analysis of the process used to perform the synthesis, place and route should confirm that the verification of the device requirements demonstrates the behaviour of the implementation of the device"*.

4.1.2.2.4 Configuration Management

STA tools support script languages for tool operation and configuration. These scripts can be subject to revision control (CVS, and SVN among others). The user-defined timing constraints file needs to be also subject to revision control as part of the configuration management process.

The further assessment of the selected STA tools will clearly identify the type of files to be subject to revision control.

STA tool version should also be tracked together with the design technology libraries used for the analysis.

4.1.3 Integrated Development Environments (IDE)

Usually, place and route tools are provided by the PLD vendor as part of an IDE. These development environments provide a comprehensive set of tools covering both design and verification activities along the design flow. These IDE may include, besides the place and route and the additional functions mentioned on § 4.1.1.3.3 , the following tools:

- Graphical HDL entry: Embedded design entry tool mixing graphics (such as schematic and state diagram for FSM) and text, providing automated generation of hierarchical HDL code
- PLD vendor-specific logical synthesis tool: Some of the main PLD vendors offer their own integrated synthesis engine optimized for their devices
- Power consumption analysis: This function gives a power estimation based on design's logic resource usage, toggle rates, and I/O loading among other factors. Its accuracy is highly dependent on the accuracy of the input data provided by the designer
- Static timing analysis (STA): Major function for timing verification, which is the process of verifying that the design meets the defined timing constraints. This analysis should be performed on the design after placement and routing in order to get accurate results (refer to § 4.1.2.2.1 and § 4.1.2.2.2 for a more detailed description). STA is closely linked to place and route processes, allowing the designer to identify failing paths -not meeting timing- to correct them
- HDL simulation: These simulators, provided by the PLD vendor or by a third party CAE vendor, are directly integrated within the place and route flow
- Embedded logic analysis: Configurable Soft IP cores for signal capture and monitor purposes which are directly instantiated into the design, allowing to view any selected internal signal or node upon preconfigured trigger conditions in real-time. Monitored signals are accessed through the device's physical resources such as its JTAG port

4.1.4 Alternative and/or Complementary Types of Tools

This category encompasses verification methods which are gaining acceptance in today's programmable logic devices industry and, consequently, may also become useful for the development of programmable AEH.

Most of these verification methods are identified in ED-80/DO-254 Appendix B, “Design assurance considerations for level A and B functions” §3.3, as advanced verification methods: Safety-specific analysis and formal methods.

Types of tools within this category include:

- HDL Rule Checkers
- Clock Domain Crossing (CDC) analysers
- Logic Equivalence Checking (LEC) tools
- Assertion-Based Verification (ABV) methods:
    - Dynamic ABV tools (dynamic approach)
    - Formal Model Checking tools (static approach)

§ 4.2 aims to assess the types of tools used to perform these advanced verification methods, highlighting the following aspects:

- Tool position within the programmable AEH design flow
- Coverage of verification objectives as defined by ED-80/DO-254 §6.2.1
- Advantages, disadvantages and limitations

4.2 Evaluation of Alternative and/or Complementary Types of Tools

This section focuses on the types of tools which are gaining acceptance in today's industry as a useful part of complex design flows. Most of them may be also suitable for the development of programmable AEH. This section provides an overview of these types of tools together with their advantages, disadvantages and limitations. Finally, this section draws a number of conclusions about the role of each type of tool within the programmable AEH design flow.

The alternative and/or complementary types of tools assessed are:

- HDL Rule Checkers, also known as HDL linters
- Clock Domain Crossing (CDC) analysers
- Logic Equivalence Checking (LEC) tools
- Assertion-Based Verification (ABV) tools

4.2.1 HDL Rule Checkers (Linters)

4.2.1.1 Introduction to HDL Rule Checking

HDL rule checking is an static verification method which performs syntax and semantic checks of the HDL source code (VHDL, Verilog and System Verilog) against a set of rules. These rules, which can be complemented by user-defined directives, cover several areas which include naming and RTL coding conventions, improvement of simulation performance, reset and clock connectivity issues, and detection of potential errors.

4.2.1.2 HDL Rule Checkers within the AEH Design Flow

HDL rule checkers, also known as HDL linters, take the HDL RTL description as an input, and may, in some cases, accept the addition of user-defined rules to be checked. These additional rules can be extracted from the coding rules recorded in the Hardware Design Standards.

The tool generates a complete report gathering all the results after checking the HDL source code.

4.2.1.3 ED-80/DO-254 Verification Objectives

HDL rule checking is a static verification method which is independent from the requirement-based verification approach. Therefore, the design is not verified against its requirements and traceability is neither assessed.

Acceptance test criteria is based on the absence of checking errors and analysis of warnings. In case of errors, the HDL source code should be corrected and a further analysis should be carried out to ensure that these modifications have no impact in terms of safety.

HDL rule checking may help in establishing conformance with Hardware Design Standards, as recommended in CM-SWCEH-001 §8.4.2.1 item (f) *"If a Hardware Description Language (HDL), as defined in ED-80/DO-254, is used, coding standards for a proper use of this*

language should be defined. […] Conformance to those standards should be established".

4.2.1.4 Advantages

HDL rule checkers can be used to lighten the RTL source code reviews against the coding rules specified in the Hardware Design Standards, increasing the productivity of the design team. The reports generated by the tool may be used as an effective way of documentation.

4.2.1.5 Disadvantages and Limitations

HDL rule checkers do not fully replace code reviews. As a static verification method, HDL rule checking is not intended to verify the RTL source code against its functional requirements and architectural decisions as stated in CM-SWCEH-001 §8.4.2.1 item (i) *"If a Hardware Description Language (HDL) is used, an HDL code review against the conceptual design and requirements should be performed"*. Therefore, code reviews are still necessary.

HDL rule checkers are automated tools which may require tool assessment involving additional reviews of the outputs or even a basic tool qualification.

4.2.1.6 Conclusions

HDL rule checkers are not a replacement of HDL code reviews, but they can contribute to lighten the review process, especially for large designs.

4.2.2 Clock Domain Crossing (CDC) Analysers

4.2.2.1 Introduction to CDC Analysis

In today's multi-clock designs, synchronize signals that cross unrelated time domains have become an important issue. Clock Domain Crossing (CDC) may be at the origin of the following failures:

- Metastability: This issue may occur when the data input changes too close to the clock edges (setup or hold violation). In such cases, the FF cannot decide whether its output should be a logic '1' or a logic '0' for a long time
- Race/skew errors: The same signal is not arriving to all the destination FF at the same time, which means that only some of them are detecting the event
- Data loss: The destination clock domain may not register the source data in the very first cycle due to metastability
- Data inconsistency: Multiple signals are crossing from one clock domain to another, and each signal is synchronized separately

CDC analysis may become important to identify timing errors which are not detected neither by functional simulation nor by STA (because most of STA tools treat CDC paths as exceptions and ignore them). The use of CDC analysis tools may be useful to perform this analysis.

4.2.2.2 CDC within the AEH Design Flow

Figure 4.3 shows the CDC analysis tool within the programmable AEH design flow. The CDC analysis is performed on the HDL RTL source code.

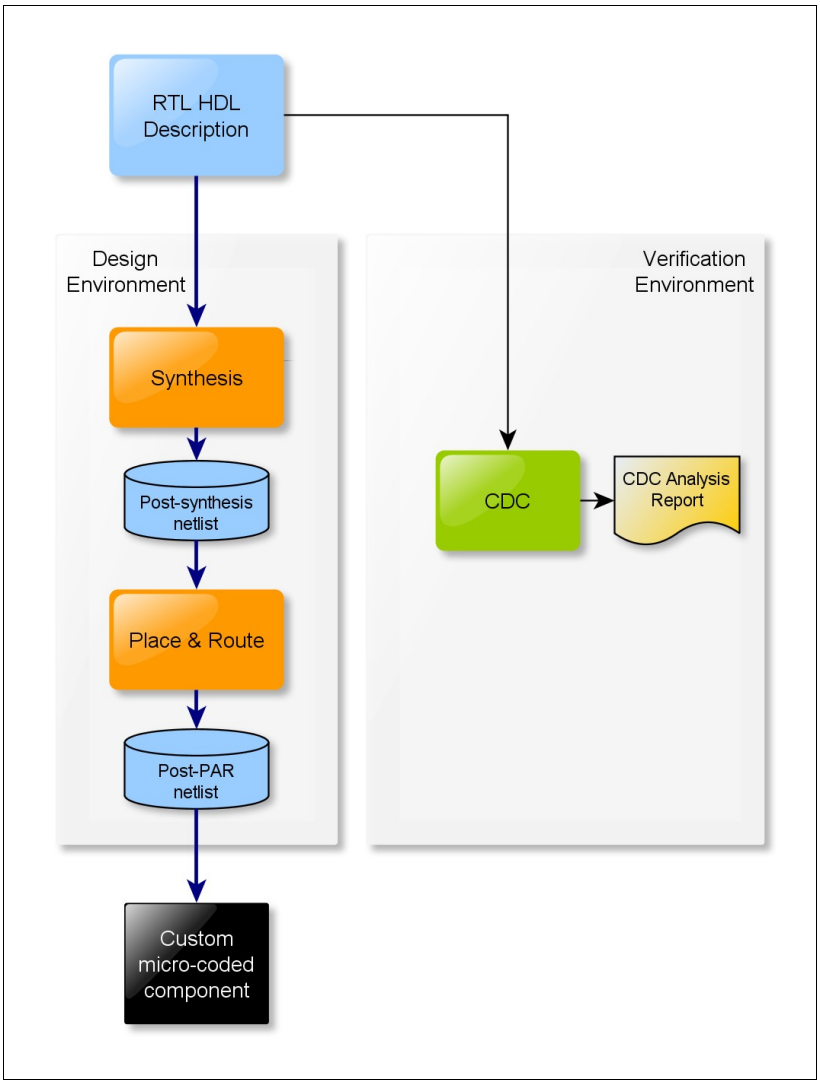

Figure 4.3. CDC Analysis Tool within AEH Design Flow

4.2.2.3 ED-80/DO-254 Verification Objectives

CDC analysis can be used to verify that the design meets some specific requirements regarding timing performance.

Acceptance test criteria is based on the absence of checking errors and analysis of warnings. In case of errors, the HDL source code should be corrected and a further analysis should be done to ensure that these modifications have no impact in terms of safety.

4.2.2.4 Advantages

Some CDC analysis tools feature automated metastability injection which can be combined with functional simulation in order to find complex CDC reconvergence bugs.

This type of tool is useful to identify all the CDC paths for a more detailed analysis, and particularly helpful for large designs.

4.2.2.5 Disadvantages and Limitations

The CDC analysis is directly performed on the RTL HDL code. Therefore, an inappropriate use of design tools -synthesis and place and route- may introduce errors in CDC paths which were originally well designed and flagged as correct after CDC analysis.

A case study is the replication of the second FF of a synchronizer made of two cascaded FF. This replication may be introduced by the synthesis tool due to fanout limitations, and will induce data inconsistency along the design.

The CDC analysis tools generate large reports including multiple false negatives. The analysis of these false negatives may be significantly time-wasting.

These automated tools may require tool assessment involving additional reviews of the outputs or even a basic tool qualification.

4.2.2.6 Conclusions

CDC tools are more suited for the design of large ASICs involving multiple clock domains, where physical tests are not possible and errors should be found before tape-out to avoid very expensive redesigns.

A CDC efficient analysis may be achieved through the combination of the following best practices and verification methods:

- Proper use of Hardware Design Standards: Most of the CDC errors come from poor design descriptions. Therefore, the proper use of Hardware Design Standards is essential. This includes coding rules and guidelines related to:
  - ✔ Coherent naming convention to identify the CDC paths and ease the analysis
  - ✔ Clock and reset design practices for proper synchronization
  - ✔ Clock domain crossing design: Synchronization techniques for single and multiple signals to avoid reconvergence issues, data loss and inconsistency
- RTL analysis and review to assess the right implementation of design standards
- Use of extra timing constraints specific to CDC
- Functional simulation
- STA to identify the CDC paths
- Constrain synthesis and place and route tools to ensure that no optimization is made on CDC paths
- Review synthesis and place and route reports to ensure that no optimization is made on CDC paths

For programmable AEH requiring multiple clock domains, CDC analysis tools may be a useful complement but should not replace the above mentioned verification means.

4.2.3 Logic Equivalence Checking (LEC)

4.2.3.1 Introduction to LEC

Logic Equivalence Checking (LEC) is a static verification method used to formally prove that two representations of a design have the same behaviour. This method can be applied at different levels within the design flow, comparing the RTL HDL description, the netlist generated after synthesis, and the netlist after place and route.

The RTL HDL description used as reference source, also known as "golden netlist", should be the design fully verified.

In some cases, this method is used as an additional step in the verification process to check that Engineering Change Orders (ECO) have not introduced any error.

4.2.3.2 LEC Within the AEH Design Flow

Figure 4.4 shows the LEC verification method within the programmable AEH design flow.

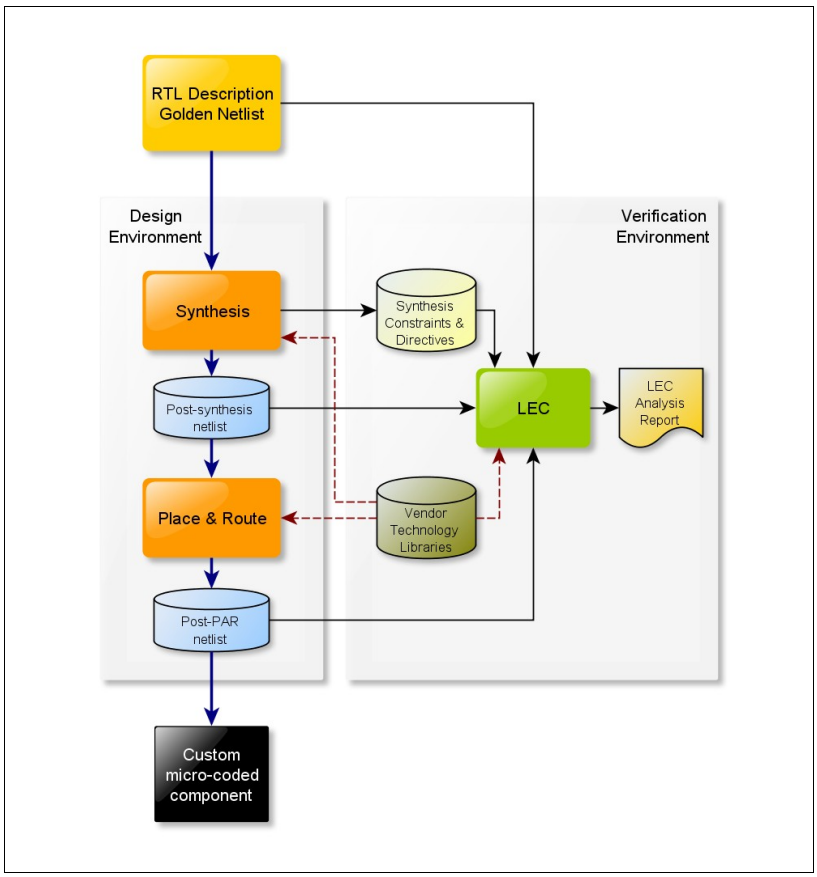

Figure 4.4. LEC Analysis within AEH Design Flow

LEC can be used in two steps:

- RTL to Synthesis Verification: The post-synthesis netlist is compared against the original RTL HDL description (golden netlist). To perform this process, the LEC tool requires additional information concerning synthesis constrains and optimization directives. Such data is provided by the synthesis tool
- Synthesis to place and route Verification: This method compares the netlist generated by the place and route tool with the netlist generated after synthesis.

4.2.3.3 ED-80/DO-254 Verification Objectives

LEC analysis is a static verification method which is independent from the requirement-based verification approach. Therefore the design is not verified against its requirements, neither traceability is assessed.

Acceptance test criteria is based on the absence of checking errors and analysis of warnings. In case of errors, and assuming that the HDL source code has been fully verified (golden netlist), the settings of the tool that generated the faulty netlist should be reassessed.

LEC analysis may help in assessing the equivalence of the detailed design data through the different stages of the design flow (synthesis and place and route), as recommended in CM-SWCEH-001 §8.4.2.1 item (a) *"The correctness of requirements, conceptual design data and detailed design data (including HDL or schematics) should be verified in order to ensure that detailed design data correctly and completely represent the device behaviour specified in the requirements"*.

4.2.3.4 Advantages

LEC tools may provide evidence that both synthesis and place and route tools produced an output equivalent to the original RTL source, increasing the confidence on these types of tools. This evidence can be used for tool assessment and qualification purposes of the design tools.

The LEC analysis may reveal non-used RTL HDL code or unreachable logic.

When combined with STA, LEC analysis can lighten post-place and route simulation.

4.2.3.5 Disadvantages and Limitations

Some vendor-specific macros and functions may not be supported or produce problems which may cause the LEC tool to report false differences:

- Block RAMs (especially large dual-port block RAMs), ROM and PLL functions. These macros should be declared as black boxes in order to be ignored by LEC
- Multipliers instantiated as macros or implemented in logic may not solve in some cases
- Tristate registers which may require manual matching

Some optimization techniques introduced by the synthesis tools such as retiming may not be accepted by LEC tools, whose output will show logic differences which are difficult to justify.

LEC methodology was introduced for the design of ASICs. The first generation of these tools required complex setup and big learning curves for their use within the PLD design flow. In

order to overcome this challenge, the current generation of LEC tools implement advanced mapping techniques which need detailed information concerning synthesis constrains and optimization directives. This information is directly provided by the synthesis tool, increasing the risk of loosing independence.

The LEC tools also uses vendor-specific libraries. Both LEC and synthesis tool vendors should prove that there are no sensitive common points between their tools.

4.2.3.6 Conclusions

LEC is not a replacement for post-place and route simulation; however, combined with STA, it may be used as a way of lightening this level of simulation, which can be partially performed.

The known issues involving the interfaces between LEC and synthesis tools may add important design constraints.

Some of LEC advantages concerning RTL HDL code debugging may be also attained by combining other verification methods, such as design reviews, elemental analysis (code coverage) or HDL rule checkers.

LEC tools can be used as an effective means for design tool assessment, for both synthesis and place and route tools. For this purpose, independence, especially between LEC and synthesis tools, needs to be assessed.

LEC tools for the development of custom micro-coded components are continuously evolving, therefore reducing the number of unresolved cases and increasing efficiency. This evolution, together with the increasing complexity of programmable AEH, makes LEC analysis a verification method to be taken into account in the foreseeable future.

4.2.4 Assertion-Based Verification (ABV) Tools

4.2.4.1 Introduction to ABV

Assertion-Based Verification (ABV) is a methodology which verifies the predefined properties of the design.

A property is the description of the required design behaviour, while the assertion is a directive to the verification tool aiming to verify the property. Both terms may be used interchangeably in the verification flow.

There are two main types of properties:

- Safety properties: Design characteristics bounded in time which should always hold true. It is equivalent to the statement “bad things never happen”. Example: “A request is followed by acknowledge within 4 clock cycles”
- Liveness properties: Design characteristics that involve infinite sequences. It is equivalent to the statement “good things eventually happens”. Example: “If external reset is deasserted, then internal reset is eventually deasserted too”

4.2.4.2 Two Approaches for ABV: Dynamic and Static

ABV can be performed in two ways, depending on the method used to verify the properties:

- Dynamic approach: Test vectors are applied to the DUV which is instantiated in a testbench containing all written simulation scenarios. This method is also known as Dynamic ABV and works together with functional simulation
- Static approach: The DUV is modelled as a mathematical entity using formal techniques in order to verify its properties. This method is also known as Formal Model Checking and does not require test vectors

Assertions are not only limited to the RTL HDL description of the DUV; they can also be integrated within some verification means used for simulation, such as BFM used in simulation testbenches.

There are assertions which are synthesizable and specific for gate level descriptions, enlarging the scope of application of ABV within the design flow.

4.2.4.3 ABV Within the AEH Design Flow

Figure 4.5 shows Dynamic ABV and Formal Model Checking methods integrated within the programmable AEH design flow.

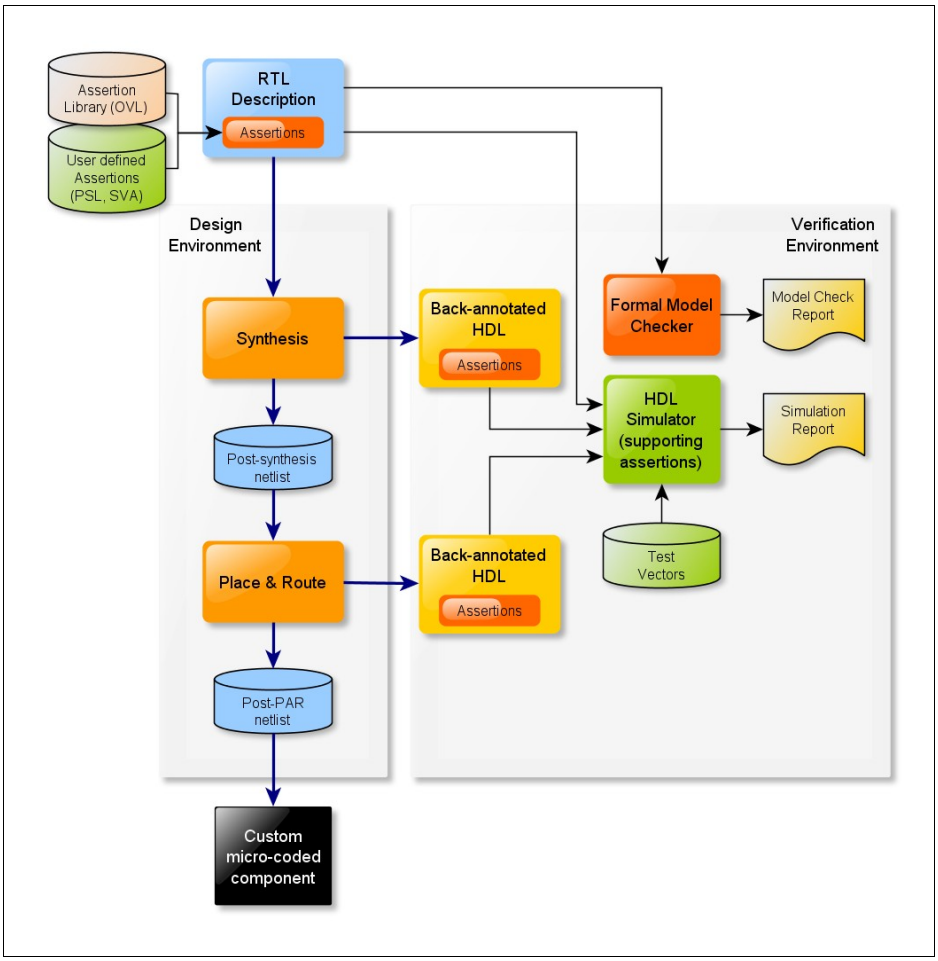

Figure 4.5. ABV within AEH Design Flow

Assertions can be written by the RTL designer using the following industry standard languages:

- Open Verification Library (OVL): Vendor and language-independent library of predefined assertions which allows the use of the same set of assertions across different tools and flows
- Property Specification Language (PSL): Language developed by Accellera to capture design behaviour in an executable, formal and unambiguous manner. PSL was standardized in September 2004 by the IEEE 1850 working group
- System Verilog Assertions (SVA): Set of constructs integrated within System Verilog which are meant to build assertions and to couple them with the RTL design and verification means used for simulation
- Open Vera Assertions (OVA): Open Vera is an open source verification language for the development of simulation testbenches, assertions and properties. OVA provides a declarative method to describe sequences of events and to test them for their occurrence

4.2.4.4 ED-80/DO-254 Verification Objectives

ABV methodology aim to prove that the design meets the requirements, considering the definition of property as a description of the design specified behaviour. Therefore, this verification means meets the verification process objectives stated in ED-80/DO-254 §6.2.1, in the same way functional simulation does when requirement-based test cases are used to generate the test vectors for simulation.

4.2.4.5 Advantages

ABV methodology meets ED-80/DO-254 verification objectives since properties can be directly mapped to the requirements.

Some assertion languages (PSL) and libraries (OVL) are standardized, making possible the use of assertions together with the main HDL used in the programmable AEH design flow, such as Verilog, VHDL and System Verilog.

Assertions improve the observability of the design, creating several observation points and enabling internal state, datapath and error pre-conditions coverage analysis.

Dynamic ABV has the following advantages:

- Tool assessment: Since the tool used for Dynamic ABV is a HDL simulator supporting assertion languages, no special considerations are needed, and the tool can be assessed as it is done when performing functional HDL simulation (refer to § 4.1.2.1.4)
- Assertions can be propagated through the design flow, from RTL to gate level, enabling them for post-synthesis and post-place and route simulations

Furthermore, Formal Model Checking provides a comprehensive coverage, allowing the complete verification of both safety and liveness properties. It can also increase the verification of robustness, by exercising corner cases which may be very difficult to define using requirement-based test cases.

4.2.4.6 Disadvantages and Limitations

Assertions in general may not have the necessary level of detail to verify low-level functions and ensure a high degree of code coverage.

Dynamic ABV can lack properties coverage, since it only covers safety properties which also depend on time (each test case is bounded in time) and datapath (only selected paths are verified).

Besides, Formal Model Checking may present the following limitations:

- Tool assessment: Formal Model Checking relies on complex mathematical algorithms. This dependence may increase the difficulty of assessing the tool
- Low support among PLD vendors: This verification method is still considered a not mature technology and, therefore, it has not been supported by some of the main PLD vendors, which claim slow customer acceptance

4.2.4.7 Conclusions

Dynamic ABV can be a useful complement for functional simulation, especially in terms of robustness verification.

The selection of the assertion language is a key element and it should take into account several parameters, including its own maturity, the HDL used for the RTL description, verification tool support, and the characteristics of the assertion language itself (how it interacts with the testbench, flexibility and functional coverage support).

Formal Model Checking is an interesting methodology due to its comprehensiveness but it lacks some integration in the PLD design environments and may also cause additional concerns in terms of tool assessment and qualification. Similarly to other formal approach techniques such as LEC, this formal technique will be more helpful as complexity of programmable AEH increases.

The effectiveness of both ABV approaches is highly dependent on the quality of the properties and assertions which should be defined by someone different to the RTL designer in order to ensure independence.

4.3 Selection of the Types of Tools Target of the Study

This section aims to select the types of tools -identified in the previous sections- which should be considered subject to the SHARDELD study after evaluating their relevance and complexity in terms of functional features and configuration. Selection has been made on the basis of IOxOS Technologies experience with most of the types of tools identified. The feedback from PLD tool vendors and FAEs has also been taken into account for this purpose.

4.3.1 Design Tools

Design entry tools for conceptual design, such as text/graphical editors and MBD tools, are not considered subject to this study due to their relatively low level of complexity and configuration options. Regardless of the tool used for design entry, the output -which is the RTL HDL description- should be independently reviewed against the hardware requirements and also to determine whether the coding rules specified in the Hardware Design Standards have been followed. These reviews, which are also mentioned in CM-SWCEH-001 §8.4.2.1 items (a), (f) and (i), make no further tool assessment necessary.

Concerning the tools to carry out the detailed design process -synthesis and place and route tools- their complexity is significantly higher and their outputs may be harder to assess.

Synthesis tools feature additional functions aiming at optimizing the design timing and area which can lead to unexpected gate level outputs. These include redundant paths for architectural mitigation purposes that are removed for area optimization, well-designed synchronizers intended for CDC management which are replicated due to fanout limitations, and FSM implemented with deadlock states among other undesired results. Some directives which may be related to safety critical applications, such as the *safe* attribute for FSM, may have a different interpretation depending on the tool used. Furthermore, synthesis attributes are non-standardized, that may lead the synthesis tool to ignore them.

Place and route tools also have a high degree of complexity together with an advanced set of options which could cause errors and unexpected behaviours if they are not properly used.

Even though proper HDL coding style reduces the impact of design tool interpretation and optimization, it is still necessary to control the design tool configuration and functionalities. Both design tools generate comprehensive reports that need to be reviewed. The content of these reports -and particularly the warnings- should be duly analysed and justified.

In addition, IOxOS Technologies Hardware Design Standards [INT-01] record guidelines for synthesis and place and route tools, which reinforces the fact that these type of tools, mandatory for the development of programmable AEH, should be considered for specific assessment. These guidelines show compliance with CM-SWCEH-001 §8.4.2.1 item (f) statement: *"HDL coding standards usually include but are limited to: Guidelines to ensure the design will synthesize properly, rules to address the limits of design and verification tools, […] and guidelines to reuse lessons learned from previous developments"*.

4.3.2 Verification Tools

HDL simulators, which are extensively used in programmable AEH design flows, present a relatively low level of complexity in terms of configuration and their outputs can be more easily assessed. The effectiveness of these tools depends more on the quality of the testbenches used for simulate the DUV.

Concerning the alternative and/or complementary types of tools described in § 5.2 , most of the potential issues involving these verification tools may be mitigated by a proper coding technique (described in the Hardware Design Standards) and the right use of the design tools (synthesis and place and route). Besides, their relevance and degree of integration within programmable AEH design flows is still unclear, contrary to the identified design tools, which are mandatory.

For these reasons, no specific assessment should be considered for the before mentioned verification tools.

However, there are some aspects concerning the integration of these verification tools within the design flow (for example, simulation libraries management), or their tool assessment and qualification criteria, which need to be taken into account. Consequently, generic guidelines for these types of tools should be provided even if they are not selected as a target of the SHARDELD study.

Regarding STA tools, those embedded in the design environments (IDE) provided by PLD vendors will be specifically assessed due to their high degree of interdependence with the design tools covered in the study. One item of the consultation to be carried out in the following task of the SHARDELD study will concern the use and experience with third party STA tools. Depending on the results of this consultation, selected third party STA tools may also be specifically assessed.

4.4 Conclusions of Task 1

Design tools identified in § 4.1.1 , which are intended for detailed design process, i.e. synthesis, and place and route, have a level of complexity and configuration options which could lead to potential safety impact when designing programmable AEH. Therefore, both type of tools, which carry out mandatory processes for the development of programmable AEH, should be subject to a specific assessment in the framework of the SHARDELD study.

Specific assessment will consist on a detailed evaluation, in terms of configuration and embedded functions, of commercial CAE tools selected by means of a commercial tool survey and a consultation with aircraft and equipment manufacturers to be performed in the following task of the SHARDELD study.

As a result, the specific assessment will provide the following information:

- Guidelines and best practices for tool usage
- Tool limitations due to functional features and configuration options
- Technical features and benefits
- Known technical issues and their impact on safety
- Tool integration within design life cycles requiring compliance with ED-80/DO-254 guidance:
  - ✔ Data availability from tool vendors
  - ✔ Recommendations for tool assessment and qualification
  - ✔ Recommendations for configuration management
  - ✔ Relevant service experience

As stated in § 4.1.1.2.3 , § 4.1.1.3.3 , and § 4.1.3 , both synthesis and place and route tools may include features other than the main design functions they are intended for. The assessment of these additional functionalities, which may cover both design and verification processes, will depend on the results of the consultation with aircraft and equipment manufacturers.

Regarding verification tools identified in § 4.1.2 :

- For HDL simulators, no detailed assessment will be done due to the relatively low level of complexity of these tools and the simple approach to tool assessment and qualification suggested in § 4.1.2.1.4 . Nevertheless, a generic assessment of HDL simulators will be performed to provide the following information:
  - ✔ Guidelines and best practices for tool usage
  - ✔ Highlight advantages, disadvantages and limitations
  - ✔ Overall recommendations for tool integration within design life cycles requiring compliance with ED-80/DO-254 guidance
- STA tools integrated within design environments (IDE) will be subject to a specific assessment due to their significant interaction with place and route processes

With respect to alternative and/or complementary types of tools described in § 5.2 , a generic assessment of some of the identified types of tools -depending on the results of the consultation with aircraft and equipment manufacturers- will be provided, complementing the evaluation described in

this interim report. Additionally, the consultation will include questions about the alternative methods which are already adopted for the development of programmable AEH. The results will indicate the degree of integration of these methodologies, as well as their acceptance within the programmable AEH designer community.

Table 4.1 summarizes the type of tools identified, together with the required level of assessment in the framework of the SHARDELD study.

Type of Tools and Required Level of Assessment			
ED-80/DO-254 Related Process	***Type of Tool***	***Level of Assessment***	***Rationale***
Conceptual Design (§5.2)	Design Entry	No further assessment required	Low level of complexity Tool assessment done by RTL HDL code review
Detailed Design (§5.3)	Synthesis	Specific assessment	High level of complexity and configuration options These types of tools are mandatory for the development of programmable AEH The assessment of additional functions included in the tools will depend on the results of the consultation with aircraft and equipment manufacturers
	Place and route	Specific assessment	
Dynamic Verification (§6)	HDL simulators	Generic assessment	Relatively low level of complexity Simple approach to tool assessment
	Dynamic ABV		
Static Verification (§6)	STA	Specific assessment of STA tools integrated into design environments (IDE)	Configuration settings and/or improper use may have a significant impact on the design and verification processes Significant interdependence with place and route processes
	HDL Rule Checkers	Generic assessment depending on consultation results	Potential issues easily mitigated with the proper use of design tools, validation methods such as design reviews and well-defined RTL HDL coding style Some of this tools are not yet widely used for the development of programmable AEH
	CDC Analysers		
	LEC		
	Formal Model Checkers	Generic assessment depending on consultation results	Lack of integration in the programmable AEH design environment

Table 4.1. Summary of Types of Tools and Required Level of Assessment

5. Task 2: Survey Available Commercial Tools

The second task of the SHARDELD study aims to identify the tools available on the market, within the types of tools selected in the previous task, which are relevant for the study. It also aims to assess these tools in order to determine their limitations, benefits, and the different methods followed by the programmable AEH designer community to fulfil ED-80/DO-254 objectives in terms of tool assessment and qualification.

To achieve this objective, a consultation has been addressed to relevant aircraft and equipment manufacturers. The consultation is organized into the following four sections, each one including an specific questionnaire:

- Section 1: Identification of commercial CAE tools within the types of tools selected for specific assessment (synthesis, place and route and STA) and HDL simulation tools
- Section 2: Tool Assessment and Qualification approach and methods for the selected tools
- Section 3: Identification of alternative and/or complementary types of tools used in the programmable AEH design flow
- Section 4: Identification of commercial custom micro-coded components used for the development of programmable AEH (CPLDs, FPGAs and structured ASICs)

The consultation form and its four questionnaires is annexed to this report in §8.

The following sections describe the activities performed to carry out this task.

5.1 Custom Micro-coded Components used for the Development of Programmable AEH:

This section aims to identify the main vendors and device types used for the development of programmable AEH, increasing the effectiveness when considering commercial CAE tools provided by PLD vendors.

IOxOS Technologies has experience developing safety critical airborne application using devices from the following vendors: Altera, Microsemi (Actel), and Xilinx. Three types of devices are provided: FPGAs, CPLDs and Structured ASICs, which shares the same tools within the design flow.

The preliminary results of the consultation of aircraft and equipment manufacturers revealed that the three PLD vendors above mentioned are the sole providers of programmable AEH targeting applications requiring ED-80/DO-254 compliance. Other PLD vendors such as Atmel or Lattice were not mentioned as providers for programmable AEH.

Table 5.1 summarizes the PLD vendors and the type of devices used as programmable AEH, including those that were not referred in the consultation.

PLD Vendors and Type of Devices used as Programmable AEH			
PLD Vendor	***Type of Device***	***Technology***	***Devices***

PLD Vendors and Type of Devices used as Programmable AEH			
Altera	CPLD	SRAM / non-volatile / reprogrammable	MAX Series
	FPGA	SRAM / volatile / reprogrammable	Stratix, Arria and Cyclone Series
	Structured ASIC	Standard cell / non-volatile / one-time programmable (OTP)	HardCopy Series
Atmel	FPGA	SRAM / volatile / reprogrammable	AT40K Series
Lattice Semiconductor	CPLD	Flash / non-volatile / reprogrammable	MachXO Series
	FPGA	SRAM / volatile / reprogrammable	LatticeECPx Series LatticeSC
Microsemi (formerly Actel)	FPGA	Flash / non-volatile / reprogrammable	Igloo Series ProASIC and RT ProASIC Fusion and SmartFusion
		Antifuse / non-volatile / one-time programmable (OTP)	RTAX and RTSX-SU Series
Xilinx	CPLD	SRAM / non-volatile / reprogrammable	CoolRunner Series Spartan-3AN
	FPGA	SRAM / volatile / reprogrammable	Spartan Series Virtex Series

Table 5.1. Summary of PLD Vendors and Type of Devices used as Programmable AEH

Antifuse technology (Microsemi) and radiation-hardened SRAM technology (Xilinx) are proven technologies to mitigate SEU effects. However, their expensive price has restricted their use in avionics, making them more appropriate for the space domain.

5.2 Commercial CAE Tools Survey

This section aims to select the commercial CAE tools available on the market, taking into account the custom micro-coded components identified in § 5.1 and the summary of types of tools described in the conclusions of the Task 1 of the SHARDELD study.

The survey is based on IOxOS Technologies own experience and feedback from PLD tool vendors and FAEs.

Available Commercial CAE Tools				
ED-80/DO-254 Related Process	***Type of Tool***	***Tool***	***Tool Vendor***	***Comments***
Conceptual Design (§5.2)	Design Entry	HDL Designer*	Mentor Graphics	Comprehensive design entry environment
		Matlab/Simulink*	MathWorks	Model-Based Design (MBD) approach
Detailed Design (§5.3)	Synthesis	Quartus II Integrated Synthesis	Altera	Bundled with Altera Quartus II IDE
		Xilinx XST	Xilinx	Bundled with Xilinx ISE design environment
		Synplify / Synplify Pro	Synopsys	Widely used third party high-end synthesis tools with long service experience Synplify Pro is embedded as synthesis engine in Lattice and Microsemi design environments
		Encounter RTL	Cadence	
		Leonardo Spectrum*	Mentor Graphics	Simple and mature synthesis tool
		Precision RTL Plus*	Mentor Graphics	Claims FPGA synthesis for Mil-Aero and safety critical applications based on its integration with a formal equivalence checker (FormalPro)
	Place and route	Quartus II	Altera	Tool provided by the PLD vendor All of them are integrated in comprehensive design environments (IDE)
		ispLEVER	Lattice	
		Libero IDE	Microsemi (Actel)	
		Xilinx ISE	Xilinx	
Dynamic Verification (§6)	HDL simulators	ModelSim DE/PE/SE	Mentor Graphics	These HDL simulators offer additional features such as code coverage (to perform elemental analysis) and assertion language support (particularly PSL) to carry out Dynamic ABV verification method
		Active HDL	Aldec	
		Incisive Enterprise Simulator*	Cadence	
		Quartus II Simulator (Qsim)*	Altera	Bundled with Altera Quartus II IDE Does not support advanced features
		ISE Simulator (ISim)*	Xilinx	Bundled with Xilinx ISE design

Available Commercial CAE Tools				
				environment Does not support advanced features
	Dynamic ABV	QuestaSim	Mentor Graphics	These HDL simulators support advanced HDL languages and features
		Riviera Pro	Aldec	
Static Verification (§6)	STA	TimeQuest	Altera	Bundled with Quartus II IDE
		SmartTime	Microsemi (Actel)	Bundled with Actel Libero IDE
		Trace Timing Analyzer	Xilinx	Bundled with Xilinx ISE IDE
		PrimeTime*	Synopsys	More appropriate for ASIC design flows
		Encounter Timing System	Cadence	
	HDL Rule Checkers	HDL Checker	Mentor Graphics	Straight integration with QuestaSim verification environment
	CDC Analysers	Questa CDC (0-In)*	Mentor Graphics	
	LEC	FormalPro*	Mentor Graphics	Straight integration with Synplify Pro synthesis tool
		Formality*	Synopsys	More appropriate for ASIC design flows
		Conformal*	Cadence	

Table 5.2. Available Commercial CAE Tools

*These tools have not been mentioned in the consultation of aircraft and equipment manufacturers but are referenced in this survey due to its increasing acceptance degree in other industrial domains

5.3 Assessment of Selected Commercial Software Tools

5.3.1 Tools Needing Specific Assessment

This section covers the tools needing an specific assessment due to their complexity and configuration options. These types of tools were identified in the first task of the SHARDELD study.

This group includes the following types of tools:

- Detailed Design tools: Synthesis (Table 5.3) and Place and Route (Table 5.4)
- Static Verification tools: STA tools (Table 5.5)
- Dynamic Verification tool: HDL simulators (Table 5.6)

For each type of tool, a selection of commercial CAE tools has been done based on the results of the consultation to relevant aircraft and equipment manufacturers. These results are mainly used to elaborate the "Technical Issues" and "Integration within ED-80/DO-254 design life cycles" categories of the following comparison tables.

The "Benefits" and "Limitation" categories are elaborated based on IOxOS Technologies experience together with the results of the consultation.

Detailed Design CAE Tools: Synthesis			
Tool	Quartus II Integrated Synthesis	Xilinx XST	Synplify / Synplify Pro
Tool Vendor	Altera	Xilinx	Synopsys
Benefits	Integrated within Altera Quartus II IDE Optimized for Altera devices High integration with Altera design technology libraries	Integrated within Xilinx ISE Optimized for Xilinx devices High integration with Xilinx design technology libraries	Fast synthesis engine Comprehensive graphical viewer Support for devices from all FPGA vendors Longer service experience
Limitations	Depends on PLD vendor technology libraries (models) → assessment is required	Depends on PLD vendor technology libraries (models) → assessment is required	Depends on PLD vendor technology libraries (models) → assessment is required Most options should be disabled to keep full control of the tool output
Technical Issues	Issue: Wrong VHDL netlist for macro function generated by the tool Severity: **High** Mitigation: Post-synthesis simulation to identify the wrong netlist	Issue: Wrong synthesis of DSP blocks Severity: **High** Mitigation: Instantiate DSP directly in the RTL HDL code	Issue: Wrong "Safe FSM encoding" Severity: **High** Mitigation: Manually encode FSM to cover unreachable FSM states (SEU mitigation)
	Issue: Wrong PLL configurations and lock time Severity: **High** Mitigation: Post-synthesis simulation		
	Issue: Removed counters defined as		

Detailed Design CAE Tools: Synthesis			
	integers Severity: **High** Mitigation: Use a different data type (*std_logic_vector*)		

Integration within ED-80/DO-254 design life cycles	Tool assessment and qualification:
	• Independent assessment of tool output:
	  • Post-synthesis simulation (most common approach)
	  • Physical tests on programmed device
	• Relevant service experience*
		Tool assessment and qualification:
		• Independent assessment of tool output:
		  • Post-synthesis simulation (most common approach)
		  • Physical tests on programmed device
		• Relevant service experience*
			Tool assessment and qualification:
			• Independent assessment of tool output:
			  • Post-synthesis simulation (most common approach)
			  • Physical tests on programmed device
			• Relevant service experience*

Table 5.3. Commercial Synthesis Tools Assessment

*There are discrepancies between some companies regarding the availability of service experience data for synthesis tools

Detailed Design CAE Tools: Place and Route			
Tool	Quartus II	Xilinx ISE	Libero IDE
Tool Vendor	Altera	Xilinx	Actel
Benefits	Comprehensive design environment STA tool integrated Additional functions for design and verification, including embedded logic analysers for monitor internal signals in real-time	Comprehensive design environment STA tool integrated Additional functions for design and verification, including embedded logic analysers for monitor internal signals in real-time	Comprehensive design environment STA tool integrated Additional functions for design and verification
Limitations	Depends on PLD vendor technology libraries (models) → assessment is required	Depends on PLD vendor technology libraries (models) → assessment is required	Depends on PLD vendor technology libraries (models) → assessment is required
Technical Issues	Issue: Memory implemented with logic cells instead of BlockRAM Severity: **Medium** Mitigation: Instantiate memory blocks directly in the RTL HDL code	Issue: Wrong place and route of DSP blocks Severity: **High** Mitigation: Instantiate DSP directly in the RTL HDL code Issue: Mismatch between design libraries and timing libraries Severity: **High** Mitigation: Library updated by PLD vendor	Issue: Model library behaviour mismatch Severity: **High** Mitigation: Library updated by PLD vendor
Integration within ED-80/DO-254	Tool assessment and qualification: • Independent assessment of tool output:	Tool assessment and qualification: • Independent assessment of tool output:	Tool assessment and qualification: • Independent assessment of tool output:

Detailed Design CAE Tools: Place and Route			
design life cycles	• Post-place and route simulation • Visual inspection • Physical tests on programmed device (most common approach) • Relevant service experience**	• Post-place and route simulation • Physical tests on programmed device (most common approach) • Relevant service experience**	• Post-place and route simulation • Visual inspection • Physical tests on programmed device (most common approach) • Relevant service experience**

Table 5.4. Commercial Place and Route Tools Assessment

**Availability of data to claim for relevant service experience seems not enough for place and route tools

Static Verification CAE Tools: Static Timing Analysis			
Tool	TimeQuest	Trace Timing Analyzer	SmartTime
Tool Vendor	Altera	Xilinx	Actel
Benefits	Integrated within Altera Quartus II IDE Optimized for Altera devices High integration with Altera design technology libraries Exhaustive analysis Useful for CDC path analysis	Integrated within Xilinx ISE Optimized for Xilinx devices High integration with Xilinx design technology libraries Exhaustive analysis Useful for CDC path analysis	Integrated within Actel Libero IDE Optimized for Actel devices High integration with Actel design technology libraries Exhaustive analysis Useful for CDC path analysis
Limitations	Timing performance strong dependence on user constraints Default options may lead to poor timing estimation, particularly in following areas: • Combinational loop structures • Designs with PLL or DCM	Timing performance strong dependence on user constraints Default options may lead to poor timing estimation, particularly in following areas: • Combinational loop structures • Designs with PLL or DCM	Timing performance strong dependence on user constraints Default options may lead to poor timing estimation, particularly in following areas: • Combinational loop structures • Designs with PLL or DCM
Technical Issues	*-- Not enough data from the consultation --*	Issue: Mismatch between design libraries and timing libraries Severity: **High** Mitigation: Library updated by PLD vendor	*-- Not enough data from the consultation --*
Integration within ED-80/DO-254 design life cycles	Tool assessment and qualification: • Independent assessment of tool output: • Post-place and route simulation • Visual inspection of STA report (most common approach) • Physical tests on programmed device,	Tool assessment and qualification: • Independent assessment of tool output: • Post-place and route simulation • Visual inspection of STA report (most common approach) • Physical tests on programmed device,	Tool assessment and qualification: • Independent assessment of tool output: • Post-place and route simulation • Visual inspection of STA report (most common approach) • Physical tests on programmed device,

Static Verification CAE Tools: Static Timing Analysis			
	including endurance and robustness tests • Relevant service experience	including endurance and robustness tests • Relevant service experience	including endurance and robustness tests • Relevant service experience

Table 5.5. Commercial Static Timing Analysis Tools Assessment

Dynamic Verification CAE Tools: HDL Simulators			
Tool	Active HDL	ModelSim	
Tool Vendor	Aldec	Mentor Graphics	
Benefits	Provides different levels of verification (from RTL to gate level) High level of observability and accessibility to the DUV (including error injection) Provides all code coverage metrics required for elemental analysis Supports advanced features (assertion-based languages)	Provides different levels of verification (from RTL to gate level) High level of observability and accessibility to the DUV (including error injection) Provides all code coverage metrics required for elemental analysis Supports advanced features (assertion-based languages)	
Limitations	Timing is not verified → STA is also required Depends on PLD vendor technology libraries and simulation models → assessment and physical tests on the programmed device are required Potential lack of exhaustiveness: Verification coverage depends on requirement-based test cases Post-place and route simulations are time-consuming	Timing is not verified → STA is also required Depends on PLD vendor technology libraries and simulation models → assessment and physical tests on the programmed device are required Potential lack of exhaustiveness: Verification coverage depends on requirement-based test cases Post-place and route simulations are time-consuming	
Technical Issues	*-- Not enough data from the consultation --*	Issue: Different code coverage results depending on simulation options Severity: **Low** Mitigation: -- Issue: Different simulation results when -opt engine option used Severity: **High** Mitigation: -opt engine disabled Issue: Default resolution leads to wrong DCM behaviour Severity: **High** Mitigation: Set resolution indicated by PLD vendor for DCM	
Integration	Tool assessment and qualification:	Tool assessment and qualification:	

Dynamic Verification CAE Tools: HDL Simulators			

within ED-80/DO-254 design life cycles	• Independent assessment of tool output:
	  • Visual inspection
	  • Physical tests on programmed device
	  • Two simulators running in parallel (most common approach)
	• Relevant service experience data available
		• Independent assessment of tool output:
		  • Visual inspection
		  • Physical tests on programmed device
		  • Two simulators running in parallel (most common approach)
		  • Comparison of simulation results at different levels (RTL against post-place and route)
		• Relevant service experience data available	

Table 5.6. Commercial HDL Simulation Tools Assessment

5.3.2 Tools Needing Generic Assessment

This section covers the tools needing a generic assessment. These types of tools were described in the first task of the SHARDELD study, and are identified in ED-80/DO-254 Appendix B, "Design assurance considerations for level A and B functions" §3.3, as advanced verification methods: Safety-specific analysis and formal methods.

This group includes the following types of tools:

- Dynamic Verification: Dynamic ABV
- Static Verification:
  - HDL Rule Checkers
  - CDC Analysers
  - LEC
  - Formal Model Checkers

Table 5.7 provides a comparison table for each type of tool, including the assessment of their "Degree of integration for the development of programmable AEH" based on the preliminary results of the consultation to relevant aircraft and equipment manufacturers.

Alternative and/or Complementary CAE Tools				
Tool Type	***Degree of integration for the development of programmable AEH***	***Benefits***	***Limitations***	***Integration within ED-80/DO-254 design life cycles***
Dynamic ABV	Used by less than 30% of companies consulted to get certification credit in some cases The most common approach is an standard HDL simulator supporting PSL assertion language	Improves the observability of the design under verification Assertions can be propagated from RTL to gate level, enabling post-synthesis and post-place and route simulations Standardization of assertion languages No special consideration for tool assessment and qualification (same criteria as HDL simulators)	Strong dependence on properties defined by the user Lack of properties coverage	Same tool assessment and qualification criteria applicable to HDL Simulators
HDL Rule checkers	Used by less than 30% of companies consulted to get certification credit in some cases Basic tool qualification as the sole tool assessment approach	Reduces the effort for RTL code review against coding rules The output reports may be used for documentation purposes	Does not replace code reviews (RTL code should also be reviewed against its functional requirements) Tool assessment may require a basic tool qualification	Manual review of the output reports Basic tool qualification may be required
CDC Analysers	Used informally (technical decisions), not for getting certification credit	Automated metastability injection Reduces the effort for the identification of multiple	Only performed at RTL level (design tools may introduce errors in CDC paths flagged as correct by the CDC tool)	Manual review of the output reports Basic tool qualification may be required

Alternative and/or Complementary CAE Tools				
	CDC paths are analysed with standard STA tools	CDC paths in large designs	Large output reports including false negatives Tool assessment may require a basic tool qualification	
LEC	Not used because of lack of tool maturity and high complexity of both the tool and its qualification	Provides evidences that design tools produced an output equivalent to RTL source May reveal non-used RTL HDL code and/or unreachable logic Combined with STA, LEC can lighten post-place and route simulation	Some vendor-specific macros are may not be supported, introducing false differences Some synthesis optimization techniques are not support, introducing false differences Sensible common points between LEC and synthesis tool (loss of independence required for tool assessment)	Manual review of the output reports Basic tool qualification may be required
Formal Model Checkers	Not used because of lack of tool maturity and high complexity of both the tool and its qualification	Comprehensive coverage Increases the verification of robustness (exercising corner cases difficult to define with requirement-based test cases)	Low support among PLD vendors Lack of maturity Tool assessment and qualification difficult to achieve	Manual review of the output reports Basic tool qualification may be required

Table 5.7. Alternative and/or Complementary Tools Assessment

5.4 Conclusions of Task 2

The conclusions of the second task of the SHARDELD study are based on the results of the consultation of aircraft and equipment manufacturers.

Three PLD vendors (Altera, Microsemi and Xilinx) are widely use as providers of custom micro-coded components for programmable AEH.

Regarding synthesis tools, the consultation revealed that:

- In most cases, third vendor synthesis tools and synthesis functions embedded in the PLD vendor design environment (IDE) are used indistinctly
- Most configuration settings are left to the default value, while the ones which are modified are documented in the Hardware Design Standards of each company
- Concerning additional functions:
  - ✔ Functions related to Finite State Machines (FSM) such as FSM extraction and FSM encoding are commonly used
  - ✔ Fan-out limit is also a common option
  - ✔ Regarding optimization methods, in most cases all are disabled. Methods such as pipelining and register duplication were never mentioned by any company
  - ✔ Incremental synthesis and Triple Modular Redundancy (TMR) are not used, claiming difficulties to get certification credit and issues concerning verification complexity and logic resources consumption
- Technical design issues with the synthesis tool were detected during verification process

The answers to the questionnaire about place and route tools showed that:

- As with synthesis tools, most configuration settings are left to the default value, while the ones which are modified are documented in the Hardware Design Standards of each company
- Concerning additional functions:
  - ✔ Graphical constraint editors, routing optimization and timing-driven place and route are generally used by most of the companies consulted
  - ✔ The rest of functions are rarely used, or they are used informally
- In terms of embedded IDE tools:
  - ✔ The integrated STA tool is used by almost all the companies consulted
  - ✔ Other functions such as the power consumption analyser or the embedded logic analyser are used for debugging purposes, therefore no certification credit is claimed
- Technical design issues with the place and route tool were detected during verification process

STA tools integrated within the PLD vendor design environment are, in almost all cases, the first and sole choice of design teams when performing static timing analysis. This trend confirms the advantage of STA tools from PLD vendor against third party tools highlighted in the conclusions of the Task 1 of this study.

With regard to HDL Simulators, the consultation highlighted the following facts:

- Most configuration settings are left to the default value, while the ones which are modified are documented in the Hardware Design Standards of each company
- All the companies consulted use third party HDL simulation tools. Simulator functions embedded in IDE are not used due to low performance and lack of support for advanced features
- Code coverage is always used to assess the completion of verification testing (elemental analysis)
- Verification issues with code coverage results depending on the compilation options fixed in subsequent versions of the tool or mitigated with good HDL coding practices
- Technical issues: Some companies highlighted the tool errata document provided by the tool vendor as a useful mean to mitigate those errors

The questionnaire to determine the different approaches and criteria for tool assessment and qualification suggested that:

- Independent tool output assessment is the most common approach, followed by relevant history of the tool
- In case of performing the independent tool output assessment of synthesis, place and route, and STA tools, the simulation of the back-annotated HDL model and the physical tests on the programmed device are almost the sole methods. Formal methods such as Logic Equivalence Checking (LEC) were not used
- Regarding STA tools, the visual inspection of the tool report is also done
- For HDL simulation tools, the use of two simulators in parallel under the same verification environment is the most common approach. Manual review of the tool output is also a quite common approach
- Two additional tool assessment and qualification approaches are used by some companies for STA tools:
  - ✔ Endurance and robustness testing
  - ✔ Environmental testing of the physical hardware embedding the programmed device
- Additional tool assessment and qualification approaches used by some companies for HDL simulation tools: Comparison of simulation results at different levels (RTL against post-place and route models)

The consultation results regarding the alternative and/or complementary tools used in the development of programmable AEH indicated that:

- Two methods are being evaluated -and in some cases used to get certification credit- by some of the companies consulted:
  - ✔ HDL rule checkers
  - ✔ Dynamic ABV: Standard HDL simulators supporting PSL assertion language are used as the sole approach for this verification method
- Clock Domain Crossing (CDC) is manually checked at RTL level, therefore no specific CDC tool is used for this purpose
- The main reasons for not claiming certification credit when using these methods are the

difficulty for tool assessment and qualification and the lack of maturity

Finally, it should be signalled that DAL criteria is taken into account when selecting the programmable device technology. For DAL A and B applications, SEU immunity is the most applicable criteria, making Flash-based technology (Actel) and Structured ASICs (HardCopy Series from Altera) two of the most common technologies for this level of design assurance.

6. Task 3: Design Tools Usage Assessment

This task aims to perform the specific assessment in terms of configuration options and embedded functionalities of selected commercial CAE tools, highlighting their benefits, limitations and risks in terms of safety.

The assessment also aims at defining a list of best practices, recommendations and guidelines to maximize the tools effectiveness while reducing the safety risks. From this list, some recommendations will be derived to amend the EASA Certification Memorandum "Development Assurance of Airborne Electronic Hardware" CM-SWCEH-001 [EXT-02].

6.1 Assessment of Configuration Options and Embedded Functionalities of Selected CAE Tools

This section aims to perform the specific assessment in terms of both configuration options and embedded functionalities of the relevant commercial CAE tools selected after analysing the results of the consultation to relevant aircraft and equipment manufacturers.

For each configuration option and embedded functionality, the safety risk is also assessed and classified in three categories:

- Low: There is low impact in the implementation in terms of performance and no potential errors
- Medium: An inappropriate use may lead to potential errors that can be easily detected in the verification process and/or by reviewing the tool outputs
- High: An inappropriate use may lead to potential errors very difficult to detect

This assessment is done on the following selected commercial CAE tools:

- Synthesis tools:
  - ✔ Quartus II Integrated Synthesis (Altera)
  - ✔ Xilinx Synthesis Technology - XST (Xilinx)
  - ✔ Synplify Pro (Synopsys)
- Place and route tools / Integrated Development Environments (IDE):
  - ✔ Quartus II (Altera)
  - ✔ ISE (Xilinx)
  - ✔ Libero IDE (Actel)
- STA tools:
  - ✔ TimeQuest (Altera)
  - ✔ Trace Timing Analyzer (Xilinx)
  - ✔ SmartTime (Actel)
- HDL simulators:

- ✔ ModelSim (Mentor Graphics)
- ✔ Active HDL (Aldec)

6.1.1 Synthesis Tools

6.1.1.1 Quartus II Integrated Synthesis (Altera)

Quartus II Integrated Synthesis is the synthesis tool included within Altera's Quartus II IDE, and includes advanced integrated synthesis supporting VHDL, Verilog HDL and Altera-specific design entry languages. The tool can also write VHDL and Verilog netlists after synthesis for further simulation.

The specific assessment is performed using as a reference the version 11.1 of Altera's Quartus II software.

Table 6.1 summarizes its synthesis configuration options and functions along with their benefits, limitations and recommendations for use.

Synthesis Tools: Quartus II Integrated Synthesis - Altera				
#	*Option/Function*	*Description*	*Benefits*	*Limitations and Recommendations*
01	Optimization Technique	Specifies the goal for logic optimization during compilation Values: • Area • Balanced (default) • Speed	Allows to establish a design implementation strategy tightly coupled with the place and route process The *Area* option makes the design as small as possible, the *Balanced* tries to make a trade-off between high performance and minimal logic usage, while the *Speed* option chooses the fastest design implementation	It is recommended to set this option to its default value (*Balanced*), even if this setting is not available for all device families, in order to enable the trade-off between high performance and minimal logic usage Safety risk: **Low**
02	Auto Gated Clock Conversion	Automatically converts gated clocks in the design to use clock enable pins if available Values: • On • Off (default)	This option may be useful to prototype ASICs using FPGAs	It is recommended to set this option to its default value (*Off*) since gated clocks are highly discouraged in most Hardware Design Standards Safety risk: **Medium**
03	Auto Clock Enable Replacement	Allows the tool to find logic that feeds a register and move the logic to the register's clock enable input port Values: • On (default) • Off	When enabled (*On*), this function decreases logic element usage	It is recommended to disable this option (*Off*) to avoid negative impact in the place and route process Safety risk: **Low**
04	Auto Resource Sharing	Allows the tool to share hardware resources among many similar, but mutually exclusive, operations in the RTL HDL code Values: • On • Off (default)	When enabled (*On*), the tool merges compatible addition, subtraction, and multiplication operations in order to reduce the area required	It is recommended to disable this option (*Off*) since this option may negatively impact the timing performance Resource sharing combined with poor coding style may force the tool to implement more arithmetic operation than is required, reducing the overall performance of the

Synthesis Tools: Quartus II Integrated Synthesis - Altera				
				design In addition, this optimization may not be compatible with LEC Safety risk: **High**
05	Auto Shift Register Replacement	Allows the tool to find a group of shift registers of the same length that can be replaced with an Altera shift register macro Values: • Always • Auto (default) • Off	This option is useful for finding areas of the design that can be implemented more efficiently in order to minimize area and maximize speed	It is recommended to disable this option (*Off*) since the Altera shift register macro may use embedded block RAM to replace registers Implement logic into memory blocks may reduce the reliability of the whole design since memory blocks are more sensitive to SEU If the Altera shift register macro is required, it is preferable to instantiate it directly in the RTL HDL code In addition, this optimization may not be compatible with LEC Safety risk: **High**
06	Ignore *translate_off* and *synthesis_off* Directives	Instructs the tool to ignore all translate_*off* and *synthesis_off* directives in the RTL HDL code Values: • On • Off (default)	The translate_*off* and *synthesis_off* directives allows to synthesize designs originally written for use with other synthesis tools without needing to modify the source code	It is highly recommended to enable this option (*On*) since the RTL HDL code between these two directives is not synthesized and therefore may generate a different implementation with an unexpected behaviour Safety risk: **High**
07	Infer RAMs from Raw Logic	Instructs the tool to infer RAMs from registers and multiplexers Values: • On (default) • Off	When enabled (*On*), this option may optimize the resource logic used to implement the design It can also be useful to increase the portability of the RTL HDL code to other target devices	It is recommended to disable this option (*Off*) to control how the tool uses available memory resources A good design practice is to instantiate the Altera memory macros directly in the RTL HDL code, in order to keep control of the implemented resources even if this practice reduces the portability of the code to different target devices Implement logic into memory blocks may reduce the reliability of the whole design since memory blocks are more sensitive to SEU In addition, this optimization may not be compatible with LEC Safety risk: **High**
08	Parallel Synthesis	Option to enable parallel synthesis Values: • On (default) • Off	When enabled (*On*), this option directs the tool to use more than one processor to execute the synthesis process	Incremental compilation should be used to take advantage of this option It is recommended to set this option to its default value (O*n*) to reduce synthesis runtime

Synthesis Tools: Quartus II Integrated Synthesis - Altera				
				Safety risk: **Low**
09	PowerPlay Power Optimization	Determines how aggressively the tool optimizes the design for power Values: • Extra Effort • Normal Compilation (default) • Off	When *Normal Compilation* is set, the tool performs the following power optimizations, as long as there is no expectation of design performance reduction: • Minimization of number of memory blocks accessed during each clock cycle • Rearrangement of logic to eliminate nets with high toggle rates The *Extra Effort* option performs the following additional power optimizations: • Shut down memory blocks that are not accessed • Power-aware memory balancing	It is recommended to set this option to its default value (*Normal Compilation*) to avoid the additional power optimizations which may affect design performance It is also recommended to apply power reduction techniques directly in the RTL HDL code such as: • Adding control logic to handle Block RAM enable signals • Using LUT instead of Block RAM for small memory blocks • Control over counters Safety risk: **Low**
10	Restructure Multiplexers	Reduces the number of logic elements required to implement multiplexers in a design Values: • Auto (default) • On • Off	When enabled (*On*), this function decreases logic element usage The *Auto* option allows the tool to determine whether multiplexer restructuring should be enabled depending on the selected Optimization Technique	It is recommended to set this option to its default value (*Auto*) to avoid the optimizations which may negatively affect design clock speed Safety risk: **Medium**
11	Timing-Driven Synthesis	Allows the tool to use timing information -extracted from the timing constraint file (SDC)- during synthesis to better optimize the design Values: • On (default) • Off	Timing-Driven Synthesis increases performance by improving logic depth on critical portions of the design This option works along with the selected Optimization Technique : • Area: Optimization only for area • Balanced: Optimization of timing-critical portions, allowing only limited area increase • Speed: Optimization of timing-critical portions at the cost of increasing area	It is recommended to set this option to its default value (*On*) to enable the optimization To get better results: • Avoid asynchronous structures if possible • Provide a complete and accurate timing constraint file (SDC) • The specific target device should be selected Safety risk: **Low**
12	SDC Constraint Protection	Specifies whether the tool should protect registers from merging when they have incompatible timing constraints Values: • On • Off (default)	Potential improvements in design speed	It is recommended to set this option to its default value (*Off*) since the Timing-Driven Synthesis already covers this function Safety risk: **Low**

Synthesis Tools: Quartus II Integrated Synthesis - Altera				
13	State Machine Processing	Defines the processing style to synthesize a FSM Values: • Auto (default) • Gray • Johnson • Minimal Bits • One-hot • Sequential • User-Encoded	Allows to select the FSM encoding algorithm in order to improve area and timing The default value (*Auto*) uses one-hot encoding for FPGA devices and minimal-bits encoding for CPLDs	When enable, the State Machine Processing optimizes each FSM by removing all states and transition logic that can not be reached. The result is an FSM which can not recover from an invalid state caused by some external influence (SEU), asynchronous inputs, or a physical failure in the device In order to implement more reliable FSM, it is highly recommended to set the State Machine Processing as *User-Encoded* and encode the states of each FSM at RTL level in order to get an exact implementation of the RTL HDL code, including the expected recovery state (covered by the "others" branch of the *case* statement) Safety risk: **High**
14	Extract VHDL/Verilog State Machines	Allows the tool to extract FSM from VHDL/Verilog design files Values: • On (default) • Off	When enabled, this option allows the tool to optimize the extracted FSM to reduce area and/or improve performance When disabled, the FSM are extracted and optimized as regular logic	It is highly recommended to disable this option (*Off*) to prevent automatic FSM optimizations that may produce FSM which can not recover from invalid states Safety risk: **High**
15	Safe State Machine	Instructs the tool to implement FSM that can recover from illegal states Values: • On • Off (default)	Instructs the tool to insert extra logic to detect illegal states, and force the transition of the FSM to the recovery state The only safe recovery state forced by the tool is the reset state	It is highly recommended to set this option to its default value (*Off*) and encode the states of each FSM at RTL level in order to get an exact implementation of the RTL HDL code, including the expected recovery state (covered by the "others" branch of the *case* statement) which may be different from the reset state [EXT-04] Safety risk: **High**
16	Power-Up Don't Care	Causes registers that do not have a Power-Up Level logic option setting to power up with a don't care logic level ('X') Values: • On (default) • Off	When enabled, this option allows the tool to change the power up level of registers that do not have a defined power up condition if they can be removed with this change of level, in order to minimize the area of the design	It is recommended to disable this option (*Off*) because not all third-party synthesis tools support this optimization technique It is also recommended to use reset or preset signals rather than depending on the power-up condition Safety risk: **Low**
17	Remove Duplicate Registers	Enables or disables removal of identical registers described at RTL level Values: • On (default)	Improves area, increasing the probability that the design will fit on the device	It is recommended to set this option to its default value (*On*) to perform a first run, even if the processing time increases due to FF optimization, and check the synthesis report to identify the

Synthesis Tools: Quartus II Integrated Synthesis - Altera				
		• Off		equivalent registers removed by this option. Then, the duplicated registers identified should be removed from the RTL HDL code. At this point, the option can be disabled (*Off*) This option is not compatible with some techniques used to increase robustness such as TMR Safety risk: **High**
18	Preserve Registers	Instructs the tool not to minimize or remove a specified register during synthesis optimizations This option is applicable at register level or to design entity, and can be set in the Quartus II Assignments Editor Values: • On (default) • Off	This option is useful to preserve FFs with specific purposes, and it is ignored if the register does not drive anything	It is recommended to enable this option (*On*) to preserve intentional redundant registers when the *Remove Duplicate Registers* option is enabled Safety risk: **Medium**
19	Disable Register Merging	Prevents the specified register from merging with other registers and vice versa This option is applicable at register level or to design entity, and can be set in the Quartus II Assignments Editor Values: • On (default) • Off	This option may be used to instruct the tool to use the user-specific timing constraints on the register during synthesis This option does not prevent a register from being removed (as Preserve Registers option does)	It is recommended to use this option (*On*) when user-specific multicycle timing constraints are used Safety risk: **Medium**
20	Disable Register Merging Across Hierarchies	Specifies whether registers that are in different hierarchies are allowed to be merged if their inputs are the same Values: • Auto (default) • On • Off	Improves area, increasing the probability that the design will fit on the device	
21	Allow Shift Register Merging Across Hierarchies	Allows the tool to take shift registers from different hierarchies of the design and put them in the same RAM Values: • Always • Auto (default) • Off	Optimizes the usage of embedded memory resources	It is recommended to disable this option (*Off*) to control how the tool uses available memory resources Safety risk: **Low**
22	Allow Synchronous	Allows the tool to use synchronous clear and/or	This option helps to reduce the total number logic cells used in	It is recommended to disable this option (*Off*) to avoid negative

Synthesis Tools: Quartus II Integrated Synthesis - Altera				
	Control Signals	synchronous load signals in normal mode logic cells Values: • On (default) • Off	the design	impact in the place and route process, since synchronous control signals are shared by all logic cells in a LAB (Logic Array Block – basic building block in Altera devices) Safety risk: **Low**
23	Preserve Hierarchical Boundary	This option is available only in Quartus II software versions 8.1 and earlier	n.a.	To preserve hierarchy, Altera recommends using design partitions together with incremental compilation Safety risk: **Low**
24	Allow Any RAM/ROM/Shift Register For Recognition	Allows the tool to infer RAM/ROM/Shift Registers of any size, even if they do not meet the current minimum requirements Values: • On • Off (default)	This option may be useful to get a preliminary estimation of the implementation results	The use of this option is intended for preliminary estimation purposes, therefore it is highly recommended to set this option to its default value (*Off*) Safety risk: **Medium**
25	Synchronization Register Chain Length	Specifies the maximum number of registers in a row to be considered as a synchronization chain Value: Any non-negative integer (2 is the default value)	Protects the synchronization FF from gate level retiming and duplication	It is recommended to set the value to the synchronization chain length used in the design, to avoid modifications of the synchronization register chain due to retiming and/or duplication optimizations Safety risk: **High**
26	HDL Message Level	Specifies the type of HDL messages to be displayed, including messages that display processing errors in the RTL HDL code Values: • Level 1 • Level 2 (default) • Level 3	*Level 1* allows to display only the most important HDL messages. Warning messages are generated if there is a high probability that it points to an actual design problem *Level 2* allows to display additional messages that identify possible design problems *Level 3* allows to display all HDL info and warning messages. This level includes extra messages that suggests changes to improve the readability, style, or portability of the RTL HDL code	It is highly recommended to set the *HDL Message Level* to *Level 3* in order to display all info and warning messages. Some of these messages that are not reported as errors might be critical Safety risk: **High**
27	Perform WYSIWYG* primitive resynthesis	Specifies whether to perform WYSIWYG primitive resynthesis during synthesis, using the setting specified in the Optimization Technique option Values: • On • Off (default)	This option may be useful for resynthesizing WYSIWYG primitives in the design for better area and/or performance	It is recommended to set this option to its default value (*Off*) to avoid the modification of EDIF input files generated by other EDA tools Safety risk: **Medium**

Synthesis Tools: Quartus II Integrated Synthesis - Altera				
28	Synthesis Effort	Specifies the overall synthesis effort level of the tool Values: • Auto (default) • Fast	*Fast* option may be used to perform early timing estimations	It is highly recommended to set this option to its default value (*Auto*), to allow the tool to perform all the process steps *Fast* option may omit some steps to accomplish synthesis more quickly, having an impact in design performance and resource cost Safety risk: **Medium**
29	Synthesis Seed	Specifies the seed that the tool uses to randomly do synthesis in a slightly different way Value: Any non-negative integer (1 is the default value)	The synthesis seed can be modified when a design is close to meeting requirements, in order to get a slightly different result	It is highly recommended to set the *Synthesis Seed* to a fixed value when timing requirements are met, in order to ensure the reproducibility of the implementation process The use of synthesis scripts that changes the seed based on dynamic parameters such as date and time is also highly discouraged Safety risk: **High**
30	VHDL Version	Allows to select the VHDL version for a VHDL project Values: • VHDL 1987 • VHDL 1993 (default) • VHDL 2008	n.a.	Set the standard used in the RTL HDL code Safety risk: **Low**
31	Verilog Version	Allows to select the standard for a Verilog project Values: • Verilog-1995 • Verilog-2001 (default) • System Verilog-2005	n.a.	
32	Default Parameters	Specifies the default settings for the parameters used for synthesis	Allows to define default settings for synthesis parameters	Assignments in design files or assignments made through the tool GUI will override these defaults Safety risk: **Low**

Table 6.1. Quartus II Integrated Synthesis Tool Specific Assessment of Options and Functionalities

*WYSIWYG acronym stands for “What You See Is What You Get”

Following design-specific files should be subject to revision control:

- Project file (including the name and location of all the RTL HDL files to be synthesized and the synthesis options)
- RTL HDL source code
- Synopsys Design Constraints File (SDC)

- Tcl script to guide the synthesis process (if any)
- Design technology libraries

6.1.1.2 Xilinx Synthesis Technology - XST (Xilinx)

XST is the synthesis tool integrated within Xilinx ISE design environment and synthesizes VHDL, Verilog and mixed language designs to create Xilinx-specific netlists. The tool can also write VHDL and Verilog netlists after synthesis for further simulation.

The specific assessment is performed using as reference the version 12.4 of Xilinx ISE software.

Table 6.2 summarizes its synthesis configuration options and functions along with their benefits, limitations and recommendations for use.

Synthesis Tools: XST - Xilinx				
Synthesis Options				
#	*Option/Function*	*Description*	*Benefits*	*Limitations and Recommendations*
01	Optimization Goal	Defines the synthesis optimization strategy at global level Values: • Speed (default) • Area	Allows to establish a design implementation strategy tightly coupled with the place and route process *Speed* goal reduces the number of logic levels, while the *Area* option reduces the total amount of logic	Some options may not have an optimal result depending on the selected synthesis optimization strategy Therefore the *Optimization Goal* should be adapted to the design requirements Safety risk: **Low**
02	Optimization Effort	Defines the synthesis optimization effort level at global level Values: • Normal (default) • High • Fast	Instructs the tool to perform additional optimization techniques depending on the desired processing time	*High* optimization effort may degrade the implementation performance, while the *Fast* option only leads to faster processing times Xilinx highly recommends to set this option to its default value (*Normal*) Safety risk: **Medium**
03	Power Reduction	Enables synthesis optimization techniques to reduce power consumption Values: Check box disabled by default	Allows to minimize power consumption reducing the number of simultaneously active block RAM elements by using RAM enable features	It is recommended to set this option to its default value (disabled) to keep full control on the RAM implementation It is also recommended to apply power reduction techniques directly in the RTL HDL code such as: • Adding control logic to handle Block RAM enable signals • Using LUT instead of Block RAM for small memory blocks • Control over counters Safety risk: **Low**
04	Use Synthesis Constraint File	Instructs the tool to process the user XST Constraint File (XCF)	Allows to enable the user constraint file	It is highly recommended to use at least one constraint file including user defined timing constraints

Synthesis Tools: XST - Xilinx				
		Values: Check box enabled by default		Safety risk: **Medium**
05	Keep Hierarchy	Defines the generation of a hierarchical netlist, preserving the HDL hierarchy through the synthesis and place and route processes Values: • Yes • No (default) • Soft	Preserving the hierarchy may speed up processing When this option is disabled, the result is a flatten design which can improve the synthesis results by optimizing the entity boundaries The *Soft* option keeps hierarchy only at synthesis level	It is recommended to set this option to its default value (No) to improve the implementation results Safety risk: **Low**
06	Netlist Hierarchy	Controls the form in which the final netlist is generated, allowing a hierarchical netlist despite the value of the Keep Hierarchy option Values: • As Optimized (default) • Rebuilt	This option may optimize the design implementation by grouping basic inferred macros together with higher complexity macros	It is recommended to set this option to its default value (*As Optimized*) to not override the *Keep Hierarchy* option value Safety risk: **Low**
07	Global Optimization Goal	Allows to optimize the design by regions in terms of timing Values: • AllClockNets (default) • Inpad To Outpad • Offset In Before • Offset Out After • Maximum Delay	*AllClockNets* option optimizes the period of the entire design *Inpad To Outpad* optimizes the maximum delay from input pad to output pad *Offset In Before* optimizes the maximum delay from input pad to clock *Offset Out After* optimizes the maximum delay from clock to output pad *Maximum Delay* incorporates all previously mentioned constraints	These timing constraints are globally applied to the entire design and are overridden by constraints specified in the User Constraint File (UCF) It is recommended to set this option to its default value (*AllClockNets*) or to *Maximum Delay* and define a comprehensive User Constraint File (UCF) Safety risk: **Medium**
08	Generate RTL Schematic	Allows the tool to generate a netlist file representing a Register Transfer Level (RTL) design structure Values: • Yes (default) • No • Only	There is no impact on the implementation The netlist file can be viewed with the RTL Viewer or the Technology Viewer tools for visual inspection of the synthesized design	There is no impact on the implementation The netlist file can be viewed with the RTL Viewer or the Technology Viewer tools for visual inspection of the synthesized design Safety risk: **Low**
09	Read Cores	Allows the tool to read EDIF and NGC core files for timing estimation and device utilization control Values: Check box enabled by default	The tool can optimize better the logic around the core	For some cores, such as the PCI core, requires a different optimization of their interconnection logic. Therefore this option should be disabled when using these cores Safety risk: **Low**
10	Write Timing Constraints	Allows the tool to write timing constraints to the synthesized	Potential improvements in optimization	It is recommended to set this option to its default value (disabled) and

Synthesis Tools: XST - Xilinx				
		file (NGC) Values: Check box disabled by default		define the timing constraints for the place and route process in the User Constraint File (UCF) Safety risk: **Medium**
11	Cross Clock Analysis	Allows the tool to perform timing optimizations across clock domains Values: Check box disabled by default	Potential improvements in optimization	It is highly recommended to set this option to its default value (disabled) to avoid undesired optimizations on the synchronization logic One of the recommendations of this study concerning CDC paths is to constraint the synthesis tool to ensure that no optimization is made on them (refer to § 4.2.2.6) Safety risk: **High**
12	LUT-FF Pairs Utilization Ratio	Defines the area size of LUT-FF pairs that the tool should not exceed during timing optimization Values: A percent of total numbers (100% is set by default)	This option can be useful when the target device logic resources (area) may be not enough to implement the design	It is recommended to set this option to its default value (100%) to avoid additional area constraints on the design A good design practice is to foresee a target device with enough logic resources to keep a comfortable margin and avoid logic-reduction techniques such as mapping logic to block RAM Safety risk: **Low**
13	BRAM Utilization Ratio	Defines the number of block RAM components that the tool should not exceed during synthesis Values: A percent of total numbers (100% is set by default)	This option allows to reserve block RAM for inferred RAM components	It is recommended to set this option to its default value (100%) to avoid unexpected use of distributed RAM (logic block configured as memory) instead of block RAM (dedicated memory) A good design practice is to define the type of RAM (block/distributed) at RTL level. Block RAM are more recommended for large sized memories, while distributed RAM should be used for small sized memories and FIFOs It is important to note that they are operated differently: In distributed RAM the read operation is asynchronous while in block RAM is synchronous Safety risk: **Medium**
14	DSP Utilization Ratio	Restricts the number of DSP blocks that the tool uses to implement inferred functions Values: A percent of total numbers (100% is set by default)	This option is used in collaborate workflows to reserve DSP resources for each separate component	It is recommended to set this option to its default value (100%) and foresee the available DSP resources in a early stage of the design flow, ensuring that the selected device can accommodate all instantiated DSP blocks

Synthesis Tools: XST - Xilinx				
				The option is overridden in DSP implementation is forced on inferred macros Safety risk: **Medium**
15	Generic, Parameters	Redefine Generics (VHDL) and Parameters (Verilog) in the top-level design block	Allows to modify the design without modifying the source code	It is highly recommended not to use this option, and make these modifications at RTL level instead of at synthesis level to keep a better traceability of the project baseline an avoid implementations with an unexpected configuration Safety risk: **High**
16	Verilog Macros	Redefine Verilog macros	Allows to modify the design configuration without modifying the source code	It is recommended not to use this option, and make these modifications at RTL level instead of at synthesis level to keep a better traceability of the project baseline an avoid implementations with an unexpected configuration Safety risk: **High**
HDL Options				
#	***Option/Function***	***Description***	***Benefits***	***Limitations and Recommendations***
17	Automatic FSM Extraction and Encoding Algorithm	Enables FSM extraction and specific synthesis optimizations Values: • Auto (default) • One-hot • Compact • Sequential • Gray • Johnson • speed1 • None	Allows to select the FSM encoding algorithm in order to improve area and timing The default value (*Auto*) lets the tool to select the best coding technique for each individual state machine	When enable, the FSM extraction optimizes each FSM by removing all states and transition logic that can not be reached. The result is an FSM which can not recover from an invalid state caused by some external influence (SEU), asynchronous inputs, or a physical failure in the device In order to implement more reliable FSM, it is highly recommended to disable the Automatic FSM Extraction (*None*) and encode the states of each FSM at RTL level in order to get an exact implementation of the RTL HDL code, including the expected recovery state (covered by the "others" branch of the *case* statement) Safety risk: **High**
18	Safe Implementation	Implements FSM in Safe Implementation mode, generating additional logic that forces the FSM to a valid state (recovery state) Values: • Yes • No (default)	Avoids the tool to remove the transition logic for unreachable states A safe recovery state is defined by the tool (reset state or power-up state if the FSM does not have an initialization signal) or by the user	It is highly recommended to set this option to its default value (*No*) and encode the states of each FSM at RTL level in order to get an exact implementation of the RTL HDL code, including the expected recovery state (covered by the "others" branch of the *case* statement) [EXT-04]

Synthesis Tools: XST - Xilinx				
				Safety risk: **High**
19	Case Implementation Style	Instructs the tool how to interpret Verilog *case* statements Values: • None (default) • Full • Parallel • Full-parallel	This option (for Verilog designs only) avoids latches inferred by uncompleted *case* statements (*Full* and *Full-parallel* values). It is also used to avoid priority encoders (*Parallel* and *Full-parallel* values)	It is highly recommended to set this option to its default value (*None*) to force the tool to implement the exact behaviour of the *case* statements, which has already been reviewed Safety risk: **High**
20	FSM Style	Allows to select the type of resources (LUT or block RAM) for the implementation of FSM Values: • LUT (default) • Block RAM	FSM implemented with block RAM resources may be faster and more compact	It is recommended to set this option to its default value (*LUT*). Block RAM resources are more sensitive to phenomena such as SEU than LUT Safety risk: **High**
21	RAM Extraction	Enables or disables the inference of RAM macros Values: • Yes (default) • No	When enabled, this option allows the inference of RAM macros	It is recommended to disable this option (*No*) and declare at RTL level the type of RAM for each entity Safety risk: **Low**
22	RAM Style	Controls the way the tool implements the inferred RAM macros Values: • Auto (default) • Distributed • Block	Allows the user to force the use of distributed or block RAM resources to implement inferred RAM	This option is not applicable when RAM Extraction is disabled as recommended In case RAM Extraction is enabled, then it is recommended to set this option to its default value (*Auto*) to allow the tool the selection of the more appropriate resource in terms of HDL description style (synchronous or asynchronous data read) and availability Safety risk: **Low**
23	ROM Extraction	Enables ROM macro inference from *case* statements in which all assigned contexts are constant values Values: • Yes (default) • No	When enabled, this option allows the inference of ROM macros	It is recommended to disable this option (*No*) and declare at RTL level the type of ROM for each entity Safety risk: **Low**
24	ROM Style	Controls the way the tool implements the inferred ROM macros Values: • Auto (default) • Distributed • Block	Allows the user to force the use of distributed or block RAM resources to implement inferred ROM	This option is not applicable when ROM Extraction is disabled as recommended In case ROM Extraction is enabled, then it is recommended to set this option to its default value (*Auto*) to allow the tool the selection of the more appropriate resource in terms of HDL description style (synchronous or asynchronous data

Synthesis Tools: XST - Xilinx				
				read) and availability Safety risk: **Low**
25	Automatic BRAM Packing	Packs two small block RAM components into a single block RAM primitive as dual-port block RAM Values: • Yes • No (default)	Potential improvement in terms of block RAM usage	It is recommended to set this option to its default value (*No*) to avoid unexpected behaviour of the implemented memory structures Safety risk: **Medium**
26	Shift Register Extraction	Enables the inference of Shift Register macro Values: • Yes (default) • No	Allows to use dedicated hardware resources to implement shift registers	It is recommended to set this option to its default value (*Yes*) and check the inferred macro The inferred shift register macros use LUT resources instead of Block RAM Safety risk: **Medium**
27	Resource Sharing	Enables or disables resource sharing of arithmetic operators Values: • Yes (default) • No	Minimizes the number of operators. This optimization uses two similar arithmetic resources which are never used at the same time to implement a single operator In most cases resource sharing improves area	It is recommended to disable this option (*No*) since this option may negatively impact the timing performance Resource sharing combined with poor coding style may force the tool to implement more arithmetic operation than is required, reducing the overall performance of the design In addition, this optimization may not be compatible with LEC Safety risk: **High**
28	Use DSP Block	Enables or disables the use of DSP block resources Values: • Auto (default) • Yes • No	Allows the use of DSP blocks to implement arithmetic logic (especially for multipliers and accumulators) to optimize the overall use of logic resources	Structures around the arithmetic logic should be protected to prevent their implementation within the DSP block It is recommended to disable this option (*No*) Safety risk: **Low**
29	Asynchronous To Synchronous	Allows to transform asynchronous resources into synchronous Values: Check box disabled by default	Allows to transform asynchronous resources into synchronous without changing the RTL HDL code, in order to assess the potential of those resources Applies to inferred sequential elements only	It is highly recommended to set this option to its default value (disabled) Most of the transformations are not needed if Hardware Design Standards are properly used (reset design practices) Safety risk: **High**
Xilinx Specific Options				
#	***Option/Function***	***Description***	***Benefits***	***Limitations and Recommendations***
30	Add I/O Buffers	Enables or disables I/O buffer insertion	Automatically generates the I/O primitives that are connected to	It is recommended to disable this option when the tool synthesizes an

Synthesis Tools: XST - Xilinx				
		Values: Check box enabled by default	I/O ports of the top-level module	internal module that is instantiated later in a larger design Safety risk: **Medium**
31	Max Fanout	Limits the fanout of nets and signals Values: Integer number (100'000 is set by default)	A conservative maximum fanout may improve timing results This property can be used along with *Register Duplication* to improve timing performance by replicating registers with high fanout	This value depends on the selected target device, therefore it is recommended to set it according to the device The *Max Fanout* property is not a technology limit, but only a guide for the synthesis tool This property may not be respected by the tool, especially when the limit is small. Therefore *Max Fanout* property should be given a reasonable value Safety risk: **Low**
32	Number of Clock Buffers	Controls the maximum number of clock buffers (BUFG) elements created by expressions Values: Integer number (default value depends on target device and is equals to the maximum number of available BUFG elements)	This option may be used to preserve BUFG elements when the tool synthesizes an internal module that is instantiated later in a larger design	It is recommended to set this option to its default value Safety risk: **Low**
33	Register Duplication	Enables or disables register replication Values: • Yes (default) • No	Improves timing performance by replicating registers with high fanout Helps to meet the *Max Fanout* constraint by replicating any register exceeding the applied Max Fanout value	When register duplication is enabled, the RTL HDL code should include attributes to avoid replication of some registers such as the FF used to synchronize clock domains (refer to § 4.2.2.5) Xilinx recommends to disable this option and perform manual register duplication in the RTL HDL code, since in some cases registers may not be automatically replicated as expected by using this option together with the *Max Fanout* property Safety risk: **High**
34	Equivalent Register Removal	Enables or disables removal of equivalent registers described at RTL level Values: Check box enabled by default	Improves area, increasing the probability that the design will fit on the device Equivalent FF are not removed if they are instantiated from a Xilinx primitive library	It is recommended to set this option to its default value (enabled) to perform a first run, even if the processing time increases due to FF optimization, and check the synthesis report to identify the equivalent registers removed by this option. Then, the duplicated registers identified should be removed from the RTL HDL code. At this point, the option can be disabled

Synthesis Tools: XST - Xilinx				
				This option is not compatible with some techniques used to increase robustness such as TMR Safety risk: **High**
35	Register Balancing	Enables FF retiming by moving FF and latches across logic to increase clock frequency Values: • Yes • No (default) • Forward • Backward	Improves timing performance by means of two optimizations: • Forward register balancing: A set of FFs at the inputs of a LUT is moved to a single FF at its output • Backward register balancing: A FF at the output of a LUT is moved to a set of FFs at its input	It is highly recommended to set this option to its default value (No) for the following reasons: • This optimization can move combinational logic across different clock domains • Design verification may become more complicated because register names, position, and functionality no longer match the RTL description This function is affected by the Keep Hierarchy option (when the design is flattened -Keep Hierarchy disabled-, FF may leave the block boundaries) Safety risk: **High**
36	Move First Flip-Flop Stage	Controls the retiming of registers with paths coming from primary inputs Values: Check box enabled by default when Register Balancing is enabled	Improves timing performance	This option is available only when Register Balancing is enabled Besides the undesired effects of Register Balancing, this option may also increase the input to clock timing. To avoid this effect, the "offset in before" timing property should be properly constrained in the UCF For these reasons, it is highly recommended to disable this option (disabled) Safety risk: **High**
37	Move Last Flip-Flop Stage	Controls the retiming of registers with paths going to primary inputs Values: Check box enabled by default when Register Balancing is enabled	Improves timing performance	This option is available only when Register Balancing is enabled Besides the undesired effects of Register Balancing, this option may also increase the clock to output timing. To avoid this effect, the "offset in after" timing property should be properly constrained in the UCF For these reasons, it is highly recommended to disable this option (disabled) Safety risk: **High**
38	Pack I/O	Packs FF into the I/Os to	When the default value is set	It is recommended to set this option

Synthesis Tools: XST - Xilinx				
	Registers into IOBs	improve input and output path timing Values: • Auto (default) • Yes • No	(*Auto*), the tool packs FF into the I/Os depending on the Optimization Goal settings: • Area: The tool packs FF as tightly as possible to the IOBs • Speed: Registers are moved to the IOBs provided they are not covered by timing constraints	to its default value (*Auto*) Checking the RTL viewer to identify the registers moved to the IOBs is also a good practice Safety risk: **Low**
39	LUT Combining	Merges LUT pairs with common inputs to single dual-output LUT6 elements Values: • No (default) • Auto • Area	This optimization may improve design area The option *Auto* tries to make a trade-off between area and speed	It is recommended to set this option to its default value (*No*) to avoid undesired optimizations together with a reduction of design speed Safety risk: **Medium**
40	Reduce Control Sets	Reduces the number of control sets, which are collections of control signals (clock, clock enable and set/reset) Values: • No (default) • Auto	This optimization, that applies only to synchronous control signals, may reduce design area, improve the packing process and reduce the number of slices (even if the number of LUTs increases)	It is recommended to set this option to its default value (*No*) to avoid undesired optimizations Safety risk: **Medium**
41	Use Clock Enable	Enables or disables clock enabling in FFs Values: • Auto (default) • Yes • No	This option may optimize logic The option *Auto* tries to make a trade-off between using a dedicated clock enable input of a FF, and putting clock enable logic on the input of the FF	It is recommended to disable this option (*No*) and instantiate, if necessary, the clock enable logic directly in the RTL HDL code Safety risk: **Medium**
42	Use Synchronous Set	Enables or disables the synchronous set function in FFs Values: • Auto (default) • Yes • No	This option may optimize logic The option *Auto* tries to make a trade-off between using dedicated synchronous set input of a FF input, and putting synchronous set logic on the D input of a FF	It is recommended to disable this option (*No*) and instantiate, if necessary, the synchronous set logic directly in the RTL HDL code Safety risk: **Medium**
43	Use Synchronous Reset	Enables or disables the synchronous reset function in FFs Values: • Auto (default) • Yes • No	This option may optimize logic The option *Auto* tries to make a trade-off between using dedicated synchronous reset input of a FF input, and putting synchronous reset logic on the D input of a FF	It is recommended to disable this option (*No*) and instantiate, if necessary, the synchronous reset logic directly in the RTL HDL code Safety risk: **Medium**
44	Optimize Instantiated Primitives	Enables or disables the optimization of instantiated Xilinx library primitives Values: Check box disabled by default	Allows the optimization of some instantiated Xilinx library primitives	It is recommended to set this option to its default value (disabled) to avoid undesired optimizations Safety risk: **Medium**

Table 6.2. XST Synthesis Tool Specific Assessment of Options and Functionalities

Following design-specific files should be subject to revision control:

- Project file (including the name and location of all the RTL HDL files to be synthesized and the synthesis options)
- RTL HDL source code
- XST Constraints File (XCF)
- (Optionally) Core files in neither NGC or EDIF format
- Tcl script to guide the synthesis process (if any)
- Design technology libraries

6.1.1.3 Synplify Pro (Synopsys)

Synplify Pro is a third party synthesis tool which is also integrated within Actel Libero IDE design environment. It accepts high-level designs written in Verilog and VHDL to create PLD vendor-specific netlists. The tool can also write VHDL and Verilog netlists after synthesis for further simulation.

The specific assessment is performed using as reference the version E-2010.09A-1 software. Some options which are specific to a given technology or a single device family are not included. For these options, Synopsys recommends to refer to the vendor documentation.

Table 6.3 summarizes its synthesis configuration options and functions along with their benefits, limitations and recommendations for use.

Synthesis Tools: Synplify Pro - Synopsys				
Device Mapping Options				
#	*Option/Function*	*Description*	*Benefits*	*Limitations and Recommendations*
01	Fanout Guide	Specifies the global fanout limit for the whole design Value: Any non-negative integer (24 is the default value)	A conservative maximum fanout may improve timing results This tool may try replication or signal buffering to respect the fanout limit	This value depends on the selected target device, therefore it is recommended to set it according to the device The *Fanout Guide* property is not a technology limit, but only a guide for the synthesis tool This property may not be respected by the tool, especially when the limit interferes with optimization. Therefore *Max Fanout* property should be given a reasonable value The RTL HDL code should include attributes to avoid replication of some registers such as the FF used to synchronize clock domains (refer to § 4.2.2.5) Safety risk: **Medium**
02	Disable I/O Insertion	Enables or disables the insertion of I/O pads (inputs, outputs and bidirectionals) in the output netlist Values: Check box disabled by default	Automatic I/O insertion may be useful to estimate how much logic resources are used before synthesizing an entire FPGA	It is recommended to disable the automatic I/O insertion and instantiate I/O pads directly in the HDL RTL code for the pins that require them Safety risk: **Medium**
03	Update Compile Point Timing Data	Controls whether or not changes to a locked compile point force remapping of higher level compile points, taking into account the new timing model Values: Check box disabled by default	This option is used in incremental design flows	This option is unavailable in Synplify F-2011.09 or higher Incremental design flows may involve an additional effort in order to assess that design modifications do not disturb other parts of the design This methodology increases the complexity of the project baseline,

Synthesis Tools: Synplify Pro - Synopsys				
				traceability and configuration management processes since several iterations with different settings are required to generate the final implementation Therefore, it is not recommended to use Incremental design flows unless the methodology and the supporting processes are properly mastered to ensure the reproducibility of the implementation process Safety risk: **High**
04	Promote Global Buffer Threshold	Specifies the fanout load threshold to promote signals to global signals (also referenced as high drive elements) Value: Any non-negative integer (50 is the default value)	The tool assigns the available global buffers to drive promoted signals using the following priority: • Clock • Asynchronous set/reset signals • Enable and Data signals This option, applicable to both ports and nets, allows to reserve the use of global clock networks to high fanout nets	This value depends on the selected target device, therefore it is recommended to set it according to the device Automatic global promotion is recommended only for high fanout nets. Driving high fanout nets with a clock network may improve timing and routability Safety risk: **Low**
05	Operating Conditions	This option allows to specify an operating condition for certain devices Values: Depending on selected technology. The Actel operating condition can contain specifications for military, commercial and industrial, with designations of worst, typical and best case	Different operating conditions cause differences in device performance, affecting the following processes and data: • Optimization (when timing constraints are used) • Timing analysis • Timing reports	It is recommended to set this parameter to the selected specification (commercial / industrial / military) deigned for worst-case Safety risk: **Medium**
06	Annotated Properties for Analyst	Annotates the design with generic non-timing instance properties and timing properties Values: Check box enabled by default	The annotated properties are viewable in the RTL View and the HDL Analyst software tool	It is recommended to set this option to its default value (enabled) if HDL Analyst software tool is used for further code analysis Safety risk: **Low**
07	Max Number of Critical Paths in SDF	Specifies the maximum number of critical paths in a forward-annotated constraint file (SDF) Value: Any non-negative integer (4000 is the default value)	The SDF file displays a prioritized list of the worst-case paths in a design, which can be used by the place and route tool to improve timing and performance	It is recommended to perform several iterations with different values to achieve the best performance possible Safety risk: **Low**
08	Conservative Register Optimization	Instructs the tool to use less restrictive register optimizations Values: Check box disabled by default	When enabled, this option allows to reduce process runtime by applying less restrictive register optimization when area is not the main constraint	It is recommended to set this option to its default value (disabled) to avoid unexpected optimizations Safety risk: **Medium**

Synthesis Tools: Synplify Pro - Synopsys				
09	Resolve Mixed Drivers	Instructs the tool to resolve nets driven by VCC or GND and active drivers Values: Check box disabled by default	This option is useful to resolve mixed drivers. If a net is driven by VCC or GND and active drivers, enabling this option will connect the net to VCC or GND	It is recommended to set this option to its default value (disabled) and resolve the identified mixed drivers directly in the RTL HDL code Safety risk: **Medium**
10	Verification Mode	This option ensures compatibility with some LEC tools Values: Check box disabled by default	When enabled, this option ensures that synthesis output files are compatible with the Verplex Conformal Logic equivalence Checker (LEC) tool	When this option is enabled, then *Retiming* and *Pipelining* settings should be disabled since these optimizations may not be compatible with LEC It is recommended to set this option to its default value (disabled) if no LEC tool is used Safety risk: **Low**
Optimization Switches				
#	***Option/Function***	***Description***	***Benefits***	***Limitations and Recommendations***
11	FSM Compiler	This option allows the tool to recognize, extract and optimize the state machines in the design Values: Check box enabled by default	When enabled, this option allows the tool to optimize the extracted FSM to reduce area and/or improve performance The FSM Compiler has a special encoding directive, "*safe*", that will add logic such that if the FSM should ever reach an invalid state, it will be forced to the reset state When disabled, the FSM are extracted and optimized as regular logic	It is highly recommended to disable this option (disabled) to prevent automatic FSM optimizations that may produce FSM which can not recover from invalid states The "*safe*" directive does not allow to define the recovery state which is forced to the reset state Disabling *FSM Compiler* also allows to control the FSM implementation by specifying the number of states, the recovery state and the encoding directly in the RTL HDL code [EXT-04] Safety risk: **High**
12	FSM Explorer	This option allows the tool to explore different encoding styles and select the best suited for the design Values: Check box disabled by default	When enabled, this option allows the tool to select the encoding style that better fits the FSM	This option is unavailable in Synplify F-2011.03 or higher When enabled, *FSM Explorer* runs *FSM Compiler* (even if it is disabled) to extract the FSM information it needs Therefore it is highly recommended to set this option to its default value (disabled) to prevent automatic FSM optimizations that may produce FSM which can not recover from invalid states Safety risk: **High**
13	Resource Sharing	This option determines whether the synthesis process will use resource sharing techniques Values: Check box enabled by default	Minimizes the number of operators. This optimization uses two similar arithmetic resources which are never used at the same time to implement a single operator	It is recommended to disable this option (disable) since this option may negatively impact the timing performance Resource sharing combined with

Synthesis Tools: Synplify Pro - Synopsys				
			In most cases resource sharing improves area	poor coding style may force the tool to implement more arithmetic operations than is required, reducing the overall performance of the design In addition, this optimization may not be compatible with LEC Safety risk: **High**
14	Pipelining	This option uses register balancing and pipeline registers on multipliers and ROMs Values: Check box disabled by default	When enabled, this option may improve timing performance	It is highly recommended to set this option to its default value (disabled) for the following reasons: • This optimization can move combinational logic across different clock domains • Design verification may become more complicated because register names, position, and functionality no longer match the RTL description Safety risk: **High**
15	Retiming	This option uses retiming to improve timing performance Values: Check box disabled by default	When enable, this option may improve timing performance of sequential circuits without having to modify the RTL HDL source code The improvement is achieved by means of two optimizations: • Forward register balancing: A set of FFs at the inputs of a LUT is moved to a single FF at its output • Backward register balancing: A FF at the output of a LUT is moved to a set of FFs at its input	It is highly recommended to set this option to its default value (disabled) for the following reasons: • This optimization can move combinational logic across different clock domains • Design verification may become more complicated because register names, position, and functionality no longer match the RTL description Safety risk: **High**
Constraints				
#	*Option/Function*	*Description*	*Benefits*	*Limitations and Recommendations*
16	Frequency (MHz)	This option sets the default global frequency Values: • Frequency (MHz) (default) • Auto Constraint (Optimize to obtain maximum frequency)	*Frequency* value sets the default global frequency in MHz When *Auto Constraint* is set, and no clocks are defined, the tool automatically constraints the design to achieve the best possible timing	It is recommended to use the *Frequency* option, setting a global value which will be overridden with the individual clock constraints defined in the SDC file Safety risk: **Low**
17	Use Clock Period for	Determines whether the default constraints are used for I/O	When enabled, the tool considers any explicit constraint (default	It is recommended to set this option to its default value (disabled) and let

Synthesis Tools: Synplify Pro - Synopsys				
	Unconstrained I/O	ports that do not have user defined constraints Values: Check box disabled by default	behaviour) and, for all ports without explicit constraints, it uses constraints based on the clock period of the attached register	the designer define a comprehensive set of explicit timing constraints Safety risk: **Low**
18	Constraint Files	Specifies which constraint files are used for the implementation	Allows to define one or more constraint files (SDC)	It is highly recommended to use at least one constraint file (SDC) including user defined timing constraints Safety risk: **Low**
Optional Output File Options				
#	***Option/Function***	***Description***	***Benefits***	***Limitations and Recommendations***
19	Write Mapped Verilog/VHDL Netlist	Generates mapped Verilog/VHDL netlist files Values: Check box disabled by default	This model is used to perform the post-place and route simulation	It is recommended to enable this option (enabled) to generate the post-place and route simulation model, in case post-place and route simulations are planned in the verification process Safety risk: **Low**
20	Write Vendor Constraint File	Generates a vendor-specific constraint file for forward annotation Values: Check box enabled by default	The synthesis constraints are mapped to the appropriate vendor constraints To forward-annotate timing constraints, timing parameters (clock period, max delay, and input/output delay among others) should be set using the tool's constraint editor (Scope)	If this option is used, it is recommended to review the forward-annotated constraint file to check that all the input timing constraints were properly converted to be supported by the target device Safety risk: **Low**
21	Write Verification Interface Format (VIF) File	Generates a Tcl file interface for better integration with verification tools Values: Check box disabled by default	Improves the integration of LEC tools, reducing their setup work by using the information provided in the VIF file	It is recommended to enable this option (enabled) especially if LEC tools are used in the verification flow Safety risk: **Low**
Timing Report				
#	***Option/Function***	***Description***	***Benefits***	***Limitations and Recommendations***
22	Number of Critical Paths	Sets the number of critical paths to be displayed on the timing report	This option allows to generate timing reports that are easier to review	It is recommended to adapt the number of critical paths to the design complexity Safety risk: **Low**
23	Number of Start/End Points	Sets the number of Start/End points to be displayed on the critical path section of the timing report		It is recommended to adapt the number of Start/End points to the design complexity Safety risk: **Low**
VHDL				
#	***Option/Function***	***Description***	***Benefits***	***Limitations and Recommendations***
24	Top Level Entity	Sets the name of the top level VHDL entity	n.a.	If the top level does not use the default work library to compile

Synthesis Tools: Synplify Pro - Synopsys				
				VHDL files, it is necessary to specify the library file where the top level entity can be found Safety risk: **Low**
25	Default Enumeration Encoding	Specifies the default enumeration encoding to be used Values: • Default (default) • Onehot • Gray • Sequential	This property only applies to enumerated types When *Default* is selected, the tool assigns an encoding style based on the number of states: • Sequential (0 to 4 states) • Onehot (5 to 24 states) • Gray (more than 24 states) For FSMs, the *FSM Compiler* (when enabled) automatically determines the state machine encoding	It is highly recommended to set this option to the most robust encoding style (*Onehot*) and encode the enumerated types directly in the RTL HDL code to keep control of the encoding style Safety risk: **High**
26	Push Tri-states	Instructs the tool to push tri-states across process boundaries Values: Check box enabled by default	When enable, this option pushes tri-states through objects such as multiplexers, registers, latches, buffers, nets and tri-state buffers, and propagates the high-impedance state Pushing tri-states to the periphery of the design improves timing results because the tool uses tri-state output buffers	Pushing tri-states may increase the design resources needed to implement the design It is recommended to insert the tri-state output buffers directly in the RTL HDL code and disable this option to avoid unexpected structures (such as multiplexed tri-states) Safety risk: **Low**
27	Synthesis On/Off Implemented as Translate On/Off	Enables the *synthesis_on* and *synthesis_off* directives Values: Check box disabled by default	The *synthesis_on* and *synthesis_off* directives allows to synthesize designs originally written for use with other synthesis tools without needing to modify the source code	It is highly recommended to set this option to its default value (disabled) since the RTL HDL code between these two directives is not synthesized and therefore may generate a different implementation with an unexpected behaviour Safety risk: **High**
28	VHDL 2008	Enables the use of VHDL 2008 language standards Values: Check box disabled by default	n.a.	It is recommended to enable this option if VHDL 2008 standards are used in the RTL HDL code Safety risk: **Low**
29	Generics	Shows generics extracted with the *Extract Generic Constants* function	This option allows to override the default and set new values for the generic constant without modifying the RTL HDL code This option may be useful to try different values to find the best performance	It is highly recommended not to modify generic values using this function to avoid mismatches between the RTL HDL code and the final implementation Safety risk: **High**
Verilog				
#	***Option/Function***	***Description***	***Benefits***	***Limitations and Recommendations***

Synthesis Tools: Synplify Pro - Synopsys				
30	Top Level Module	Sets the name of the top level Verilog module	n.a.	It is recommended to type the module name before performing the synthesis process. The tool will use that module as top level
31	Verilog Language	Allows to select the standard for a Verilog project Values: • Verilog 95 • Verilog 2001 (default) • System Verilog	n.a.	Set the standard used in the RTL HDL code Safety risk: **Low**
32	Push Tri-states	Instructs the tool to push tri-states across process boundaries Values: Check box enabled by default	When enable, this option pushes tri-states through objects such as multiplexers, registers, latches, buffers, nets and tri-state buffers, and propagates the high-impedance state Pushing tri-states to the periphery of the design improves timing results because the tool uses tri-state output buffers	Pushing tri-states may increase the design resources needed to implement the design It is recommended to insert the tri-state output buffers directly in the RTL HDL code and disable this option to avoid unexpected structures (such as multiplexed tri-states) Safety risk: **Low**
33	Allow Duplicate Modules	Allows to use duplicate modules in the design Values: Check box disabled by default	When enabled, the last definition of the module is used by the tool and any previous definitions are ignored	It is highly recommended to set this option to its default value (disabled) to avoid configuration management issues Safety risk: **High**
34	Multiple File Compilation Unit	Allows to use duplicate modules in the design Values: Check box disabled by default	When enable, this option allows the tool to use the compilation unit for modules defined in multiple files, reducing the process runtime	It is highly recommended to set this option to its default value (disabled) to avoid configuration management issues and file order dependencies difficult to trace Safety risk: **High**
35	Compiler Directives and Parameters	Shows parameters extracted with the *Extract Parameters* function	This option allows to override the default and set new values for the parameters without modifying the source code This option may be useful to try different values to find the best performance	It is highly recommended not to modify parameters using this function to avoid mismatches between the RTL HDL code and the final implementation Safety risk: **High**

Table 6.3. Synplify Pro Synthesis Tool Specific Assessment of Options and Functionalities

Following design-specific files should be subject to revision control:

- Project file (including the name and location of all the RTL HDL files to be synthesized and the synthesis options)
- RTL HDL source code
- Synopsys Design Constraints File (SDC)

- Tcl script to guide the synthesis process (if any)
- Design technology libraries

6.1.2 Place and Route Tools / Integrated Development Environments (IDE)

6.1.2.1 Quartus II Integrated Development Environment (Altera)

The Quartus II IDE involves separate steps of synthesis (performed by the Quartus II Integrated Synthesis tool assessed in § 6.1.1.1), place and route (referenced by Altera as fitter) and STA (TimeQuest Timing Analyzer assessed in § 6.1.3.1). The synthesis process is unable to anticipate the routing delays seen by the fitter. To better optimize the implementation results, the Quartus II software integrates physical synthesis optimizations that take those routing delays into consideration and focus timing-driven optimizations, increasing the integration of the fitting and synthesis process.

The specific assessment is performed using as reference the version 11.1 of Altera's Quartus II software.

Table 6.4 summarizes the device, physical synthesis and the fitter options and functions along with their benefits, limitations and recommendations for use.

Place & Route Tools: Quartus II Design Environment - Altera				
Device and Pin Options				
#	*Option/Function*	*Description*	*Benefits*	*Limitations and Recommendations*
01	General – Auto-restart Configuration After Error	Directs the device to restart the configuration process automatically if a data error is encountered Values: Check box enabled by default	Enabling this option, the device can restart the configuration process automatically if an error occurs	It is recommended to disable this option and analyze the error that stopped the configuration Safety risk: **Low**
02	General – Release clears before tri-states	Directs the device to release the clear signal on registered logic cells and I/O cells before releasing the output enable override on tri-state buffers Values: Check box disabled by default	Enabling this option allows the registers to operate before the output pins are active (initialized)	It is recommended to set this option to its default value (disabled) if there is no specific requirement concerning the start-up state of the affected pins Safety risk: **Low**
03	General – Enable User-supplied Start-up Clock	Directs the device to use a user-supplied clock for initialization Values: Check box disabled by default	Some device configuration schemes may require an external clock source	It is recommended to set this option to its default value (disabled) if there is no specific requirement concerning the device configuration scheme Safety risk: **Low**
04	General – Enable Device-wide Reset	Enables a Device Clear pin to be used as a global reset by an external source Values: Check box disabled by default	This option, when enabled, provides an external reset source that clears all the device registers When this option is disabled, the Device Clear pin can be used as a user I/O	It is recommended to set this option to its default value (disabled) and define a reset strategy in the RTL HDL code Safety risk: **Low**

Place & Route Tools: Quartus II Design Environment - Altera				
05	General – Enable Device-wide Output Enable	Enables a Device Output Enable pin to be used as an global output enable signal Values: Check box disabled by default	This option, when enabled, provides an external output enable signal that forces, when asserted, all device outputs to tri-state When this option is disabled, the Device Output Enable pin can be used as a user I/O	It is recommended to set this option to its default value (disabled) and define a global output enable signal in the RTL HDL code Safety risk: **Low**
06	General – Enable INIT_DONE output	Enables the device INIT_DONE pin Values: Check box disabled by default	This option, when enabled, provides a pin to monitor when initialization is complete and the device is in user mode When this option is disabled, the INIT_DONE pin can be used as a user I/O	It is recommended to enable this option if external monitoring of the device operational mode (configuration/user) is required Safety risk: **Low**
07	Reserve All Unused Pins	Allows the tool to specify the reserve state of all unused pins on the device Values: • As Input Tri-stated • As Output Driving Ground (default) • As Output Driving an Unspecified Signal • As Input Tri-stated with Bus-hold Circuitry • As Input Tri-stated with Weak Pull-up Resistor	This option defines the reserve state of the unused pins It is also possible to assign the reserve state to unused pins individually (Floorplanner editor)	The default value depends on the selected device family. It is important to note that different settings will consume different amounts of power. Therefore it is recommended to select the value depending on the selected device and power consumption requirements A good approach is to set the *As Input Tri-stated with Weak Pull-up Resistor* value. In this configuration unused pins are not floating and they will not cause short-circuit is they are accidentally tied to VDD or GND Safety risk: **Medium**
08	Dual-Purpose Pins	Specifies how dual-purpose pins should be used after device configuration is complete Values: Default settings for each pin depend on the current configuration scheme	This option allows to increase the number of available user I/Os	It is recommended to use dual-purpose pins as user I/Os only if no more I/Os are available Safety risk: **Low**
09	Capacitive Loading	Specifies values for capacitive loading per I/O standard	Capacitive Loading provides a simple I/O timing model to the TimeQuest analyzer to scale the output timing from the output buffer to the device pin	The characterization of the capacitive loading is recommended to get more accurate I/O timing results Setting the *Capacitive Loading* to *No Load* (0 pF) may be useful to perform further signal integrity simulations Safety risk: **Low**
10	Board Trace Model	Specifies board trace, termination, and capacitive load parameters for each I/O standard	Board Trace Model provides an advanced timing model to the TimeQuest analyzer to perform the Advance I/O Timing analysis,	These settings affect Advanced I/O Timing only, and are used instead of Capacitive Loading

Place & Route Tools: Quartus II Design Environment - Altera				
11	I/O Timing	Allows to describe a board trace and termination network as a set of capacitive, resistive and inductive assignments	including the differential I/O standards	The characterization of a Board Trace Model is recommended to get more accurate I/O timing results Safety risk: **Low**
12	Voltage	Specifies voltage options for the device (default I/O standard)	Allows to specify the default I/O standard to be used for pins on the target device	It is recommended to set this option to its default value, which depends on the selected target device, and define the I/O standard of each I/O Bank in the SDC file Safety risk: **Low**
13	Pin Placement	Specifies options for adjusting the voltage of LVTTL/LVCMOS input pins Values: Check box “Allow Voltage Overdrive...” disabled by default	Voltage overdrive allows to place input pins with LVTTL or LVCMOS (3.3V/2.5V) I/O standards inside an I/O bank powered with lower VCCIO voltage (1.8V/1.5V) When enabled, this option may allow the following optimizations: • Reduction of device power consumption • Reduction of the number of different voltages required • Increase of device pin placement flexibility	It is highly recommended to disable this option to avoid I/O bank overdriving, which may lead to higher leakage currents, causing the design not to work as intended Safety risk: **High**
14	Error Detection CRC – Enable Error Detection CRC	Specifies error detection CRC and CRC ERROR pin usage (if available) for the selected device Values: Check box disabled by default	This option, when enabled, performs a check that determines whether one of the following errors occurred in the programming data of the device: • Single error • Double-adjacent bit error • Uncorrectable errors Error detection can be performed during configuration or when the device is in user mode This error detection feature may be useful to mitigate configuration memory errors caused by SEU The Structured ASIC devices from Altera do not have configuration circuitry and therefore do not need this feature	Enabling this option may reduce device speed in some device families The CRC ERROR pint should be only used during user mode error detection This feature may be used as an effective mean to mitigate the risks of SEU only in the configuration memory cells (the most SEU sensitive device resources together with on-chip RAM cells). The following resources are not covered by this feature: • On-chip memory RAM cells → Mitigation: Built-in ECC • Registers and FF in the device core → Mitigation: Hamming code for FSM, triplication • I/O registers → Altera claims that I/O registers robustness make no contribution to FIT rate The recovery from CRC errors should be done at an upper level

Place & Route Tools: Quartus II Design Environment - Altera				
				(usually at system level) Safety risk: **High**
15	Error Detection CRC – Enable Open Drain on CRC ERROR Pin	Sets the CRC ERROR pin as an open-drain pin Values: Check box disabled by default	Turning on this option allows to decouple the voltage level of the CRC ERROR pin from the VCCIO voltage, providing voltage leveling advantages	It is recommended to set this option to its default value (disabled). Configure this output pin as open-drain may be useful to implement logic functions such as active-low wired-OR in order to combine the CRC ERROR pin with other error signals. This approach is not recommendable since the CRC ERROR should have its own dedicated line given its criticality If this option is enabled then an external pull-up resistor should be connected to the CRC ERROR pin for its proper functioning Safety risk: **High**
16	Error Detection CRC – Enable Internal Scrubbing	Specifies internal scrubbing usage for the selected device Values: Check box disabled by default	When enabled, this option allows the device to correct single error or double adjacent error within the core configuration memory while the device is still running	This option is only available on newest devices (Stratix V and later) It is highly recommended to set this option to its default value (disabled) and manage the error recovery at an upper level Safety risk: **High**
17	Error Detection CRC – Error Check Frequency	Sets the error check frequency by dividing the internal clock by the selected value Values: From 2 up to 256 in steps of power of 2 values	The error detection process can be slowed down to have enough time to read the device register storing the error detection information	The error check frequency should be adapted to the logic designed to monitor the error detection information Safety risk: **Medium**
Physical Synthesis Settings				
#	*Option/Function*	*Description*	*Benefits*	*Limitations and Recommendations*
18	Netlist Optimizations	Specifies whether the tool should perform advanced netlist optimizations, such as gate-level retiming or physical synthesis This option is applicable at single nodes or to design entity, and can be set only in the Quartus II Assignments Editor Values: • Always Allow • Never Allow • Default (default)	Setting this option to *Always Allow* value, allows the tool to modify the node or entity, even if doing so affects the timing or performance of the design The *Never Allow* value prevents the tool from modifying the node or entity The *Default* value allows the tool to duplicate, move or change the synthesis of the node or entity, or allows register retiming during netlist optimization only if doing so does not negatively affect the timing or performance of the design	The *Always Allow* value is not recommended by Altera, since it may have a negative impact in both timing and area results It is recommended to set this option to its default value (*Default*) and select the specific netlist optimization options to be performed from the Quartus II Settings Dialog Box (entity level) or the Assignment Editor (node level) Safety risk: **High**
19	Perform Physical	Instructs the tool to increase	The tool can reduce the number	It is recommended to set this option

Place & Route Tools: Quartus II Design Environment - Altera				
	Synthesis for Combinational Logic (Physical Synthesis)	performance by performing physical synthesis on combinational logic Values: Check box disabled by default	of combinational logic level in critical paths to improve performance	to its default value (disabled) This option may not be compatible with some techniques used to increase robustness such as TMR Safety risk: **Medium**
20	Perform Register Retiming	Instructs the tool to increase performance by using register retiming to perform physical synthesis on registers Values: Check box disabled by default	Register retiming improves the delay of synchronous sequential circuits by moving registers across combinational logic	It is highly recommended to set this option to its default value (disabled) since this optimization: • May move combinational logic across different clock domains if clocks are not constrained individually • May move synchronization FF if the Synchronization Register Chain Length is not properly set • May increase the effort of design verification because register names, position and functionality no longer match the RTL description Safety risk: **High**
21	Effort Level	Specifies the level of physical synthesis optimization to be performed This setting applies to Physical Synthesis for Combinational Logic and Register Retiming Values: • Fast • Normal (default) • Extra	The option *Fast* instructs the tool to use a lower level of physical synthesis to keep a smaller increase in compilation time The default level (*Normal*) increases the compilation time by an average of two or three times The *Extra* value forces the tool to use a higher level of physical synthesis optimization	It is recommended to set this option to its default value (*Normal*) Safety risk: **Medium**
22	Perform Automatic Asynchronous Signal Pipelining	Instructs the tool to insert pipeline stages for asynchronous clear and/or load signals to increase timing performance Values: Check box disabled by default	This option is useful for asynchronous signals that are failing recovery and removal timing because they feed registers using a high-speed clock	It is highly recommended to set this option to its default value (disabled), identify the critical paths and apply optimization techniques directly in the RTL HDL code Safety risk: **High**
23	Perform Register Duplication	Instructs the tool to duplicate registers and combinational logic based on placement information provided by the fitter process Values: Check box disabled by default	Improves timing performance by replicating registers with high fanout Helps to meet the Max Fanout constraint by replicating any register exceeding the applied Max Fanout value	Registers that are part of a synchronization chain or that are driven by a register in another clock domain are protected against duplication by default It is recommended to enable this option (enabled) if : • An improvement of timing performance is

Place & Route Tools: Quartus II Design Environment - Altera				
				needed and, • Registers as the before mentioned are identified and checked after synthesis Safety risk: **High**
24	Perform Physical Synthesis for Combinational Logic (Physical Synthesis for Density)	Instructs the tool to reduce area by performing physical synthesis optimizations on combinational logic during placement and routing Values: Check box disabled by default	The tool can detect and remove duplicate combinational logic or register nodes to reduce area	It is recommended to set this option to its default value (disabled) This option may not be compatible with some techniques used to increase robustness such as TMR Safety risk: **Medium**
25	Perform Logic to Memory Mapping	Allows the tool to reduce area by mapping logic and registers into unused memory blocks during placement and routing Values: Check box disabled by default	This optimization may improve design area	It is recommended to set this option to its default value (disabled) Implement logic into memory blocks may have an impact on the design reliability since memory blocks are more sensitive to SEU Safety risk: **High**
Place and Route (Fitter) Settings				
#	***Option/Function***	***Description***	***Benefits***	***Limitations and Recommendations***
26	Timing-Driven Compilation - Optimize Hold Timing	Instructs the tool to optimize hold time within a device to meet user specific timing requirements and assignments Values: • I/O Paths and Minimum TPD Paths • All Paths (default) • Disabled	Directs the tool to place logic elements in the device to meet the timing constraints defined in the SDC file	It is recommended to set this option to its default value (*All Paths*) Safety risk: **Medium**
27	Timing-Driven Compilation - Optimize Multi-Corner Timing	Instructs the tool to consider all corner timing delays, including both fast-corner timing and slow-corner timing to meet timing requirements at both corners Values: Check box disabled by default	When enabled, this option directs the tool to place logic elements in the device to meet the timing constraints defined in the SDC file considering the following operating conditions: • Slow-corner at maximum and minimum specified temperatures: Slowest manufactured device for a given speed grade, operating under low-voltage conditions • Fast-corner at minimum specified temperature: Fasted manufactured device for a given speed grade, operating under high-voltage conditions	It is highly recommended to enable this option in order to perform the compilation taking into account different operating conditions depending on Process-Voltage-Temperature (PVT) parameters Due to process variation and changes in operating conditions, delays on some paths can be significantly smaller than those in the slow-corner. This can result in hold time violations on those paths Multi-Corner Timing may identify these timing issues, covering the range of the device's operating conditions and providing more accurate timing results Safety risk: **High**

Place & Route Tools: Quartus II Design Environment - Altera				
			If the option is disabled, only slow-corner timing is considered	
28	PowerPlay Power Optimization	Determines how aggressively the tool optimizes the design for power Values: • Extra Effort • Normal Compilation (default) • Off	When *Normal Compilation* is set, the tool performs the following power optimizations, as long as there is no expectation of design performance reduction: • Minimization of number of memory blocks accessed during each clock cycle • Rearrangement of logic to eliminate nets with high toggle rates The *Extra Effort* option performs the following additional power optimizations: • Shut down memory blocks that are not accessed • Power-aware memory balancing	It is recommended to set this option to its default value (*Normal Compilation*) to avoid the additional power optimizations which may affect design performance Safety risk: **Medium**
29	Fitter Effort	Specifies the level of place and route effort to be performed Values: • Standard Fit • Fast Fit • Auto Fit (default)	The *Standard Fit* option do not decrease fitter effort, maximizing the timing performance (Fmax) regardless of the timing requirements *Fast Fit* value decreases the fitter effort, with a reduction of approximately 10% in the design's maximum operating speed The default option, *Auto Fit*, instructs the tool to reduce the fitter effort only after meeting timing and routing requirements. It is possible to get some margin by setting the Desired Worst-Case Slack option (integer value in ns)	The *Fast Fit* option reduces the compilation time in designs which are easy to route. However, when routing complexity increases, this option may increase compilation time or cause fitting to fail It is recommended to use *Standard Fit* or *Auto Fit* values depending on timing closure criteria. In designs where timing requirements can be easily met, the *Standard Fit* option can result in longer compilation times than using the *Auto Fit* option Safety risk: **Low**
30	Limit to One Fitting Attempt	Controls how many fitting attempts the tool tries to get a fit Values: Check box disabled by default	When disable, the tool performs three attempts to get a fit In designs where timing and routing requirements can be easily met, enabling this option may reduce the compilation time	It is recommended to set this option to its default value (disabled) to get better timing and routing results Safety risk: **Low**
31	Seed	Specifies the seed that the tool uses to randomly determining the initial placement for the current design Value: Any non-negative integer (1 is the default value)	The seed can be modified when a design is close to meeting requirements, in order to get a slightly different result	It is highly recommended to set the Seed to a fixed value when timing and routing requirements are met, in order to ensure the reproducibility of the implementation process The use of place and route scripts that changes the seed based on

Place & Route Tools: Quartus II Design Environment - Altera				
				dynamic parameters such as date and time is also highly discouraged Safety risk: **High**
32	Allow Single-Ended Buffer for Differential XSTL Input	Allows the pin with Differential-XSTL I/O standard to be used with a single-ended input buffer Values: • On • Off (default)	n.a.	It is recommended to set this option to its default value (*Off*) and modify the I/O standard of the pin to keep coherency Safety risk: **Low**
33	Auto Delay Chains	Allows the tool to choose the optimal delay setting to meet I/O timing requirements Values: • On (default) • Off	When enabled, this option may reduce setup timing violations Enabling this option does not override delay chains specified in the user constraints file	It is recommended to set this option to its default value (*On*) to benefit from automatic delay chain settings to meet timing requirements Safety risk: **Low**
34	Auto Global Clock	Allows the tool to choose the global clock signal Values: • On (default) • Off	When enabled, the tool chooses the signal that feeds the most clock inputs to FFs as a global clock signal that is made available throughout the device on the global routing paths	It is recommended to disable this option (*Off*) and specify all global and/or regional clock signals directly in the RTL HDL code Safety risk: **Low**
35	Auto Global Register Control Signals	Allows the tool to choose the global register control signals Values: • On (default) • Off	When enabled, the tool chooses the signals that feeds the most control signal inputs to FFs as global control signals that are made available throughout the device on the global routing paths	It is recommended to disable this option (*Off*) and specify all global register control signals directly in the RTL HDL code Safety risk: **Low**
36	Auto Merge PLLs	Allows the tool to automatically find and merge together two compatible PLLs driven by the same clock source Values: • On (default) • Off	This option is useful for decreasing the total number of PLLs in a design that did not fit into the target device during compilation	If the number of available PLLs is enough, then it is recommended to disable this option (*Off*) to keep the implementation in line with the RTL description Safety risk: **Low**
37	Auto Packed Registers	Allows the tool to combine a register and a combinatorial function, or to implement registers using I/O cells, RAM blocks, or DSP blocks instead of logic cells Values: • Off (default) • Sparse • Sparse Auto • Normal • Minimize Area • Minimize Area with Chains • Auto (default)	This option controls how aggressively the tool combines registers with other function blocks to reduce the area of the design Registers are combined with I/O cells to improve I/O timing, and with DSP and RAM blocks to reduce the area required for placing the design	It is recommended to set this option to its default value (*Off*) to prevent the tool from implementing registers using RAM blocks Implement logic into memory blocks may have an impact on the design reliability since memory blocks are more sensitive to SEU In addition, the options *Normal*, *Minimize Area,* and *Minimize Area with Chains* have different behaviours depending on the targeted device family Safety risk: **High**

Place & Route Tools: Quartus II Design Environment - Altera				
38	Auto RAM to MLAB Conversion	Controls whether the tool is able to convert RAM blocks to use LAB locations Values: • On (default) • Off	The MLAB is implemented using registers. This resource can be used to increase the device embedded memory	It is recommended to disable this option (*Off*) and define the ram style directly in the RTL HDL code MLAB and dedicated RAM blocks have slight power-up and initialization differences The Quartus II Integrated Synthesis tool does not map inferred memory to MLAB resources unless the RTL HDL code specifies the appropriate ram style attribute Safety risk: **Medium**
39	Auto Register Duplication	Allows the tool to automatically duplicate registers within a LAB containing empty logic cells Values: • On • Off • Auto (default)	Turning on this option allows the *Logic Cell Insertion – Logic Duplication* option to improve the routability of the design	This option is not compatible with routing back-annotation and can also make LEC tools to report logic differences It is recommended to disable this option (*Off*) Safety risk: **Medium**
40	Clamping Diode	Turns on the clamping diode of a pin to limit overshoot voltage for the pin input operation Values: • On • Off (default)	The clamping diode is turned on by default for PCI and PCI-X I/O standards and turned off for 3.3V LVTTL and LVCMOS I/O standards	It is recommended turn on and turn off clamping diodes directly in the physical constraints file Safety risk: **Medium**
41	Enable Beneficial Skew Optimization	Allows the tool to insert skew on globally routed clock signals Values: • On (default) • Off	Inserting skew may improve design performance	It is recommended to disable this option (*Off*) since its verification may be complex Safety risk: **Medium**
42	Enable Bus-Hold Circuitry	Enables bus-hold circuitry during device operation Values: • On • Off (default)	When this option is enabled, a pin retains its last logic level when it is not driven, instead of going to high impedance logic level	This option should not be used at the same time as the *Weak Pull-Up Resistor* option This option may introduce mismatch issues between functional and post-place and route simulation It is recommended to set this option to its default value (*Off*) Safety risk: **Medium**
43	Equivalent RAM to MLAB Paused Read Capabilities	Controls whether a RAM block implemented in MLAB cells should have equivalent paused read behaviour as the RAM block implemented in dedicated block RAM Values: • Care (default)	When enabled (*Care*), this option avoids different paused read behaviour of RAM outputs when implemented in MLAB cells or in dedicated block RAM, since the tool only places RAM blocks into MLAB cell locations if this results in equivalent dedicated block RAM implementation	It is recommended to set this option to its default value (*Care*) to avoid different RAM output paused read behaviour, even if the tool mapping flexibility decreases Safety risk: **Medium**

Place & Route Tools: Quartus II Design Environment - Altera				
		• Don't Care		
44	Equivalent RAM to MLAB Power Up	Controls whether a RAM block implemented in MLAB cells should have power-up conditions equivalent to block RAM implementation Values: • Auto (default) • Care • Don't Care	If this option is set to *Care*, the tool does not convert RAM blocks to MLAB unless they have equivalent power-up conditions to a block RAM implementation	It is recommended to set this option to *Care* to avoid different power-up conditions, even if the tool mapping flexibility decreases Safety risk: **Medium**
45	Final Placement Optimizations	Specifies whether the tool performs final placement optimizations Values: • Always • Automatically (default) • Never	The final placement optimizations may improve timing and routing results, but also may require longer compilations time	It is recommended to set this option to its default value (*Automatically*) to benefit from placement optimizations Safety risk: **Low**
46	Fit Attempts to Skip	Controls the fit attempts the tool skips Value: Any non-negative integer (0 is the default value)	This option may be useful to save compilation time when multiple attempts are necessary	It is recommended to set this option to its default value (*0*) to let the tool perform the default three attempts to get a fit, in order to get better timing and routing results, even if the compilation time increases Safety risk: **Low**
47	Fitter Aggressive Routability Optimizations	Specifies whether the tool aggressively optimizes for routability Values: • Always • Automatically (default) • Never	Aggressive routability optimizations may decrease design speed, but may also reduce wire usage and routing time The default setting (*Automatically*) lets the tool decide whether to perform these optimizations based on routability and timing requirements	It is recommended to set this option to its default value (*Automatically*) to benefit from placement optimizations Safety risk: **Low**
48	Force Fitter to Avoid Periphery Placement Warnings	Instructs the tool to treat periphery placement warnings as errors Values: • On • Off (default)	When this option is enabled, the tool attempts to find a placement for the design that corrects the placement warnings	It is recommended to set this option to its default value (*Off*) and analyse the warnings Safety risk: **Medium**
49	I/O Placement Optimizations	Instructs the tool to optimize the location of I/Os that do not already have a pin location assigned to them Values: • On (default) • Off	I/O placement optimizations may improve timing and routing results, but also may require longer compilations time	It is recommended to set this option to its default value (*On*) to benefit from placement optimizations Safety risk: **Low**
50	Logic Cell Insertion – Logic Duplication	Allows the tool to automatically insert buffer logic cells between two nodes	This option, that works only when the *Auto Register Duplication* is enabled, allows to improve the	This option is not compatible with routing back-annotation and can also make LEC tools to report logic

Place & Route Tools: Quartus II Design Environment - Altera				
		without altering the functionality of the design Values: • Auto (default) • Off • On	routability of the design	differences It is recommended to disable this option (*Off*) Safety risk: **Medium**
51	M144K Block Read Clock Duty Cycle Dependency	Allows to specify whether the M144K memory block read operations depend upon read clock's duty cycle Values: • On • Off (default)	This option prevents the M144K memory blocks from locking when driven by a read clock with a very narrow pulse	This option, when enabled, may degrade the performance of the M144K memory blocks Clocks in general should have balanced duty cycles, therefore the use of narrow pulses as read memory clocks is not recommended It is recommended to set this option to its default value (*Off*) Safety risk: **Medium**
52	MLAB Add Timing Constraints For Mixed-Port Feed-Through Mode Setting Don't Care	Allows to specify whether STA should evaluate timing constraints between the write and the read operation of the MLAB memory block Values: • On • Off (default)	Performing a write and read operation simultaneously at the same memory address might result in metastability because no timing constraints between those operations exist by default Turning on this option introduces timing constraints between the write and read operations on the MLAB memory	This option, when enabled, may degrade the performance of the MLAB memory It is recommended to set this option to its default value (*Off*) and guarantee by design that there are no read and write operations performed simultaneously at the same memory address Safety risk: **Medium**
53	Maximum Number of Clocks of any Type Allowed	Specifies the maximum number of clocks of any type (global, periphery and regional) that can be used by the design Values: • Non-negative integers • -1 to set no limits (default)	This option is used in collaborate workflows to reserve clock resources for each separate component	It is recommended to set this option to its default value (*-1*) to allow the tool using all the clocks supported by the device Safety risk: **Low**
54	Optimize Design for Metastability	This option may improve the reliability of the design Values: • On (default) • Off	When this setting is enabled, the tool aims to increase the output setup slacks of synchronizer registers in the design, which can exponentially increase the design reliability	This option takes effect only if Altera's STA tool is being used for timing-driven compilation It is recommended to set this option to its default value (*On*) and review the timing reports to check the synchronizers detected by the tool Safety risk: **High**
55	Optimize IOC (Input Output Cell) Register Placement for Timing	Controls whether the tool optimizes the I/O pin timing by automatically packing registers into I/Os to minimize I/O to register and register to I/O delays	This option affects only pins with setup and clock-to-output timing requirements When the *Normal* option is enabled, the tool will opportunistically pack registers	It is recommended to set this option to its default value (*Normal*) to improve I/O timing, and review the mapping report to check the registers placed into IOC This option requires the *Optimize*

Place & Route Tools: Quartus II Design Environment - Altera				
		Values: • Normal (default) • Off • Pack All I/O Registers	into I/Os that should improve I/O timing The *Pack All I/O Registers* option instructs the tool to aggressively try to pack any register connected to an I/O into an IOC unless prevented by user constraints	*Timing* option to be enabled Safety risk: **Medium**
56	Optimize Timing	Controls whether the tool optimizes to meet the user's maximum delay timing requirements Values: • Normal Compilation (default) • Off	Turning this option off may help fit designs that have extremely high interconnect requirements and can also reduce compilation time	It is highly recommended to set this option to its default value (*Normal Compilation*) to take into account the design's timing requirements and enable other timing optimizations Safety risk: **Medium**
57	Optimize Timing for ECOs	Controls whether the tool optimizes to meet the user's maximum delay timing requirements during ECO compiles Values: • On • Off (default)	Turning this option off may help fit designs that have extremely high interconnect requirements and can also reduce compilation time	It is highly recommended to enable this option (*On*) to take into account the design's timing requirements and enable other timing optimizations Safety risk: **Medium**
58	Perform Clocking Topology Analysis During Routing	Instructs the tool to perform an analysis of the design's clocking topology Values: • On • Off (default)	When enabled, this option may adjust the optimization approach on paths with significant clock skew, improving hold timing	It is recommended to enable this option (*On*) to get better timing results, even if the compilation time increases Safety risk: **Low**
59	Placement Effort Multiplier	Controls how much time the tool spends in placement Value: Any non-negative floating point number (1.0 is the default value)	A higher value increases compilation time but may improve placement quality	It is recommended to adapt the placement effort to the design complexity, considering that: • Values smaller than 1.0 can reduce compilation time but reducing placement quality and design performance (not recommendable) • Values greater than 1.0 increase placement time and placement quality, but may reduce routing time Therefore the recommendable minimum value is 1.0 Greater values may also be considered since they can improve placement quality. For example, a value of 4.0 increases placement time by approximately 2 to 4 times, but may improve quality Safety risk: **Medium**

Place & Route Tools: Quartus II Design Environment - Altera				
60	Programmable Power Maximum High-Speed Fraction of Used LAB Tiles	Sets an upper limit on the fraction of the LAB tiles used by the design that can be high-speed Value: Floating point number between 0.0 and 1.0 (default value)	A value of 1.0 means that there is no restriction on the number of high-speed tiles Values lower than 1.0 may reduce power consumption by forcing some resources into low-power mode	Values lower than 1.0 may degrade timing results by forcing some timing critical resources into low-power mode It is recommended to set this option to its default value (*1.0*) Safety risk: **Medium**
61	Programmable Power Technology Optimization	Controls how the tool configures tiles to operate in high-speed mode or low-power mode Values: • Automatic (default) • Force All Tiles with Falling Timing Paths to High-Speed • Minimize Power Only	The *Minimize Power Only* value specifies that the tool should set the maximum number of tiles to operate in low-power mode to decrease the overall power consumption *Force All Tiles...* value specifies the tool to set al paths that are failing timing into high-speed mode to improve timing results	For designs that meet timing, the *Force All Tiles...* option should be similar to the *Automatic* setting For designs that fail timing, the use of the *Force All Tiles...* value may not increase the speed of the design and increase static power consumption It is recommended to set this option to its default value (*Automatic*) and redesign failing timing paths to close timing Safety risk: **Medium**
62	Regenerate Full Fit Report During ECO Compiles	Instructs the tool to regenerate the place and route report during ECO compiles Values: • On • Off (default)	When disabled, this option will decrease the compilation time	It is recommended to enable this option (*On*) to get an updated place and route report Safety risk: **Low**
63	Router Effort Multiplier	Controls how quickly the tool tries to find a valid routing solution Value: Any floating point number ≥ 0.25 (1.0 is the default value)	A value higher than 1.0 increases compilation time but may improve routing quality Values lower than 1.0 can reduce compilation time but also reducing routing quality	It is recommended to adapt the placement effort to the design complexity but keeping values greater than or equal to 1.0 Safety risk: **Medium**
64	Router Timing Optimization Level	Controls how aggressively the tool tries to meet timing requirements Values: • Normal (default) • Maximum • Minimum	Setting this option to *Maximum* may increase design speed slightly and also increase the compilation time The *Minimum* value can reduce compilation time but also reducing design speed	It is recommended to set this option to *Maximum* or *Normal* values to get better timing results Safety risk: **Medium**
65	SSN Optimization	Controls the Simultaneous Switching Noise (SSN) optimization setting Values: • Extra Effort • Normal Compilation • Off (default)	When set to *Normal Compilation*, this option performs SSN optimizations which should not impact design performance The *Extra Effort* option may affect design performance This option may be useful to define pin location of the design	It is recommended to set this option to its default value (*Off*) if pin location is already defined at board level, otherwise the option should be set to *Normal Compilation* *Safety risk:* ***Medium***
66	Treat Bidirectional Pin	Instructs the tool to process bidirectional pins as output	When enabled, this option uses the input path for feedback from	It is recommended to set this option to its default value (*Off*)

Place & Route Tools: Quartus II Design Environment - Altera				
	as Output Pin	pins Values: • On • Off (default)	the output path	Safety risk: **Medium**
67	Weak Pull-Up Resistor	Enables the weak pull-up resistor when the device is operating in user mode Values: • On • Off (default)	This option, when enabled, pulls a high-impedance bus signal to VCC	This option should not be used at the same time as the *Enable Bus-Hold Circuitry* option This option may introduce mismatch issues between functional and post-place and route simulation It is recommended to set this option to its default value (*Off*) Safety risk: **Medium**

Table 6.4. Quartus II Design Environment Tool Specific Assessment of Options and Functionalities

Following design-specific files should be subject to revision control:

- Project file (including the name and location of the synthesized netlist(s) and the place and route options)
- Synopsys Design Constraints file (SDC)
- Tcl script to guide the place and route process (if any)
- Design technology libraries

6.1.2.2 ISE Integrated Development Environment (Xilinx)

The Xilinx ISE IDE integrates the synthesis function (performed by the XST assessed in § 6.1.1.2), place and route and STA (Trace Timing Analyzer assessed in § 6.1.3.2). The ISE software also includes several functions to resynthesize the design, applying physical synthesis optimizations that take the routing delays into consideration and focus timing-driven optimizations, increasing the integration of the place and route and synthesis process. The physical synthesis options are merged with the map properties.

The specific assessment is performed using as reference the version 12.4 of Xilinx ISE software.

Table 6.5 summarizes its place and route options and functions along with their benefits, limitations and recommendations for use.

Place & Route Tools: ISE Integrated Development Environment - Xilinx				
Translate Properties				
#	*Option/Function*	*Description*	*Benefits*	*Limitations and Recommendations*
01	Use LOC Constraints	Specifies whether or not to use location constraints (LOC) found in the input netlist or User Constraint File (UCF) Values: Check box enabled by default	When disabled, this option allows the tool to ignore: • LOC information that contains the relative placement of one CLB (Configurable Logic Block) to another • Invalid LOC information that would result in errors	It is recommended to set this option to its default value (enabled) to process the LOC constraints included in the User Constraint File (UCF) and find potential placement errors LOC constraints allow to manually place some design elements, such as block RAM, to improve performance Safety risk: **Medium**
02	Netlist Translation Type	Specifies how source files are used by the tool Values: • Timestamp (default) • On • Off	When *Timestamp* option is selected, the tool translates the input netlist only if it is newer than the output (if there is any)	It is recommended to set this option to its default value (*Timestamp*) to better control the implementation process Safety risk: **Low**
03	Macro Search Path	Specifies the search path to add to the list of directories to search when resolving file references. This option also supplies paths for macros or other directories containing design files	It is possible to specify multiple search paths in order to better organize the project file structure	It is recommended to specify the file structure to keep track of the different design files involved in the project Safety risk: **Low**
04	Create I/O Pads from Ports	Specifies whether or not to add PAD properties to all top level port signals Values: Check box disabled by default	This option, when enabled, inserts PAD properties if the input design file does not contain them	This option should be adapted to the design strategy. For designs aiming to be portable, it is recommended to enable the option to let the tool adding PAD properties If the option is enabled and the design already contains PAD properties, then the tool reports an

Place & Route Tools: ISE Integrated Development Environment - Xilinx				
				error and stops the implementation Safety risk: **Low**
05	Allow Unexpanded Blocks	Specifies whether or not the tool continues to run if it encounters a block in the design that cannot be expanded to its correspondent primitive (NGD format) Values: Check box disabled by default	When this option is enabled, the tool generates a warning instead of an error if a block cannot be expanded, and writes the NGD file containing the unexpanded block This option may be useful to get early estimations of the implementation, performing the Translate process even if the design is still missing lower level modules	It is highly recommended to set this option to its default value (disabled) in order to detect blocks that cannot be expanded or missing design files Safety risk: **Low**
06	User Rules File for Netlister Launcher	Specifies the location and name of the user file to control how the tool parses files	The user rules file determines the acceptable netlist input files, the netlist readers and their default options	Using this kind of user rules files is a good practice to improve the configuration management aspects of the project Safety risk: **Low**
07	Allow Unmatched LOC Constraints	Specifies whether or not the tool ignores LOC constraints in the User Constraint File that do not match the net names in the netlist Values: Check box disabled by default	This option may be useful to get early estimations of the implementation, performing the Translate process even if LOC constraints do not match the net names	It is highly recommended to set this option to its default value (disabled) in order to detect net names mismatches due to incorrect LOC constraints or nets removed by the synthesis tool due to unexpected optimizations Safety risk: **High**
08	Allow Unmatched Timing Group Constraints	Specifies whether or not the tool ignores timing group constraints in the User Constraint File that cannot be found in the netlist Values: Check box disabled by default	This option may be useful to get early estimations of the implementation, performing the Translate process even if some timing group constraints are not found	It is highly recommended to set this option to its default value (disabled) in order to detect incorrect timing group constraints or logic removed by the synthesis tool due to unexpected optimizations Safety risk: **High**
Map Properties				
#	*Option/Function*	*Description*	*Benefits*	*Limitations and Recommendations*
09	Perform Timing-Driven Packing and Placement	Specifies whether or not the tool gives priority to timing critical paths during packing in the map process Values: Check box disabled by default	User generated timing constraints, recorded in the User Constraint File, are used to drive the packing and placement operations When enable, this option allows the tool to place the design as part of the map process	It is recommended to enable this option if there is a User Constraint File. In the absence of user timing constraints, the tool enters into Performance Evaluation Mode which may obtain results that are not necessarily the most optimal Timing-driven packing and placement is not an option for Xilinx latest devices (Virtex-5 and Virtex-6). For these devices, the design is automatically placed as part of the map process Safety risk: **Medium**
10	Placer Effort	Specifies the effort level to	The *Standard* option gives the	It is recommended to set this option

Place & Route Tools: ISE Integrated Development Environment - Xilinx				
	Level	apply to the map process Values: • Standard • High (default)	fastest run time with the lower mapping effort. This setting is appropriate for a less complex design *High* value gives the longest run time with the best mapping results. This setting is appropriate for a more complex design	to its default value (*High*) in order to get the better mapping results, regardless of the design complexity Despite the increment of the mapping process run time, having a higher quality placement can reduce the time needed to route and meet timing, resulting in an overall runtime reduction Safety risk: **Medium**
11	Placer Extra Effort	Specifies the extra effort level for timing-driven packing Values: • None (default) • Normal • Continue on Impossible	When the *Normal* option is selected, the tool runs until timing constraints are met unless they are found to be impossible to meet The *Continue on Impossible* setting continues working to improve timing until no more progress is made, even if timing constraints are impossible to meet	This option is only available if the *Placer Effort Level* is set to *High* It is recommended to set this option to the *Normal* value to focus on meeting timing constraints Safety risk: **Medium**
12	Starting Placer Cost Table (1-100)	Specifies a mapping initialization value to begin the map attempts Value: Any non-negative integer up to 100 (1 is the default value)	A cost table is a random seed to placement Each subsequent map attempt is assigned an incremental value based on the mapping initialization value The Starting Placer Cost Table can be modified when a design is close to meeting requirements, in order to get a slightly different result	It is highly recommended to set the starting placer cost table to a fixed value when timing and routing requirements are met, in order to ensure the reproducibility of the implementation process Safety risk: **High**
13	Extra Cost Tables	Specifies cost tables for use with highly utilized designs Value: Any non-negative integer (0 is the default value)	These cost tables can be used in conjunction with the *Starting Placer Cost Table* property in order to improve the placement of the design	This option, which is only available for Xilinx latest devices, is intended for highly congested designs only However, it may increment the compilation time and substantially reduce the quality of timing results For these reasons, it is highly recommended to set the extra cost table to its default value (0), in order to disable the option Safety risk: **High**
14	Combinatorial Logic Optimization	Physical synthesis property Instructs the tool to run a process that revisits the combinatorial logic within a design to see if any improvement can be made that will improve the overall quality of results Values: Check box disabled by default	This option re-synthesizes the placed and routed critical paths in the design to improve timing and area Timing constraints and logic packing information are considered to run this process	This option is available only when *Perform Timing-Driven Packing and Placement* is used It is recommended to enable this option (enabled) if a reduction of area and/or an improvement of timing are needed Safety risk: **Low**

Place & Route Tools: ISE Integrated Development Environment - Xilinx				
15	Register Duplication	Physical synthesis property Instructs the tool to replicate the registers to help control fanout Values: Check box disabled by default	Improves timing performance by replicating registers with high fanout Helps to meet the fanout constraints	This option is available only when *Perform Timing-Driven Packing and Placement* is used Xilinx recommends to disable this option and perform manual register duplication in the RTL HDL code, since in some cases registers may not be automatically replicated as expected by using this option together with the *Max Fanout* property Safety risk: **High**
16	Global Optimization	Physical synthesis property Allows the tool to perform global optimization routines on the fully assembled netlist before mapping the design Values: • Off (default) • Speed • Area • Power	Global optimization includes the following optimization routines: • Logic remapping and trimming • Logic and register replication and optimization • Logic replacement of tristates This option can be used to improve performance (*Speed*), reduce area utilization (*Area*) or reduce dynamic power consumption (*Power*) The use of this option is most effective when designs are built from multiple netlists or when the full set of optimization techniques are not used during the synthesis phase	The use of global optimization techniques may extend the runtime for the map process When optimizing for area or power, there may be a trade-off in timing performance It is recommended to set this option to its default value (Off) especially if formal methods are used for verification such as LEC (refer to § 4.2.3.5) Safety risk: **Medium**
17	Retiming	Physical synthesis property Allows the tool to move registers to increase the overall clock frequency Values: Check box disabled by default	When enabled, this option moves registers forward or backwards through the logic to balance out the delays in a timing path, aiming to increase the overall clock frequency	This property is only available when the Global Optimization option is enabled It is highly recommended to set this option to its default value (disabled) for the following reasons: • This optimization can move combinational logic across different clock domains • Design verification may become more complicated because register names, position, and functionality no longer match the RTL description Safety risk: **High**
18	Equivalent Register Removal	Physical synthesis property Allows the tool to remove registers with redundant	When enabled, this option removes the registers with redundant functionality after examining if their removal will	It is recommended to set this option to its default value (disabled) and enable the equivalent option in the synthesis tool only if *Register*

Place & Route Tools: ISE Integrated Development Environment - Xilinx				
		functionality in order to increase clock frequencies Values: Check box disabled by default	improve the overall timing	*Duplication* is disabled and no register is replicated directly in the RTL HDL code Safety risk: **High**
19	Ignore User Timing Constraints	Controls the use of timing constraints found in the User Constraint File (UCF) during placement Values: Check box disabled by default	When enabled, timing constraints in the UCF are ignored by the tool, which will run without timing constraints For some devices, timing constraints are automatically generated depending on the setting of the *Timing Mode* option	It is highly recommended to set this option to its default value (disabled) in order to process the user timing constraints Safety risk: **Medium**
20	Timing Mode	This property is enabled when the *Ignore User Timing Constraints* is enabled Selects whether or not the tool should automatically generate timing constraints Values: • Performance Evaluation (default) • No Timing Driven	When the *Performance Evaluation* mode is selected, timing constraints for all internal clocks are generated automatically and dynamically adjusted during the map process to increase performance The *No Timing Driven* mode instructs the tool to run with no timing constraints	The *Performance Evaluation* mode is only available on Virtex-5 devices It is recommended not to use this option, disabling the *Ignore User Timing Constraints* property Safety risk: **Medium**
21	Trim Unconnected Signals	Specifies whether or not to trim unconnected components and nets from the design before mapping occurs Values: Check box enabled by default	Leaving unconnected components and nets may be useful for estimating the logic resources required on partially finished designs	It is recommended to set this option to its default value (enabled) to remove unnecessary logic It is also recommended to check the map report in order to identify the trimmed logic, with the purpose of removing the unnecessary logic from the RTL HDL code Safety risk: **Medium**
22	Replicate Logic to Allow Logic Level Reduction	Allows the tool to replicate logic to reduce fanout Values: Check box enabled by default	When enabled, this option allows to replicate logic elements such as single drivers that drive multiple loads to reduce fanout and enable a mapping strategy that may more readily meet timing constraints	Xilinx recommends to disable this option and perform manual register duplication in the RTL HDL code, since in some cases registers may not be automatically replicated as expected by using this option together with the *Max Fanout* property Safety risk: **High**
23	Allow Logic Optimization Across Hierarchy	Instructs the tool to ignore any *Keep Hierarchy* properties set for synthesis in order to perform optimization across any hierarchy boundaries Values: Check box disabled by default	This option is used to preserve the signals that span the hierarchical boundaries for the purpose of simulation, or to ensure that optimizations do not affect the behaviour of a design using partitions. These optimizations may improve timing performance	It is recommended to set this option to its default value (disabled) to avoid optimization crossing hierarchy boundaries Safety risk: **Medium**
24	Optimization Strategy (Cover Mode)	Specifies the criteria used during the cover phase of the mapping process	The cover phase assigns the logic to CLB function generators (LUTs)	The *Speed* option may produce a large increase in the number of LUTs

Place & Route Tools: ISE Integrated Development Environment - Xilinx				
		Values: • Area (default) • Speed • Balanced • Off	The *Area* option is used to reduce the number of LUTs, while the *Speed* setting it helps achieving timing constraints The *Balanced* option may be used to balance area and speed, reducing both the number of LUTs and levels of LUTs	It is recommended to set this option to *Balanced* in order to take advantage of the trade-off between area and timing Safety risk: **Medium**
25	Generate Detailed MAP Report	Instructs the tool to generate a detailed MAP report Values: Check box disabled by default	The detailed MAP report displays important information such as redundant blocks that were removed and signals that were merged during the mapping process	It is recommended to enable this option (enabled) to generate the detailed MAP report, in order to get useful information to check mapping optimizations Safety risk: **Low**
26	Use RLOC Constraints	Instructs the tool to use the RLOC information found in the User Constraint File (UCF) Values: Check box enabled by default	When enabled, the RLOC constraints, that contain the relative placement of one CLB to another, are processed	It is highly recommended to set this option to its default value (enabled) in order to process the user RLOC timing constraints Safety risk: **Medium**
27	Pack I/O Registers/Latches into IOBs	Controls the packing of FFs or latches within an I/O cell Values: • Off (default for Virtex-5 devices) • For Inputs Only • For Outputs Only • For Inputs and Outputs (default for all other devices)	The registers within the I/O cells may decrease clock-to-in and clock-to-out times, allowing the design to run faster The architecture and higher voltage of these registers made them more robust against SEU	It is recommended to enable this option for inputs and outputs, in order to benefit of the advantages of I/O cell registers (better timing and protection against SEU) In some cases, the architecture does not allow all registers to be packed into IOBs, therefore it is a good practice to review both the map report and the Technology Viewer to check the registers packed into IOBs Safety risk: **Medium**
28	Disable Register Ordering	Specifies whether or not the tool uses register ordering optimization Values: Check box disabled by default	This option is no longer available in Xilinx ISE v.11.1 and higher	
29	Maximum Compression	Instructs the tool to pack the design logic as densely as possible Values: Check box disabled by default	When enable, this option allows the tool to map the design using less logic resources (higher density packing)	This option is only available on Virtex-5 devices It is recommended to set this option to its default value (disabled) to avoid negative performance in the place and route process: higher delays and more unrouted nets Safety risk: **Medium**
30	CLB Pack Factor Percentage	Specifies how densely logic will be partitioned using a percentage value Value: Any non-negative	When the set value is lower than 100, this option allows the tool to map the design using less logic resources (higher density packing)	This option is not available on Virtex-5 devices This option can not be applied when the *Perform Timing-Driven Packing*

Place & Route Tools: ISE Integrated Development Environment - Xilinx				
		integer up to 100 (100 is the default value)		*and Placement* property is enabled It is recommended to set this option to its default value (100%) to avoid negative performance in the place and route process: higher delays and more unrouted nets Safety risk: **Medium**
31	Tri-state Buffer Transformation Mode	Specifies the type of bus transformation to be performed by the tool Values: • Off (default) • On • Aggressive • Limit	This option is no longer available in Xilinx ISE v.11.1 and higher	
32	LUT Combining	Instructs the tool to perform LUT combining optimization Values: • Off (default) • Auto • Area	LUT combining optimization merge LUT pairs with common inputs into single dual-output LUTs in order to improve design area The *Auto* option tries to make a trade-off between area and speed, while the *Area* option performs maximum LUT combining to provide an implementation as smaller as possible	This option is only available on Virtex-5 devices The LUT combining optimization may reduce design speed It is recommended to set this option to its default value (Off) to avoid unexpected results in timing performance Safety risk: **Medium**
33	Map Slice Logic into Unused Block RAMs	Specifies whether or not the tool attempts to place LUTs and FFs into unused block RAMs Values: Check box disabled by default	This optimization may improve design area	It is recommended to set this option to its default value (disabled) Implement logic into memory blocks may have an impact on the design reliability since memory blocks are more sensitive to SEU Safety risk: **High**
34	Power Reduction	Specifies whether or not the tool optimizes placement during timing-driven packing and placement to reduce the power consumed by the design Values: Check box disabled by default	This option allows the tool to reduce the power consumed by the design	This option can be applied only when the *Perform Timing-Driven Packing and Placement* property is enabled, and it is only available for Xilinx latest devices It is recommended to set this option to its default value (disabled) and use power reduction techniques directly in the RTL HDL code such as: • Adding control logic to handle Block RAM enable signals • Using LUT instead of Block RAM for small memory blocks • Initializing registers Safety risk: **Medium**

Place & Route Tools: ISE Integrated Development Environment - Xilinx				
35	Power Activity File	Specifies a simulation file to guide the tool when it optimizes the design for poser reduction	This file may be useful to achieve better power reduction results	The efficiency of this file depends on the quality and exhaustiveness of the performed simulation Safety risk: **Low**
36	Enable Multi-Threading	Instructs the tool to enable multi-threading Values: • Off (default) • 2	This option, when enabled, allows the tool engine to take advantage of multi-core processor workstations in order to speed up place and route times	Multi-threading has evolved positively in the latest releases of Xilinx ISE, showing a much better performance in desktop computers compared with laptops featuring the equivalent hardware configuration It is recommended to run the same mapping process with and without multi-threading, check the output equivalence and assess the processing time before using this option Safety risk: **Low**
Place & Route Properties				
#	***Option/Function***	***Description***	***Benefits***	***Limitations and Recommendations***
37	Place & Route Mode	Specifies the type of place and route to be performed Values: • Normal Place and Route (default for all other devices) • Place Only • Route Only (default for Virtex-5 and Virtex-6) • Reentrant Route	The *Normal Place and Route* option allows the user to set the effort levels When the *Place Only* option is selected, the router process does not run (The whole PAR process should be run at least once to use this option) When the *Route Only* option is selected, the placer process does not run and the current placement is kept (The whole PAR process should be run at least once to use this option) The *Reentrant Route* option keeps the placement and routing. The router runs one time using the existing routing as a starting point	Xilinx does not recommend the *Reentrant Route* option (only for advanced flows) The *Route Only* option is not recommended when the design was placed using the *Perform Timing-Driven Packing and Placement* property It is recommended to set this option to the default value selected for each device family Safety risk: **Medium**
38	Place & Route Effort Level (Overall)	Specifies the effort level to apply to the place and route process Values: • Standard • High (default)	The *Standard* option gives the fastest run time with the lower place and route effort. This setting is appropriate for a less complex design *High* value gives the longest run time with the best place and route results. This setting is appropriate for a more complex design	It is recommended to set this option to its default value (*High*) in order to get the better place and route results, regardless of the design complexity Safety risk: **Medium**
39	Extra Effort (Highest PAR level only)	Specifies the extra effort level for the place and route Values: • None (default)	When the *Normal* option is selected, the tool runs until timing constraints are met unless they are found to be impossible to meet	This option is only available if the *Place & Route Effort Level* is set to *High* It is recommended to set this option

Place & Route Tools: ISE Integrated Development Environment - Xilinx				
		• Normal • Continue on Impossible	The *Continue on Impossible* setting continues working to improve timing until no more progress is made, even if timing constraints are impossible to meet	to the *Normal* value to focus on meeting timing constraints Safety risk: **Medium**
40	Starting Placer Cost Table (1-100)	Specifies a placement initialization value to begin the place and route attempts Value: Any non-negative integer up to 100 (1 is the default value)	Each subsequent map attempt is assigned an incremental value based on the placement initialization value The Starting Placer Cost Table can be modified when a design is close to meeting requirements, in order to get a slightly different result	If a number is specified for the *Starting Placer Cost Table* property in the map settings, the same number is used for place and route It is highly recommended to set the placement initialization value to a fixed value when timing and routing requirements are met, in order to ensure the reproducibility of the implementation process Safety risk: **High**
41	Ignore User Timing Constraints	Controls the use of timing constraints found in the User Constraint File (UCF) during place and route Values: Check box disabled by default	When enabled, timing constraints in the UCF are ignored by the tool, which will run without timing constraints For some devices, timing constraints are automatically generated depending on the setting of the *Timing Mode* option	It is highly recommended to set this option to its default value (disabled) in order to process the user timing constraints Safety risk: **Medium**
42	Timing Mode	This property is enabled when the *Ignore User Timing Constraints* is enabled Selects whether or not the tool should automatically generate timing constraints Values: • Performance Evaluation (default) • No Timing Driven	When the *Performance Evaluation* mode is selected, timing constraints for all internal clocks are generated automatically and dynamically adjusted during the place and route process to increase performance The *No Timing Driven* mode instructs the tool to run with no timing constraints	The *Performance Evaluation* mode is only available on Virtex-5 devices It is recommended not to use this option, disabling the *Ignore User Timing Constraints* property Safety risk: **Medium**
43	Use Bonded I/Os	Specifies whether or not the tool places internal I/O logic into bonded I/O sites in which the I/O pad is not used Values: Check box disabled by default	This option, when enabled, allows the tool to route through bonded I/O sites (I/O locations on the die connected to the package)	It is highly recommended to set this option to its default value (disabled) in order to avoid logic placed in bonded sites connected to external signals, power or ground, which may cause unexpected behaviour of the implemented design Safety risk: **High**
44	Generate Asynchronous Delay Report	Specifies whether or not to generate an asynchronous delay report when the place and route process is run Values: Check box disabled by default	This report contains a list of all nets in the design and the delays of all loads on the net	It is recommended to enable this option (enabled) to generate the asynchronous delay report, in order to get useful information for clock analysis Safety risk: **Low**
45	Generate Clock Region Report	Specifies whether or not to generate a clock region report when the place and route	This report contains information on the resource utilization of each clock region and lists any clock	It is recommended to enable this option (enabled) to generate the clock region report, in order to get

Place & Route Tools: ISE Integrated Development Environment - Xilinx				
		process is run Values: Check box disabled by default	conflicts between global clock buffers in a clock region	useful information to verify the implemented clock strategy Safety risk: **Low**
46	Generate Post-Place & Route Simulation Model	Specifies whether or not to generate a post-place and route simulation model when the place and route process is run Values: Check box disabled by default	This model is used to perform the post-place and route simulation A dedicated dialog box allows to configure this model, setting the selected HDL language, and the name of the design instance among other options	It is recommended to enable this option (enabled) to generate the post-place and route simulation model, in case post-place and route simulations are planned in the verification process Safety risk: **Low**
47	Generate Post-Place & Route Power Report	Specifies whether or not to generate a post-place and route power report when the place and route process is run Values: Check box disabled by default	This report provides information about the design power consumption and thermal dissipation	It is recommended to enable this option (enabled) to generate the post-place and route power report, in order to get useful information to check the effectiveness of power optimization techniques Safety risk: **Low**
48	Power Reduction	Specifies whether or not to optimize routing to reduce power consumption Values: Check box disabled by default	This option allows the tool to reduce the power consumed by the design	It is recommended to set this option to its default value (disabled) and use power reduction techniques directly in the RTL HDL code such as: • Adding control logic to handle Block RAM enable signals • Using LUT instead of Block RAM for small memory blocks • Initializing registers Safety risk: **Medium**
49	Power Activity File	Specifies a simulation file to guide the tool when it optimizes the design for poser reduction	This file may be useful to achieve better power reduction results	The efficiency of this file depends on the quality and exhaustiveness of the performed simulation Safety risk: **Low**
50	Enable Multi-Threading	Instructs the tool to enable multi-threading Values: • Off (default) • 2 • 3 • 4	This option, when enabled, allows the tool engine to take advantage of multi-core processor workstations in order to speed up place and route times	Multi-threading has evolved positively in the latest releases of Xilinx ISE, showing a much better performance in desktop computers compared with laptops featuring the equivalent hardware configuration It is recommended to run the same mapping process with and without multi-threading, check the output equivalence and assess the processing time before using this option Safety risk: **Low**

Table 6.5. ISE Integrated Development Environment Tool Specific Assessment of Options and Functionalities

Following design-specific files should be subject to revision control:

- Project file (including the name and location of the synthesized netlist(s) and the place and route options)
- User Constraints File (UCF)
- Tcl script to guide the place and route process (if any)
- Design technology libraries

6.1.2.3 Libero Integrated Development Environment (Actel)

The Actel Libero IDE integrates third party synthesis tools (such as Synplify Pro from Synopsis assessed in § 6.1.1.3), the place and route tool (Actel Designer) and STA (SmartTime assessed in § 6.1.3.3).

The specific assessment is performed using as reference the version 9.1 of Actel Libero IDE software.

Table 6.6 summarizes its place and route options and functions (referred by Actel as compile and layout) along with their benefits, limitations and recommendations for use.

Place & Route Tools: Libero IDE Integrated Development Environment - Actel				
Compile Options				
#	*Option/Function*	*Description*	*Benefits*	*Limitations and Recommendations*
01	Abort Compile if errors are found in the Physical Design Constraints (PDC)	This option stops the flow if any error is reported in reading the PDC file Values: Check box enabled by default	When enabled, this option allows to detect errors in the PDC If this option is disabled, the errors are reported as warnings. However the flow always stop in case of Tcl errors or wrong local clock assignments Disabling the option can be useful to do an early estimation of the design compilation even if the PDC is not well-defined yet	It is highly recommended to set this option to its default value (enabled) in order to ensure a valid PDC file Safety risk: **Medium**
02	Display object names that are no longer found after netlist matching is performed on the design	Displays netlist objects in the PDC file that are not found in the imported netlist during the Compile ECO mode (Incremental design flow) Values: Check box disabled by default	Incremental design flows are based on compile points and requires specific settings for the synthesis tool (“difference-based” incremental synthesis) Incremental design flows are useful for team-oriented design and also to reduce the overall runtime process	Incremental design flows may involve an additional effort in order to assess that design modifications do not disturb other parts of the design This methodology increases the complexity of the project baseline, traceability and configuration management processes since several iterations with different settings are required to generate the final implementation Therefore, it is not recommended to use Incremental design flows unless the methodology and the supporting processes are properly mastered to ensure the reproducibility of the implementation process Safety risk: **High**
03	Limit the number of displayed messages to:	Defines the maximum number of errors/warnings to be displayed in the case of reading ECO constraints Value: Any non-negative integer (10000 is the default value)	This option is used in incremental design flows, and allows to control the quantity of information displayed on the tool console	
04	Demote global nets whose fanout is less than:	Enables the global clock demotion of global nets to regular nets Values: • Check box disabled by default	When enabled, this option allows to reserve the use of global clock networks to high fanout nets	Automatic global demotion is recommended only for small fanout nets The clock tree architecture, including pin assignment and/or placement constraints, should be

Place & Route Tools: Libero IDE Integrated Development Environment - Actel				
		• Text box to select the fanout value: Any non-negative integer (12 is the default value)		clearly defined before enabling automatic global demotion It is recommended to run Compile with this option set to its default value (disabled) and analyze the results in terms of timing and routability before attempting a complete place and route with automatic global demotion Safety risk: **Medium**
05	Promote regular nets whose fanout is greater than:	Enables the global clock promotion of regular nets to global clock networks Values: • Check box disabled by default • Text box to select the fanout value: Any non-negative integer (200 is the default value) • Text box to set the maximum number of nets to promote: Any non-negative integer (0 is the default value)	When enabled, this option allows to reserve the use of global clock networks to high fanout nets	Automatic global promotion is recommended only for high fanout nets. Driving high fanout nets with a clock network may improve timing and routability The clock tree architecture, including pin assignment and/or placement constraints, should be clearly defined before enabling automatic global promotion It is recommended to run Compile with this option set to its default value (disabled) and analyze the results in terms of timing and routability before attempting a complete place and route with automatic global promotion Safety risk: **Medium**
06	Limit the number of shared instances between any two non-overlapping local clock regions to:	Defines the maximum number of shared instances allowed to perform the legalization Values: Text box to select the value: Any non-negative integer in the range of 0-1000 (12 is the default value, while 0 prevents legalization)	Actel devices organize clock regions in quadrants. Legalization ensures that the number of different clock nets used in every region is less or equal to the number of clock resources available in that region This option allows adding more flexibility to perform the legalization process	It is recommended to set this option to its default value (12) Higher values may lead to a large number of shared instances, which may indicate floorplanning problems Safety risk: **Low**
07	When inserting buffers to legalize shared instances between non-overlapping local clock regions, limit the buffers' fanout to:	Defines the maximum fanout value used during buffer insertion for clock legalization Values: Text box to select the value → Any non-negative integer in the range of 0-1000 (12 is the default value, while 0 prevents legalization)	This option allows to control and disable clock legalization	It is recommended to set this option to its default value (12) to allow clock legalization Safety risk: **Medium**
08	Combine registers into I/Os whenever possible	Instructs the tool to pack registers into I/O cells Values: Check box disabled by default	The registers within the I/O cells may decrease clock-to-in and clock-to-out times, allowing the design to run faster The architecture and higher voltage of these registers made	It is recommended to enable this option, in order to benefit of the advantages of I/O cell registers (better timing and protection against SEU) In some cases, the architecture does

Place & Route Tools: Libero IDE Integrated Development Environment - Actel				
			them more robust against SEU	not allow all registers to be packed into I/O cells, therefore it is a good practice to review the Technology Viewer to check the registers packed into I/O cells Safety risk: **Medium**
09	Delete buffers and inverter trees whose fanout is less than:	This option enables buffer tree deletion on the global signals from the netlist Values: • Check box disabled by default • Text box to select the fanout value: Any non-negative integer (12 is the default value)	When enabled, this option allows to reserve the use of buffers for high fanout nets	It is recommended to set this option to its default value (disabled) to avoid automatic buffer tree deletion of high fanout nets Safety risk: **Medium**
10	Limit the number of displayed high fanout nets to:	Enables FFs net sections in the compile report and defines the number of high fanout nets to be displayed in the report Values: Text box to select the value → Any non-negative integer (10 is the default value)	This option allows to generate compile reports easier to review	It is recommended to set this option to its default value (10) Safety risk: **Low**
Layout Options				
#	*Option/Function*	*Description*	*Benefits*	*Limitations and Recommendations*
11	Timing-Driven	Timing-Driven layout's primary goal is to meet timing constraints Values: Check box disabled by default	Directs the tool to place logic elements in the device to meet the timing constraints defined in the SDC file by the user or automatically generated by the tool Timing-Driven layout typically delivers better performance than standard layout	It is highly recommended to enable this option in order to consider timing constraints when performing the layout process Standard layout targets efficient usage of logic resources but ignores timing constraints Safety risk: **Medium**
12	Power-Driven	Power-Driven layout's primary goal is to reduce dynamic power while still maintaining timing constraints Values: Check box disabled by default	When enabled, this option helps to reduce dynamic power	This option is only available when *Timing-Driven* option is selected It is recommended to do a first run of Timing-Driven layout with the *Power-Driven* option disabled in order to get a VCD (Value Change Dump) after post-place and route simulation of the back-annotated netlist. The VCD file can be imported and its simulation vectors be used in a second run of Timing-Driven layout with *Power-Driven* option enabled. It is important to verify that timing constraints are met in both runs

Place & Route Tools: Libero IDE Integrated Development Environment - Actel				
				Safety risk: **Medium**
13	Run Place	This option allows the tool to run the placer function during layout Values: Check box enabled by default if Layout has not been run before	When disabled, allows to unload the Layout process in incremental design flows	It is recommended to run always the placer function in order to keep fixed settings for Layout Safety risk: **Medium**
14	Place Incrementally	This option allows the use of previous placement data as the initial placement for the next run Values: Check box disabled by default	Incremental design flows are useful for team-oriented design and also to reduce the overall runtime process	Incremental design flows may involve an additional effort in order to assess that design modifications do not disturb other parts of the design This methodology increases the complexity of the project baseline, traceability and configuration management processes since several iterations with different settings are required to generate the final implementation Therefore, it is not recommended to use Incremental design flows unless the methodology and the supporting processes are properly mastered to ensure the reproducibility of the implementation process Safety risk: **High**
15	Lock Existing Placement (Fix)	This option allows to preserve previous placement as the initial placement for the next run Values: Check box disabled by default		
16	Run Route	This option allows the tool to run the router function during layout Values: Check box enabled by default if Layout has not been run before	When disabled, allows to unload the Layout process in incremental design flows	It is recommended to run always the router function in order to keep fixed settings for Layout Safety risk: **Medium**
17	Route Incrementally	This option allows to reroute a design when some nets failed to route. It can also be used when the input netlist has undergone an ECO Values: Check box disabled by default	Incremental design flows are useful for team-oriented design and also to reduce the overall runtime process	Incremental design flows may involve an additional effort in order to assess that design modifications do not disturb other parts of the design This methodology increases the complexity of the project baseline, traceability and configuration management processes since several iterations with different settings are required to generate the final implementation In addition, incremental routing should only be used if a low number of nets fail to route (Actel advises less than 50) Therefore, it is not recommended to use Incremental design flows unless the methodology and the supporting processes are properly mastered to

Place & Route Tools: Libero IDE Integrated Development Environment - Actel				
				ensure the reproducibility of the implementation process Safety risk: **High**
18	Use Multiple Passes	This option allows the tool to run layout multiple times with different seeds Values: Check box disabled by default	When enabled, this option may improve layout quality by selecting from a greater number of layout results The following reports are saved for each path in order to support later analysis: • Timing • Maximum delay timing violations • Minimum delay timing violations • Power	This option increases the process runtime and should be properly configured to be efficient It is recommended to use this option to improve layout results Safety risk: **Low**
19	Multiple Passes: Number of Passes	Sets the number of iteration Values: Slider bar ranging from 1 to 25 (5 is the default value)	Allows to modulate the effort of the *Multiple Passes* function	It is recommended to set this option to its default value (5) to find a balance between process runtime and layout quality Safety risk: **Low**
20	Multiple Passes: Start at seed index (1 – 101):	Specifies the seed that the tool uses to randomly determining the initial placement for the current design Value: Any non-negative integer (1 is the default value)	The seed can be modified when a design is close to meeting requirements, in order to get a slightly different result	It is highly recommended to set the Seed to a fixed value when timing and routing requirements are met, in order to ensure the reproducibility of the implementation process Safety risk: **High**
21	Multiple Passes Measurement	Selects the measurement criteria when comparing layout results Values: One of the following four criteria should be selected: • Slowest Clock • Specific Clock • Timing Violations • Total Power	*Slowest Clock* criteria uses the slowest clock frequency in the design in a given pass as the performance reference for the layout pass The *Specific Clock* option selects to use a specific clock from a list of identified clocks as the performance reference for all passes *Timing Violation* criteria selects the pass that best meets the slack or timing violations constraints. This criteria, which requires user timing constraints (SDC), has the following configuration settings: • Maximum Delay: Examines timing violations (slacks) obtained from maximum delay analysis • Minimum Delay: Examines timing violations (slacks)	It is recommended to set the *Timing Violation* criteria to determine the best pass with the following options: • Maximum Delay • Select by Total Negative Slack. It is a more comprehensive analysis, and in case of no negative slack exists, then the worst slack is used to evaluate that pass • Disable *Stop on first pass without violations* check box in order to obtain better timing results (greater amount of positive slack) even if process runtime may be longer *Timing Violation* criteria considers user timing constraints Safety risk: **Medium**

Place & Route Tools: Libero IDE Integrated Development Environment - Actel				

			obtained from minimum delay analysis
			• Select by:
			  • Worst Slack
			  • Total Negative Slack
			• Stop on first pass without violations (no negative slacks reported)
			Total Power criteria selects the pass that produces the lowest total power (static + dynamic) out of all layout passes	

22	Save Design File For Each Pass	This option allows saving design data for all passes Values: Check box disabled by default	When enabled, this option allows recording design data for every passes, not only the best This data may be useful to analyse the result of each pass in more detail	It is recommended to enable this option, even if it requires more disk space, since generates useful data for further analysis Safety risk: **Low**
Advanced Layout Options for Timing-Driven Layout				
#	*Option/Function*	*Description*	*Benefits*	*Limitations and Recommendations*
23	High Effort Layout	This option turns on netlist optimizations of combinational logic to obtain better performance Values: Check box disabled by default	The placer may change the mapping of the logic components, preserving the original functionality of the design, in order to improve the overall performance This option can be combined with *Multiple Passes* mode to achieve better performance	Layout runtime increases and names and types of the combinational core logic primitives may change In incremental design flows, if the *Lock Existing Placement* option is enabled, then the placer runs in regular effort mode It is recommended to set this option to its default value (disabled) in order to avoid unexpected optimizations Safety risk: **Medium**
24	Sequential Optimization	This option turns on netlist optimizations of sequential cells in the High Effort Layout mode Values: Check box disabled by default	When enabled, this option enables register retiming, moving registers forward or backwards through the logic to balance out the delays in a timing path, aiming to increase the overall clock frequency	The names of registers may change unless they are assigned a physical constraint, referred in a timing constraint, or have a preserve property It is recommended to set this option to its default value (disabled) for the following reasons: • This optimization can move combinational logic across different clock domains • Design verification may become more complicated because register names, position, and functionality no longer match the RTL

Place & Route Tools: Libero IDE Integrated Development Environment - Actel				
				description Safety risk: **High**
25	Router: Repair Minimum Delay Violations	This option allows the router to attempt to repair paths that have minimum delay violations or hold time violations Values: Check box disabled by default	This option enables an additional route that will attempt to repair paths that have minimum delay and hold time violations No additional logic is inserted	This function is only suited to repair paths with small hold and minimum delay violations (0 to 3 ns) To repair paths with large timing violations, manual placement and/or RTL HDL code modification may be needed Safety risk: **Medium**

Table 6.6. Libero Integrated Development Environment Tool Specific Assessment of Options and Functionalities

Following design-specific files should be subject to revision control:

- Project file (including the name and location of the synthesized netlist(s) and the place and route options)
- Physical Design Constraints file (PDC)
- Tcl script to guide the place and route process (if any)
- Design technology libraries

6.1.3 Static Timing Analysis Tools

6.1.3.1 TimeQuest Timing Analyzer (Altera)

TimeQuest Timing Analyzer is the advanced STA tool integrated within Altera's Quartus II IDE. A timing netlist should be generated before running the TimeQuest analyzer, specifying the timing model and device speed grade.

The specific assessment is performed using as reference the version 11.1 of Altera's Quartus II software.

Table 6.7 summarizes its options and functions along with their benefits, limitations and recommendations for use.

STA Tools: TimeQuest Timing Analyzer - Altera				
Create Timing Netlist Settings				
#	*Option/Function*	*Description*	*Benefits*	*Limitations and Recommendations*
01	Input Netlist	Specifies the type of timing netlist the tool uses to calculate path delays Values: • Post-fit (default) • Post-map	The post-fit netlist contains physical synthesis optimizations and fitting, while the post-map netlist includes only logic synthesis information	It is recommended to set this option to its default value (*Post-fit*) to get more reliable timing results Safety risk: **Low**
02	Delay Model	Specifies the delay model the tool uses when performing timing analysis Values: • Slow-corner (default) • Fast-corner	The *Slow-corner* value uses the worst-case timing model to compute delays depending on the speed grade of the device specified in the Speed Grade list The *Fast-corner* value uses the best-case timing model to compute delays	It is highly recommended to set this option to the worst-case value (*Slow-corner*) to get more conservative timing results Safety risk: **Medium**
03	Zero IC Delays	Allows the tool to compute timing with no interconnection delays Values: Check box disabled by default	This option can be used early in the design process to determine if the design can meet timing requirements	It is highly recommended to set this option to its default value (disabled) to get more reliable timing results Safety risk: **Medium**
TimeQuest Timing Analyzer Settings				
#	*Option/Function*	*Description*	*Benefits*	*Limitations and Recommendations*
04	SDC Files to Include in the Project	Allows to select the Synopsys Design Constraints (SDC) file to include in the project	This function allows to identify all the constraints files involved in the project, including the constraints that may be defined within the Quartus II project file	Is it recommended to arrange the files in the order they should be read to prevent compilation errors in case of multiple files with dependencies Safety risk: **Low**
05	Enable Advanced I/O Timing	Instructs the tool to use Advance I/O Timing to generate I/O timing results	Timing results are based on the Board Trace Model specified for each pin by means of the Capacitive Loading and/or Board	It is recommended to enable this option if a Board Trace Model is provided

STA Tools: TimeQuest Timing Analyzer - Altera				
		Values: Check box disabled by default for device families allowing the activation of this option	Trace Model tabs of the Device and Pin Options Dialog Box (refer to § 6.1.2.1) The timing analyzer reports provide output buffer to pin delays only. The Advance I/O Timing reports provide transition time assignments, board trace delays, and signal integrity metrics	The latest Altera device families have this option always enabled Safety risk: **Low**
06	Enable Multicorner Timing Analysis during Compilation	Instructs the tool to consider all corner timing delays, including both fast-corner timing and slow-corner timing to meet timing requirements at both corners Values: Check box enabled by default	When enabled, this option directs the tool to analyze the timing constraints defined in the SDC file considering the following to cases: • Slow-corner at maximum and minimum specified temperatures: Slowest manufactured device for a given speed grade, operating under low-voltage conditions • Fast-corner at minimum specified temperature: Fasted manufactured device for a given speed grade, operating under high-voltage conditions If the option is disabled, only slow-corner timing is considered	It is highly recommended to enable this option in order to perform the timing analysis taking into account different operating conditions depending on Process-Voltage-Temperature (PVT) parameters Due to process variation and changes in operating conditions, delays on some paths can be significantly smaller than those in the slow-corner. This can result in hold time violations on those paths Multi-Corner Timing may identify these timing issues, covering the range of the device's operating conditions and providing more accurate timing results Multicorner analysis becomes more important as devices geometries become smaller than 90 nm Turning on this option does not enable multicorner analysis in the Fitter which should be set separately (refer to § 6.1.2.1) Safety risk: **High**
07	Enable Common Clock Path Pessimistic Removal	Instructs the tool to remove common clock path pessimism (CCPP) during slack computation Values: Check box enabled by default	Minimum and maximum delay variation can occur when two different delay values are used for the same clock path, resulting in an overlay pessimistic analysis This option, when enabled, accounts for the minimum and maximum delay variation associated with common clock paths during timing analysis and adds the difference to the appropriate slack equation to compensate the overlay pessimistic scenario	It is recommended to set this option to its default value (enabled) to get more realistic timing results, since it is highly unlikely to experience both extreme variations on the exact same signal Safety risk: **Medium**
08	Report Worst-case Paths during Compilation	Instructs the tool to report the worst-case slack for each clock domain in the design	This option, when enabled, may be useful to identify timing critical paths	It is recommended to enable this option (*enabled*) to generate more comprehensive timing reports

STA Tools: TimeQuest Timing Analyzer - Altera				
		Values: Check box disabled by default		It is also useful to use the tool GUI *Report Timing* setting to select the types of path to be included in the timing report for a better analysis Safety risk: **Low**
09	Tcl Script File for Customizing Reports during Compilation	Allows to select a Tcl script file that customizes reporting options	The use of scripts improves the traceability and reproducibility of the design process	The use of scripts is recommended. These scripts should be subject to revision control as part of the configuration management process Safety risk: **Low**
10	Run Default Timing Analysis before Running Custom Script	Instructs the tool to run the default timing analysis prior to running the user-specified report script Values: Check box disabled by default	n.a.	It is recommended to set this option to its default value (disabled) Safety risk: **Low**
11	Metastability Analysis	Specifies the project-wide level of metastability analysis to perform during compilation Values: • Off • Auto (default) • Forced If Asynchronous	This option controls how aggressively the tool identifies registers as synchronization register chains for the entire design The *Auto* value is used to identify valid synchronization registers that are part of a chain including more than one register that contains no combinational logic The *Forced If Asynchronous* value identifies synchronization register chains if the tool detects an asynchronous signal transfer The tool also reports the MTBF of the detected synchronization chains	It is highly recommended to set this option to its default value (*Auto*), and adapt the *Synchronization Register Chain Length* option of the Quartus II Integrated Synthesis tool (refer to § 6.1.1.1) to the synchronization chain length used in the design It is also a good practice to review the timing report and check that all the synchronization chains specified in the RTL HDL code were detected by the tool Safety risk: **High**

Table 6.7. Quartus II TimeQuest Timing Analyzer Tool Specific Assessment of Options and Functionalities

6.1.3.2 Trace Timing Analyzer (Xilinx)

Trace Timing Analyzer is the advanced STA tool integrated within Xilinx's ISE IDE. A timing netlist should be generated before running the Trace Timing analyzer, specifying the timing model and device speed grade.

This tool used a "relative minimum" method to perform timing analysis. This method consists in the application of a factor of the maximum value, which is based on characterization of the analysed device family. For new devices (Virtex-6 and following series), a multicorner analysis is performed, where the timing analysis is done considering both the slow process corner (high temperature and low voltage) and fast process corner (low temperature and high voltage).

The specific assessment is performed using as reference the version 12.4 of Xilinx ISE software.

Table 6.8 summarizes the Trace Timing Analyzer options and functions along with their benefits, limitations and recommendations for use.

STA Tools: Trace Timing Analyzer - Xilinx				
General Settings				
#	*Option/Function*	*Description*	*Benefits*	*Limitations and Recommendations*
01	Analyze Against	This option allows to select the timing paths to be analysed Values: • Design Timing Constraints • Auto-generated Timing constraints • User Specified Paths by Defining Endpoints • User Specified Paths by Defining Clock and I/O Timing	When the *Design Timing Constraints* option is selected, the tool performs a timing constraints analysis and reports on all paths covered by user timing constraints (UCF), including failing paths and fastest paths The *Auto-generated Timing Constraints* option instructs the tool to report the maximum clock frequencies for all clocks in the design, worst-case setup and hold times for inputs, worst-case clock to out times for outputs, and the worst-case timing for all clock paths. In this mode, the tool ignores user timing constraints The *User Specified Paths by Defining Endpoints* option instructs the tool to report the worst-case path delays for all paths that are selected by the user. All user constraints, except time group constraints, are ignored. Paths are selected by means of a dedicated dialog box When the *User Specified Paths by Defining Clock and I/O Timing* option is selected, the tool performs a detailed analysis of user-specified clock and I/O timing constraints. All user	It is recommended to perform the timing analysis against the user timing constraints (*Design Timing Constraints* option) and enable the analysis of unconstrained paths in order to analyse the paths which were not covered by the UCF Safety risk: **Medium**

STA Tools: Trace Timing Analyzer - Xilinx				
			constraints, except time group constraints, are ignored. Clock and I/O timing constraints are specified by means of a dedicated dialog box	
02	Timing Constraints	Allows to select the User Constraints File (UCF) to include in the project	This function allows to identify all the constraints files involved in the project	
Report Options				
#	**Option/Function**	**Description**	**Benefits**	**Limitations and Recommendations**
03	Summary Only	This option displays a summary table listing the requested delay, actual delay and logic levels for the slowest path of each constraint Values: Check box disabled by default	n.a.	This option disables the *Verbose Path Report* option It is recommended to set this option to its default value (disabled) in order to enable the *Verbose Path Report* option to generate more comprehensive timing reports Safety risk: **Low**
04	Verbose Path Report	This option allows the user to control the paths that appear in the report Sub options: • Report Failing Paths Only check box • Report Fastest Paths / Verbose Hold Paths check box	The *Report Failing Paths Only* option generates a timing constraints analysis report for paths that do not meet timing constraints The *Report Fastest Paths/Verbose Hold Paths* option allows the report of the fastest paths of the design together with hold paths	Selecting this option disables the *Summary Only* option It is recommended to enabled both *Report Failing Paths Only* and *Report Fastest Paths/Verbose Hold Paths* check boxes, in order to generate more comprehensive timing reports Safety risk: **Low**
05	Report Paths By Constraint	This option allows to control the number of worst-case paths displayed in the report according to the number of selected *Path Per Constraint* Sub options: • Path Per Constraint	This option allows to generate timing reports that are easier to review	It is recommended to adapt the number of worst-case paths displayed in the timing report to the complexity of the design Safety risk: **Low**
06	Report Paths By Endpoint Per Constraint	This option allows to control the number of worst-case paths displayed in the report according to the number of selected *Endpoints Per Constraints* and *Paths Per Endpoint* Sub options: • Endpoints Per Constraint • Paths Per Endpoint		
07	Do Unconstrained Analysis and Report Unconstrained	This option instructs the tool to report on unconstrained paths Values: Check box disabled by default	Allows to perform a timing analysis for paths not explicitly covered by existing timing constraints	It is highly recommended to enable the *Do Unconstrained Analysis and Report Unconstrained Paths* check box in order to perform a timing analysis for paths not explicitly

STA Tools: Trace Timing Analyzer - Xilinx				
	Paths			covered by existing timing constraints to determine errors in the design constraint specification Safety risk: **High**
08	Generate Additional Reports	This option controls additional sections to be included in the timing report Sub options: • Data Sheet Section • Timegroup Section • A Separate Constraints Interaction Report	These additional sections allows a more comprehensive timing report, especially the Timegroup Section, which generates a table with all the sources and destination used to analyse Timegroups	Timegroups are used to group signals with the same timing requirements. This methodology reduces runtime and processor memory usage. For this reason it is recommended to enable the *Timegroup Section* to display these group of signals Safety risk: **Low**
Device Settings				
#	***Option/Function***	***Description***	***Benefits***	***Limitations and Recommendations***
09	Specify Speed Grade	Sets the speed grade of the device Values: Speed grades available in the selected device	Allows a much accurate timing analysis It can also be used to test different speed grades in order to evaluate alternative device options This value does not interfere in the speed grade selected in the design project file	It is recommended to set the speed grade of the selected device to get more accurate timing results Safety risk: **Medium**
10	Voltage	This option allows the user to choose between the worst-case voltage value (default) and setting a range of values	This option allows relaxing the timing analysis in order to meet timing constraints easily	It is highly recommended to select the proposed default voltage, which is the worst-case value, to allow the tool perform a Process-Voltage-Temperature (PVT) analysis. This process covers the range of the device's operating conditions, providing more accurate timing results Safety risk: **Medium**
11	Temperature	This option allows the user to choose between the worst-case temperature value (default) and setting a range of values	This option allows relaxing the timing analysis in order to meet timing constraints easily	It is highly recommended to select the proposed default temperature, which is the worst-case value, to allow the tool perform a Process-Voltage-Temperature (PVT) analysis. This process covers the range of the device's operating conditions, providing more accurate timing results Safety risk: **Medium**
Filter By Net and Path Tracing				
#	***Option/Function***	***Description***	***Benefits***	***Limitations and Recommendations***
12	Find Nets	This option allows to find and select nets to exclude for the timing analysis	All nets are included in a timing analysis by default. This option allows to generate timing reports	It is recommended to exclude nets that are not relevant for timing analysis

STA Tools: Trace Timing Analyzer - Xilinx				
		Values: Search box to be used with wildcards	that are easier to review	Safety risk: **Low**
13	By Path Tracing Controls	This option allows to enable the path types to be included in the timing analysis Values: List of check boxes with the type of paths identified during place and route	This option allows to reduce runtime and generate timing reports easier to review This option can be applied when performing the four types of timing analysis selectable with the *Analyze Against* option	It is recommended to analyse the impact of the paths to be removed By way of example, it is common to exclude the asynchronous reset paths to the registered elements of the design. However, these paths should be checked to avoid potential removal/recovery errors Safety risk: **Medium**
14	By Path Components	This option allows the user to specify the path components to be included in the timing analysis Values: • List of check boxes with the type of paths identified during place and route • Search box to be used with wildcards	This option allows to reduce runtime and generate timing reports easier to review This option can be applied when performing the four types of timing analysis selectable with the *Analyze Against* option	

Table 6.8. ISE Trace Timing Analyzer Tool Specific Assessment of Options and Functionalities

6.1.3.3 SmartTime Timing Analyzer (Actel)

SmartTime Timing Analyzer is the advanced STA tool integrated within Actel's Libero IDE. SmartTime supports a range of timing constraints to provide useful analysis and efficient timing-driven layout. Most constraints that can be generated by synthesis tools are automatically passed to SmartTime in an SDC file which can also be edited by the user in the SmartTime Constraints Editor.

The SmartTime Timing Analyzer has three timing analysis views:

- Maximum Delay Analysis: Checks the setup timing of the design
- Minimum Delay Analysis: Checks the hold timing
- Bottleneck analysis: Identifies congestion points and problematic nets

The specific assessment is performed using as reference the version 9.1 of Actel's Libero software.

Table 6.9 summarizes its options and functions along with their benefits, limitations and recommendations for use.

STA Tools: SmartTime Timing Analyzer - Actel				
General Settings				
#	*Option/Function*	*Description*	*Benefits*	*Limitations and Recommendations*

01	Perform maximum delay analysis based on	Allows to select the maximum delay analysis criteria
		Values:
		• Worst (default)
		• Typical
		• Best
			These two options allow to perform static timing analysis under different operating conditions	The default settings (max. delay worst-case / min. delay best-case) does not always cover the worst-case scenario for hold check
				It is highly recommended to run twice the analysis with the following parameters in order to perform a multicorner timing analysis:
				• Run #1:
				  • Max. worst-case
				  • Min. best-case
				• Run #2:
				  • Max. best-case
				  • Min. worst-case
				Safety risk: **High**

02	Perform minimum delay analysis based on	Allows to select the minimum delay analysis criteria Values: • Worst • Typical • Best (default)		
03	Include inter-clock domains in calculations for timing analysis	This option instructs the tool to perform CDC timing analysis Values: Check box disabled by default	When enabled, this option considers the inter-clock domain as functional, performing setup and hold checks between the clock domains It is possible to deactivate specific inter-clock domains (if they are considered as not functional) by means of *False Path* constraints	It is highly recommended to enable this option to identify the inter-clock paths (if any) and reveal potential violations Some of the inter-clock domain paths are valid timing paths and some are false paths. The designer should identify these paths ans apply the timing exception as needed Safety risk: **High**

STA Tools: SmartTime Timing Analyzer - Actel				
04	Enable recovery and removal checks	This option instructs the tool to check removal and recovery time on asynchronous signals Values: Check box disabled by default	This option enhances the CDC timing analysis, performing recovery and removal time analysis of asynchronous signals to detect potential metastability and data inconsistency issues Additional sets are created in each clock domain to report the corresponding paths	It is highly recommended to enable this option, especially when asynchronous reset de-assertion is implemented (best design practices suggests to synchronize the reset de-assertion) This option was disabled by default to avoid timing discrepancies with prior SmartTime versions, and Actel recommends its activation for new designs Safety risk: **High**
Analysis View Settings				
#	*Option/Function*	*Description*	*Benefits*	*Limitations and Recommendations*
05	Limit the number of paths shown in a path set to:	Limits the number of paths shown in a path set for timing analysis Values: Any non-negative integer greater than 1 (100 is the default value)	This option allows to generate timing reports that are easier to review	It is recommended to set this option to its default value (100) or modify this number depending on the design size Safety risk: **Low**
06	Filter paths by slack value	Specifies the slack range between minimum slack and maximum slack Values: Check box disabled by default	This option allows to generate timing reports that are easier to review	It is recommended to set this option to its default value (disabled) in order display all the analysed paths Safety risk: **Low**
07	Show clock network details in expanded path	Displays the clock network details as well as the data path details in the expanded path views Values: Check box enabled by default	This option allows to generate timing reports that are easier to review	It is recommended to set this option to its default value (enabled) in order to check clock networks and data paths in more detail Safety risk: **Low**
08	Limit the number of parallel paths shown in expanded path to:	Specifies the maximum number of parallel paths to be shown for each expanded path Values: Any non-negative integer (1 is the default value)	This option allows to view and analyze parallel configurations of a violating path in the expanded path window	It is recommended to set this option to its default value (1) or modify this number depending on the design complexity and size Safety risk: **Low**
Advanced / Special Situations				
#	*Option/Function*	*Description*	*Benefits*	*Limitations and Recommendations*
09	Use loop-back in bi-directional buffers	Allows to use loop-back in bi-directional buffers Values: Check box disabled by default	When enabled, this option allows to include the loop-back timing arc of the bi-directional buffer in the timing analysis	This option is useful only when data has to be written and read during the same clock cycle, which is not a common practice. In the rest of cases, enabling this option may create false timing paths that will slightly degrade the timing analysis results For this reason it is recommended to set this option to its default value (disabled)

STA Tools: SmartTime Timing Analyzer - Actel				
				Safety risk: **Medium**
10	Break paths at asynchronous pins	Allows to break paths at asynchronous pins Values: Check box enabled by default	When enabled, this option allows to define begin and end of timing analysis for asynchronous signals	It is recommended to set this option to its default value (enabled) to finish timing calculations once the asynchronous signal reaches the pad (for outputs) or start timing calculations once the asynchronous signal reaches the pad (for inputs) Safety risk: **Medium**
11	Disable non-unate arcs in clock network	Allows to disable non-unate timing arcs in the clock networks Values: Check box enabled by default	Non-unate timing arcs are present in logic functions whose output value change cannot be predicted by the direction of the change on the input value (i.e. XOR gate)	It is recommended to set this option to its default value (enabled) to disable non-unate arcs. These non-unate timing arcs should no be part of the clock network Safety risk: **Medium**

Table 6.9. Libero IDE SmartTime Timing Analyzer Tool Specific Assessment of Options and Functionalities

6.1.4 HDL Simulator Tools

6.1.4.1 ModelSim (Mentor Graphics)

ModelSim is a logic simulation tool for verification and debugging of digital designs. Most PLD vendors include this tool as default simulator in their IDE. This simulation tool can be used in both GUI and Batch modes.

The specific assessment is performed using as reference the version 10.0b of Mentor Graphics' ModelSim DE software.

Table 6.10 summarizes its options and functions along with their benefits, limitations and recommendations for use.

HDL Simulation Tools: ModelSim – Mentor Graphics				
Design Simulation				
#	*Option/Function*	*Description*	*Benefits*	*Limitations and Recommendations*
01	Design Unit(s)	Specifies the design top level unit name	n.a.	n.a.
02	Resolution	Specifies the simulator resolution Values: Range from 1 fs to 100 seconds (1 ns is used as default value)	Lower time resolutions will make the simulation run faster	The default resolution (1 ns) may not be enough for the simulation of designs featuring some specific macros (especially clock-related macros such as PLLs) which require higher resolution to run properly It is recommended to set *Resolution* to 1 ps or lower for applications using these macros Safety risk: **Medium**
03	Optimization	Sets the optimization degree to decrease simulation runtime Values: • Level (0 to 5) • Fast • No Vital • No 1164	When enabled, this option removes constructs that are not functionally essential to make the simulation run faster Best performance depends on the optimization value selected and the HDL language	Optimization may have an impact on some debugging features, especially in code coverage and Dataflow window When enabled, *Optimization* removes signals and functions that are not functionally essential to make the simulation run fast. As a result, aggressive optimization levels may have an important impact on code coverage results since the removed lines of code are not counted in the analysis Removed signals and processes that the Dataflow window (embedded debugging tool) would normally displayed are not shown anymore It is recommended to run simulations with the optimization option disabled in order to: • have a better internal

HDL Simulation Tools: ModelSim – Mentor Graphics				
				visibility for debugging purposes and • get realistic results for code coverage metrics Safety risk: **Medium**
Runtime Options				
#	**Option/Function**	**Description**	**Benefits**	**Limitations and Recommendations**
04	Default Radix	Sets the default radix for the current simulation run Values: • Symbolic (default) • Binary • Octal • Decimal • Unsigned • Hexadecimal • ASCII	Allows to set a default radix for all signals to be displayed in the Waveform window	It is recommended to set this option to its default value (*Symbolic*) and apply the specific radix to each signal in the Waveform window Safety risk: **Low**
05	Suppress Warnings	Suppresses warnings generated by some standard packages Values: • From Synopsys Packages (check box) • From IEEE Numeric Std Packages (check box)	This option is useful to generate simulation log files easier to review Selecting *From Synopsys Package* suppresses warnings generated within the accelerated Synopsys *std_arith* packages Selecting *From IEEE Numeric Std Packages* suppresses warnings generated within the accelerated *numeric_std* and *numeric_bit* packages	It is recommended to keep warnings from these packages and analyse them Safety risk: **Medium**
06	Default Run	Sets the default run length in ns Values: Any non-negative integer	This option allows to set the simulation run time	It is recommended to set the default run length using Tcl scripts (also referenced as macros or *.do files) Simulation run time can also be managed by the HDL testbench Safety risk: **Low**
07	Iteration Limit	Sets a limit on the number of deltas within the same simulation time unit to prevent infinite looping Values: Any non-integer value (5000 by default)	This option is useful to detect infinite zero-delay loops caused by: • Loop with no exit • Series of gates with zero delay where the outputs are connected back to the inputs	It is recommended to set this option to its default value (*5000*) to detect infinite zero-delay loops Safety risk: **Medium**
08	Default Force Type	Selects the default force type for the simulation Values: • Freeze • Drive • Deposit	*Force* function allows to modify the value of a signal directly in the Waveform window	It is recommended to use the *Force* function for debugging purposes only Safety risk: **Low**

HDL Simulation Tools: ModelSim – Mentor Graphics				
09	Break Severity	Selects the assertion severity that will stop simulation Values: • Fatal • Failure (default) • Error • Warning • Note/Info	n.a.	It is recommended to set this option to *Error* value Safety risk: **Low**
10	No Message Display For	Selects the assertion type to ignore Values: • Failure/Fatal • Error • Warning • Note/Info (default)	This option is useful to generate simulation log files easier to review	It is recommended to set this option to its default value (*Note/Info*) to keep displaying warnings for a more comprehensive analysis Safety risk: **Low**
11	WLF File Size Limit	Limits the WLF file by size Values: • No Size Limit (default) • Size Limit (MBytes)	This option allows to limit the WLF file in terms of size WLF files allow to view simulation datasets for further analysis	It is recommended not to limit WLF file size in order to keep all necessary information to analyse the simulation dataset. These analysis also include waveform comparison Safety risk: **Low**
12	WLF File Time Limit	Limits the WLF file by time Values: • No Time Limit (default) • Time Limit (ns)	This option allows to limit the WLF file in terms of time	It is recommended to use this option if only a well-defined time period has to be analysed Safety risk: **Low**
13	Compress WLF Data	Compresses WLF files to reduce their size Values: Check box enabled by default	Allows to reduce the size of the WLF file while keeping its information	It is recommended to set this option to its default value (enabled) to reduce the size of the WLF file Safety risk: **Low**
14	Delete WLF File on Exit	Specifies whether the WLF file should be deleted when the simulation ends Values: Check box disabled by default	This option saves disk space	It is recommended to set this option to its default value (disabled) to keep the WLF file for further analysis Safety risk: **Low**
15	Design Hierarchy	Specifies whether to save all design hierarchy in the WLF file or only regions containing logged signals Values: • Save Regions Containing Logged Signals • Save All Regions in Design	This option allows to decrease the WLF file size by saving only regions containing signals logged for simulation (*Save Regions Containing Logged Signals* option)	It is recommended to save all the design (*Save All Regions in Design* option) in order to get a WLF file for a more comprehensive analysis. However, for waveform comparison, the *Save Regions Containing Logged Signals* option may be more appropriated Safety risk: **Low**

Table 6.10. ModelSim Tool Specific Assessment of Options and Functionalities

Following design-specific files should be subject to revision control:

- Project file (including simulation settings)
- Tcl script to guide the simulation process (if any)
- Design technology libraries. Some of the libraries used for RTL and post-place and route simulations are technology-dependent, and therefore are delivered as part of the PLD vendor-specific IDE. It is recommended to map and compile directly these libraries, which are usually installed in the IDE directory tree, instead of making local copies, in order to avoid two versions of the same item with the related configuration management issues

6.1.4.2 Active HDL (Aldec)

Active HDL is a RTL and gate-level mixed-language simulator for the deployment and verification of digital designs. This simulation tool can be used in both GUI and Batch modes.

The specific assessment is performed using as reference the version 9.2 of Aldec's Active HDL software.

Table 6.11 summarizes its options and functions along with their benefits, limitations and recommendations for use.

HDL Simulation Tools: Active HDL – Aldec				
Simulation Settings				
#	Option/Function	Description	Benefits	Limitations and Recommendations
01	Disable Timing Checks	Disables Verilog and VITAL timing checks	This option allows to speed the simulation up	It is recommended to enable timing checks to perform post-place and route simulation Safety risk: **Medium**
02	Disable Timing Check Messages	Disables displaying the timing constraint error messages	This option is useful to generate simulation log files easier to review	It is recommended to enable timing check messages in order to collect and analyse them Safety risk: **Low**
03	Enable XTrace	Enables XTrace debugging tool	This option allows using the XTrace tool for debugging purposes	n.a.
04	Generate Data for Advanced Dataflow	Turns on data generation for the Advanced Dataflow viewer	Disabling this option decreases memory consumption during initialization of simulation	n.a.
05	Split Net vectors in VCD File	Splits net vectors in VCD file	This option makes the VCD file compatible with the power estimator tool from Xilinx	The VCD file can be imported and its simulation vectors be used in a second run of the place and route process to get a more accurate power estimation Safety risk: **Low**
06	Retval Memory Size	Specifies an area in memory (in MBytes) for values returned by HDL routines	Increasing the default value may be required for constructs returning large amount of data	It is recommended to set this option to its default value, which is large enough Structures requiring larger Retval memory size are not recommended Safety risk: **Low**
07	Stack Memory Size	Specifies an area in memory (in MBytes) for the stack of the simulation kernel Value: 32 MBytes	This option allows to allocate more memory in case the simulation process requires it	n.a.
08	Select Resolution	Specifies the simulator resolution Values: Combination of a number and a time unit: fs, ps,	Lower time resolutions will make the simulation run faster If no value is selected, the tool determines automatically the	It is recommended to set *Select Resolution* to 1 ps or lower depending on the resolution required by macros used in the

HDL Simulation Tools: Active HDL – Aldec				
		ns, us, ms (1 ps is used as default value)	resolution from timescale directives (if any) embedded in the RTL HDL code or applying 1 ps by default	design Safety risk: **Medium**
09	Current Resolution	Displays the current simulation resolution	The simulation resolution is displayed when the simulation is initialized	n.a.
VHDL Settings				
#	**Option/Function**	**Description**	**Benefits**	**Limitations and Recommendations**
10	Disable IEEE Warning Generation	Disables warning generation from IEEE libraries	This option is useful to generate simulation log files easier to review	It is recommended to keep warnings from these packages and analyse them Safety risk: **Medium**
11	Enable VITAL Acceleration	Enables an optimization algorithm that speeds up simulation	Designs implementing built-in acceleration for VITAL models can benefit of this acceleration	VITAL acceleration requires an internal modelling methodology which may increase both coding and reviewing efforts Safety risk: **Low**
12	Ignore VITAL Glitches	Disable the display of VITAL glitch warnings in the Console Window	This option is useful to generate simulation log files easier to review	It is highly recommended to keep glitch warnings in order to analyse and find the cause Glitch is a short pulse on the signal waveform that is usually undesired and may cause unexpected design behaviour Safety risk: **Low**
13	Disable VITAL Glitch Messages	Disable printing VITAL glitch messages to the Console window	This option is useful to generate simulation log files easier to review	
Verilog Settings				
#	**Option/Function**	**Description**	**Benefits**	**Limitations and Recommendations**
14	Delay Selection	This option allows selecting timing delay from Verilog min:typ:max expressions Values: • Min Delay • Typ Delay • Max Delay	This option allows to perform post-place and route simulations under different operating conditions	It is highly recommended to run twice the post-place and route simulation selecting minimum and maximum delays Safety risk: **Medium**
15	Set Initial Value for Regs	Sets the initial value for Verilog registers Values: • '0' • '1' • 'Z'	This option allows overriding the default initialization (before reset or power up) for the register which is defined as an unknown value ('X') This option has no effect on Verilog memories	It is recommended to keep the default initialization value ('X'), especially if Waveform comparison with other HDL simulator output is foreseen Safety risk: **Low**
16	Verilog Optimization	Enables SLP accelerated simulation	SLP is a simulation acceleration technology for Verilog that significantly reduces simulation runtime	It is recommended to disable SLP optimization, since it may block access to design objects during simulation and prevent recording signal history, or displaying object values

HDL Simulation Tools: Active HDL – Aldec				
				Safety risk: **Low**
17	Disable Pulse Error and Warning Messages	Disables the display of warning messages that are generated if path pulse errors occur	This option is useful to generate simulation log files easier to review	It is highly recommended to keep pulse error and warning messages in order to analyse them and find the cause Safety risk: **Medium**
18	Rejection Limit	Specifies the module path pulse rejection limit as a percentage of path delay	These arguments are only supported for SLP accelerated netlists	It is recommended to disable SLP optimization, since it may block access to design objects during simulation and prevent recording signal history, or displaying object values Safety risk: **Medium**
19	Error Limit	Specifies the module path pulse error limit as a percentage of path delay		
20	Pulse Error Filtering	Allows to select the propagation style of pulse errors Values: • On Event • On Detect • None	When *On Event* propagation style is selected, the leading edge of a pulse is scheduled to happen after a timing delay When the *On Detect* propagation style is selected, a pulse error is visible on the output immediately after is detected The *None* value disables pulse error filtering	It is recommended to disable *Pulse Error Filtering* (*None*) in order to keep all pulse errors for further analysis Safety risk: **Medium**
Generic/Parameters				
#	***Option/Function***	***Description***	***Benefits***	***Limitations and Recommendations***
21	Name/Value	Displays the list of VHDL generics / Verilog parameters and new values to assign	These options allow to extract and override the default values and set new values for the generics / parameters without modifying the source code This option may be useful to try different values to simulate different configurations	It is highly recommended not to modify generics / parameters using this function to avoid mismatches between the RTL HDL code and the final implementation Safety risk: **Medium**
22	Override Default Value	Allows to override values of generics that received explicit values in generic maps		
Trace/Debug				
#	***Option/Function***	***Description***	***Benefits***	***Limitations and Recommendations***
23	Select Signals	Allows to select signals to be dumped for post simulation analysis purposes	This option allows to decrease the post simulation database file size by saving only selected signals	It is recommended to select I/O ports and representative internal signals for a more comprehensive analysis Safety risk: **Low**
24	Save Full Signal History	Enables the Full Signal History Mode	Disabling these two options save disk space	It is recommended to enable this mode to keep the post simulation database file for further analysis Safety risk: **Low**
25	Preserve File with Signal History for Post Simulation Debug	Saves the post simulation database file		
26	File with	Specifies the name and location	n.a.	n.a.

HDL Simulation Tools: Active HDL – Aldec				
	Simulation Results	of the post simulation database file		
27	Enable Show Event Source	Enables the *Show Event Source* debug function	This option allows using the *Show Event Source* for debugging purposes When this option is enabled, it is possible to find a source of events on objects displayed in the Waveform viewer	n.a.

Table 6.11. Active HDL Tool Specific Assessment of Options and Functionalities

Following design-specific files should be subject to revision control:

- Project file (including simulation settings)
- Tcl script to guide the simulation process (if any)
- Design technology libraries. Some of the libraries used for RTL and post-place and route simulations are technology-dependent, and therefore are delivered as part of the PLD vendor-specific IDE. It is recommended to map and compile directly these libraries, which are usually installed in the IDE directory tree, instead of making local copies, in order to avoid two versions of the same item with the related configuration management issues

6.2 Best Practices, Recommendations and Guidelines

The results of the assessment of configuration options and embedded functionalities of selected CAE tools performed in § 6.1 have been used as a the basis for the elaboration of the following best practices, recommendations and guidelines, which aim to maximize the CAE tools effectiveness while reducing the safety risks arising from their improper use.

These best practices, recommendations and guidelines are tagged for traceability purposes as follows:

[*ID-nn*]

where:

- *ID* indicates the scope of each recommendation:
  - ✔ *SYN*: Applicable to both synthesis tools and physical synthesis functions embedded in Integrated Development Environments (IDE)
  - ✔ *PAR*: Applicable to place and route tools and/or Integrated Development Environments (IDE)
  - ✔ *STA*: Applicable to Static Timing Analysis tools
  - ✔ *SIM*: Applicable to HDL simulators
- *nn* indicates the recommendation number, a value from 01 to 99

6.2.1 Synthesis Tools

The following best practices, recommendations and guidelines apply both to synthesis tools and to physical synthesis functions embedded in Integrated Development Environments (IDE).

[*SYN-01*] The reproducibility of the implemented item is a key issue for design tools in terms of configuration management, as stated in ED-80/DO-254 §7, *"The configuration management process is intended to provide the ability to consistently replicate the configuration item, regenerate the information if necessary and modify the configuration item in a controlled fashion if modification is necessary"*. Some synthesis tools include in their algorithms pseudo random procedures to improve timing closure. Therefore, in order to ensure the reproducibility of the synthesis tool output, it is essential to keep control on these algorithms by managing the following items:

- The seed used to define the starting point of the algorithm should be set to a fixed value (Table 6.1 item #29)
- The synthesis tool software version, including its service pack, and associated design technology libraries should be identified as part of the project baseline. Different software versions may generate different results with the same design and settings ([EXT-05])
- The configuration of the hardware platform which runs the synthesis tool, including the operative system, should also be included within the project baseline, since the pseudo random seed generation algorithm also depends on the floating point unit of the CPU. Consequently, any difference in the architecture of the processor will lead to a different result ([EXT-05])

[*SYN-02*] The following synthesis optimization options may lead to unexpected results, which may have an impact on the overall implementation and, therefore, they should be handled with special attention:

- Resource sharing (Table 6.1 item #04; Table 6.2 item #27; Table 6.3 item #13)
- Register duplication (Table 6.2 item #33; Table 6.4 items #23, #39; Table 6.5 items #15, #22)
- Equivalent register removal (Table 6.1 item #17; Table 6.2 item #34; Table 6.5 item #18)
- Retiming (Table 6.2 items #35, #36, #37; Table 6.3 item #15; Table 6.4 items #18, #20; Table 6.5 item #17; Table 6.6 item #24)
- Pipelining (Table 6.3 item #14; Table 6.4 item #22)
- Asynchronous to synchronous transformations of sequential logic (Table 6.2 item #29)

In addition, these optimizations may not be compatible with formal verification tools such as LEC or robustness techniques such as TMR.

[*SYN-03*] Finite State Machines (FSM) and synchronizers to handle Clock Domain Crossing (CDC) are particularly sensitive to synthesis optimization techniques; therefore, these techniques may represent a high risk in terms of safety and should be avoided as far as possible when these structures are used in the design:

- Finite State Machines (FSM) are widely used as control mechanisms. The optimization performed by synthesis tools may lead to unreliable implementations which can lock the entire design if an undefined/illegal state is reached (typically due to SEU or asynchronism issues):
  - ✔ *Safe* directive for FSM may not be effective and may create a wrong sense of safety. In fact, this directive is interpreted in different ways depending on the selected synthesis tool (i.e. definition of safe recovery state). For this reason it is highly recommended to disable it. True fault tolerant FSMs should be implemented directly in the RTL HDL code, adding robustness mechanisms, for instance encoding states with Hamming code and the corresponding recovery logic, extra parity bits, or implementing TMR on the FFs storing the state (Table 6.1 item #15; Table 6.2 item #18; Table 6.3 item #11)
  - ✔ It is also highly recommended to disable all the options aimed at identifying, extracting and optimizing FSMs in order to keep control of the FSM behaviour. The fact of encoding the states of each FSM directly in the RTL HDL code, including the recovery state defined by the designer, may lead to more reliable and fault tolerant FSMs (Table 6.1 items #13, #14; Table 6.2 items #17, #19, #20; Table 6.3 items #11, #12)
- Clock Domain Crossing (CDC) is a critical issue which may be at the origin of metastability, race/skew errors and data loss and inconsistency. Even if STA tools provide means to identify and analyze CDC paths, it is important to define well the synchronization techniques in the RTL HDL code and to use appropriate attributes and directives in order to avoid unexpected optimizations from synthesis tools. Retiming (also known as register balancing) and register duplication may have a critical impact on synchronizers; consequently, it is highly recommended to disable them (Table 6.1 item #25, Table 6.2 item #11)

[*SYN-04*] Optimizations inferring RAM from logic should be avoided as far as possible. Implementing logic into memory blocks may have an impact on the design reliability since memory blocks are more sensitive to SEU (Table 6.1 items #05, #07; Table 6.2 item #20; Table 6.4 items #25, #37; Table 6.5 item #33).

[*SYN-05*] In general, it is better to instantiate logic functions than to allow the synthesis tool to infer them in order to keep control on the implementation. This approach also allows a better integration of formal verification tools such as LEC (Table 6.1 items #05, #07; Table 6.3 items #02, #26, #32; Table 6.4 item #38).

[*SYN-06*] RTL viewers should be used to check the synthesized results, especially for sensitive structures such as clock tree distribution, FSMs, CDC synchronizers, and I/O cells (Table 6.2 items #08, #38; Table 6.5 item #27).

[*SYN-07*] Reduction of power consumption can be better achieved by applying power-aware design techniques directly in the HDL RTL code, rather than by using aggressive synthesis optimization techniques (Table 6.1 item #09; Table 6.2 item #03).

[*SYN-08*] Timing-Driven synthesis is highly recommended, together with a comprehensive set of user timing constraints (Table 6.1 item #11; Table 6.2 item #04; Table 6.3 item #18).

[*SYN-09*] It is recommended to set synthesis effort to high levels, provided the tool features the option, in order to allow the tool to perform all the process steps and to get better results, even if processing time is increased (Table 6.1 item #28).

[*SYN-10*] The compatibility between the files exchanged among CAE tools from different vendors should be checked. This is the case, for instance, of constraint files generated by the synthesis tool for an specific PLD technology (Table 6.3 item #20).

[*SYN-11*] Incremental design flows are not recommended unless the methodology and the supporting processes are properly mastered to ensure the reproducibility of the implementation process. Incremental design flows may involve an additional effort in order to assess that design modifications do not disturb other parts of the design. This methodology increases the complexity of the project baseline, traceability and configuration management processes, since several iterations with different settings are required to generate the final implementation (Table 6.3 items #03, #34).

[*SYN-12*] The use on any option allowing to override the design configuration or its generic constant values without modifying the RTL HDL code is highly discouraged (Table 6.2 items #15, #16; Table 6.3 items #29, #33, #35).

[*SYN-13*] Directives aimed to prevent portions of HDL RTL code from being synthesized (i.e. *synthesis_on*/*synthesis_off* or *translate_on*/*translate_off*) should be avoided whenever it is possible, since their use may generate a different implementation with an unexpected behaviour. In addition, these directives may not be supported by some CAE tools. Therefore, it is highly recommended to set the tool options in order to ignore these directives, to analyse the HDL RTL code concerned by these directives, and to make any necessary changes to remove them (Table 6.1 item #06; Table 6.3 item #27).

[*SYN-14*] Report options should be configured with the purpose of generating comprehensive synthesis reports for further analysis (Table 6.1 item #26; Table 6.3 items #22, #23).

6.2.2 Place and Route Tools / Integrated Development Environments (IDE)

The following best practices, recommendations and guidelines apply to place and route tools and/or Integrated Development Environments (IDE).

[*PAR-01*] The reproducibility of the implemented item is a key issue for design tools in terms of configuration management, as stated in ED-80/DO-254 §7, *"The configuration management process is intended to provide the ability to consistently replicate the configuration item, regenerate the information if necessary and modify the configuration item in a controlled fashion if modification is necessary"*. Place and route tools base their algorithms in pseudo random processes. Therefore, in order to ensure the reproducibility of the place and route tool output, it is essential to keep control on these algorithms by managing the following items:

- The seed used to define the starting point of the algorithm should be set to a fixed value (Table 6.4 item #31; Table 6.5 items #12, #13, #40; Table 6.6 item #20)
- The place and route tool software version, including its service pack, and associated design technology libraries should be identified as part of the project baseline. Different software versions may generate different results with the same design and settings ([EXT-05]; [EXT-06])
- The configuration of the hardware platform which runs the place and route tool, including the operative system, should also be included within the project baseline, since the pseudo random seed generation algorithm also depends on the floating point unit of the CPU. Consequently, any difference in the architecture of the processor will lead to a different result ([EXT-05]; [EXT-06])

[*PAR-02*] Technology viewers should be used to check the place and route results, especially for sensitive structures such as clock tree distribution, FSMs, CDC synchronizers, and I/O cells (Table 6.5 item #27; Table 6.6 item #20).

[*PAR-03*] Reduction of power consumption can be better achieved by applying power-aware design techniques directly in the HDL RTL code, rather than by using aggressive place and route optimization techniques (Table 6.4 items #28, #61; Table 6.5 items #34, #48; Table 6.6 item #12).

[*PAR-04*] Timing-Driven place and route is highly recommended together with a comprehensive set of user timing constraints. Multi-corner analysis for placement should be also enabled to take into account different operating conditions depending on Process-Voltage-Temperature (PVT) parameters (Table 6.4 items #26, #27, #54; Table 6.5 item #09; Table 6.6 item #11).

[*PAR-05*] The use on any option allowing to ignore unmatched physical and/or timing constraints defined by the user is highly discouraged. These options may prevent the tool from detecting incorrect constraints or logic removed by the synthesis tool due to unexpected optimizations (Table 6.5 items #07, #08).

[*PAR-06*] It is recommended to pack I/O registers into I/O cells for inputs and outputs in order to benefit from the advantages of I/O cells: better timing performance and protection against SEU (Table 6.4 item #55; Table 6.5 item #27; Table 6.6 item #08).

[*PAR-07*] I/O voltage overdrive and using bonded I/Os are highly discouraged, since these options may lead to electrical problems, such as higher leakage current, causing the design not to work as intended (Table 6.4 item #13; Table 6.5 item #43).

[*PAR-08*] Embedded CRC error detection mechanisms should be enabled, provided the tool features them, in order to detect errors in the configuration memory cells for SRAM based PLDs. However, internal scrubbing is not recommended, and error recovery should be managed at an upper level (Table 6.4 items #14, #15, #16, #17).

[*PAR-09*] It is recommended to set place and route effort to high levels in order to allow the tool to perform all the process steps and to get better results, even if processing time is increased. Multiple place and route passes may also be enabled; however, the number of passes should be set to the default one in order to reach a balance between process runtime and layout quality (Table 6.4 items #29, #59, #63; Table 6.5 items #10, #11, #38, #39; Table 6.6 items #18, #19, #23).

[*PAR-10*] Incremental design flows are not recommended unless the methodology and the supporting processes are properly mastered to ensure the reproducibility of the implementation process. Incremental design flows may involve an additional effort in order to assess that design modifications do not disturb other parts of the design. This methodology increases the complexity of the project baseline, traceability and configuration management processes since several iterations with different settings are required to generate the final implementation (Table 6.6 items #02, #03, #14, #15, #17).

[*PAR-11*] Report options should be configured with the purpose of generating comprehensive place and route reports for further analysis (Table 6.4 items #55, #62; Table 6.5 items #04, #21, #25, #27, #44, #45, #47; Table 6.6 items #10, #18).

6.2.3 Static Timing Analysis Tools

The following best practices, recommendations and guidelines apply to Static Timing Analysis tools.

[*STA-01*] It is recommended to check if clock constraints have been applied correctly. In some cases, the design tools may not find the specified clock nets and therefore apply the default clock frequency (Table 6.7 item #04; Table 6.8 items #01, #02, #07).

[*STA-02*] Inter-clock domain analysis should be enabled, provided the tool features it, in order to identify CDC paths and to reveal potential violations (Table 6.7 item #11; Table 6.9 items #03, #04).

[*STA-03*] Multicorner timing analysis should be enabled, provided the tool features it, in order to report the best and worst-case scenarios (most STA tools evolved from "relative minimum" methods to multicorner analysis). Multicorner analysis can also be performed by running several timing analysis, combining different values for maximum and minimum delay (Table 6.7 items #02, #06; Table 6.8 items #10, #11; Table 6.9 items #01, #02).

[*STA-04*] Report options should be configured to generate comprehensive timing reports for further analysis, including failing paths, fastest paths, and worst-case paths among other parameters (Table 6.7 items #08, #11; Table 6.8 items #04, #05, #06, #08, #12, #13, #14; Table 6.9 items #05, #06, #07, #08).

6.2.4 HDL Simulation Tools

The following best practices, recommendations and guidelines apply to HDL simulation tools.

[*SIM-01*] Simulation resolution should be set taking into account the macro functions used by the design in order to find the optimum value to guarantee simulation functionality and performance (Table 6.10 item #02; Table 6.11 item #08).

[*SIM-02*] Optimization settings intended to decrease simulation runtime may have a significant impact in elemental analysis -code coverage does not count the removed functions- and may also limit the access to internal signals, making the design more difficult to debug. Therefore, optimization should be handled with special attention (Table 6.10 item #03; Table 6.11 items #11, #16).

[*SIM-03*] Report options should be configured to generate comprehensive simulation log files for further analysis. Note/Info level messages should also be kept in the simulation log file, since they may contain relevant data concerning the design (Table 6.10 items #05, #10; Table 6.11 items #02, #10, #13, #17, #20).

[*SIM-04*] Simulation datasets should be stored for further analysis and waveform comparison (Table 6.10 items #11, #12, #13, #14, #15; Table 6.11 items #23, #24, #25).

6.3 Conclusions of Task 3

A specific assessment in terms of configuration options and embedded functionalities of the selected commercial CAE tools has been performed. This specific assessment has revealed a significant number of configuration options and embedded functionalities which may have a high impact in the design implementation.

As a result of this assessment, a set of best practices, recommendations and guidelines are described in § 6.2 , whose fulfilment will contribute to:

- Reduce the safety risks arising from undesired optimizations and/or improper use of configuration options and functionalities
- Ensure the reproducibility of the CAE tool outputs by identifying the key items to be placed under configuration management, which includes the hardware and software environments used to run the CAE tools, together with key implementation parameters such as place and route seeds
- Increase the design reliability through a controlled use of tool settings and functions, especially when implementing particularly sensitive structures such as FSMs, CDC synchronizers and CRC error detection mechanisms
- Ease the verification process by diminishing errors emerging from unexpected optimizations, by generating more comprehensive reports for further analysis, and also through a proper configuration of the verification tools assessed
- Increase the awareness in the use of design and verification tools

In addition, the consultation of aircraft and equipment manufacturers revealed that the use of default settings of CAE tools is a common practice. However, the specific assessment of these tools showed that default configuration options and functions proposed by CAE tools may not be the most appropriate for a given design, particularly for safety critical implementations.

For each selected tool, the Hardware Design Standards should describe a set of recommended settings, which may differ from the default ones. Deviations from these recommended settings should be assessed in order to determine whether or not they are applicable to a specific design.

Some of these best practices, recommendations and guidelines may be promoted, with the agreement of EASA, in order to amend the EASA Certification Memorandum “Development Assurance of Airborne Electronic Hardware” CM-SWCEH-001 [EXT-02].

7. Conclusions of the SHARDELD Study

The first task of the SHARDELD study identified the most relevant types of tools used for the development of programmable AEH. For each type of tool, a short description was provided together with methods for tool assessment and qualification and configuration management.

Alternative and/or complementary tools suitable for the development of programmable AEH were also assessed, in order to provide an overview of these types of tools together with their advantages, disadvantages and limitations.

The study also determined that the following types of tools need to be subject to a specific assessment on the basis of complexity and/or configuration criteria:

- Design tools intended for detailed design process, more specifically synthesis, and place and route tools, since they have a level of complexity and configuration options which could lead to potential safety impact when designing programmable AEH. In addition these tools carry out mandatory processes for the development of programmable AEH
- Verification tools, such as HDL simulators and STA tools integrated within design environments (IDE), due to their significant relevance for the verification activities

The second task of the SHARDELD study identified the tools available on the market, within the types of tools selected in the previous task, which were relevant for the specific assessment. It also determined their limitations, benefits, and the different methods followed by the programmable AEH designer community to fulfil ED-80/DO-254 objectives in terms of tool assessment and qualification. In order to achieve this, a consultation was addressed to relevant aircraft and equipment manufacturers covering the following topics:

- Identification of commercial CAE tools within the types of tools selected for specific assessment (synthesis, place and route and STA) and HDL simulation tools
- Tool Assessment and Qualification approach and methods for the selected tools
- Identification of alternative and/or complementary types of tools used in the programmable AEH design flow
- Identification of commercial custom micro-coded components used for the development of programmable AEH (CPLDs, FPGAs and structured ASICs)

Aside of helping to identify and to assess commercial CAE tools, the results of the consultation also revealed the following facts:

- The use of default settings of CAE tools is a common practice
- The tool assessment and qualification approaches of the companies consulted endorse the methods identified in the first task of the SHARDELD study, and reveal the independent tool output assessment as the most common approach, followed by relevant history of the tool
- Alternative and/or complementary tools still have a low degree of integration and acceptance within the programmable AEH designer community, mainly for the lack of maturity of these tools and also for the difficulty for tool assessment and qualification

Finally, the third task of the SHARDELD study performed a specific assessment in terms of configuration options and embedded functionalities of the commercial CAE tools selected in the previous task. This specific assessment found a significant number of configuration options and embedded functionalities

which may have a high impact in the design implementation.

As a result of this assessment, a set of best practices, recommendations and guidelines were elaborated, whose fulfilment will certainly contribute to reduce the safety risks arising from undesired optimizations and/or improper use of configuration options and functionalities, to ensure the reproducibility of the CAE tool outputs, to increase the design reliability, to ease the verification process, and to increase the awareness in the use of design and verification tools.

Among these best practices, recommendations and guidelines, the following may be promoted, with the agreement of EASA, in order to amend the EASA Certification Memorandum "Development Assurance of Airborne Electronic Hardware" CM-SWCEH-001 [EXT-02]:

- CAE tools evolve faster than the technology they are intended for; therefore, it is recommended to adopt a proactive approach regarding CAE tools release management, checking for known issues and workarounds provided by the CAE tool vendor for each release and updating the project baseline when appropriate
- Designers should have a sound knowledge of the CAE tools within the programmable AEH design flow; they should identify the data exchanged among the different tools, and they should know the attributes and directives available in order to control better tool settings and their impact on the implementation
- Default configuration options and functions suggested by CAE tools may not be the most appropriate for a given design, particularly for safety critical implementations. For each selected tool, the Hardware Design Standards should describe a set of recommended settings, which may differ from the default ones. Deviations from these recommended settings should be assessed in order to determine whether or not they are applicable to a specific design
- CAE tools feature optimization settings which may contribute to improve the overall performance in terms of timing, logic resources usage and power consumption. It is important to reduce the freedom of interpretation, the modifications of the implementation and the lack of visibility which may be introduced by these optimization settings, by understanding how the optimizations are performed and also through the direct implementation of some of these optimizations in the RTL HDL code whenever is possible
- Finite State Machines (FSM) and synchronizers to handle Clock Domain Crossing (CDC) are particularly sensitive to optimization techniques such as FSM-specific optimizations, retiming and register duplication; therefore, these techniques may represent a high risk in terms of safety and should be avoided as far as possible
- Incremental design flows are not recommended unless the methodology and the supporting processes are properly mastered to ensure the reproducibility of the implementation process
- Due to the pseudo random nature of synthesis and place and route algorithms, the reproducibility of their outputs can only be ensured by a consistent configuration management process, including the following items:
  - ✔ The seed used to define the starting point of the algorithm
  - ✔ The design tool software version and service pack, and the associated design technology libraries
  - ✔ The hardware and software environment running the synthesis and place and route tools
- Design technology libraries may be shared among some of the CAE tools within the programmable AEH design flow. For instance, HDL simulators require technology libraries

provided by place and route tools in order to compile and simulate PLD vendor-specific macro functions. Therefore, it is important to properly identify and manage the interdependencies of CAE tools due to these libraries

- The use of scripts is recommended to enhance the configuration management process. Tcl is the recommended scripting language because it is supported by most of CAE tools
- Reports generated by CAE tools should be analysed exhaustively in order to identify potential issues and justify the displayed warnings and notes

Conclusively, the application of these best practices, recommendations and guidelines will lead to a better understanding, control and reliable use of the CAE tools assessed for the development of programmable AEH, in order to achieve a higher degree of equivalence between the implemented hardware item and its RTL HDL description.

However, it is important to note that all the CAE tools assessed in this study rely on models (technology libraries), and these models are not perfect. In addition, the custom micro-coded component implementing the hardware item may be subject to foundry and device manufacturing issues not detectable by the verification tools object of this assessment. Consequently, physical testing will continue to be necessary for the purpose of certification credit. Increased awareness in the use of CAE tools may contribute to alleviate the level of verification coverage of the design requirements achieved by physical tests on the custom micro-coded component, and may also be used to justify the design requirements not covered by physical test. Nevertheless, precise criteria for such alleviation were not discussed in the framework of the SHARDELD study. Therefore, an additional research project could be carried out in the near future in order to define these precise criteria.

8. Annex A: Aircraft and Equipment Manufacturers Consultation

The core activity of the second task of the SHARDELD study is the consultation of relevant aircraft and equipment manufacturers mainly based in Europe and North America , which helped to identify the main PLD vendors and device types -stated in § 5.1 -, together with commercial CAE tools used for the design and verification of programmable AEH compliant with ED-80/DO-254 DAL A, B and C.

The consultation also aims to compile information about tool assessment and qualification criteria for the selected CAE tools. This information is compared with the tool assessment and qualification methods described in the first task of the SHARDELD study, showing how the methods are combined and adding, in some cases, new strategies.

Finally, the use of alternative and/or complementary methods in the development of programmable AEH is also assessed.

8.1 Consultation form

This section records the consultation form, as it was delivered to aircraft and equipment manufacturers mainly based in Europe and North America.

The consultation is organized into four sections, each one including an specific questionnaire:

- Section 1: Identification of commercial CAE tools within the types of tools selected for specific assessment (synthesis, place and route and STA) and HDL simulation tools
- Section 2: Tool Assessment and Qualification approach and methods for the selected tools
- Section 3: Identification of alternative and/or complementary types of tools used in the programmable AEH design flow
- Section 4: Identification of commercial custom micro-coded components used for the development of programmable AEH (CPLDs, FPGAs and structured ASICs)

8.1.1 Introduction

The development of programmable Airborne Electronic Hardware (AEH) using these components for safety critical applications requires a design flow which involves commercial Computer-Assisted Engineering (CAE) tools with a high degree of complexity.

The way these tools are used for both design and verification activities may have a impact in terms of safety.

The aim of the SHARDELD study is to identify the most relevant commercial CAE tools which are used for the development of programmable AEH, with the purpose of recording their advantages and limitations, safety benefits and risks, and elaborating a comprehensive list of best practices, recommendations and guidelines in order to reduce the safety risks while keeping their effectiveness.

This consultation aims to identify commercial CAE tools used for the development of programmable AEH compliant with ED-80/DO-254 DAL A, B and C.

IOxOS Technologies and the EASA undertake to maintain the anonymity of the aircraft and equipment manufacturers involved in this consultation.

8.1.2 Section 1: Commercial tools for the development of programmable AEH

The following questions aim to identify the CAE tools available on the market for the development of the programmable AEH. Furthermore, the consultation intends to determine whether the additional functions available on these tools are used and how they are used.

Within the framework of this consultation, the type of tools to be identified are the following:

- Synthesis tools
- Place and route tools
- Static Timing Analysis (STA) tools

- HDL simulation tools

Questionnaire #1

Synthesis Tools:

1. Which commercial synthesis tools does your design team normally use to develop programmable AEH?
2. Does your design team use a predefined set of functionalities? If yes, are they identified in your own Hardware Design Standards?
3. Does your design team use a predefined set of configuration settings? If yes, are they default configuration settings provided by the synthesis tool manufacturer or customised by your design team and described in your own Hardware Design Standards?
4. Does your design team use the following additional functions provided by the synthesis tool? Please indicate whether the use of these additional functions is restricted depending on the DAL of the application:
    - Graphical editors for Finite State Machine (FSM) inspection and edition
    - Triple Modular Redundancy (TMR)
    - Automated FSM extraction and optimization
    - FSM encoding
    - Fan-out limit
    - Incremental synthesis
    - Optimization methods (resource sharing, retiming, pipelining and/or register duplication)
    - Other additional functions (please indicate)
5. Did your design team experience major technical issues when using commercial synthesis tools? If yes, please indicate, if possible, which kind of design errors were detected as well as their potential severity.

Place and Route Tools:

6. Which commercial place and route tools does your design team normally use to develop programmable AEH?
7. Does your design team use a predefined set of functionalities? If yes, are they identified in your own Hardware Design Standards?
8. Does your design team use a predefined set of configuration settings? If yes, are they default configuration settings provided by the place and route tool manufacturer or customised by your design team and described in your own Hardware Design Standards?

9. Does your design team use the following additional functions provided by the place and route tool? Please indicate whether the use of these additional functions is restricted depending on the DAL of the application:
    - Graphical constraint editor
    - Physical synthesis (as an enhancement of logic synthesis)
    - Incremental compilation
    - Floorplanning
    - Optimization for speed, area or balanced (please indicate)
    - Timing-driven place and route
    - Other additional functions (please indicate)
10. The place and route tools are usually provided by the PLD vendor as part of an Integrated Development Environment (IDE). Please indicate which of the following tools or functions provided by the IDE are being used to develop programmable AEH. In the case of tools or functions such as the power consumption analyser, the embedded logic analysers, or other functions, please indicate whether they are used informally (e.g. to facilitate technical decisions) or with the purpose of getting certification credit:
    - Graphical HDL entry
    - PLD vendor-specific logical synthesis
    - Power consumption analyser
    - Static Timing Analysis tool
    - PLD vendor-specific or integrated third party HDL simulator
    - Embedded logic analyser
    - Other functions (please indicate)
11. Did your design team experience major technical issues when using commercial place & route / IDE tools? If yes, please indicate, if possible, which kind of design errors were detected as well as their potential severity.

Static Timing Analysis (STA) Tools:

12. Does your design team use STA tools from third party vendors (non PLD vendors)? If yes, specify which tool.
13. Does your design team use a predefined set of functionalities? If yes, are they identified in your own Hardware Design Standards?
14. Does your design team use a predefined set of configuration settings? If yes, are they default configuration settings provided by the STA tool manufacturer or customised by your design team and described in your own Hardware Design Standards?
15. When using an STA tool from a third party vendor, are the results compared with the STA function provided by the PLD vendor? If yes, did you notice any major differences?

16. Does your design team use the STA tool to analyse Clock Domain Crossing (CDC) paths? Are the results of such analysis comprehensive or did the design team identify additional analyses (manual, simulation...) to be performed (e.g. to verify synchronization)? If not, does your design team use a third party CDC analysis tool?
17. Did your design team experience major technical issues when using commercial STA tools? If yes, please indicate, if possible, which kind of verification errors were detected as well as their potential severity.

HDL Simulation Tools:

18. Which commercial HDL simulation tools does your design team use to develop programmable AEH?
19. Does your design team use a predefined set of configuration settings? If yes, are they default configuration settings provided by the HDL simulation tool manufacturer or customised by your design team and described in your own Hardware Design Standards?
20. Does your design team use code coverage for elemental analysis purposes as defined in ED-80/DO-254 Appendix B §3.3?
21. Is the code coverage feature used to perform verification activities other than assessing the completion of verification testing? If yes, please indicate which activities.
22. Did your design team experience major technical issues when using commercial HDL simulation tools? If yes, please indicate, if possible, which kind of verification errors were detected as well as their potential severity.

8.1.3 Section 2: Tool assessment and qualification approach and criteria

The following questions aim to determine the different approaches and criteria considered for the tool assessment and qualification (as established in ED-80/DO-254 §11.4) of the development tools identified in Section 1.

Questionnaire #2

1. Which of the following three approaches is the most commonly used when carrying out tool assessment and qualification of tools for the development of programmable AEH?
    - Independent tool output assessment
    - Relevant history of the tool
    - Tool qualification

Synthesis Tools:

2. When performing the independent assessment of the synthesis tool output (ED-80/DO-254 §11.4.1 item 3), which of the following methods are normally used by your design team?
    - Post-synthesis simulation of the back-annotated HDL model
    - Visual inspection of the synthesis output (e.g. using the integrated graphical viewer)
    - Use of Logic Equivalence Checking (LEC)
    - Other methods (please indicate)
3. Is relevant service experience (ED-80/DO-254 §11.4.1 item 5) data provided by the synthesis tool vendor upon request?
4. Does your design team have any experience performing basic tool qualification (ED-80/DO-254 §11.4.1 item 7) of a synthesis tool? If yes, did the synthesis tool vendor provide comprehensive data in order to carry out the qualification?

Place and Route Tools:

5. When performing the independent assessment of the place and route tool output, which of the following methods are normally used by your design team?
    - Post-place and route simulation of the back-annotated HDL model
    - Visual inspection of the place and route output (e.g. using the integrated graphical viewer)
    - Physical tests on the programmed device

- Use of Logic Equivalence Checking (LEC)
- Other methods (please indicate)

6. Is relevant service experience data provided by the place and route tool vendor upon request?
7. Does your design team have any experience performing basic tool qualification of a place and route tool? If yes, did the place and route tool vendor provide comprehensive data in order to carry out the qualification?

Static Timing Analysis (STA) Tools:

8. When performing the independent assessment of the STA tool output, which of the following methods are normally used by your design team?
   - Post-place and route simulation of the back-annotated HDL model
   - Visual inspection of the STA output report
   - Physical tests on the programmed device
   - Other methods (please indicate)
9. Is relevant service experience data provided by the STA tool vendor upon request?
10. Does your design team have any experience performing basic tool qualification of an STA tool? If yes, did the STA tool vendor provide comprehensive data in order to carry out the qualification?

HDL Simulation Tools:

11. When performing the independent assessment of the HDL simulation tool output, which of the following methods are normally used by your design team?
    - Manual review
    - Two HDL simulators running in parallel under the same verification environment
    - Physical tests on the programmed device
    - Other methods (please indicate)
12. Is relevant service experience data provided by the HDL simulation tool vendor upon request?
13. Does your design team have any experience performing a basic tool qualification of an HDL simulation tool? If yes, did the HDL simulation tool vendor provide comprehensive data in order to carry out the qualification?

8.1.4 Section 3: Alternative / complementary types of tools used in the programmable AEH design flow

The following questions aim to identify alternative and/or complementary methods and types of tools used in the development of programmable AEH. Most of these verification methods are identified in ED-80/DO-254 Appendix B §3.3 as advanced verification methods.

Questionnaire #3

1. Does your design team use one or more of the following alternative and/or complementary verification methods for the development of programmable AEH? If yes, please indicate, if possible, the main advantages and limitations of the selected method.
    - HDL Rule Checkers
    - Clock Domain Crossing (CDC) analysers
    - Logic Equivalence Checkers (LEC)
    - Assertion-Based Verification (ABV) methodologies:
        - Dynamic ABV: HDL simulation combined with advanced HDL languages and/or specific assertion languages (please indicate the advanced HDL and/or assertion languages)
        - Formal Model Checkers
    - Other methodologies (please indicate which methodologies and tools)
2. Are these alternative and/or complementary verification methods used informally (e.g. to facilitate technical decisions) or with the purpose of getting certification credit?
3. When using any alternative and/or complementary verification method informally (i.e. out of the ED-80/DO-254 design life cycle), please indicate the main reason for not claiming certification credit:
    - Lack of tool maturity
    - Tool reports with multiple false negatives
    - High complexity of the tool
    - Tool assessment and qualification difficult to achieve
    - Other reasons (please indicate)
4. Does your design team have any experience carrying out tool assessment and qualification of one or more of the alternative and/or complementary verification tools identified in question 1? If yes, please indicate the tools and the assessment and qualification approach.

8.1.5 Section 4: Commercial custom micro-coded components used as programmable AEH

The following questions aim to identify the custom micro-coded components available on the market for their use as programmable AEH in applications requiring compliance with ED-80/DO-254 DAL A, B and C.

Within the framework of this consultation, programmable AEH applies to the following custom micro-coded components or PLD (Programmable Logic Devices): CPLDs, FPGAs and structured ASICs.

Questionnaire #4

1. Does your design team develop programmable AEH for ED-80/DO-254 DAL A, B or C compliant applications?
2. Are DAL criteria considered when selecting the PLD device and its technology (SRAM-based / flash-based)? If yes, please indicate which considerations are taken into account.

EUROPEAN AVIATION SAFETY AGENCY
AGENCE EUROPÉENNE DE LA SÉCURITÉ AÉRIENNE
EUROPÄISCHE AGENTUR FÜR FLUGSICHERHEIT

Postal address
Postfach 101253
50452 Cologne
Germany

Visiting address
Ottoplatz 1
50679 Cologne
Germany

Tel +49_221_89990-000
Fax +49_221_89990-999
Mail info@easa.europa.eu
Web easa.europa.eu