FPGA Advantage®

Product Comparison Table

FPGA Advantage is a complete FPGA HDL Design solution supporting VHDL and Verilog on PCs and Unix workstations. To meet the diverse needs of FPGA designers, FPGA Advantage is available in multiple versions:

- **FPGA Advantage Single Vendor Personal HDL Design** is an offering of FPGA Advantage that supports only one FPGA vendor, either Xilinx or Altera. This combination of **HDL Author™, Model*Sim*® Personal Edition** (PE), and **LeonardoSpectrum™** has versions available in VHDL or Verilog, and no mixed-HDL, Text or Graphics design entry editors, and Level 3 synthesis functionality. This offering only supports Windows platforms.

- **FPGA Advantage for Personal HDL Design** is the price/performance leader for PC-based FPGA design flows. This combination of **HDL Author™, Model*Sim*® Personal Edition** (PE), and **LeonardoSpectrum™** has versions available in VHDL, Verilog, or mixed-HDL with Text and/or Graphics design entry editors, and either Level 2 or Level 3 synthesis functionality. This offering of FPGA Advantage supports Windows platforms.

- **FPGA Advantage with Personal Simulation** is the FPGA design flow consisting of **HDL Designer™**, **Model*Sim* Personal Edition**, and **LeonardoSpectrum**. This version is available for VHDL, Verilog, or mixed-HDL with Text and/or Graphics design entry editors and has the option of Level 2 or Level 3 synthesis functionality. This flavor of FPGA Advantage supports Windows platforms.

- **FPGA Advantage for HDL Design** represents the premium product, with the full set of features necessary for cutting-edge FPGA development on both PCs and Unix workstations. This version includes **HDL Designer**, **Model*Sim* SE** (Special Edition) and **LeonardoSpectrum** Level 3 with **LeonardoInsight™** and is available in many HDL language combinations with Text and/or Graphics design entry editors.

FPGA Advantage Configurations Comparison

General

Product Feature	Description	FPGA Advantage			
		Single Vendor Personal HDL Design (All Configurations)	**Personal HDL Design (All Configurations)**	**Personal Simulation (All Configurations)**	**HDL Design (All Configurations)**
Language support	Complete VHDL or Verilog	√	√	√	√
	Mixed HDL		**option**	**option**	**option**
UNIX platform	Solaris 7/8, HP-UX11.0				√
Windows platform	98, NT, 2000, XP	√	√	√	√
Floating configuration			√	√	√
Node-Locked configuration		√	√	√	

FPGA Advantage Configurations Comparison

Design Creation and Management (HDL Designer Series)

Product Features	Description	FPGA Advantage										
		Single Vendor Personal HDL Design		Personal HDL Design (L2 and L3)			Personal Simulation (L2 and L3)			HDL Design		
		Text	Graphics	Text	Graphics	Pro	Text	Graphics	Pro	Text	Graphics	Pro
Flow Chart Editor	Graphical creation of any HDL description using algorithmic notation e.g. test-benches		√		√	√		√	√		√	√
Truth Table Editor	Advanced Editor for creation of tabular input/output matrices with complex input conditions and arbitrary output actions		√		√	√		√	√		√	√
State Diagram Editor	State diagram creation		√		√	√		√	√		√	√
Block DiagramEditor	Hierarchical structure entry	√	√	√	√	√	√	√	√	√	√	√
Symbol with Tabular IO Editor	Graphical-based and table-based design interface creation	√	√	√	√	√	√	√	√	√	√	√
IBD™ Editor	Interface-Based Design™: table-based editor to describe complex design hierarchies and interconnections	√		√		√	√		√	√		√
Text Editors	Internal text editors (ESview for Windows, NEdit for UNIX) or Interface to external editors such as vi, Emacs, TextPad, UltraEdit, etc.	√	√	√	√	√	√	√	√	√	√	√
ModuleWare Library	Technology-independent, language-independent, parameterized macros	√	√	√	√	√	√	√	√	√	√	√
Convert to Graphics (Block and IBD)	Convert the existing textual views in FPGA Advantage database to block diagrams and IBD views						√		√	√		√
Convert to Graphics (Block, Flow Chart, and State Diagram)	Convert the existing textual views in FPGA Advantage database to block diagrams, state diagrams and flow charts							√	√		√	√
HDL Import (Text)	Import legacy designs and create textual views	√	√	√	√	√	√	√	√	√	√	√

FPGA Advantage Configurations Comparison

Design Creation and Management (HDL Designer Series)

Product Features	Description	FPGA Advantage										
		Single Vendor Personal HDL Design		Personal HDL Design (L2 and L3)			Personal Simulation (L2 and L3)			HDL Design		
		Text	Graphics	Text	Graphics	Pro	Text	Graphics	Pro	Text	Graphics	Pro
HDL Import (Block and IBD)	Import legacy textual designs into FPGA Advantage database, automatically convert the hierarchical information to block diagrams and IBD views						√		√	√		√
HDL Import (Block, Flow Chart and State Diagram)	Import legacy textual designs into FPGA Advantage database, automatically generate block diagrams, state diagrams and flow charts							√	√		√	√
Revision Control	Full version control with RCS and CVS, or interface to Rational ClearCase, Microsoft Visual SourceSafe, Synchronicity DesignSync and ClioSoft SOS	√	√	√	√	√	√	√	√	√	√	√
Cause	Relate waveform events to graphical source		√		√	√		√	√		√	√
Animation	Debugging features of a flow chart or a state diagram		√		√	√		√	√		√	√
Show Graphics	Cross highlight from Source window to HDS graphical diagram	√	√	√	√	√	√	√	√	√	√	√
HTML Export	Web-based HTML exports on any graphical view for documentation purposes						√	√	√	√	√	√
OLE	OLE (Object Linking and Embedding) for Window based documentation tools including the Microsoft Office applications and Adobe FrameMaker	√	√	√	√	√	√	√	√	√	√	√
C/C++ Integration	VHDL wrapper generation for C or C++ models for creation, simulation and debug with ModelSim	√	√	√	√	√	√	√	√	√	√	√
Design Browser	Explorer type view of entire design	√	√	√	√	√	√	√	√	√	√	√
Custom Tool/Flows	Add custom tools and flows	√	√	√	√	√	√	√	√	√	√	√

FPGA Advantage Configurations Comparison

Simulation (ModelSim PE and SE)

Product Feature	Description	FPGA Advantage			
		Single Vendor Personal HDL Design (All Configurations)	Personal HDL Design (All Configurations)	Personal Simulation (All Configurations)	HDL Design (All Configurations)
Waveform Compare	Completely configurable waveform comparison. Compare continuously or at specific times	**option**	**option**	**option**	√
Performance Analyzer	Optimizes designs to improve simulation performance (not available on Windows 95/98)				√
Code Coverage	Analyzes coverage of the HDL code, improves effectiveness of a set of vectors	**option**	**option**	**option**	√
Advanced Debugging Features	Bus contention and float checking, stability and toggle checking, power analysis, extended VCD, VCD for VHDL, checkpoint-restore				√
	Enhanced DataFlow Window	**option**	**option**	**option**	√
Language Neutral Licensing (LNL)	LNL enables VHDL or Verilog simulation with single license				**HDL (LNL) licensing selection**
Accelerated VITAL and Verilog Primitives	Up to 5x performance gains				√
Customizable, User-Expandable GUI	User definable menus, buttons, other GUI enhancements				√
VHDL “C” Interface	Interface Model*Sim* to third-party software and/or custom “C” code				√
C Debugger	Build-in C debugger, supports C/C++ code				√
Synopsys LMG Model Support	SmartModels and hardware modeler				√

FPGA Advantage Configurations Comparison
Simulation (ModelSim PE and SE)

Product Feature	Description	FPGA Advantage			
		Single Vendor Personal HDL Design (All Configurations)	Personal HDL Design (All Configurations)	Personal Simulation (All Configurations)	HDL Design (All Configurations)
ASIC Vendor Sign-Off Qualified	See www.model.com/svp/vend.asp for a complete list of supported ASIC vendor libraries				√
Single kernel Simulation	Mixed-language support	√	√	√	√
Optimized Native Compile	The architecture of producing coding that is both platform independent and native compiled	√	√	√	√
Waveform File Compression	Save WLF files in compressed mode	√	√	√	√
Standard & Extended VCD support (Verilog)	VCD files contain simulation events represented using 4 states: 0, 1, X and Z. This format is excellent for capturing vectors. Ability to write out multiple VCD files	√	√	√	√
PLI	PLI 1.0, 2.0	√	√	√	√
VCD file viewing utility	VCD to WLF utility-any standard VCD file converted to ModelSim WLF format for viewing	√	√	√	√
SDF	1.0, 2.0, 2.1, 3.0	√	√	√	√
Next generation GUI	Drag and drop, integrated source code editing, etc.	√	√	√	√
TCL	TCL (Tool Command Language) scripting	√	√	√	√

FPGA Advantage Configurations Comparison

Synthesis (LeonardoSpectrum Level 2 and Level 3)

Product Feature	Description	FPGA Advantage					
		Single Vendor Personal HDL Design (Text, Graphics)	Personal HDL Design (Pro, Text, Graphics)		Personal Simulation (Pro, Text, Graphics)		HDL Design (Pro, Text, Graphics)
		Level 3	Level 2	Level 3	Level 2	Level 3	Level 3
Hierarchy Manipulation		√		√		√	√
Bottom-up and Team Design		√		√		√	√
Batch Mode with Command Line Interface		√		√		√	√
Command Line Interface		√		√		√	√
Multi Pass Optimization		√		√		√	√
Mixed HDL Design	Support for Intellectual Property	√		√		√	√
Critical Path Optimization		√		√		√	√
Interactive TCL Scripting		√		√		√	√
Incremental Synthesis	Block Based Design	√		√		√	√
Fully Interactive		√		√		√	√
TimeCloser		√		√		√	√
TrueTiming Optimization		√		√		√	√
FPGA Targeting		√		√		√	√
LeonardoInsight	Cross probing, schematic viewing, schematic fragments generator, etc	**option**	**option**	**option**	**option**	**option**	√
Design Hierarchy Browser		√	√	√	√	√	√
Technology re-targeting		√	√	√	√	√	√
SDF Back-annotation		√	√	√	√	√	√
Project support		√	√	√	√	√	√
Post P&R timing analysis		√	√	√	√	√	√
Advanced constraints		√	√	√	√	√	√

FPGA Advantage Configurations Comparison
Synthesis (LeonardoSpectrum Level 2 and Level 3)

Product Feature	Description	FPGA Advantage					
		Single Vendor Personal HDL Design (Text, Graphics)	Personal HDL Design (Pro, Text, Graphics)		Personal Simulation (Pro, Text, Graphics)		HDL Design (Pro, Text, Graphics)
		Level 3	Level 2	Level 3	Level 2	Level 3	Level 3
Single pass optimization		√	√	√	√	√	√
FSM optimization		√	√	√	√	√	√
Global constraints		√	√	√	√	√	√
Top-down design		√	√	√	√	√	√
RAM, ROM and counter inferencing		√	√	√	√	√	√
P&R encapsulation		√	√	√	√	√	√
Source code editor		√	√	√	√	√	√
Hierarchy Preservation or Dissolve		√	√	√	√	√	√
Critical Path Optimization		√	√	√	√	√	√
TCL Scripting		√	√	√	√	√	√
Template insertion		√	√	√	√	√	√
Muilt-FPGA Vendor Support		Xilinx or Altera	√	√	√	√	√

FPGA Advantage Advanced Capabilities

Design Entry and Management with HDL Designer Series

HDL Designer Series provides a complete design creation and management environment for FPGA design.

The design entry methods include not only graphical editors such as block diagram, state diagram, flow chart and truth table, but also most productive editors (the IBD editor and Tabular IO editor) for describing large and complex design interfaces, hierarchies and interconnections.

The management environment provides the integration of the version control system. File Registration, Custom Tools and Flows creation expand the ability of the management to other data types and tools outside of FPGA Advantage.

The IBD Editor

The IBD, or Interface-Based Design editor, is a table-based design entry method for complex design hierarchies and interconnection description. It simplifies the definition of design interconnect for blocks, macros, components and Intellectual Property. The compact tabular format displays the design structure and interconnections, generates the HDL netlists automatically, and makes the documentation easy and simple.

The IBD editor is available in Text and Pro versions.

Revision Control

Version control is critical to large project teams that need to manage several versions of their designs; without version control, typically only one version of any design unit is available. Multiple versions of design objects can be stored in the shared repository and accessed for viewing or edit. An audit trail is maintained to record which engineer made changes with associated comments stored for each revision. A locking mechanism prevents concurrent edits being made by different users and indicates when another user locks an object.

A generic interface to version control management facilities, which provide check in, check out, locking, history and tagging controls to support team design projects. The interface is supplied with the GNU revision control system (RCS) and CVS, but also supports Rational ClearCase, Microsoft Visual SourceSafe, Synchronicity DesignSync and ClioSoft SOS.

Debugging Features with ModelSim

HDL Designer Series works with ModelSim seamlessly. After invoking ModelSim from HDL Designer Series, additional menu functions are added to the HDL Designer Series and Model*Sim* session! You can enable the debugging features such as animation (for state diagrams and flow charts), cross-probing, cause-and-effects, etc. With the built-in C debugger in ModelSim, you can debug your C/C++ code more efficiently and seamlessly.

Cross-Referencing

If you are navigating through the HDL source in the Source window and suddenly realize you need to locate the diagram that generated the HDL, you can position your cursor on any line of code and hit *the Show Graphics* menu item. Immediately the graphical diagram (block, state, flow-chart or table) will be immediately opened and the specific graphical element highlighted.

When generating or compiling an HDL netlist and you find error/warning messages, you can quickly correlate to the source and correct the problems.

Block-based Design Support

Design teams must take a modular-based design approach for complex design, where the design is partitioned at the HDL text or graphical level. Each member of the team develops and implements their own piece of the design independently; creates their own HDL or graphics with timing constraints; and individually simulates, debugs, synthesizes and runs place and route. When an individual design is ready, design manager can simply re-elaborate and link each individual design to the top level. This process allows small functional changes to be quickly incorporated into the final place and route while preserving overall timing performance.

Team & User Preferences Control

Team preferences control allows the design manager or administrator to set and control the main design flow environment for his/her design team. These preferences include: version management setup, HDL filename template, generation properties, file registration, custom tools and custom flows. User preference control allows individual designers to set and control his/her own design environment. These preferences include: text and diagram creation, compiler setting, HDL generation checks and how each type of graphical diagram is displayed.

UNIX Platform & Floating Configuration

HDL Designer Series supports both PC platforms (Windows 98, NT, 2000 and XP) and Unix platforms (Solaris and HP-UX). Any mix of PC and Unix platforms is supported: the license server can be either PC or Unix, and the client nodes can be either PC or Unix. Complete interoperability between platforms provides a common user interface, identical functionality, and even identical graphical database objects.

Simulation with ModelSim

Model*Sim* SE has been designed to increase your productivity with features, capacity, and performance for the most demanding designs. In addition to core features such as optimized direct compiled code, single-kernel mixed-language simulation, platform independent compilation, and complete standards support, Model*Sim* SE has a number of advanced features.

Performance Analyzer

The performance analyzer built into Model*Sim* allows you to analyze the simulation run and show what part of the model is taking the simulation time. The profiling will work for all modeling languages supported at all levels of abstraction. It can uncover problems such as a VITAL gate level cell that has not been globally accelerated due to it not being level-1 compliant. It will uncover processes with unnecessary signals on the sensitivity list causing it to be triggered too much. It can detect testbenches code that is not necessary for a particular test but that is still taking simulation time. Design bottlenecks can also be easily located.

Code Coverage

The coverage within Model*Sim* is line coverage. ModelSim counts the number of times a particular executable line of code has been visited during simulation. With Model*Sim* Code Coverage, there is no extra tool to put in your design flow or to learn (it's just another familiar looking Model*Sim* window). The code does not need to be instrumented since it is connected directly to the simulation kernel.

Waveform Compare

The ModelSim Waveform Compare feature analyzes the results of two simulations to quickly identify design errors. Waveform Compare is completely configurable. Any number of signals or regions can be compared either continuously or relative to a signal edge. You can set tolerances for any signal, set start and end times for the compare, set a limit for the total number of comparisons, and view the compare results graphically or in text.

Signal Spy

The ModelSim Signal Spy feature can check or force any signal in any part of your design, VHDL or Verilog, without having to change or modify your original design code, thus simplifying your testbench development.

Enhanced Dataflow Window

The Enhanced Dataflow window in ModelSim provides a powerful X-tracing feature that allows designers to automatically trace the root cause of any unknown signal and generate the schematic of any or all portions of the design. The Enhanced Dataflow window also cross-links to the Source, Wave and other standard ModelSim windows to enable significant debug productivity throughout.

Advanced Debugging Features

Model*Sim* SE has a number of advanced debugging features that are particularly useful for designing and debugging large designs. Many of these features are used by ASIC vendors as part of their signoff flow, and are useful in helping to shorten the signoff process:

Float Checking — Detects when a bus has been in a high-impedance state for longer than a user-specified amount of time.

Stability Checking — For synchronous designs, detects when circuit activity has not settled within a user-defined time period. Provides a report on pending driver events.

report on pending driver events.

Bus Contention Detects conflicts on design nodes (signals or nets) that have multiple drivers.

Toggle Checking Counts the number of transitions to 0 and 1 on specific nodes.

Power Analysis Creates a power analysis data file that can be fed into power analysis tools.

Extended VCD Standard VCD files contain simulation events represented using 4 states: 0, 1, X and Z. While this format is excellent for capturing vectors, it does not provide information such as signal strength and driving direction for bi-directional ports. Extended VCD provides this additional information in an 8-state format.

Checkpoint-restore This feature greatly enhances productivity, particularly during the debugging phase for large designs with long test sequences. For example, checkpointing a design after loading a large SDF file and running initialization code can save hours of simulation time during subsequent simulation runs.

Language-neutral licensing

Language neutral licensing (LNL) enables simulation of either VHDL or Verilog with a single license and single simulator. Model*Sim* SE/LNL gives you flexibility to meet all your language demands now and in the future. For example, an LNL license enables you to incorporate third party IP, regardless of the language, without requiring the purchase of additional simulators or additional licenses. LNL also improves your utilization of simulation licenses as the ratio of VHDL to Verilog simulators you require changes between project phases or between projects.

Accelerated VITAL and Verilog Primitives

Model*Sim* SE contains built-in primitives for standard VITAL packages, resulting in a performance increase of up to 5x for VHDL gate-level (VITAL) designs. Built-ins for standard Verilog primitives result in up to a 3x performance gain for Verilog gate-level designs.

Next Generation TCL/TK-based GUI

Model*Sim* release 5.0 introduced a Tcl/Tk-based GUI that sets the standard in simulator interfaces. Some examples of the power and flexibility of this interface:

Editable source window Edit your source code in the Model*Sim* source window, recompile the file then restart the simulation all without leaving Model*Sim*.

Cross highlighting of Compiler messages Click on an error message in theModel*Sim* transcript window and the line of code that caused the error is brought up and highlighted in the source window.

Language-independent Compile manager Allows for easy compilation of VHDL and/or Verilog code.

Drag and drop Most Model*Sim* debugging windows allow dragging and dropping of objects between windows. For example, grab a structural block from the structure window and drop its underlying signals into the wave window; select various elements from the signal window and drop them into the list window; select a signal from the wave window and drop it into the dataflow window.

Multiple windows

Model*Sim* SE supports multiple copies of each debugging window. For example, you can create two or more wave windows for viewing different signals and/or different simulation times.

Wave window

Powerful features are available such as grouping of separate signals in the wave window into a new bus; searching for next/previous signal edge (or expression of signal edges/values); quick-access buttons for zooming, searching, and setting new cursors.

Notepad editor

A simple notepad editor for editing macro files or HDL source files. Alternatively, the “edit” Model*Sim* command can be used to launch your favorite editor from the Model*Sim* prompt

Customizable, user-expandable GUI

Because Model*Sim* uses industry standard Tcl/Tk as the basis for its GUI and scripting language, any number of customizations to the look, feel, and functionality of Model*Sim* can be made. Company project, or user-specific menus and buttons can easily be added to Model*Sim* to provide for the most efficient simulation and debugging environment.

Synthesis with LeonardoSpectrum

LeonardoSpectrum combines power with ease of use. If desired, you can perform 100% push-button synthesis using LeonardoSpectrum. LeonardoSpectrum's exclusive F.A.S.T. (FPGA Architecture Specific Technology) algorithm helps designers obtain the best possible Quality-of-Results (QoR). FlowTabs™ automatically walk new users from HDL coding all the way to SDF back annotation. The exclusive SynthesisWizard™ and QuickSetup features make first time FPGA synthesis simple.

As your synthesis skills grow and the complexity of your circuits increase, Power Tabs offer you additional control over the synthesis process. Power Tabs enable experienced designers to "shove" a design into a smaller, less expensive FPGA or "push" a circuit for maximum speed. Power Tabs provide the features and control you need when "push" inevitably comes to "shove."
LeonardoSpectrum preserves your design hierarchy. This enables you to exploit the advantages of a hierarchical-based design approach, including the use of both incremental design and synthesis capabilities. With .lib support and more than 200 FPGA and ASIC technologies fully supported, LeonardoSpectrum isn't going to run out of gas as your designs become larger, faster, and more challenging.

LeonardoInsight

LeonardoInsight augments your detailed design knowledge with powerful debugging and analysis capabilities. You gain valuable insight into how different synthesis options effect your results. With LeonardoInsight, you can interactively analyze the impact of different constraint settings, coding styles, and technology parameters on your overall QoR. LeonardoInsight offers designers advanced analysis tools to track down and solve circuit bottlenecks and area budget violations.

P&R Integrator

LeonardoSpectrum "encapsulates" FPGA vendor place and route (P&R) tools, allowing you to go from schematics to HDL to back annotation from within the same familiar GUI. LeonardoSpectrum uses its deep synthesis knowledge to automatically create SmartScripts™ - "intelligent" P&R scripts for vendor tools resulting in better circuit implementations.

Incremental Synthesis

Incremental synthesis allows you to recompile only the modules that do not meet the timing constraint, and preserve all other modules with results that met the constraints. This capability do not only reduces compilation time, but it also preserves the timing behavior of certain sections of a design when other parts of the design are under development.

Support for Multiple Design Flows

Whatever your preferred design flow, flat or hierarchical, top-down or bottom up, LeonardoSpectrum supports it. If you have IP to reuse or dozens of designers working on modules that will need to be stitched together in a bottom-up fashion, LeonardoSpectrum will fully support you.

True Hierarchical Support, Incremental Synthesis

LeonardoSpectrum was designed to support true hierarchical design. Its hierarchy browser allows you to access, manipulate, constrain and interchange hierarchical blocks. Hierarchical design enables you to perform incremental design and synthesis. You can modify individual modules at the RTL level, then re-synthesize and re-optimize while preserving netlist information in surrounding blocks; significantly reducing compile times on complex, multi-block designs. An incremental approach is also supported during synthesis. Constraints can be "tightened" on sub-blocks and re-optimized to fine tune timing or area goals following place and route operations. Block-level design provides the key to efficient interaction between synthesis and place and route.

TimeCloser

TimeCloser is an automated flow to identify and improve timing performance on true critical paths. TimeCloser annotates physical timing data after place and route, and uses this physical data to perform second pass timing optimizations on the true critical paths only to improve design performance.

TrueTiming Optimization

TrueTiming optimization uses replicate logic and clustering techniques to reduce route delay and improve device performance.

F.A.S.T. Optimization

LeonardoSpectrum's proprietary F.A.S.T. algorithm guarantees the highest QoR of any FPGA synthesis tool available today. F.A.S.T. evaluates factors such as your target technology, constraints, number of serial elements, along with a dozen other design characteristics, and automatically selects an optimization strategy to obtain the best QoR.

Scripting

LeonardoSpectrum offers significantly more powerful scripting capabilities than its competitors. Powerful scripting enables you to set up and enforce customized design flows, company-wide. Scripts can access the entire design database. You get complete data manipulation capability. Conditionals enable users to automate large, complex tasks.

Partitioning

LeonardoSpectrum makes it easy to partition designs for prototyping purposes. Often it's desirable to prototype a large FPGA or ASIC in multiple, smaller FPGAs. True hierarchical support makes it easy to group and ungroup design elements, which can then be targeted for single or multiple chips.

LeonardoSpectrum is the only synthesis tool that offers a truly seamless, technology-neutral environment, from FPGAs to ASICs. This allows retargeting circuits:

- When acquiring or re-using IP
- When re-targeting FPGA code to an ASIC device
- When cheaper and/or faster devices become available
- When a decision is made to retarget to a new device in "mid-design"
- In order to partition and prototype ASICs using FPGAs

Support for Schematics

If you need to re-draw old schematics, with LeonardoSpectrum you can perform automatic, error-free re-targeting of existing schematics to newer field programmable devices, quickly and easily.

FPGA vendor support

LeonardoSpectrum supports all market leading FPGA vendors, such as Actel, Altera, Atmel, Cypress, Lattice, Minc, QuickLogic and Xilinx