Chapter 1
Visual Elite Design Flow

The Visual Elite™ tool provides you, as a SoC, ASIC, and FPGA developer and system designer, a comprehensive design environment. Working in this environment, you can capture design concepts graphically and textually, verify design behavior through simulation and debugging, and generate executable specifications for hardware implementation and HDL synthesis.

Use of Visual Elite spans the complete scope of the overall design cycle, from concept to implementation, including the system analysis and validation and hardware implementation and verification phases of most of today’s complex projects.

The Visual Elite tool provides support for VHDL, Verilog, and SystemC modeling through multi-language textual and graphical editors. These enable capturing designs as block diagrams, state diagrams, flowcharts, algorithmic flowchart state machines, and truth tables. The tool also provides the capability of transforming existing textual HDL designs into graphical representations.

The Visual Elite tool enables you to start with the capture of high-level specifications, to move on to verify those specifications via simulation, and to continue by partitioning a design, decomposing it into functional blocks and smaller subsystems, and verifying each one of them. Further partitioning is carried out until the descriptive level is suitable for mapping to specific hardware or software resources. As this level of detail is reached, the high-level SystemC models can be replaced by equivalent HDL blocks, or can be used as input to high-level synthesis tools.

The main topics presented in this section are:

- System Analysis and Validation
- Hardware Implementation and Verification

System Analysis and Validation

Today, engineers are turning increasingly to SystemC for capturing electronic designs. The adoption of these software languages has been driven by the need to remove a major barrier to design verification inherent to the HDLs. Using SystemC enables designs to be depicted at a higher level of abstraction while also providing a single language to capture initially uncommitted portions of the design; portions which can ultimately be implemented as either hardware or software.
With Visual Elite, you can graphically capture design units in SystemC, while being able to link such units to others created in HDL. The result is a major step forward in design flexibility and, for the first time, the ability to work in an environment whose scope is truly system-wide.

In Visual Elite, SystemC design units are created in the Block and State Diagram editors. Block diagrams are used to depict the basic structure of the design architecture, taking a design from its highest, most abstract level, down to the physical level. As a design hierarchy is progressively developed, each one of its component elements at the various levels is represented as a block, whether it corresponds to a processor, memory unit, data flow component, or any other hardware component. In addition, a block diagram depicts the signals and communication channels through which data flows between its blocks during the execution of algorithms and protocols.

The Visual Elite tool, in conjunction with the Design Checker utility, can be used to easily locate design errors after compilation and before simulation.

**Hardware Implementation and Verification**

The Visual Elite tool provides you, as an ASIC and FPGA IP developer, a clear path from concept to implementation. Working in this environment, you can capture design concepts graphically and textually, verify design behavior through simulation and debugging, and generate synthesizable VHDL or Verilog appropriate for their selected synthesis tool.

The tool incorporates four graphical editors to enable capturing designs as block diagrams, state diagrams, flowcharts or algorithmic flowchart state machines (AFSMs), and truth tables. It also provides textual editors for writing designs in VHDL or Verilog, as well as a tool for transforming existing textual designs into graphical representations.

**Design Capture** — As shown in Figure 1-1, work generally commences with design capture using one or more graphical editors. For each functional block of the design, you can use the most appropriate editor:

- Block diagrams can be used in either a “top-down” design strategy in order to develop architectural partitioning of an idea, or “bottom-up” to enable gate-level schematic or even gate-level capturing. Ultimately, any given design can incorporate both of these methodologies. Within block diagrams, you can also integrate macro components — configurable, technology-independent logic gates, and sequential logic components — such as counters, multiplexers, etc.

- State machine diagrams are particularly useful in developing and debugging behavioral models by logically describing control flow.

- Flowcharts represent the activities of your design algorithmically, and can be used to describe sequential flow or algorithmic flowchart state machines (AFSMs).

- Truth tables can be used to capture blocks of combinational, as well as sequential logic.
In addition, the tool includes a text editor to edit and display new, revised, or imported textual units written in VHDL or Verilog.

**Figure 1-1. Visual Elite Design Flow**

**Text to Graphics** — The HDL to Graphics capabilities of Visual Elite enhance design reusability, documentation, and maintenance, by translating existing designs written in VHDL or Verilog to the appropriate Visual Elite graphic description: block diagrams from structural modules; state diagrams or algorithmic flowchart state machines from modules characterized by logic control; and flowcharts from algorithmic modules.

**Design Management** — The Visual Elite tool includes facilities for project management, version control, and access control at the library, unit, and design tree level. All units created using the various editors are stored in libraries, the contents of which are conveniently accessed and manipulated by means of the hierarchical browser.
**Extension Language** — The Visual Elite Extension Language (VEL) is a powerful, CFI-compliant, scheme-based programming language that enables extending and customizing the functionality of Visual Elite. VEL enables you to greatly enhance your productivity with scripts that execute repetitive actions automatically. VEL can also support managers, project administrators, and CAD developers in tailoring Visual Elite to a changing design environment.

**Documentation** — The Visual Elite tool supports a number of utilities which enable you to document your designs. These include the incorporation of graphics and design-related data in FrameMaker documents, and the generation of HTML-compatible graphical representations of your designs.

**Compilation** — Compilation translates a unit or stimulus object into input for simulation or code generation. During this stage, an analyzer examines the design for errors. The analyzer checks the logic and syntax of all data being compiled, from a single design unit up to an entire design tree, and the compiler generates the appropriate objects.

**Design Verification** — The HDL designs that you create can be verified using the Visual Elite internal simulator, or any one of the major external simulation environments which it supports. Simulated VHDL-based designs may include SWIFT models and external units. Before simulation, you define a stimulus for the design being tested using the Visual Elite waveform editor, or in text. Working in tandem with the debugger to enable rapid analysis, you can view simulation results in graphical wave displays. You can set breakpoints, modify the values of existing signals and variables, single-step through different execution threads, and trace various activities. Statistical summaries of design activity are available both as graphical displays and as print-outs. The “Cause & Effect” feature of Visual Elite enables you to immediately view the design source of any given simulation result.

**Code Generation** — Once the design has been verified, the Visual Elite tool lets you produce VHDL or Verilog-format code, or an EDIF netlist, for whatever further processing you intend. With the Code Manager, you can control various attributes of the code generated by Visual Elite, including the code’s style, headers, naming conventions, and more. You have the option of choosing between synthesizable HDL code and HDL code that is intended only for external simulation.