Using Active-VHDL Design Entry and Behavioral Simulation with Atmel IDS 6.0

A Guided Tour Using Atmel's Averager Example Design

Introduction

Active-VHDL 3.5 now supports Atmel AT6K and AT40K FPGA behavioral simulation libraries. For existing users of Active-VHDL, who may wish to retain Active-VHDL's familiar design entry and project management environments while targeting Atmel FPGA families, this document presents a method in which designs can be moved through the design flow across independent synthesis and implementation toolsets.

In the area of synthesis, many synthesis software developers now support the AT6K and AT40K FPGA families, including, for the PC platform:

- Everest Design System (software and license included with Atmel IDS install)
- Exemplar Logic Leonardo Spectrum
- Synopsys FPGA Express
- Synplicity Synplify
- Orcad Express

For this guided tour, it is assumed that the user is familiar with or will further explore the Active-VHDL Design Entry and Simulation environments, and the EDS synthesis tool mentioned above. Please refer to the CAE Interfaces section in the IDS Tutorial for additional details regarding use of these synthesis tools.

The following sections describe a data averager design written in VHDL. It is first opened in Active-VHDL, and then subsequently optimized with one of the above synthesis tools, before implementation into an AT40K FPGA using Atmel's IDS software.

The basic flow described in the following pages represents the fastest path to implementation in an Atmel FPGA. While there is a cursory overview of pre-synthesis simulation, the user is strongly encouraged to perform pre-layout and post-layout simulation as well as to utilize the effective design management features of Active-VHDL.

Example Design

In this guided tour, the "averager" design will be used. This design implements a waveform smoothing function represented by:

$$y(n) = \frac{1}{M} \sum_{(i=0)}^{(M-1)} x_{(n-i)}$$

The smoothing function is used in many DSP applications to filter out high frequency spikes. These spikes are the source of noise commonly found in communications channels. They can be eliminated by taking the moving average of sample values arriving at the input of the system.

The design in this example is an "averager" circuit with eight moving points. The hardware specific details of the design are described below.

INTERFACE: The "averager" design has 8-bit input and output data lines. The design also has a clock pin and a reset pin.

Design Entry and Behavioral Simulation with Atmel IDS 6.0

Application Note

Rev. 1445A–02/00

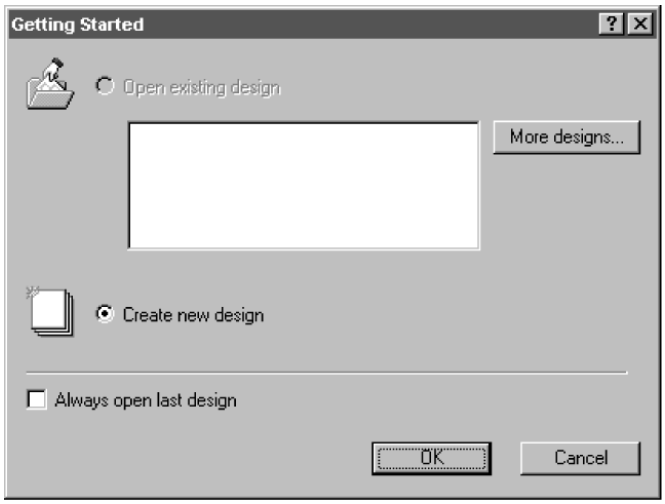
ASSUMPTIONS AND OTHER DETAILS: Only positive values can appear on the input data lines (the input is assumed to be level-shifted). During the implementation of the system, a valid output waveform can be expected on the ninth clock. However, since the outputs of the last and intermediate stages are registered, the actual output waveform starts occurring from the eleventh clock cycle onwards.

This assumes use of IDS 7.0 or above.

Active-VHDL Design Setup

Make sure the Active-VHDL tool has been installed on the path that has no space in between. Otherwise, it will give errors.

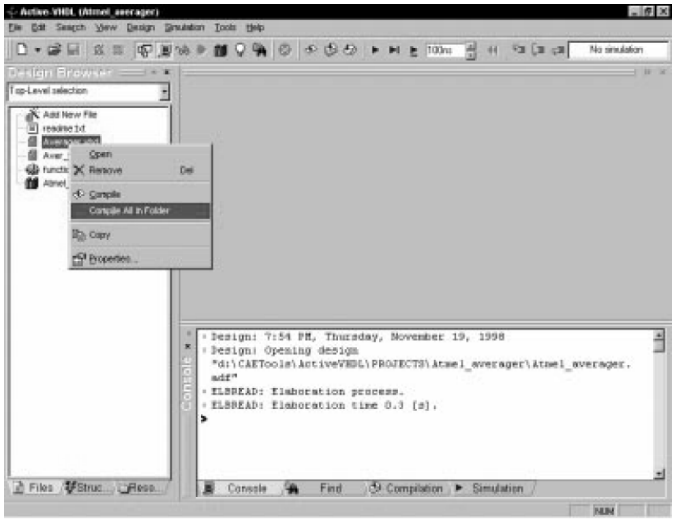
Upon invoking Active-VHDL, the user is presented with the Getting Started dialog. To access the “averager” example design, select “More designs.”



The Design Explorer is launched, allowing the user to access all registered designs including those located in the Samples folder. Click on OPEN and go to atmel/examples/at40k/activeVHDL/. Double-click on “averager”.

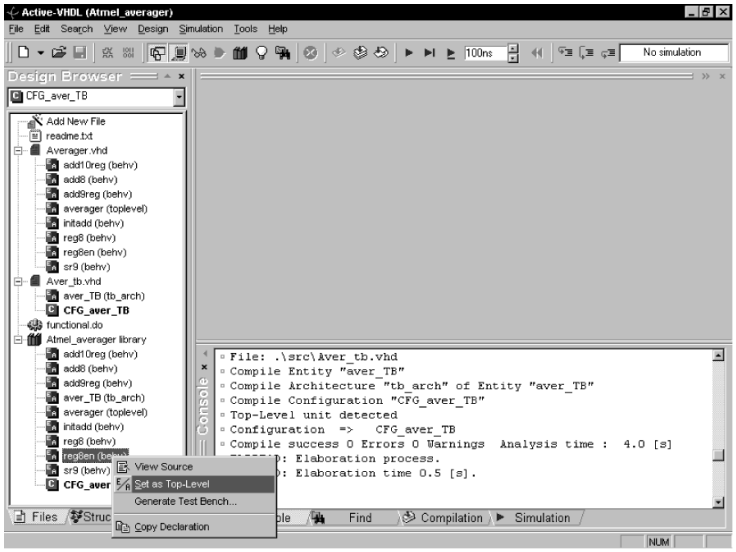
The behavioral source file, testbench, and functional simulation macro (script) are listed in the Design Browser.

To compile the source file averager.vhd and testbench Aver_tb.vhd, right-click on averager.vhd and select the “Compile All” in “Folder” option from the context sensitive menu.



The green check representing the VHDL files indicates that these files have been successfully compiled.

It is also possible to change the top-level unit to later allow the simulation of a given subdesign. To change the top-level unit, right-click on a unit from among the library contents and select the “Set as Top-Level” option from the context sensitive menu.

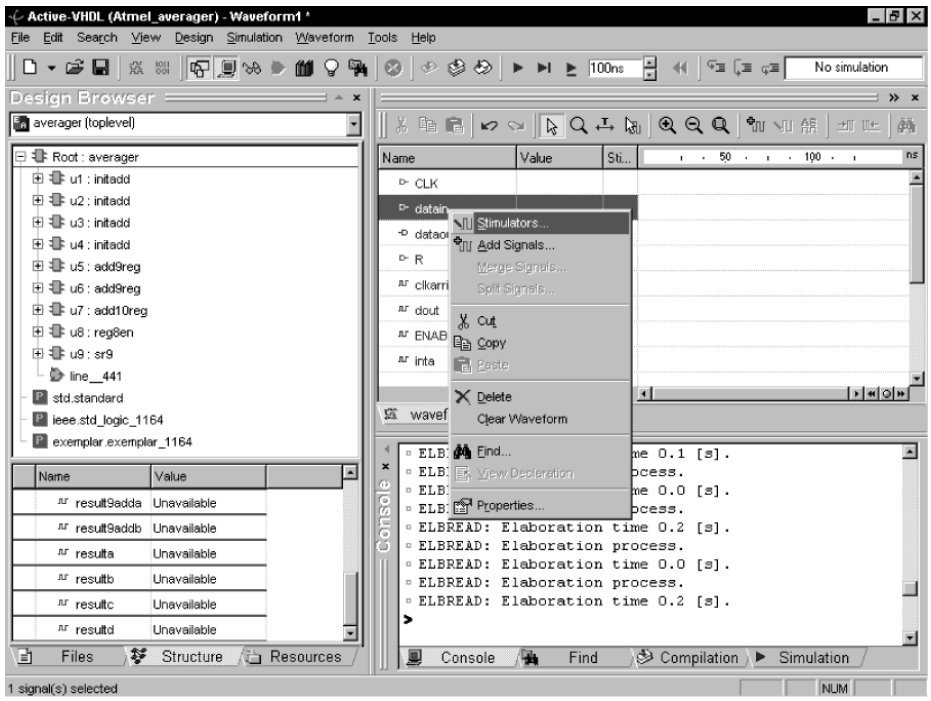


To proceed in this guided tour, please ensure that averager is selected as the top level. A unit’s status as top level is indicated by boldface type.

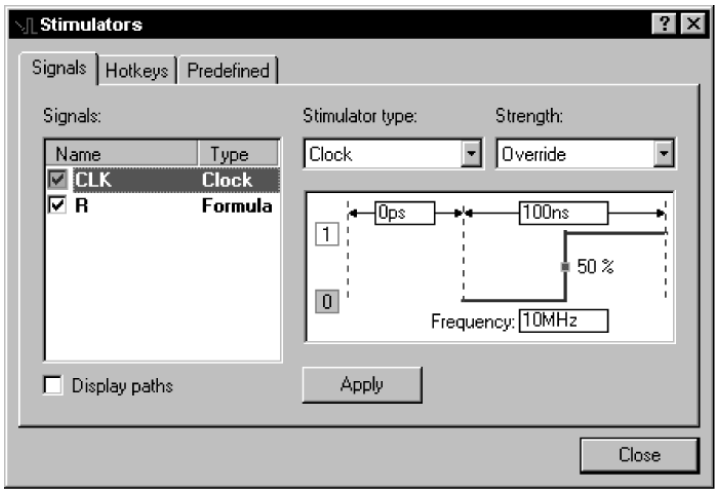
Active-VHDL Pre-synthesis Simulation

The structure of the top level unit, in this case averager, is viewed by selecting the “Structure tab of the Design Browser”.

Clicking on the “Root : Averager” item shows all the signals and ports under this design level. A new waveform window can be invoked by selecting the File -> New -> Waveform option from the drop-down menus. Signals from the listing window can be dragged individually or as a group over to the waveform window.

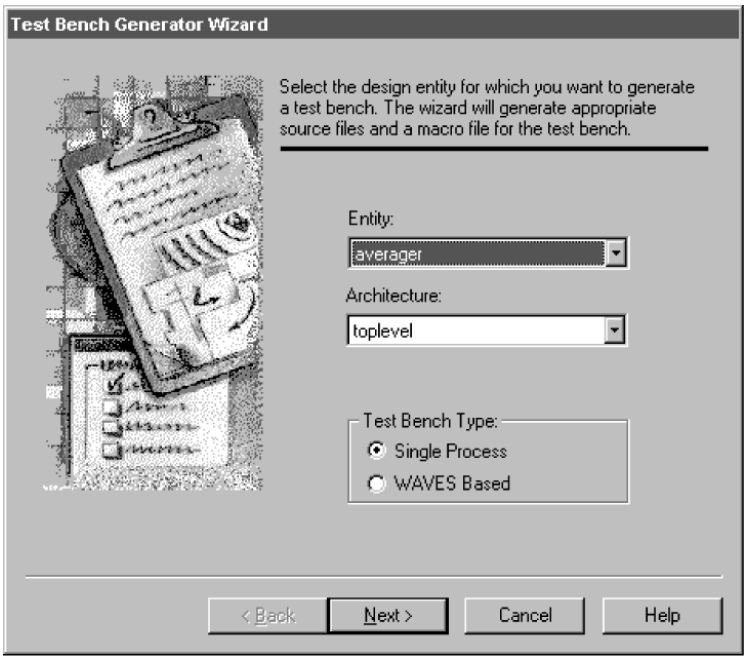


Right-click on any input signal in the waveform window and select the “Stimulators…” option from the context sensitive menu to launch the Stimulators window. Additional signals can be dragged from the waveform window to the Stimulators window.



A functional waveform can be generated by way of applying Clock-, Formula-, Predefined and/or Hotkeys-driven stimulus over simulation time. This functional waveform can be saved and later used for (1) testbench generation and (2) automated comparison against pre-layout and post-layout simulation waveforms.

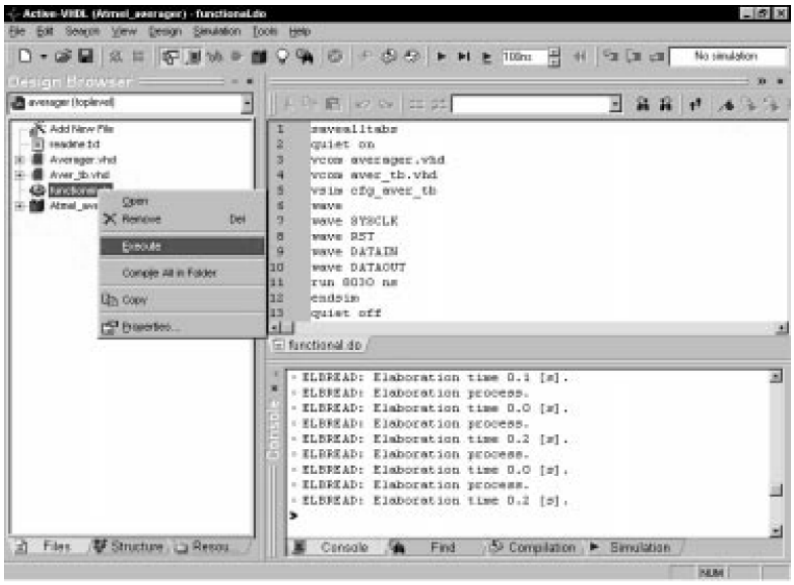
The Test Bench Generator Wizard may then be invoked by selecting the “Generate Test Bench…” option from the Tools drop down menu.

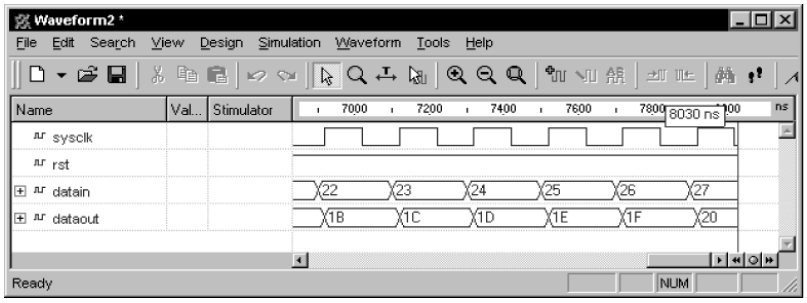


For this guided tour we have provided the testbench aver_tb.vhd. Select “Cancel” from any screen within the wizard to exit and proceed.

The Test Bench Generator Wizard is capable of providing not only the test bench source code, but also an accompanying configuration file (.vhd) and an executable macro (script) file (.do).

Right-click on the “functional.do” macro and select the “Execute” option from the context-sensitive menu. This macro (1) compiles all source files including the test bench, (2) initializes simulation, (3) adds the UUT (unit-under-test) ports to the waveform window, and (4) runs the simulation for a specified simulation period.



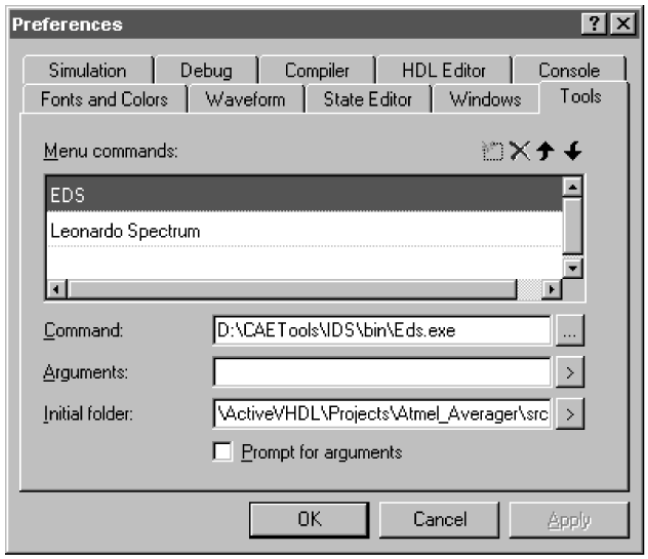


Synthesis

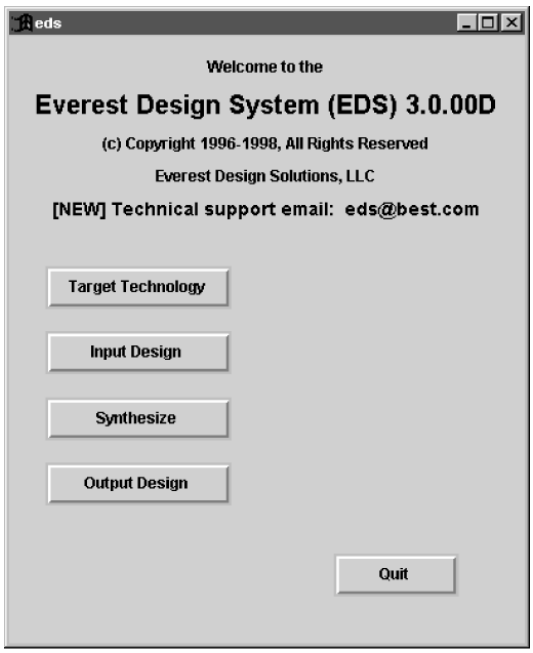
Synthesis may be performed by any tool that supports Atmel's AT6K and/or AT40K technology. EDS is introduced below in its GUI format. Proven designs can be synthesized by writing macros (.do) that launch the tools in a batch mode. Please note that this sample design was written with constructs verified against Everest only. Performing this guided tour with other synthesis tools may result in errors.

Everest

Active-VHDL's Tools drop down menu can be setup with an option to EDS (Everest Design System). Select the "Preferences…" option from the Tools drop down menu. Provide a name for your menu command and then provide the required command line and initial folder information as per your installation of Atmel's IDS. On the command line, type in the correct path for the EDS execution file and provide the correct path for the Atmel_Averager file. Click "OK" and you will find your new menu command under the Tools drop down menu. Launch EDS by selecting the new EDS option from the Tools drop down menu.



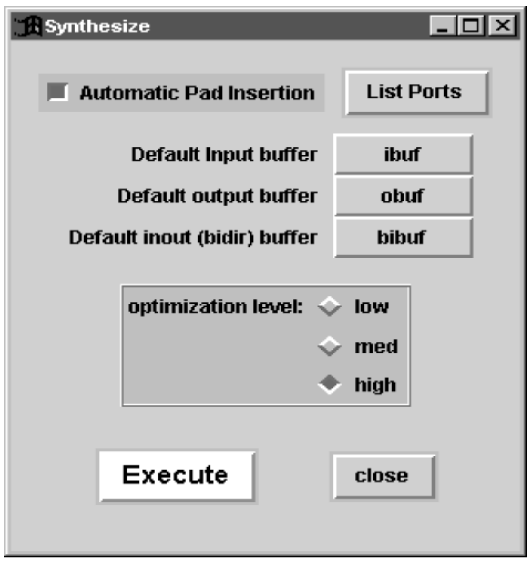
Upon entry into EDS, you will be presented with the following flow:



Click on “Target Technology” and open at40k.lib.

Close the resulting Log Output window and click on “Input Design.” Open the averager.vhd source file.

Close the resulting Log Output window and click on “Synthesize”. Select the options as shown below:

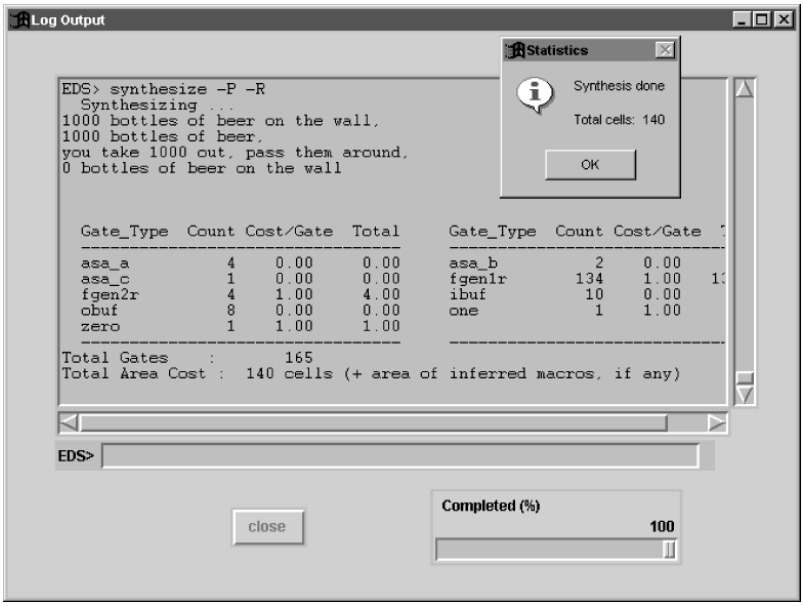


Click on “Execute” and close the resulting Statistics and Log Output dialog windows.

Click on “Output Design” and save to Averager_eds.edf. Note that the black boxes result from modules generated during synthesis as opposed to structural VHDL embedded within the source file. These black boxes are replaced by the macros generated during netlist import to Atmel’s IDS software. A full listing of macros to be resolved is located in Averager_eds.mgi. This file is written during the Output design stage.

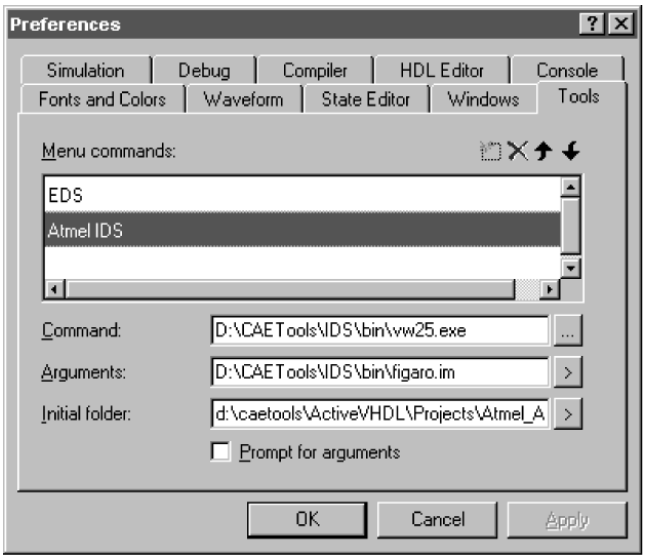
The EDS command line in the Log Output window can be used to apply advanced features such as reading in multiple design files and writing post-synthesis HDL for simulation.

Close the Log Output window and click on “Quit”.

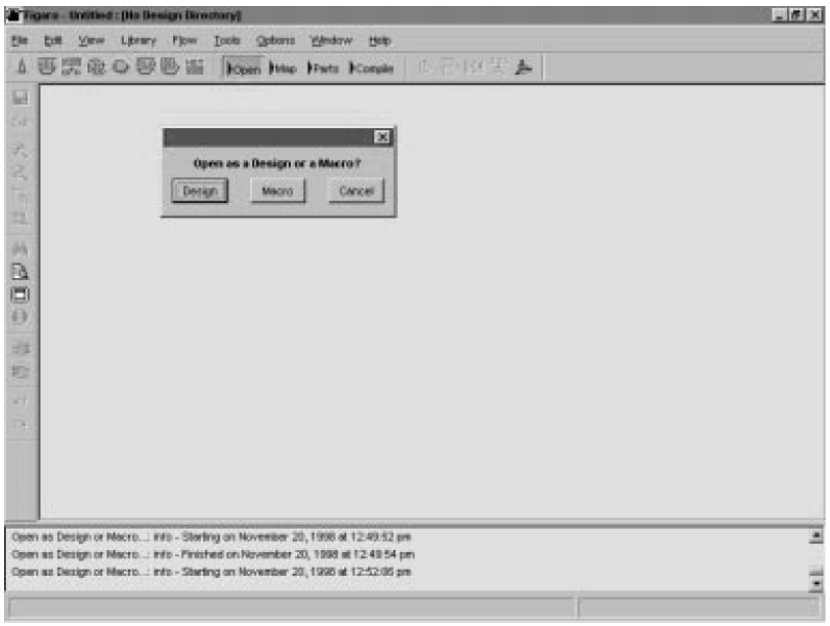


Implementation with Atmel's IDS

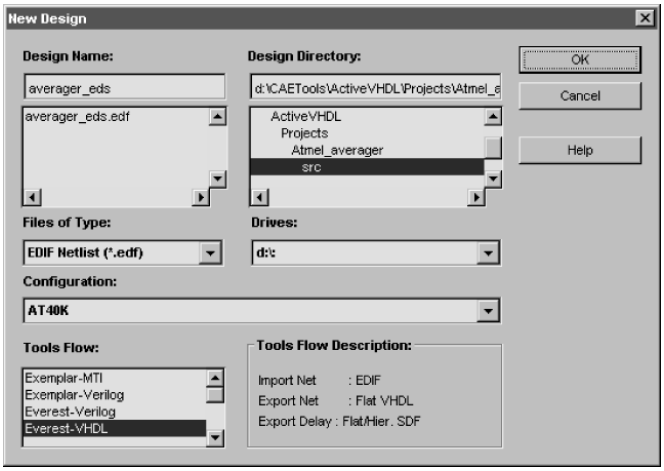
Active-VHDL's Tools drop down menu can be set up with an option to Atmel's IDS. Select the "Preferences…" option from the Tools drop down menu. Provide a name for your menu command and then provide the required command line and initial folder information as per your installation of IDS. On the command line, type in the correct path to execute Figaro and provide the correct path for the Atmel_Averager file. Click "OK" and you will find your new menu command under the Tools drop down menu. Launch IDS by selecting the new Atmel IDS option from the Tools drop down menu.



Given the IDS desktop, click on OPEN followed by Design.

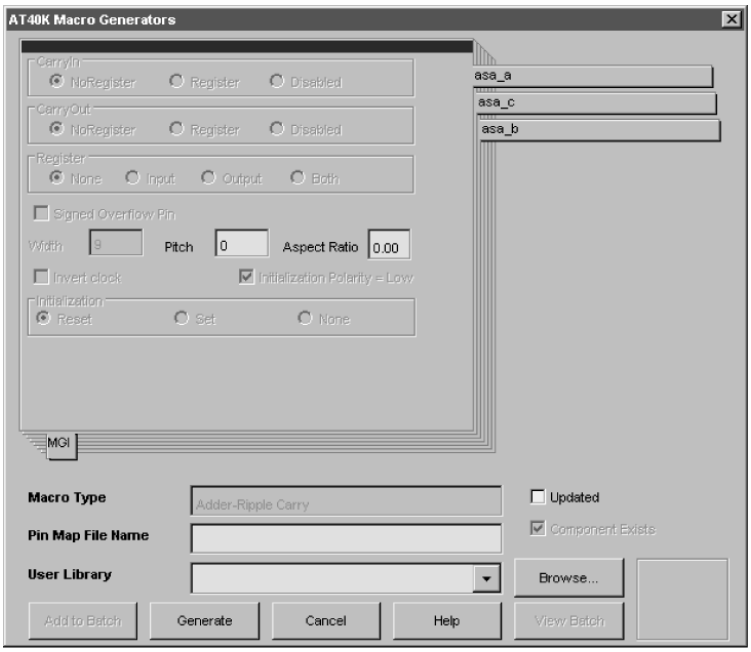


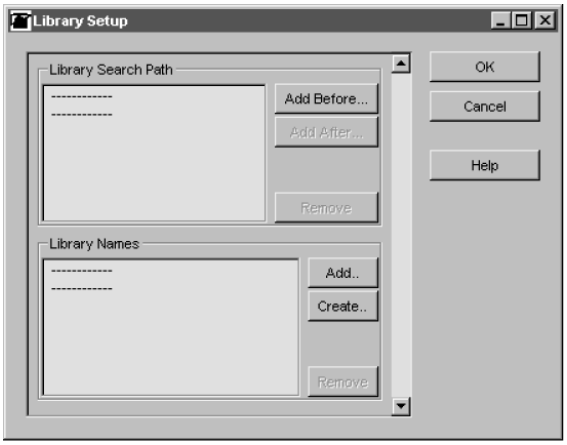
Within the Design Setup dialog window, select “New Design” and enter the information for the Averager design.



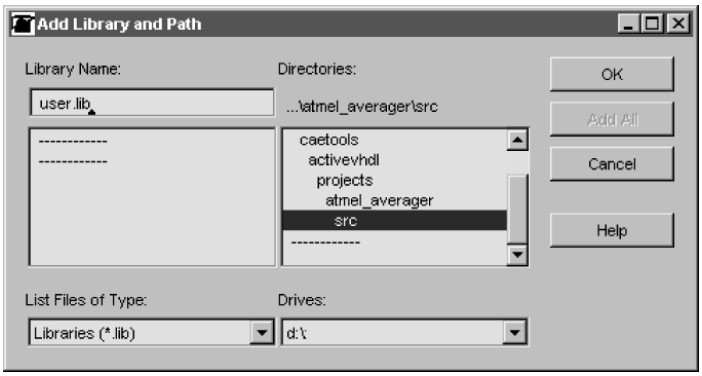
Click “OK” until you return to the IDS desktop.

When presented with the AT40K Macro Generators dialog box, click on “Browse” to set up a design library.



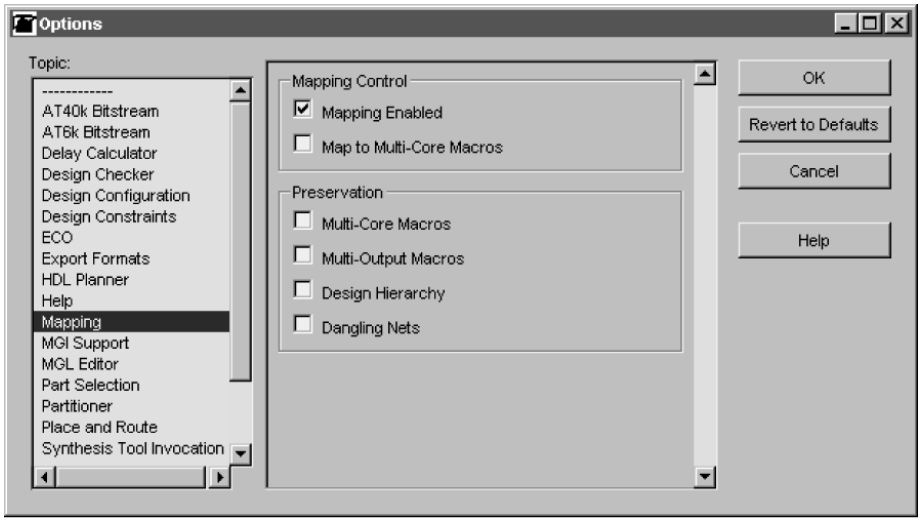


Click on “Add Before…” to create a new library as shown below:



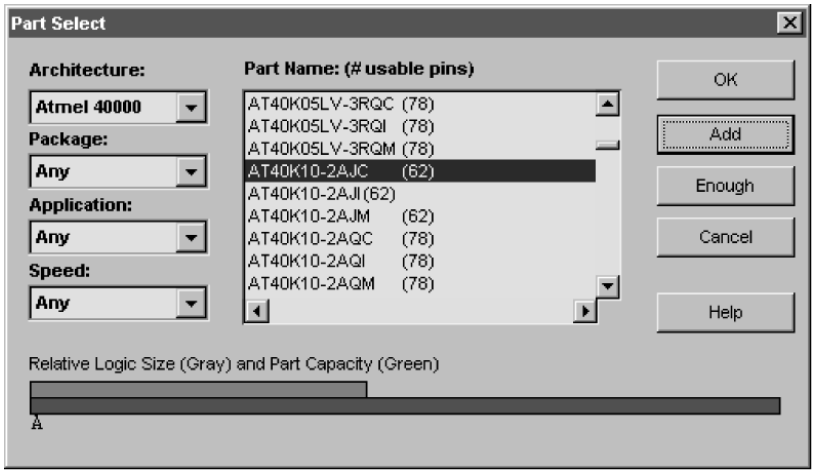
Click “OK” and “Yes” in succession until you return to the AT40K Macro Generators dialog window. Back in the AT40K Macro Generators dialog, the new library will be listed as the User Library.

Click on “Generate”. Modules identified during synthesis are now generated and stored within the new library. Upon completion of this task, click “Cancel” to close the AT40K Macro Generators dialog window. IDS will display a Design Browser window. If the Map flow button (next to Open) is disabled (on the IDS desktop), select the “Options” option from the Options drop down menu and configure the Mapping settings as indicated below.



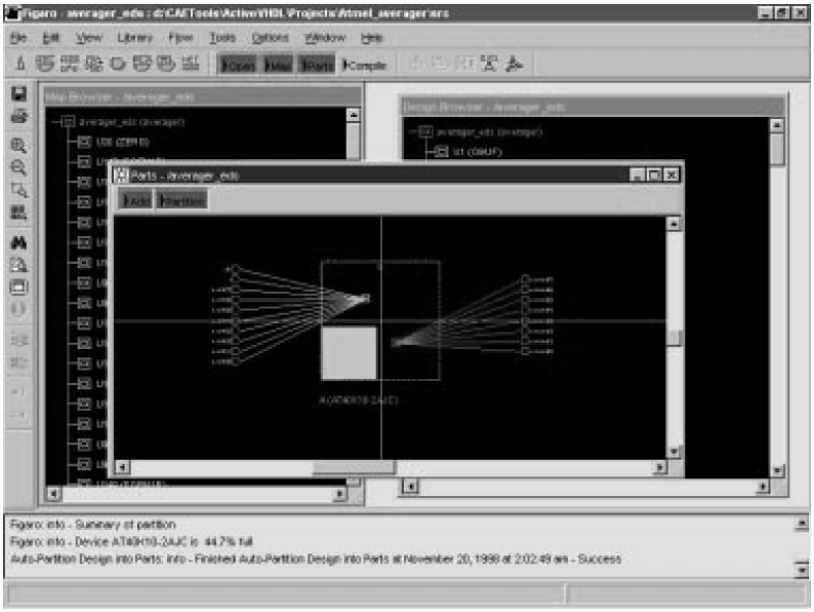
Click “OK” to exit back to the IDS desktop and continue the implementation flow by clicking on “Map”. A Map Browser results, typically with a more optimal set of pad and cell instances than that in the original Design Browser. In general, combination logic is mapped to the LUTs embedded in the core cells (FGENs) and some unregistered low input count functions may be combined into other existing core cells.

Next, click on the “Parts” flow button and select a device.



In this example, the part capacity clearly exceeds the amount of logic to be placed. You may add one such part by clicking on the “Add” button. Alternatively, you may click on the “Enough” button which will effectively add as many devices of the part type selected as needed to meet the required logic, RAM and IO counts.

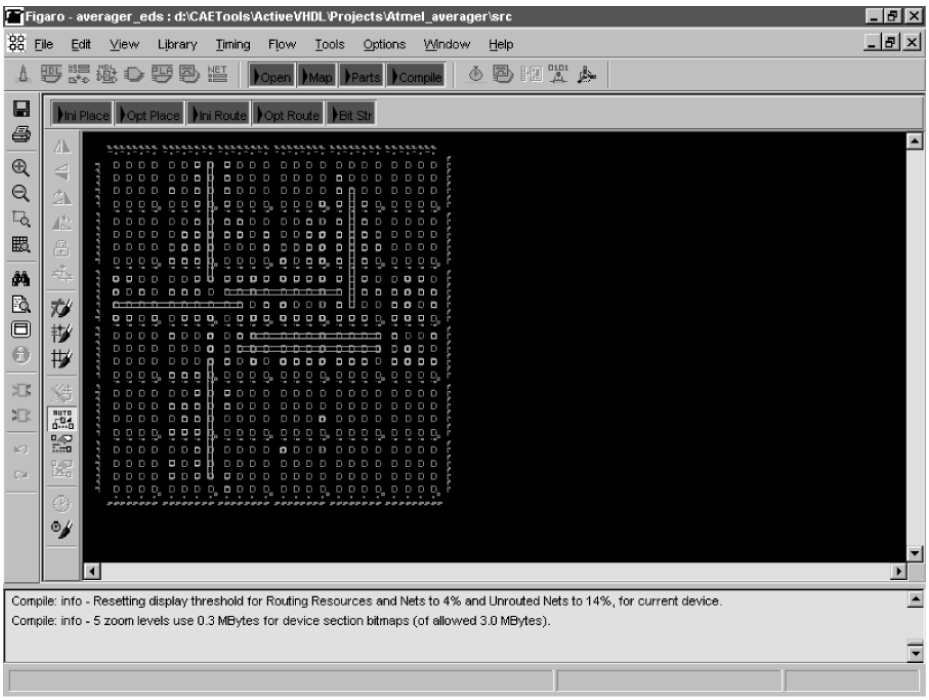
After adding one or more parts, click “OK” to exit. Then click on the “Partition” button in the Parts window. With the Parts window active, the user may assign IO pad attributes and pin assignments from the Edit drop down menu.



Before proceeding, visually inspect the Parts window. Does the number of devices make sense? Each outline represents one FPGA. Does the pink utilization bar fall outside the device outline (i.e. exceed 100% utilization for the given device)?

In the final stage of implementation, click the “Compile” button on the IDS desktop. The status of the operation is continually updated in the console section of the IDS desktop.

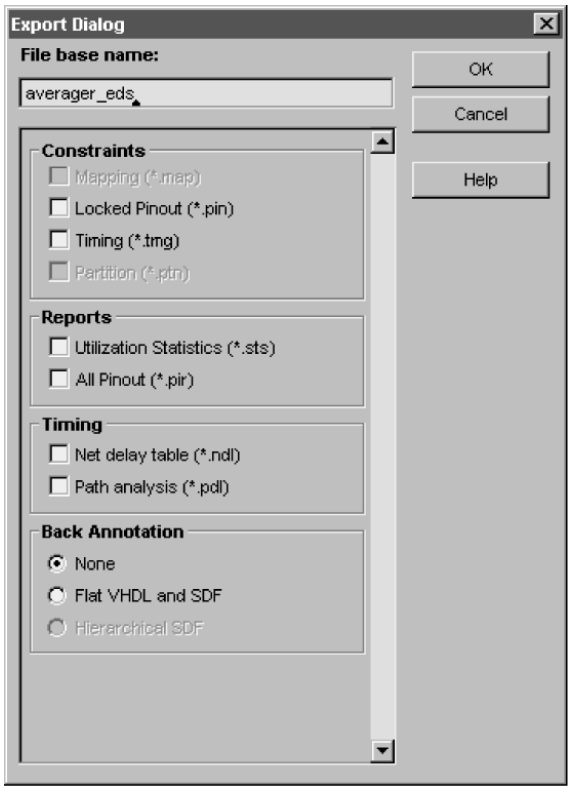
Upon completion, double-click on the device shown in the Parts window to open a new Compile Window. This window activates many of the editing and search features that would assist the user in floorplanning and resolving contentions. Powerful zoom in and out features provide excellent resolution on logic cells, RAMs, IO pads and routing resources.



Timing constraints can be applied with the Compile window active both before compilation (for the non-default option of timing-driven optimization) or after compilation (to force a path delay analysis). Where compilation fails, an option exists to force the place and route tools to try more algorithms. The default effort level is 2 out of 5 where 5 represents the maximum effort level.

If configuration fails during hardware verification, the user is advised to carefully review the bitstreaming options available under Options -> Options -> AT40K Bitstream. Refer also to the “AT40K Configuration Guide” application note.

Of final interest would be the outputs of the implementation session. Selecting the “Export…” option from the File drop down menu provides the following reports:



Please note that back annotated files are stored in the \figba subdirectory.

Some additional files automatically created from saving the design and bitstreaming are:

Averager_eds.fgd: IDS design database including placement, routing, desktop info.

Averager_eds.rct: Repeat constraints file.

Averager_eds.hxr: Intel Hex bitstream file for use in industry standard programmers.

Averager_eds.bst: Tabular ASCII-binary bitstream file for use with IDS, CPS and DOWNLD40.exe; (PC platform only).

Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe
Atmel U.K., Ltd.
Coliseum Business Centre
Riverside Way
Camberley, Surrey GU15 3YL
England
TEL (44) 1276-686-677
FAX (44) 1276-686-697

Asia
Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan
Atmel Japan K.K.
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Atmel Colorado Springs
1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Rousset
Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-4253-6000
FAX (33) 4-4253-6001

Fax-on-Demand
North America:
1-(800) 292-8635
International:
1-(408) 441-0732

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

BBS
1-(408) 436-4309

Rev. 1445A–02/00