Catapult C Lab #1

Catapult C Workflow

Introduction

This lab goes through the typical steps taken in a Catapult design session. This includes setting working directories, adding input files, setting the target technology, and generating RTL and output files. Design optimizations are not covered in this lab, but will be covered later. The main goal in this lab is for the user to familiarize themselves with the Catapult GUI and the steps taken in synthesizing a C++ design.

Workflow

Within Catapult C, the primary entry tool for constraints is the Design Bar. The Design Bar contains links to all the valid constraint managers for the design. When you start the tool, only the working directory, design setup, and input files can be specified.

Setting the working directory

You may set the working directory manually each time you invoke the tool or have Catapult C automatically set it for you.

- ✓ To manually set the directory click on the “Set Working Directory” icon in the design bar and navigate to the lab1 directory (Figure 1).

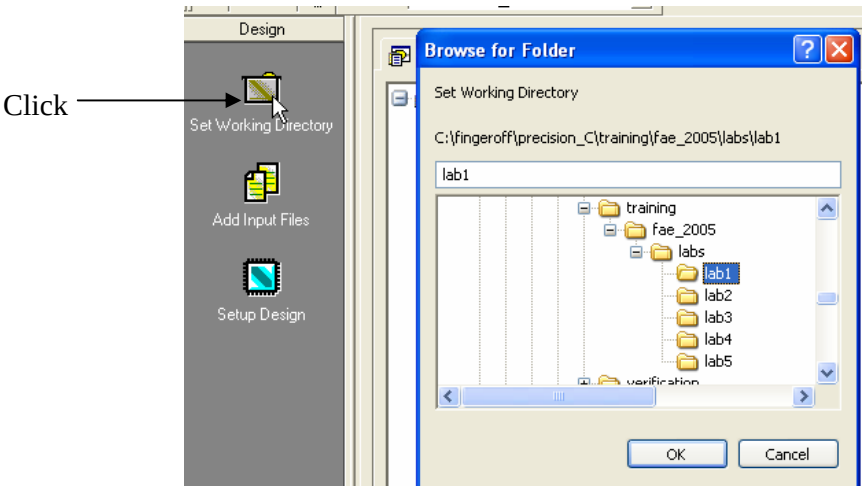

Figure 1 Setting the working directory

Setting Synthesis Options

Catapult allows you to set global options such as the default working directory, compile include paths, input and output file formats, etc.

✓ Select tools->set options->output and make sure that the VHDL and Verilog options are checked. This will instruct Catapult to output VHDL and Verilog netlists (Figure 1.a).

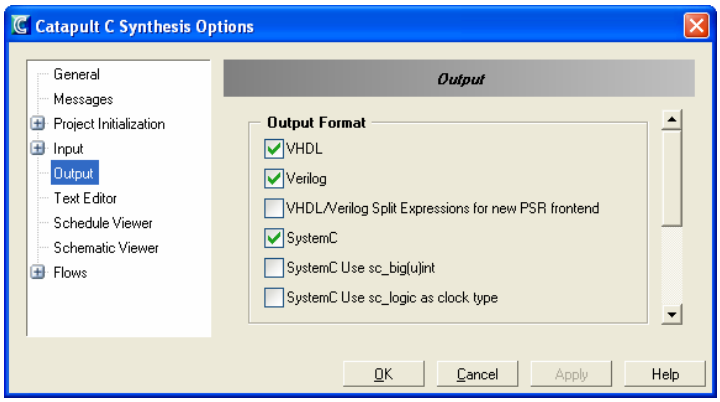

Figure 1.a Synthesis Options

✓ Click OK

✓ Select tools->save options to save the project settings

Adding Input Files

✓ Click on the “Add Input Files” icon in the design bar and add the lab1.cpp file (Figure 2).

Input files can also be added by double clicking on the input files folder in the Design Center. It is also possible to add any of the Catapult C examples that ship with the install by right clicking on the input files folder and selecting “Add Input Files From Examples”.

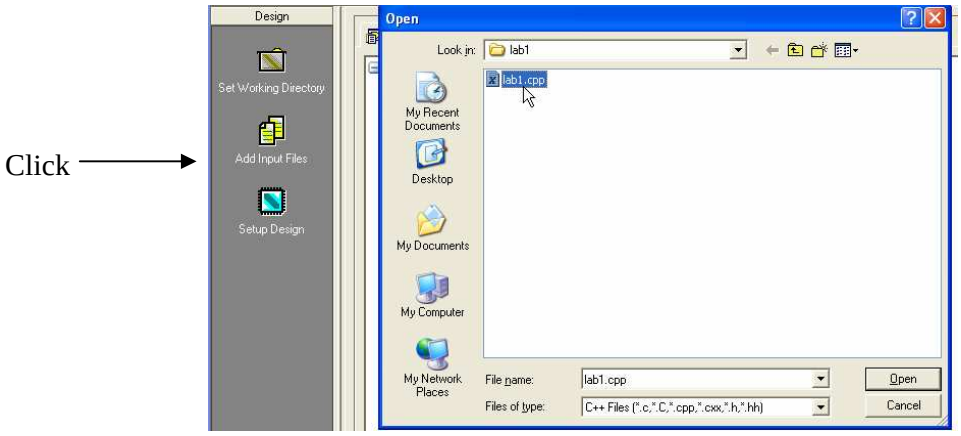

Figure 2 Adding input files

Setup Design

Clicking on the Setup Design button brings up a window that allows control of the technology and clock speed. The technology is made up of the basic library and added IP that you can make available to the tool by checking the box for that IP. Catapult C can automatically take advantage of these IP blocks, so this is one of the primary constraints for the hardware design. The clock speed is made up of two parts. The Design frequency is the target frequency of your hardware. The overhead is part of the clock period that is preserved for sharing

✓ Click on “Setup Design” in the design bar and set the technology to Altera Stratix II (Default part and speed grade). Set the clock frequency to 100MHz (Figure 3).

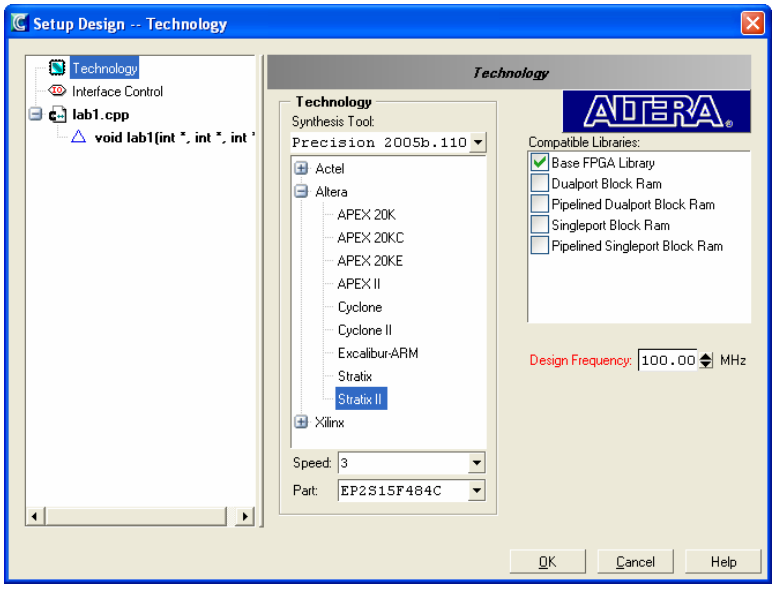

Figure 3 Setting the Technology

✓ Click on the interface Control icon in the Setup Design dialog box and check the “Start Flag Name” and “Done Flag Name” check boxes. This will instruct Catapult to insert the Start/Done process level handshake into the generated RTL (Figure 4).

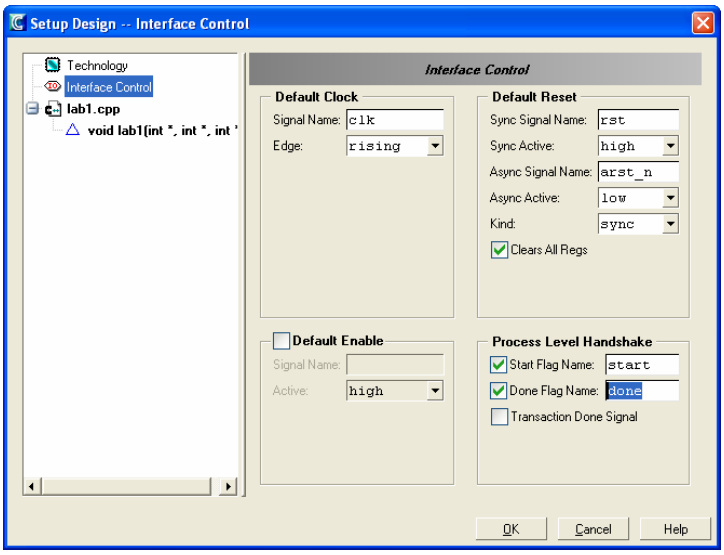

Figure 4 Adding Start/Done Handshaking

✓ Click OK

Architectural Constraints

This is covered in future labs.

Resource Constraints

This is covered in future labs.

Schedule

✓ Click on the “Schedule” icon in the design bar. This will bring up the Gantt chart, which is the main tool for algorithm analysis and is like the schematic viewer for the algorithm. It provides information on loops, components, slack, and control steps (C-steps) (Figure 5).

Information shown in the following Gantt chart is described below.

- The loop hierarchy is shown on the left hand side
- The C-Steps are roughly equivalent to states in a state machine.
- The White area represents one clock period and is to scale with respect to other operations on the Gantt chart
- The Blue boxes are the delay associated with operations.
- The red bars represent the slack for that operation in the algorithm. This is useful for finding the operations in the algorithm that are the most critical (in this case, the multiplier).
- The green bars on the right hand side indicate the individual loop execution times and are very useful for identifying bottlenecks in the algorithm.

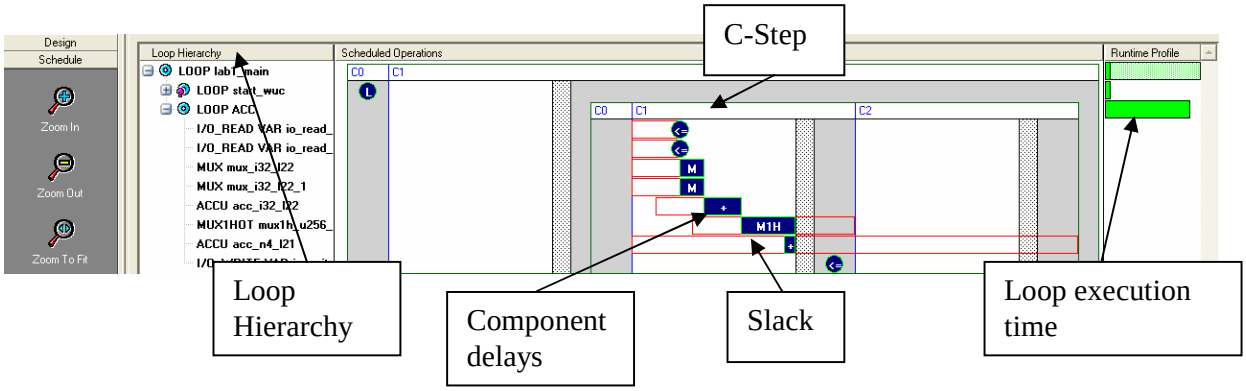

Figure 5 Gantt Chart

The Gantt chart can be used to cross-probe back to the C++ source code.

✓ Double-click on “LOOP ACC” in the loop hierarchy window. This will bring up the text editor and high light the related section of code (Figure 6).

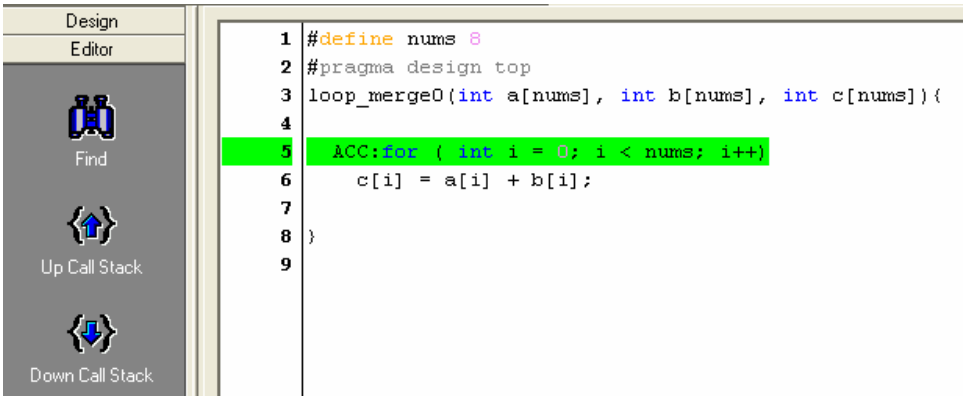

Figure 6 Cross-probing to the C++ Source

✓ GO back and examine the Gantt chart. Notice that there are three loops. The first loop “LOOP lab1_main” is the main loop and corresponds to the function call. The third loop is the “LOOP ACC” loop. However, there is another loop called “LOOP start_wuc” which does not correspond to anything in the C++ source code. This loop is inserted by Catapult and represents the START/DONE process level handshake.

✓ Spend a little time examining and cross-probing the Gantt chart.

Generate RTL

Click on the “Generate RTL” icon in the design bar. Catapult will generate a number of output files, reports, and schematics and place them in the output files folder (Figure 7).

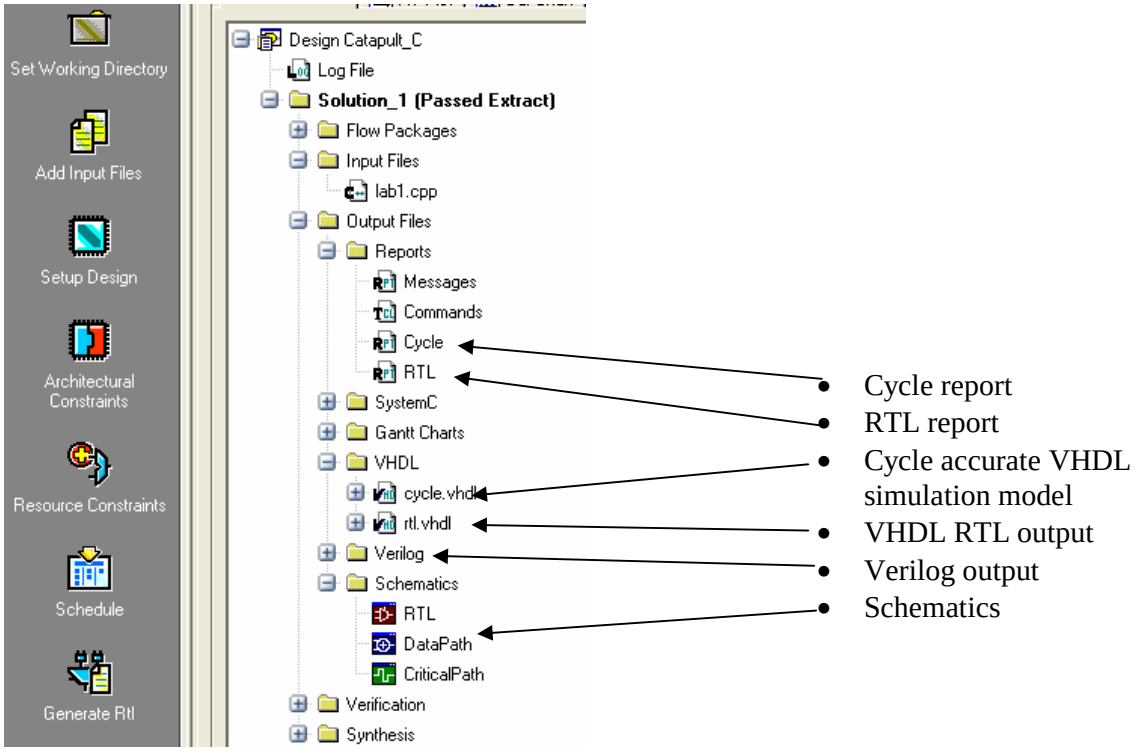

Figure 7 Output Files Folder

Reports

The Catapult C Synthesis tool generates two reports automatically for every solution.

- The Cycle report (*cycle.rpt*) is a high-level architecture/algorithm report including information on loop iterations, latency, and throughput.
- The RTL report (*rtl.rpt*) is similar to reports seen from RTL synthesis tools. The estimates for clock period and area are typically conservative and within about 20% of the area reported by RTL synthesis.

✓ Open the RTL report by double-clicking on it in the output files folder. Click on the "Bill of Materials" section to expand it. This shows a list of the types of components used to schedule the algorithm, as well as the number of components used and the area of each component.

✓ Expand the "Timing Report" section by clicking on it. This shows the critical timing path and worst-case slack, which you should see, is positive indicating that Catapult has met timing.

Cross-Probing

Catapult C Synthesis makes it easy to move between the Catapult C Synthesis GUI and the C code, and the C code and any design view. When you see **bold** text in Catapult C, you

can “cross probe” back to the C source. Double-click, or right-click and select to cross probe back to the C code. You can cross probe from the transcript, all constraint windows, the Gantt chart, all generated HDL, all reports, and the schematic.

- ✓ Double click on the data path schematic icon in the output files folder.
- ✓ Then push down into the design hierarchy by double clicking on the lab1_lab1_proc block. Note: you must click in the white space.
- ✓ Right-click on the “+” operator highlighted in light blue and select “Trace to source” (Figures 8, 9, and 10)

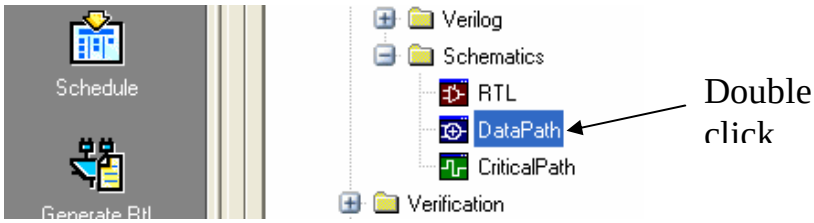

Figure 8 Data Path Schematic Icon

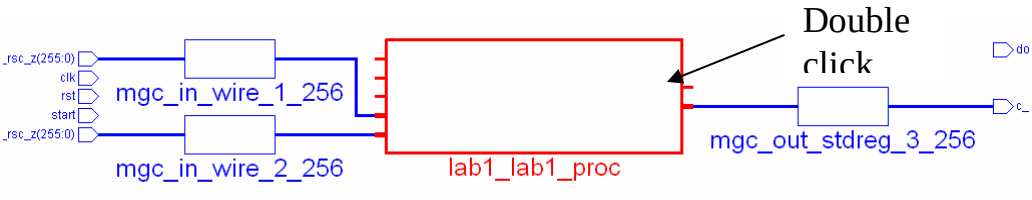

Figure 9 Top-level Data Path Schematic

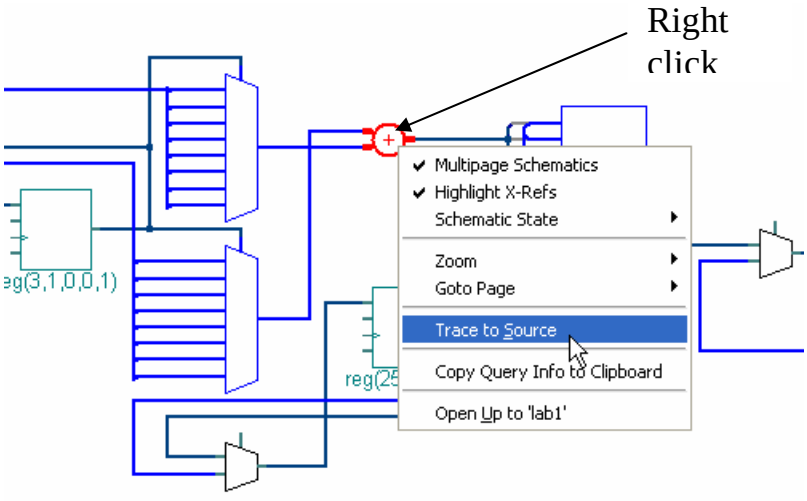

Figure 10 Cross-probing from Data Path Schematic

- ✓ Double-click on the “+” in the C source code and select the “Schedule” icon. Click OK. Note how this forward crossprobes to the Gantt chart (Figure 10a).

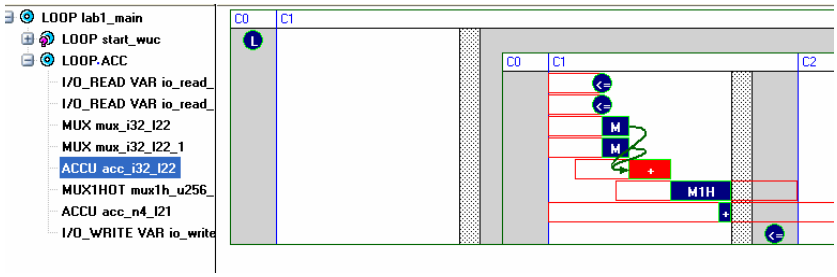

Figure 10a Forward Cross-probing

- ✓ Try cross probing from some of the other files in the output files folder (e.g. cycle report, RTL report, etc).

Transcript

The Catapult C transcript provides information on the various stages of compilations, optimization, and synthesis. It also provides warning and error messages when compilation or optimization fails. These messages can be cross-probed back to the C++ by double-clicking on the bold messages on the left. Further help information can be obtained by double-clicking on the bold help numbers on the right (Figure 11).

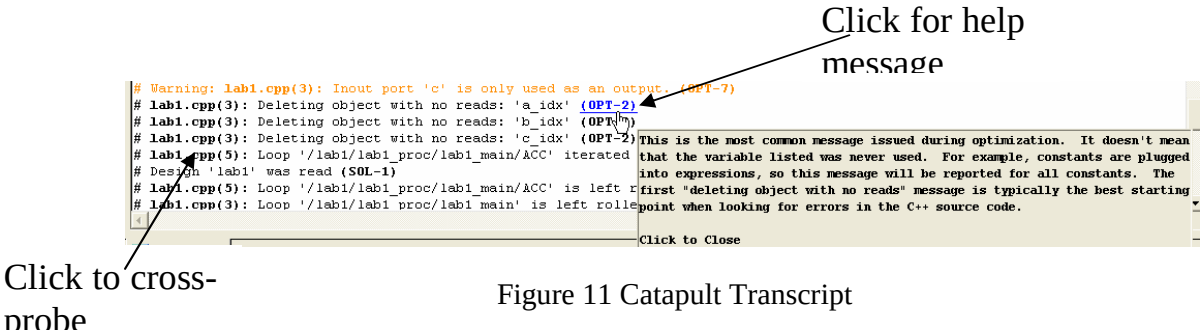

Figure 11 Catapult Transcript

- ✓ Scroll up in the transcript window and click on the bold messages on left side to cross-probe to the source code

- ✓ Click on the (OPT-2) optimization message (Figure 11)

Getting Help

In addition to getting help in the transcript, Catapult has both online help as well as extensive help available in the form of user and reference manuals, C++ style guides, and C++ to hardware concepts documentation. The online help is available in the Setup Design and Architectural constraints views by clicking on the help button. More extensive help documentation can be found in the manuals bookcase under the help menu (Figures 12 and 13).

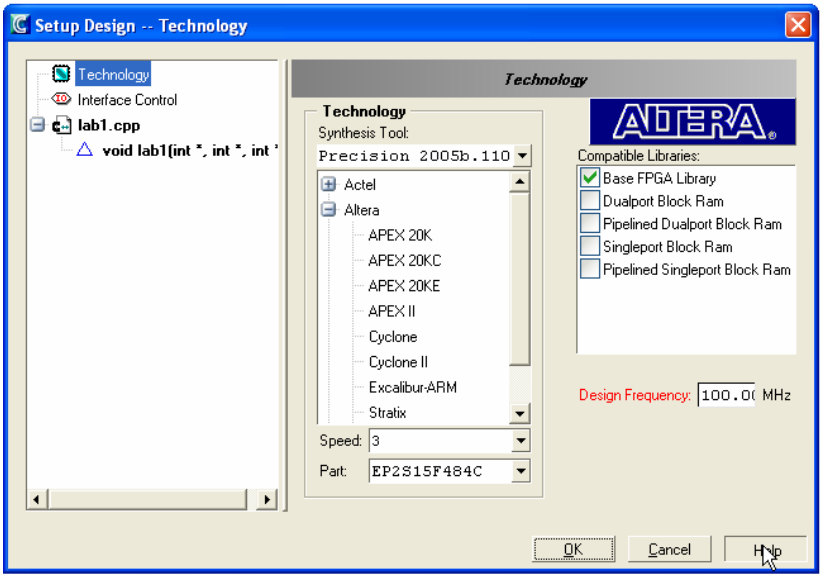

Figure 12 Online Help

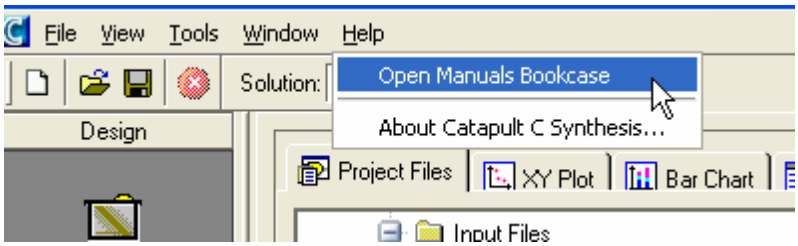

Figure 13 Help Manuals

✓ Close all windows and select File->new project

END