CprE 588
Embedded Computer Systems

Prof. Joseph Zambreno
Department of Electrical and Computer Engineering
Iowa State University

Lecture #7 – System-Level Design Tools

Introduction

- The following slides provide an introduction to the SCE tool.
- This is, by no means, a complete guide to the tool.
- More information is found on the class website (under the Resources section).

S. Abdi et al., *System-on-Chip Environment (SCE) Tutorial, Version 2.2.0 Beta*, Center for Embedded Systems, University of California-Irvine, 2003.

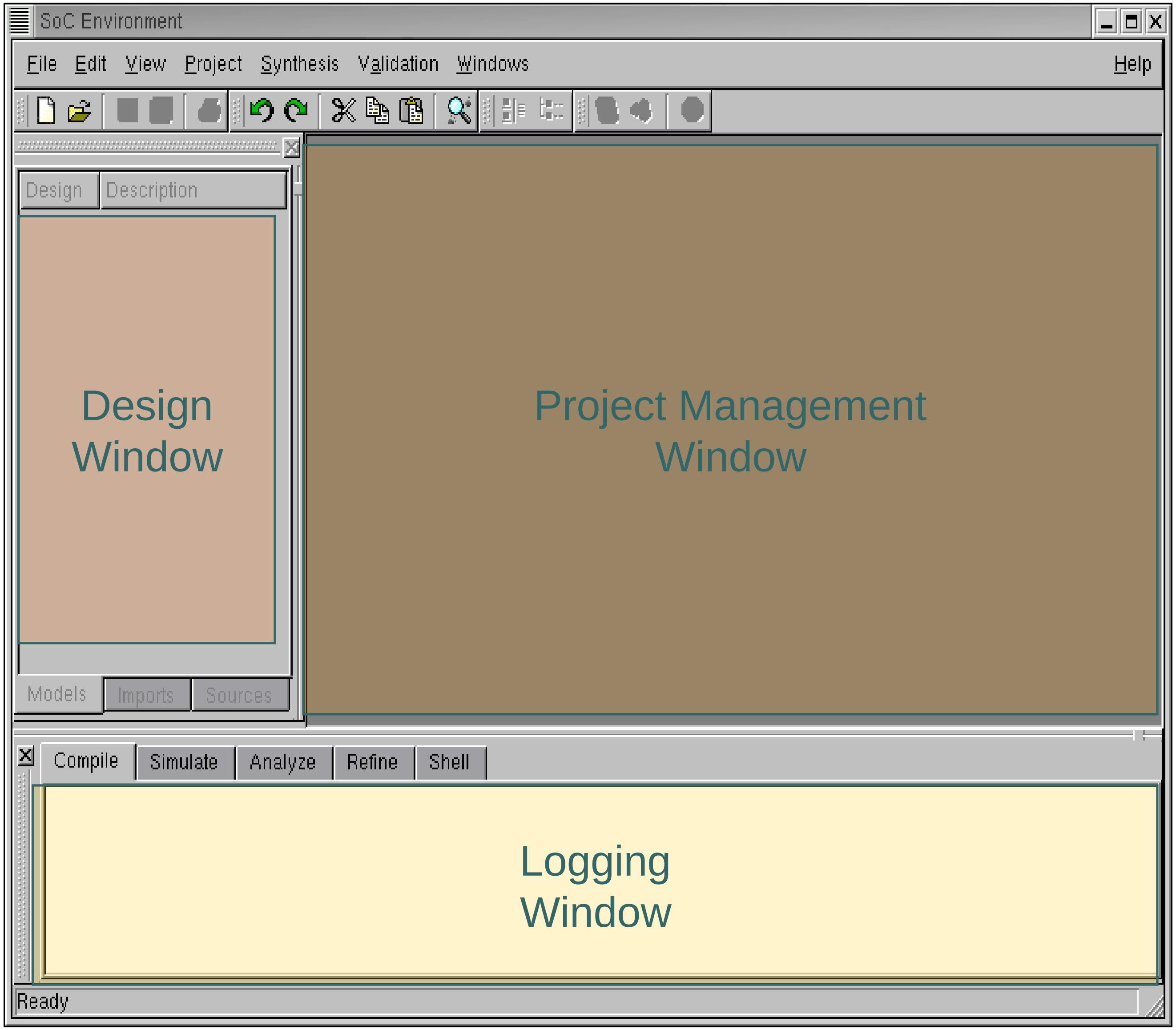
SCE Tool Setup

- The SCE Tool is only available on `systemc.ece.iastate.edu`
- Once you log into this machine, you must setup your account to use the SCE tool.
- This is a graphical tool, therefore you need to setup X11 forwarding. Instructions are found on the class website (link).
- To setup your account you must run the script “/opt/sce-200806061/bin/setup.sh”.
  - You do this by typing: `source /opt/sce-20080601/bin/setup.sh`
- To start the tool type ‘`sce &`’ at the command prompt. After a few moments, the SCE Main Window will appear.

Disclaimer

- At this point we need to clarify that the SCE tool is meant for simulation, profiling, and refinement of your design; and not for developing your first model.
- To use SCE you must have a correct SpecC model of your design beforehand. To write the SpecC code you can use any text editor, such as Emacs or Notepad.
- Knowing this you will use the parity example for the rest of this intro, with a few exceptions.
    - Download the file extras.zip and extract its contents to the parity example folder. This file contains two files:
        - io.sc: you will replace the original io.sc with this version.
        - test-vector.in: this file has a sequence of inputs for your model and is used for simulation.
    - While in the parity example folder (specification model) launch the SCE tool.
    - When you do this a window like the one on the next slide will come up.

SCE Main Window

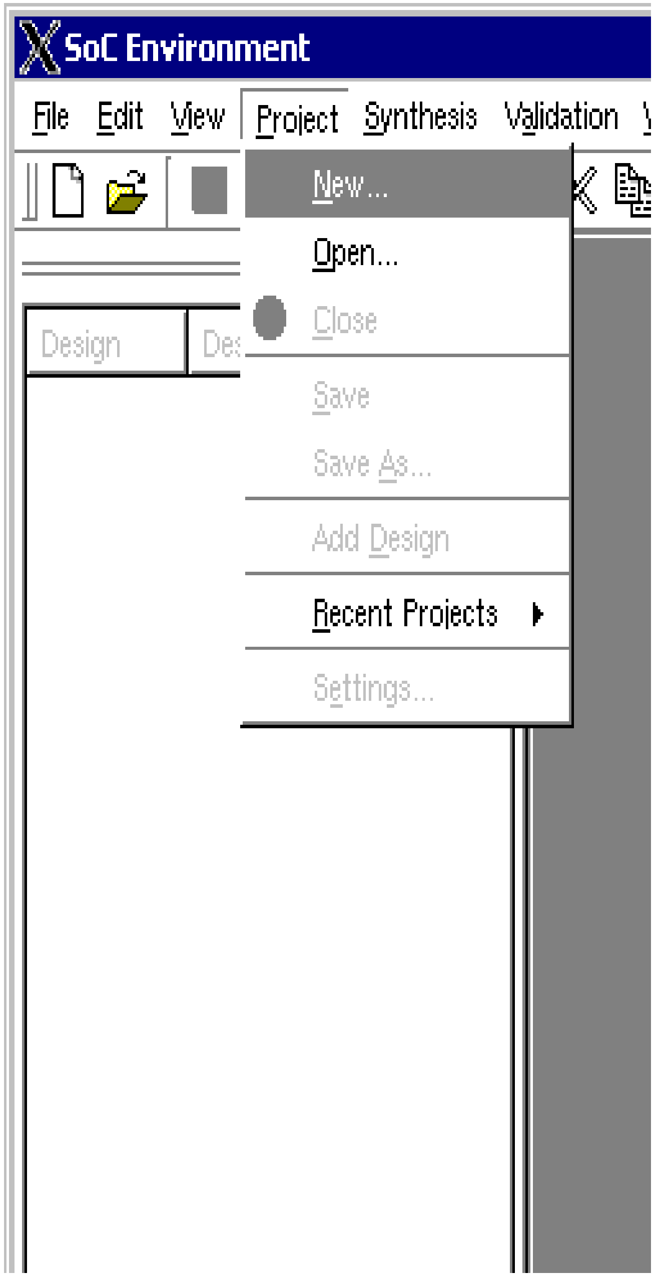

SoC Environment
File Edit View Project Synthesis Validation Windows Help
Design Description
Design Window
Models Imports Sources
Project Management Window
Compile Simulate Analyze Refine Shell
Logging Window
Ready

SCE Main Window (cont.)

- Project Management Window
  - Maintains the set of models in the open project(s).
  - Becomes active once a project is opened and a design is added.
- Design Management Window
  - Shows the hierarchy tree for the design.
  - Maintain various statistics for the design.
- Logging Window
  - Keep logs of compilation simulation analysis and refinement of models.

New Project

- Now you will a create a new project.
- Next you will setup your project and then import the parity model into this project.

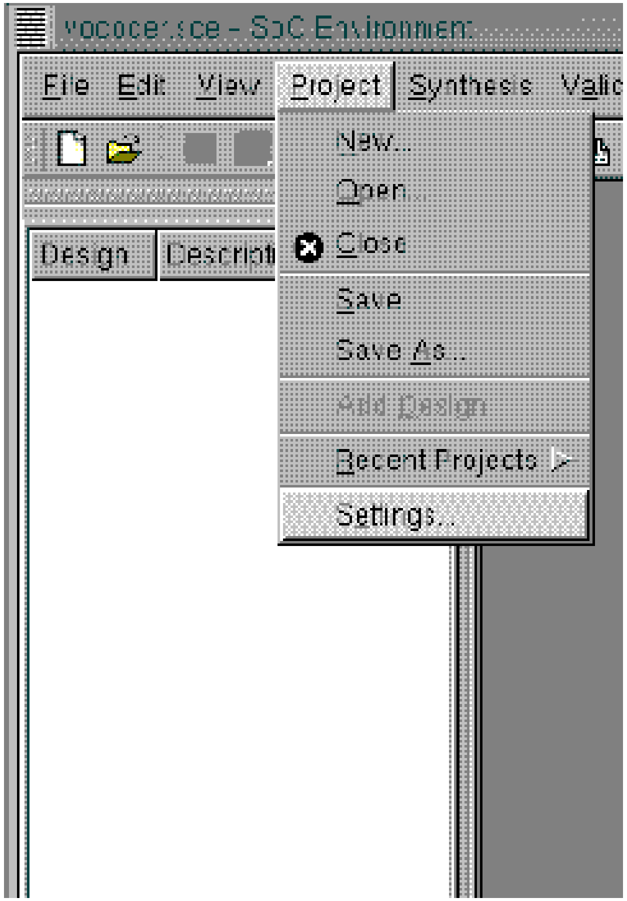


The SCE Project

- A project is associated with every design process.
- The project is used by the designer to customize the environment for a particular design process.
- Next you will look at some of the project settings

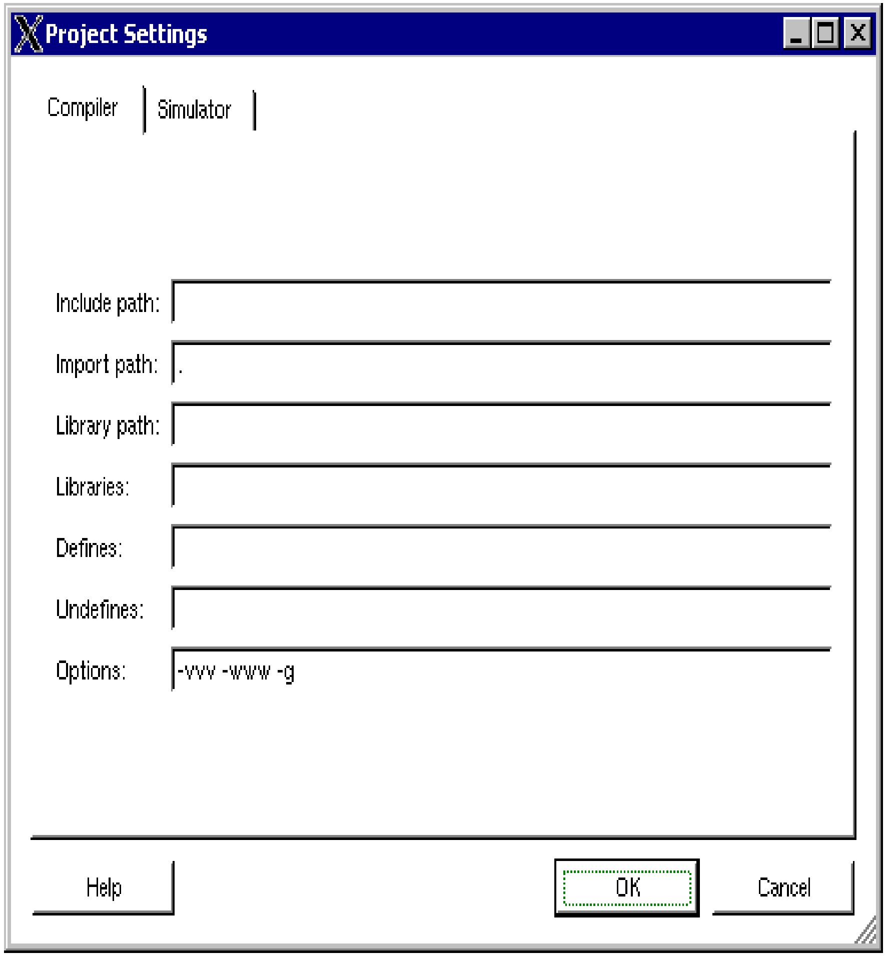
Project Settings – Compiler

- To see the project settings select:
  - ‘`Project -> Settings`’ on the top menu bar.
    - Two tabs: Compiler, for the SpecC compiler options, and Simulator, for the SpecC simulator options.
  - Include Path:
    - This is the path that is searched for header files.
  - Import Path:
    - This path is searched for files imported into the model.
  - Library Path:
    - Search path for C/C++ libraries for compilation.
  - Libraries:
    - Which libraries to link against.
  - Defines / Undefines:
    - Include here the macros to define or undefine.
  - Options:
    - Include here the options to pass to the SpecC compiler.



Project Settings – Compiler (cont.)

- You will edit the compiler settings to match the figure on the right.
  - Import path: .
  - Options: -vvv –www -g



Include path:

Import path: .

Library path:

Libraries:

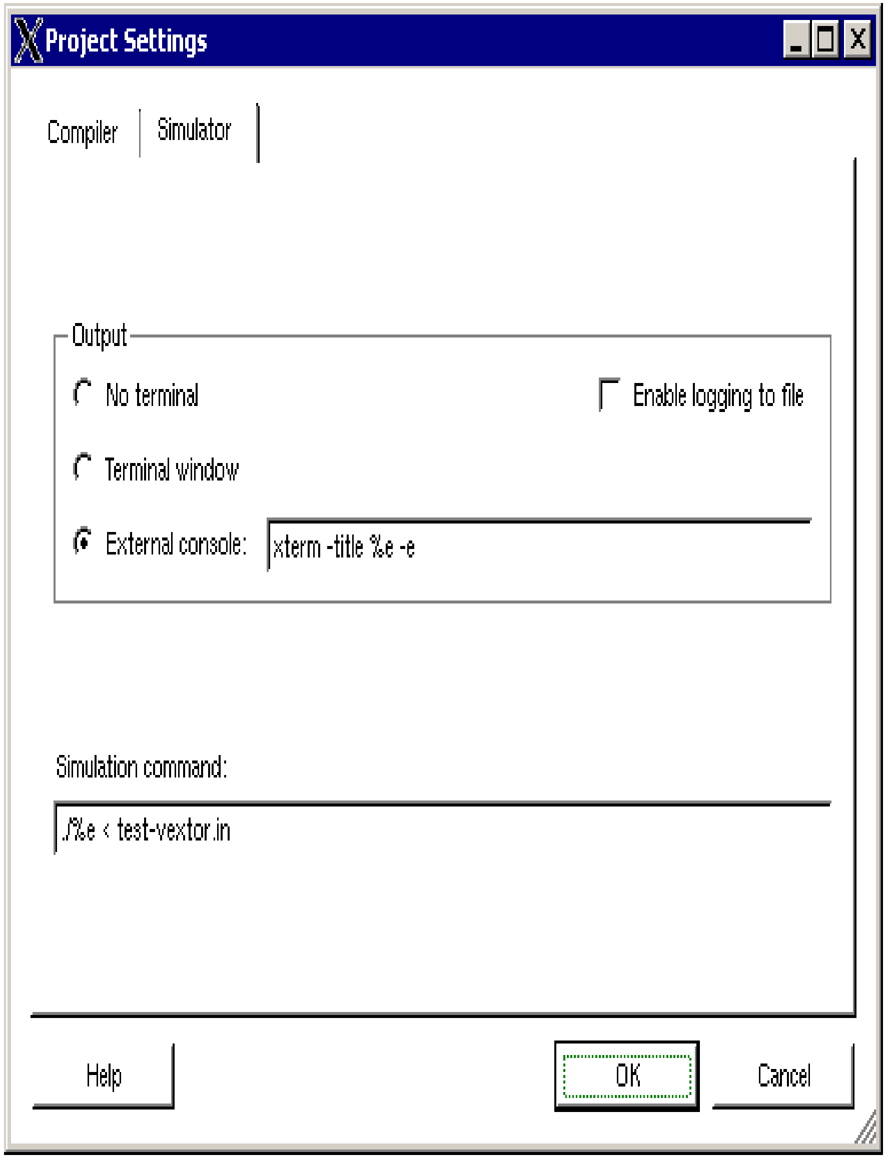
Defines:

Undefines:

Options: -vvv -www -g

Project Settings – Simulator

- SCE simulates your design by executing the command on ‘Simulation command’.
  - ./%e < test-vector.in
- Edit your settings to match the ones shown here.
- Remember that ‘test-vector.in’ is one of the files included in extras.zip



Output

○ No terminal

□ Enable logging to file

○ Terminal window

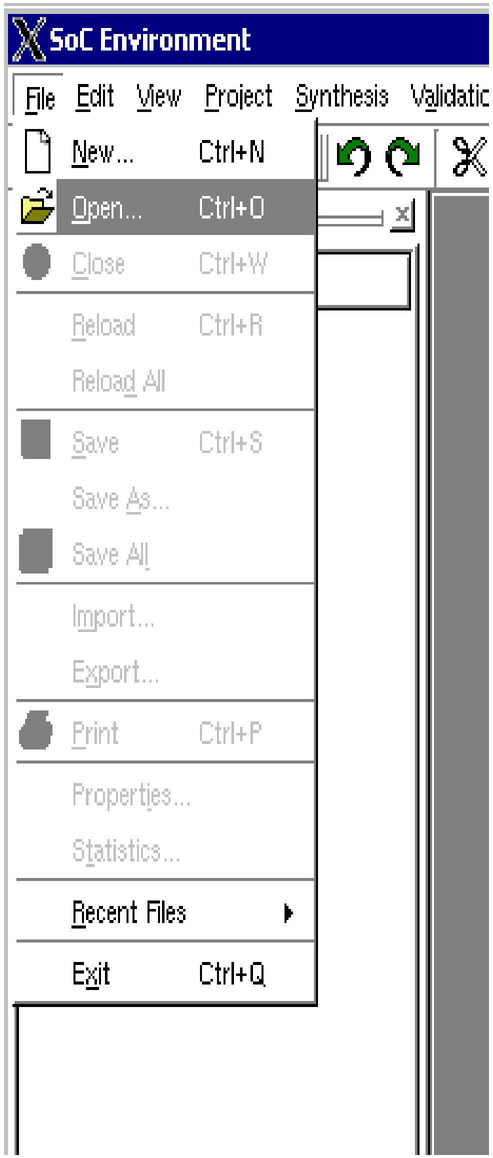
● External console: xterm -title %e -e

Simulation command:

./%e < test-vextor.in

Associate the Parity Model

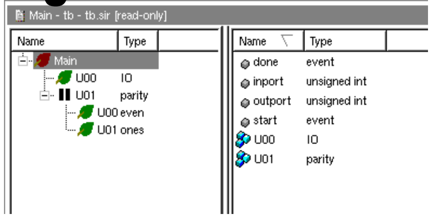
- By now you probably understand that the project keeps the compilation and simulation setting for your model, but not the model.
- To associate a model to the current project you will open the top file of your model, in this case tb.sc



Associate the Parity Model (cont.)

- As you can see in the “Project Management Window” tb.sc is composed of two behaviors:
  - IO: takes care of the I/O, but is not part of the model.
  - Parity: top behavior of the parity model.
- The final step to associate a model is to add it to the design.
- When you add the design you will see the name: tb.sir on the “Design Window”. This is the name given by SCE to your model. You may change it to something more meaningful like ParitySpec.

Management Window

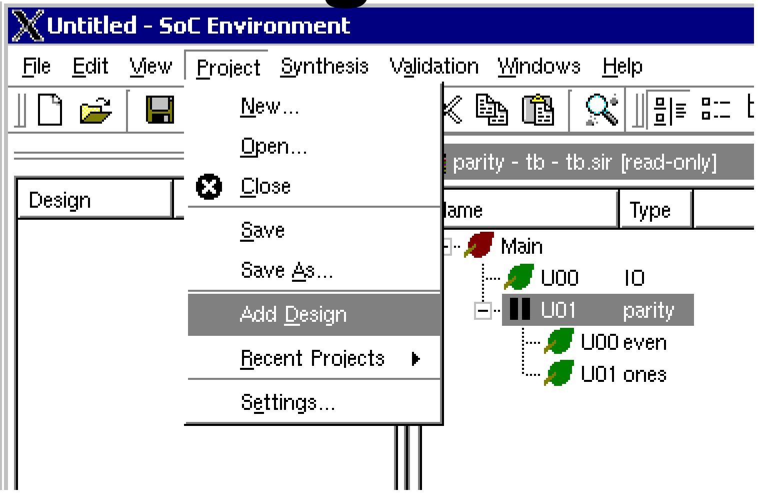


Main - tb - tb.sir [read-only]

Name	Type
Main	
U00	IO
U01	parity
U00	even
U01	ones

Name	Type
done	event
inport	unsigned int
outport	unsigned int
start	event
U00	IO
U01	parity

Add Design



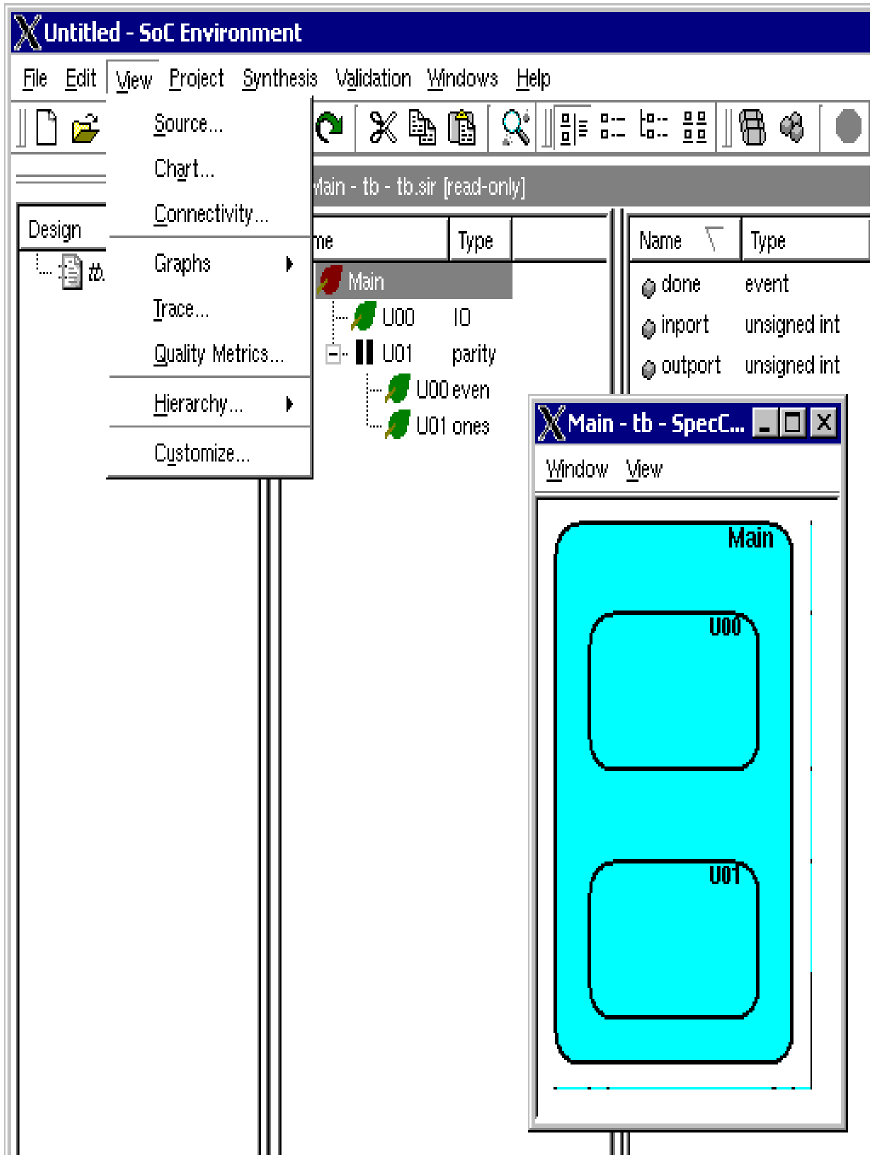
Untitled - SoC Environment

parity - tb - tb.sir [read-only]

Name	Type
Main	
U00	IO
U01	parity
U00	even
U01	ones

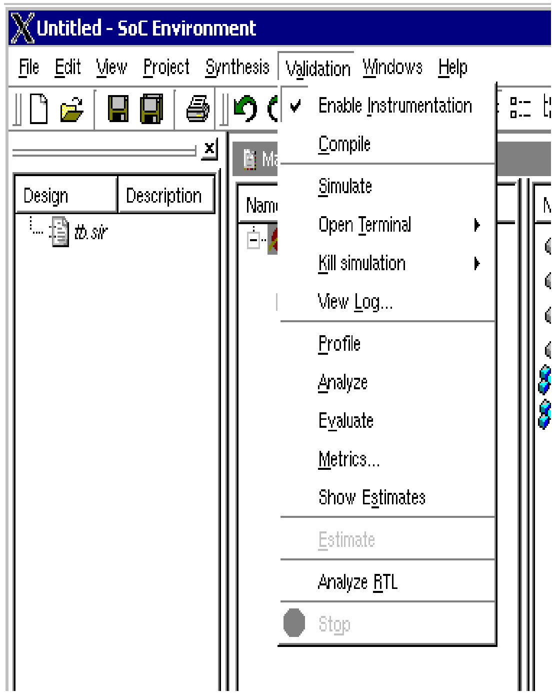
Getting Information from the Design

- At this point SCE can make a SpecChart of your design and display it for you.
- To do this select the Main level on the Project Management Window and select View->Chart
- You can traverse the design hierarchy from the different options under 'View' on the Chart window.



Compile, Simulate, and Profile

- To compile and simulate your design, simply select Compile and Simulate under Validation.
- Once you have compiled and simulated your design you may extract more information by profiling it. To profile the simulation select Validation->Profile.
- This is a functional profile from which you can determine the computation and communication utilization of your design. This information will aid you during the architecture exploration and refinement.
- You will learn more about the way SCE performs this profiling in class.



Summary

- SCE is capable of much more including automated refinement and detailed profiling down to the bit level.
- Other resources
  - System-on-Chip Environment Tutorial, Version 2.2.0 beta. (link)
  - A. Gerstlauer and Rainer Domer, "SCE Specification Model Reference Manual", *Technical Report*, Center for Embedded Computer Systems, University of California, Irvine, April 2005. (link)
  - A. Gerstlauer, D. Shin, S. Abdi, P. Chandraiah, and D. Gasjki, "Design of a MP3 Decoder using the System-On-Chip Environment (SCE)", *Technical Report CECS-07-05*, Center for Embedded Computer Systems, University of California, Irvine, November 2007. (link)