Scicos-HDL User Guide

1 About Scicos-HDL

1.1 Features

- Links The Scilab/Scicos with the Digital circuit design(EDA).
- Integrates the hardware circuit, algorithm and Scilab/Scicos environment as a plat for digital circuit design, simulation and Hardware Description Language generation.
- Enables complex signal processing combined with powerful mathematical tools.
- Inside libraries: Sequential logic library, Combinational logic library, Ipcore library, simulation library.
- Support mixed-simulation with the Scicos blocks.
- Automatically generates Description Language(VHDL and Verilog).
- Automatically generates a VHDL testbench.
- Automatically generates a Verilog testbench.
- Automatic propagation of signal names to generated HDL.
- You can specify most values in the block parameter dialog boxes using Scicos workspace.
- Mix-simulation with original Scicos blocks and Scicos-HDL blocks.
- Open interface for users to add blocks.
- Under Scilab ‘s LICENCE.
- Support: Windows XP, 2000, 98, Linux, UNIX, and Mac OS X.

1.2 General Description

Scicos-HDL integrates the hardware circuit, algorithm and Scilab/Scicos environment as a plat for digital circuit design, simulation and Hardware Description Language generation. Scicos-HDL shortens digital circuit design cycles by helping you create the hardware representation in an modeling-friendly development environment. You can combine existing Scicos blocks with Scicos-HDL blocks and to link system-level design . It is a open source project under Scilab ‘s license, the release vision 0.5 can help to design and simulate some small-scale digital circuit system with its Hardware Description Language generation. We now will make it as a tool for teaching digital circuit design. Now it supports VHDL&Verilog Languages.

Scicos-HDL now has five libraries, including : HDL_Sequential_Lib, HDL_Combinational_Lib, HDL_IPcore_Lib, HDL_Simulation_Lib, HDL_Super_Lib. See the following picture:

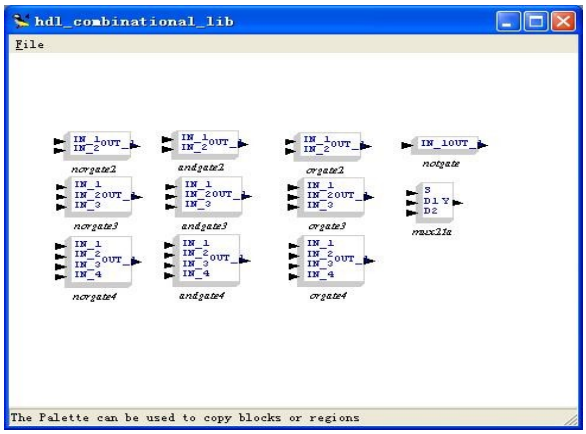


Figure 1 HDL_Combinational_Lib

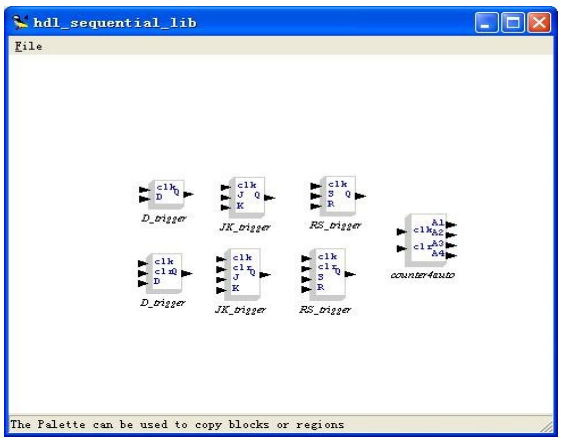


Figure 2 HDL_Sequential_Lib

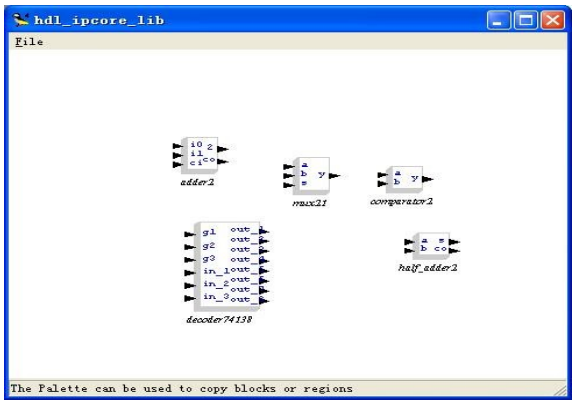


Figure 3 HDL_IPcore_Lib

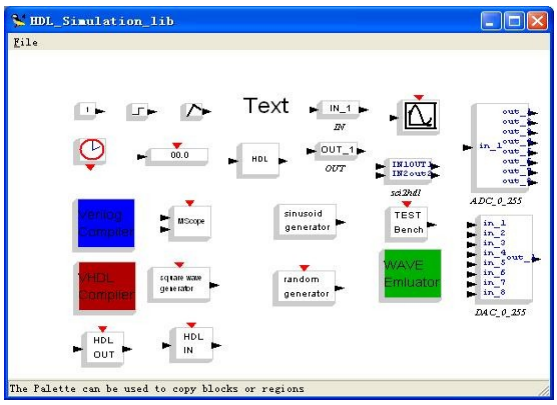


Figure 4 HDL_Simulation_Lib

1.3 Design Flow

When using Scicos-HDL, you start by creating a design model in the Scilab/Scicos software. After you have created your model, you can generate VHDL or Verilog HDL files and their simulation generate files for VHDL or Verilog HDL simulation.

The design flow involves the following steps:

1. Create a model with a combination of Scicos and Scicos-HDL blocks using the Scilab/Scicos software.
2. Simulate the model in Scicos using a Scope block to monitor the results(inside Ghdl—VHDL simulator).
3. Use the VHDL or Verilog HDL Compiler block to analyze your design and generate HDL language.
4. Use other synthesis tools to perform RTL synthesis.
5. Download to FPGA/CPLD board and test.

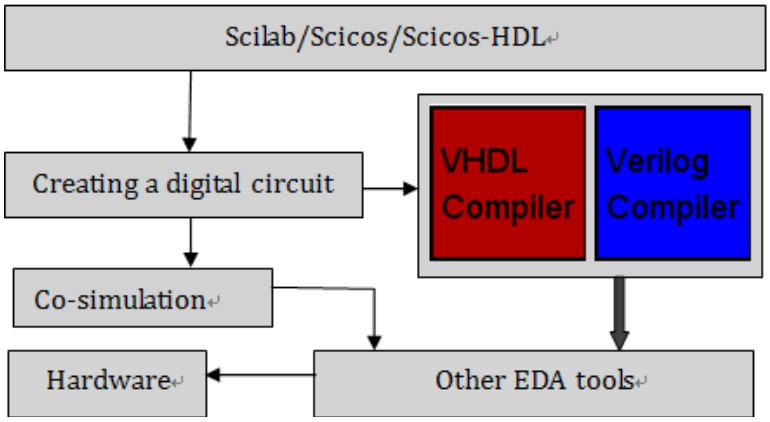


Figure 5 Design flow

2 Getting Started Tutorial

2.1 Introduction

This tutorial uses an example full adder design, fulladder.cos, to demonstrate the Scicos-HDL design flow.

The full adder is composed of some AND gate blocks, NOR gate blocks and NOT gate blocks as its entity blocks, these blocks belong to combinational logic library(HDL_Combinational_Lib); some other blocks including I/O port blocks(IN and OUT), HDL I/O ports blocks(HDL IN and HDL OUT) , simulation blocks(Square wave generator), displayer(Mscope) and VHDL and Verilog HDL Compilers, these blocks belong to simulation library(HDL_Simulation_Lib) and clock blocks from Scicos Sources palette. .

After finished the design and simulation, run Scicos-HDL Compilers to generate VHDL / Verilog HDL code of your design. By this example you will know the whole flow of using Scicos-HDL to design circuit and generate VHDL / Verilog code.

The following rules are used：

>> This sign will guide you to get into the subdirectories and select the final operation. For example: **Palette>>Palettes>>HDL_Sequential_Lib,** it means that please select the **palette** menu,click **palettes**, and then click **HDL_Sequential_Lib**.

This sign is a prompt, means there is significant information for you.

Bold-face It means the name of menu, the option of dialog box and so on, which you can click or select.

2.2 Creating the Full Adder Model

To create a new model, perform the following steps:

- Start the Scilab/Scicos.
- Choose **File >>Save As** (File menu) in the new model window.
- Browse to the directory in which you want to save the file. This directory becomes your working directory, This tutorial uses the working directory <**D:/test/**>.
- Type the file name into the File name box. This tutorial uses the name **fulladder.cos**.

- Click **Save**.
- Click the **Palette** menu in the menu bar and select the library, we need **HDL_Combinational_Lib , HDL_Simulation_Lib and Sources**.

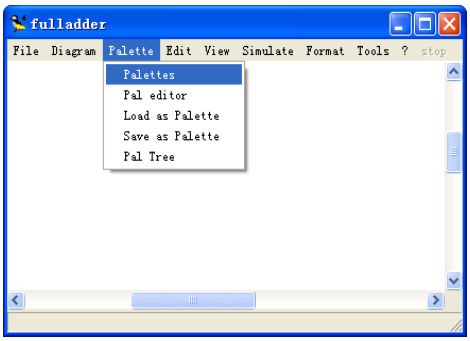
The following sections describe how to add blocks to your model and simulate the model in Scicos.

2.2.1 Add the AND gate Block

Perform the following steps to add the **AND gate** block:

- In the **Palette>>Palettes**, click **HDL_Combinational_Lib** to view the blocks .
- Double-click or drag **AND gate** block into your model (3 andgate2 blocks and 4 andgate3 blocks).

See the following picture:



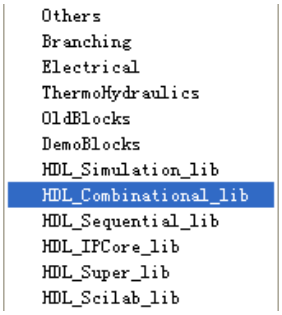


Figure 6 Open Palette

Figure 7 Click HDL_Combinational_Lib

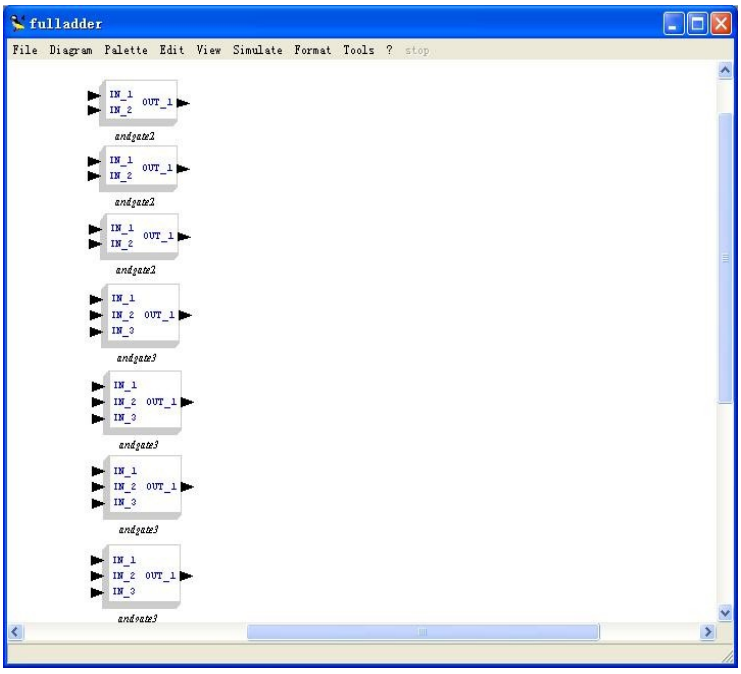


Figure 8 AND gate blocks

2.2.2 Add the NOT gate Block

Perform the following steps to add the **NOT gate** block:

- In the **Palette>>Palettes**, click **HDL_Combinational_Lib** to view the blocks .
- Double-click or drag **NOT gate** block into your model (3 notgate blocks).

See the following picture:

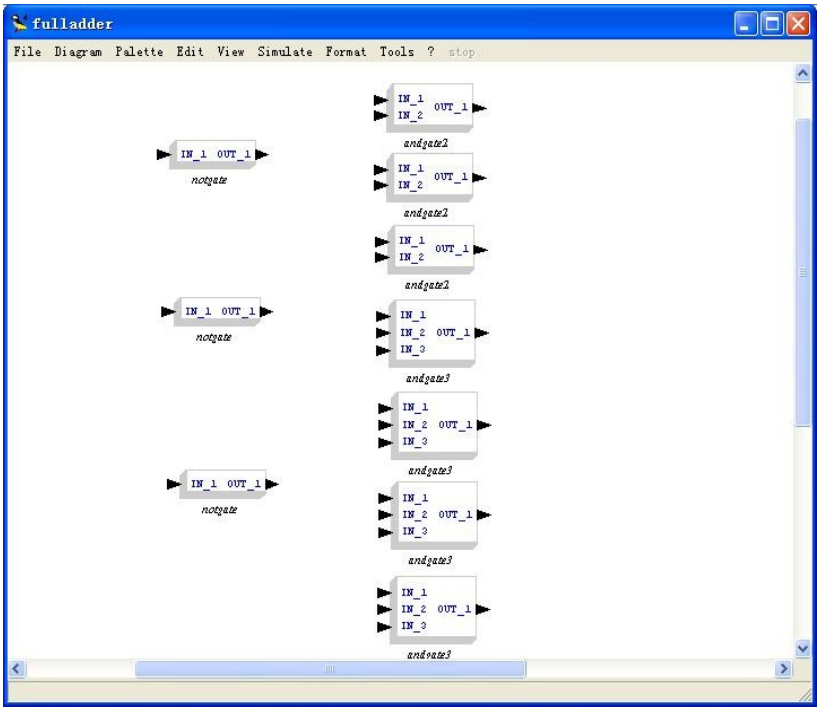


Figure 9 NOT gate blocks

2.2.3 Add the NOR gate Block

Perform the following steps to add the **NOR gate** block:

- In the **Palette>>Palettes**, click **HDL_Combinational_Lib** to view the blocks .
- Double-click or drag **NOR** gate block into your model (3 norgate blocks).

See the following picture:

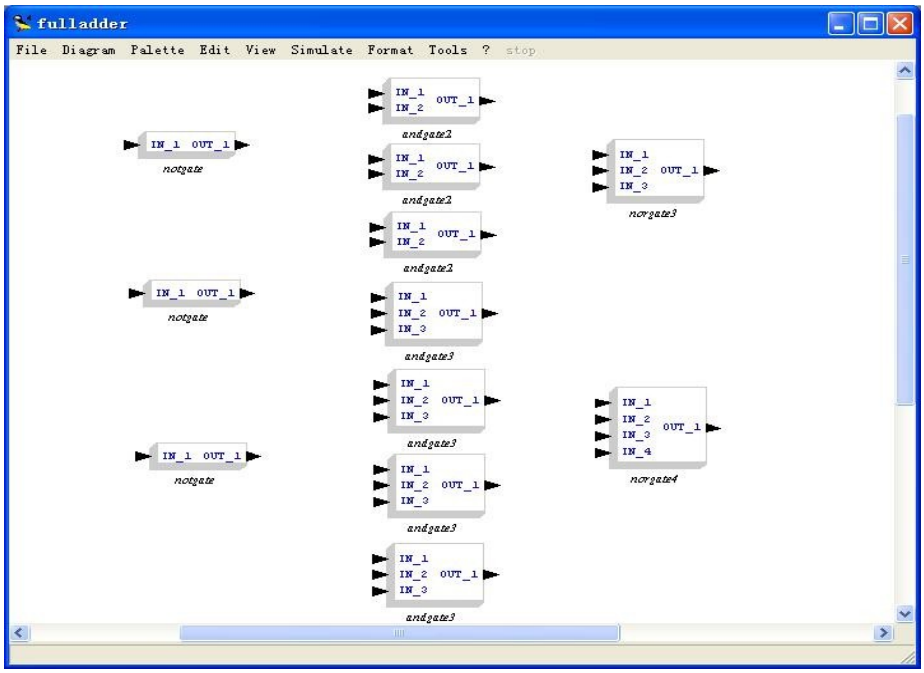


Figure 10 NOR gate blocks

2.2.4 Add the I/O port blocks(IN and OUT)

Perform the following steps to add the **I/O port blocks** block:

- In the **Palette>>Palettes**, click **HDL_Simulation_Lib** to view the blocks .
- Double-click or drag **I/O port blocks** block into your model (3 IN blocks and 2 OUT blocks).

See the following picture:

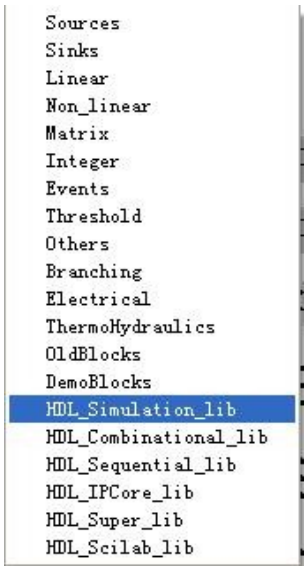


Figure 11 Click HDL_Simulation_Lib

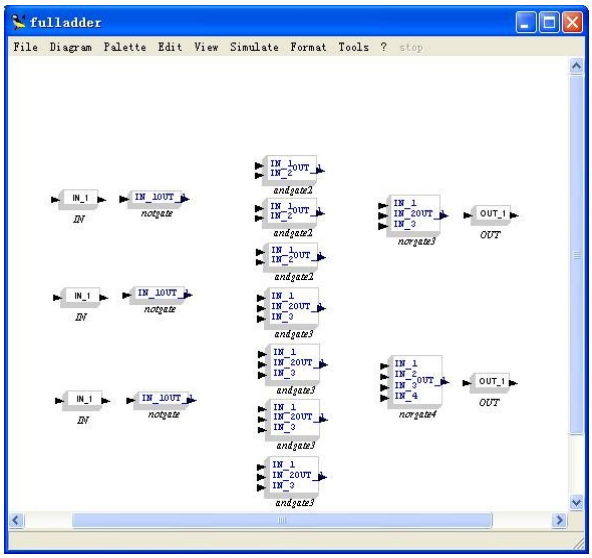


Figure 12 I/O port blocks 1

Then we change the **I/O port** blocks' name as a fulladder, see the picture next:

- Double-click the **I/O IN** blocks to display the Parameters dialog box, see figure 13.
- Change the name"**IN_1**" to "**CI**".
- Click "**OK**".

You will see the block display the name "**CI**", change the other four **I/O port** blocks(IN and OUT blocks) as well, see the figure 14:

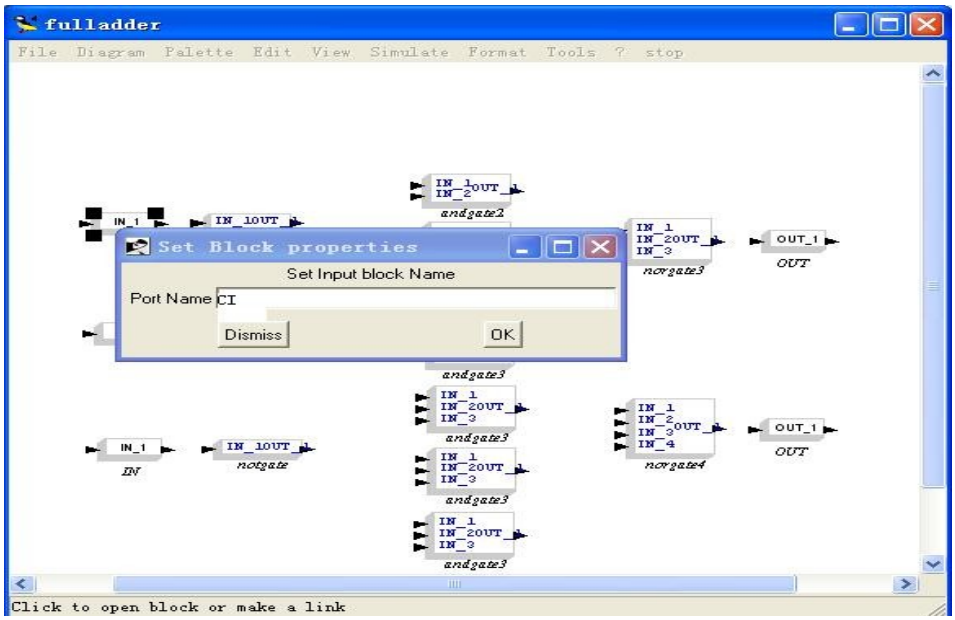


Figure 13 Change I/O blocks name

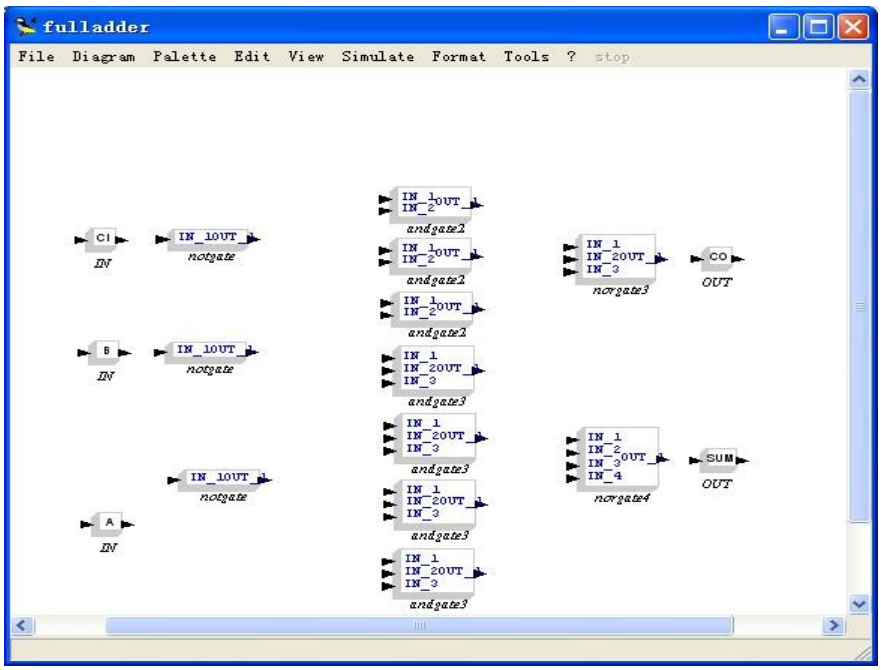


Figure 14 I/O port blocks 2

The I/O blocks must be put in every model file，the number of these blocks is equal to the whole system I/O ports number.

2.2.5 Add the HDL I/O ports blocks(HDL IN and HDL OUT)

Perform the following steps to add the **NOR gate** block:

- In the **Palette>>Palettes**, click **HDL_Simulation_Lib** to view the blocks .
- Double-click or drag **HDL IN and HDL OUT** block into your model (1 HDL IN and 1HDL OUT).
- Double-click the **HDL IN** block to display the Parameters dialog box, see figure 15.
- Change the input port number to "**3**".
- Click "**OK**".

You will see the the **HDL IN** block's input port number has been changed, change **HDL OUT** as well, see

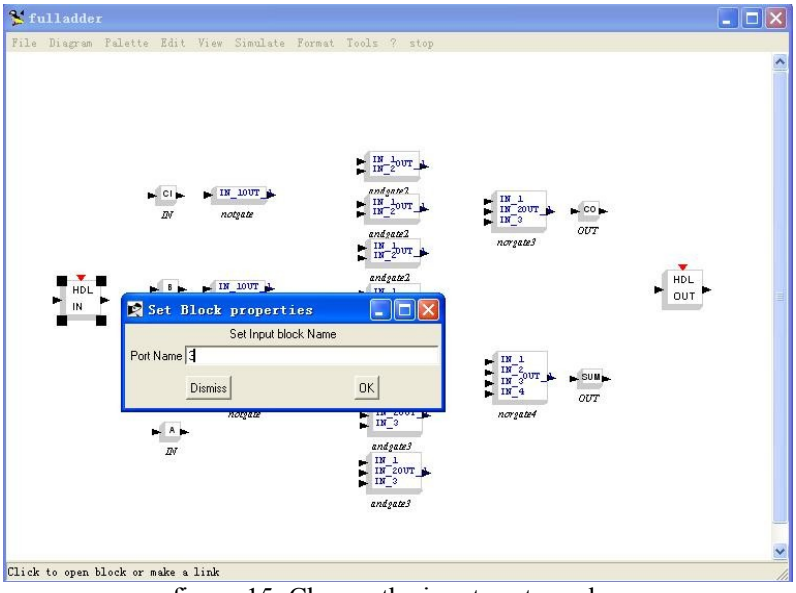


figure 15: Change the input port number

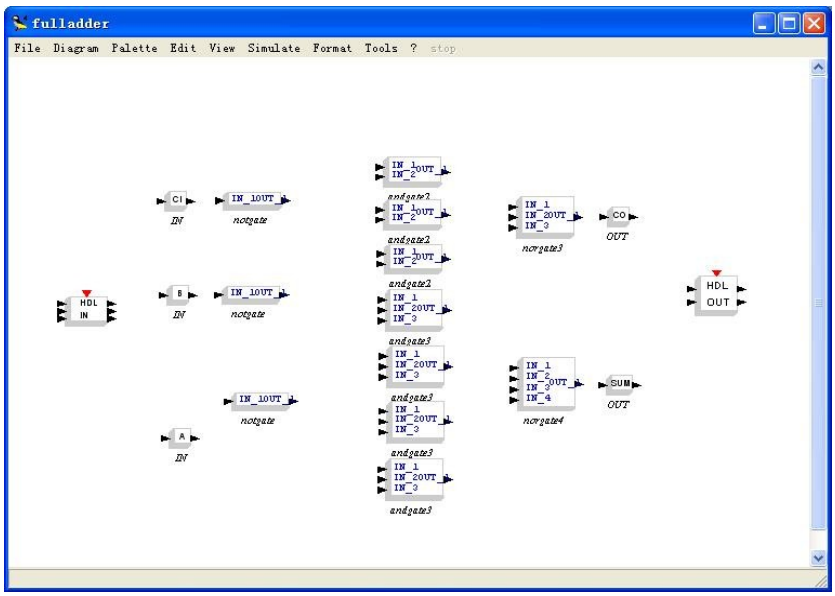


Figure 16 HDL IN AND HDL OUT

The HDL IN and HDL OUT blocks must be put in every model file，their I/O ports number is equal to the whole system I/O ports number.

2.2.6 Add the Square wave generator block

Perform the following steps to add the **Square wave generator** block:

- In the **Palette>>Palettes**, click **HDL_Simulation_Lib** to view the blocks .
- Double-click or drag **Square wave generator** block into your model (3 Square wave generator blocks).

See the following picture:

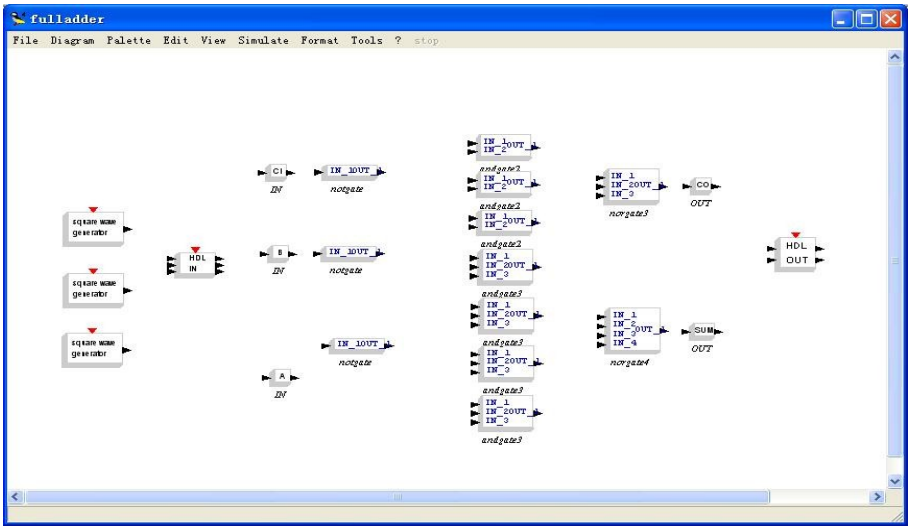


Figure 17 Square wave generator block

2.2.7 Add the displayer(Mscope) block

Perform the following steps to add the **Mscope** block:

- In the **Palette>>Palettes**, click **HDL_Simulation_Lib** to view the blocks .
- Double-click or drag **Mscope** block into your model (1 Mscope).
- Double-click the **Mscope** block to display the Parameters dialog box, see figure 18.
- Change the Parameters: **Ymin vector= -1 -1; Ymax vector = 2 2**.

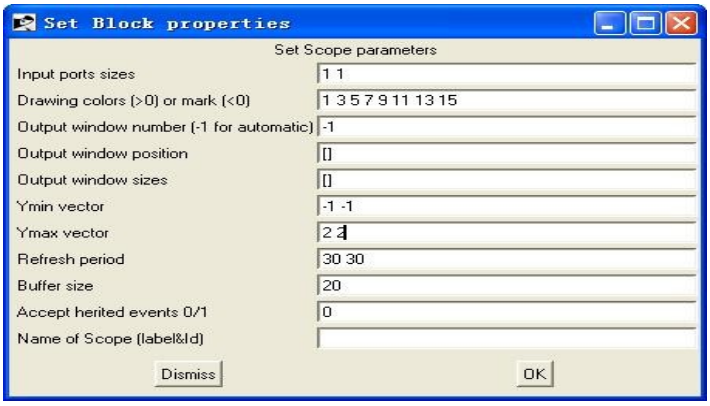


Figure 18 Change parameters

- Make sure that the input ports number of **Mscope** is equal to the number of I/O OUT blocks.
- The Scicos help document will help you how to change the parameters.

- Click "**OK**".

See the following picture:

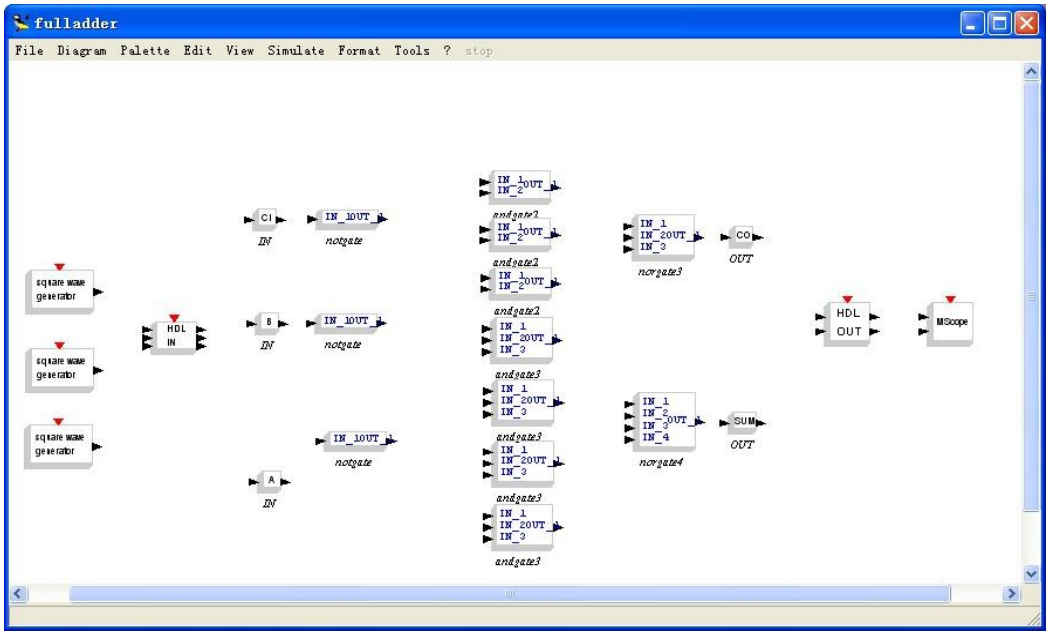


Figure 19 Mscope block

2.2.8 Add the clock block

Perform the following steps to add the **clock** block:

- In the **Palette>>Palettes**, click **Sources** to view the blocks .
- Double-click or drag **clock** block into your model (4 **clock** blocks).
- Double-click the **clock** block to display the Parameters dialog box, see figure 19.
- Change the "**Sample** and **offset"**(figure 20 show the value).
- Click "**OK**".

See the following picture:

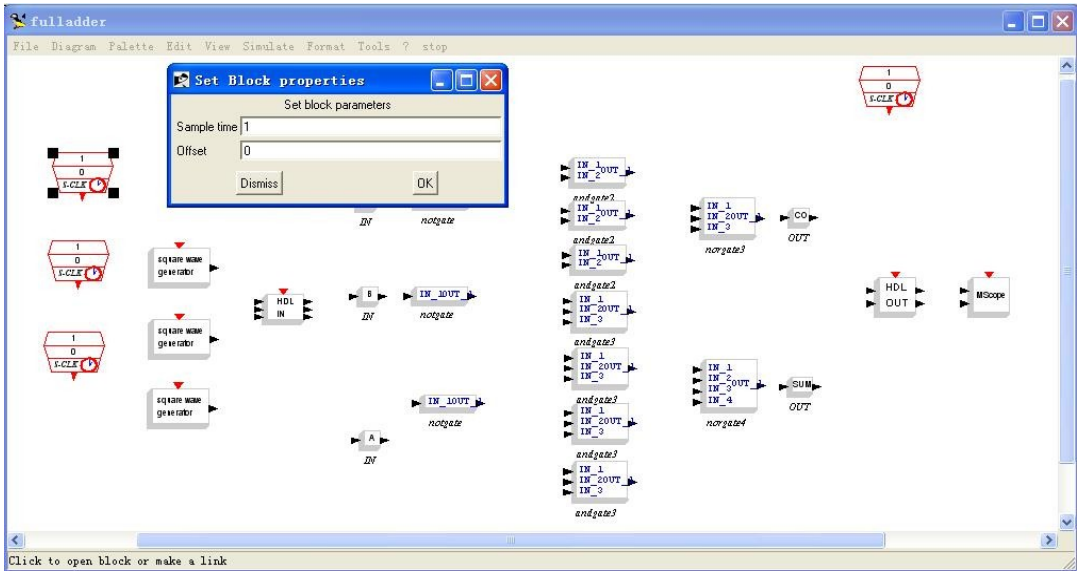


Figure 20 Clock block

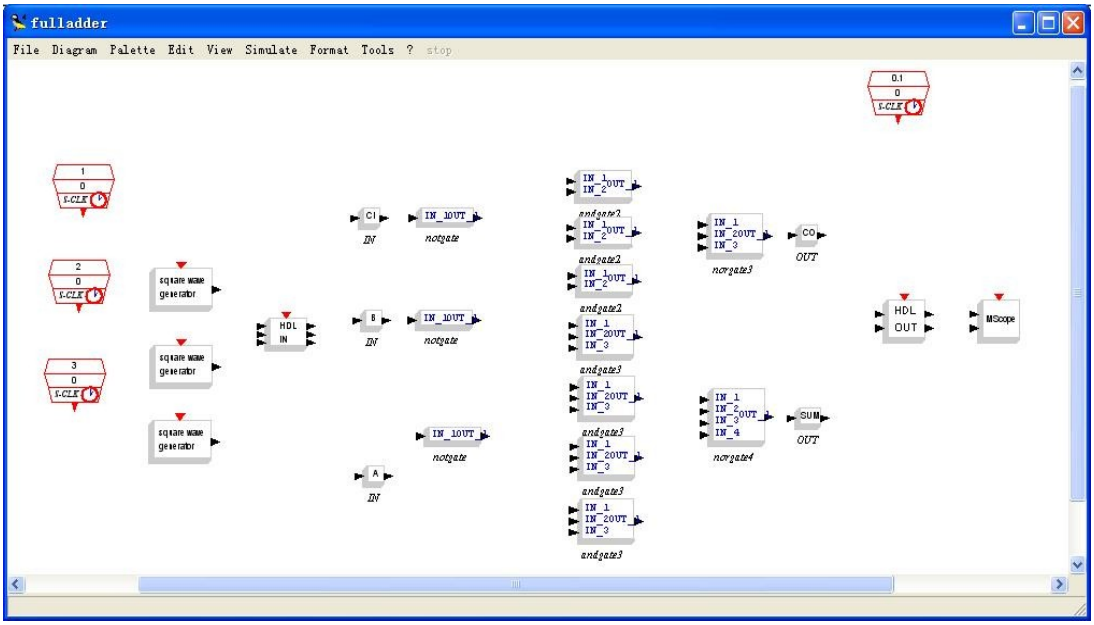


Figure 21 Clock block value

2.2.9 Make connections

Make connections to connect the complete your design as follows:

- From one black output port to one black input port(from one block's the right to one block's the left).
- From one red output to one red input port(from one block's the bottom to one block's the top).
- From one output port to more then one input ports,double click the line and then link to other input port.

See the following picture:

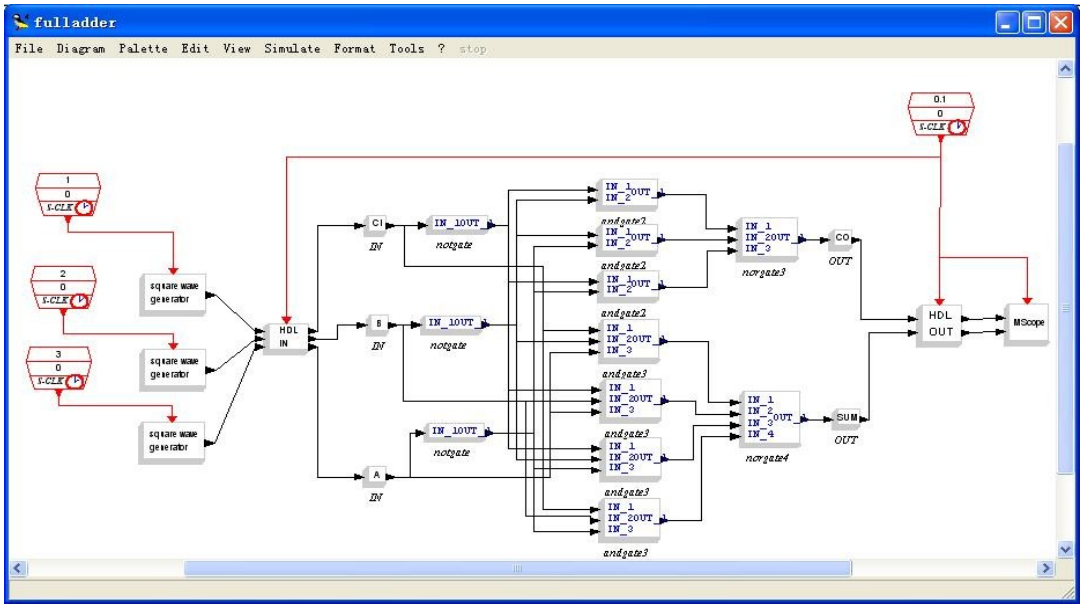


Figure 22 Connections

2.2.10 Add the VHDL and Verilog HDL Compilers block

Perform the following steps to add the **VHDL and Verilog HDL Compilers** block:

- In the **Palette>>Palettes**, click **HDL_Simulation_Lib** to view the blocks .
- Double-click or drag **VHDL and Verilog HDL Compilers** block into your model.

Now the design model file has been completely finished. See the following picture:

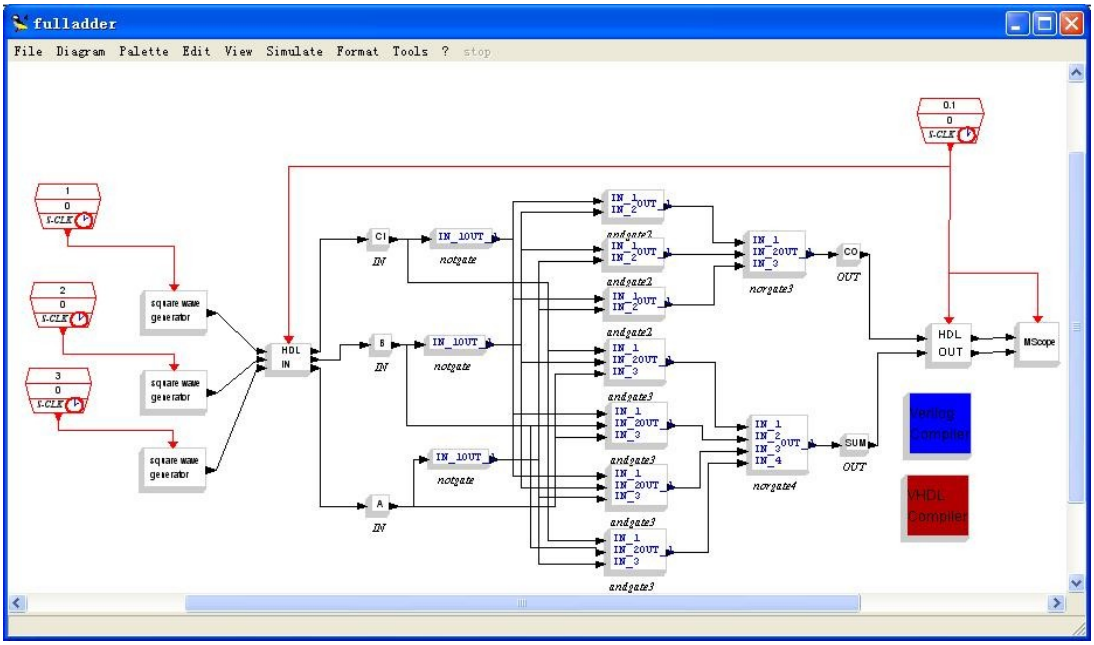


Figure 23 Compilers

2.3 Compiling the Design

To generate the VHDL and Verilog HDL codes for you design, Perform the following steps:

- save you model file.
- Create two empty folders to save the HDL codes,here we have VHDL and VERILOG folder in **D:\test** directory(Figure 24).

Figure 24 Create two empty folders

- Double-click the VHDL or Verilog HDL compiler block in your model to display the compiler dialog box (Figure 25).

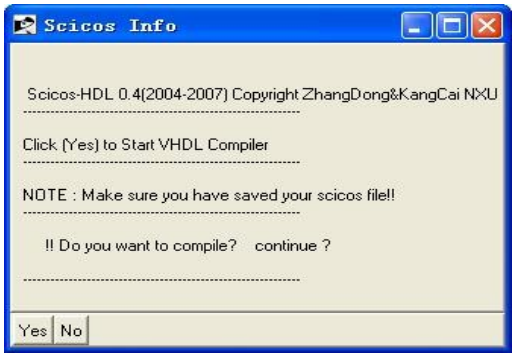


Figure 25 Dialog box

- Perform the following steps:
  - Click **Yes** to continue.
  - Choose a folder to save HDL codes and click **Yes**(Figure 26).

Figure 26 Choose a folder

  - Wait until the compiling finished and click **OK**(Figure 27).

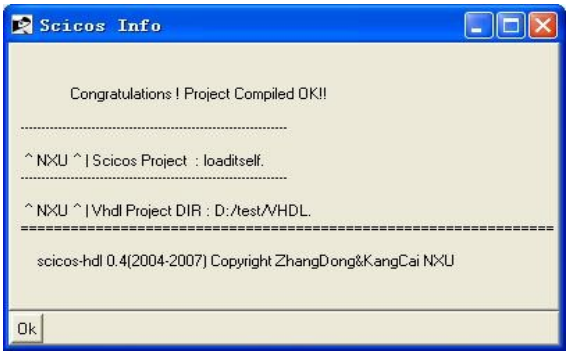


Figure 27 Compiling successfully

- Go to the folder to check the codes(Figure 28).

Figure 28 HDL file generation

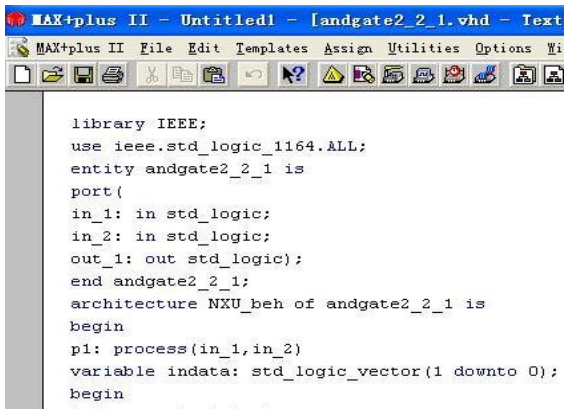


Figure 29 HDL code(part)

2.4 Simulate Your Model in Scicos

To simulate your model in the Scicos, perform the following steps:

- Choose "**Simulate>>Setup**" (top menu) to display the Configuration dialog box (Figure).

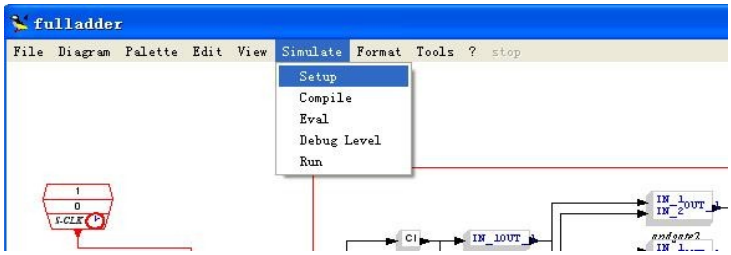


Figure 30 Simulation Setup

- Type "**50**" (or more)in the Final integration time box and click OK.

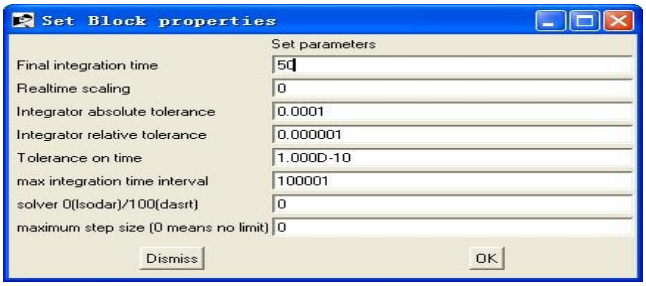


Figure 31 Change parameters

- Start simulation by choosing **Simulate >>Run** (top menu).

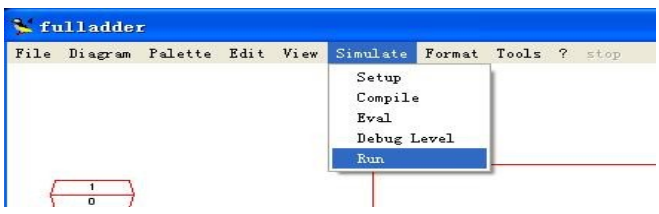


Figure 32 Run simulation

- Then Scicos will display the result.

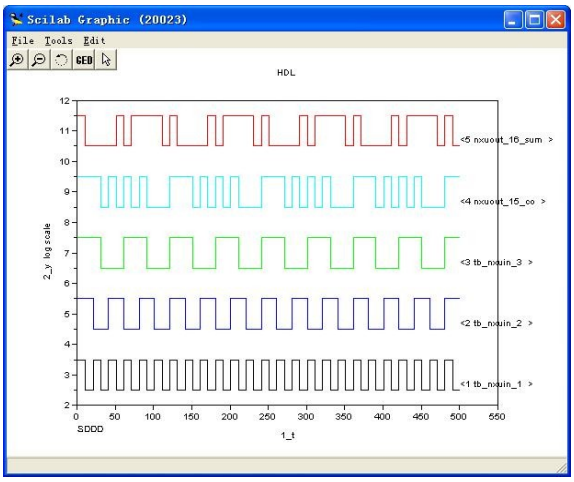


Figure 33 Result of GHDL

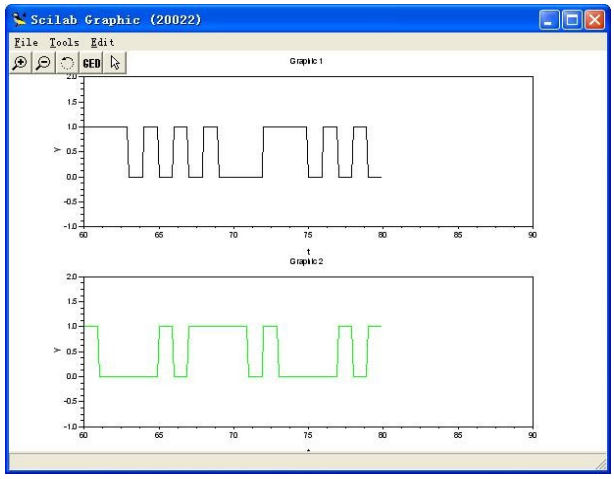


Figure34 Result of Scicos

Now we have finished the tutorial, any problem please send us an email: **scicoshdl@gmail.com**.

Some rules

- Make sure you have put the block Scicos-HDL IN as the input ports.
- Make sure you have put the block Scicos-HDL OUT as the output ports.
- Self-connected is not allowed in every block.
- Make sure the path and name of model file are correct.
- Make sure the directory of saving VHDL / Verilog code file is correct.

3 Summarize

You can use the blocks in Scicos-HDL to create and simulate a hardware implementation of a system model in Scicos in a short time. The Scicos-HDL Compiler blocks generates VHDL / Verilog HDL code. Scicos-HDL makes Scicos have hardware design and simulation function. It set up a bridge between Scilab and EDA.

****

THE END

Scicos-HDL Group 2008-04