Use xTIMEcomposer to simulate a program

IN THIS DOCUMENT

- Configure the simulator
- Trace a signal
- Set up a loopback
- Configure a simulator plugin

The xCORE simulator provides a near cycle-accurate model of systems built from one or more xCORE devices. Using the simulator, you can view a processor's instruction trace, visualize machine state and configure loopbacks to model the behavior of components connected to XMOS ports and links.

1 Configure the simulator

To configure the simulator, follow these steps:

1. Select a project in the **Project Explorer**.
2. Choose **Run ▶ Run Configurations**.
3. In the left panel, double-click **XCore Application**. xTIMEcomposer creates a new configuration and displays the default settings in the right panel.
4. In the right panel, in **Name**, enter a name for the configuration.
5. xTIMEcomposer tries to identify the target project and executable for you. To select one yourself, click **Browse** to the right of the **Project** text box and select your project in the **Project Selection** dialog box. Then click **Search Project** and select the executable file in the **Program Selection** dialog box.

   You must have previously compiled your program without any errors for the executable to be available for selection.
6. Select the **simulator** option and click the **Simulator** tab to configure additional options, as shown in Figure 1.

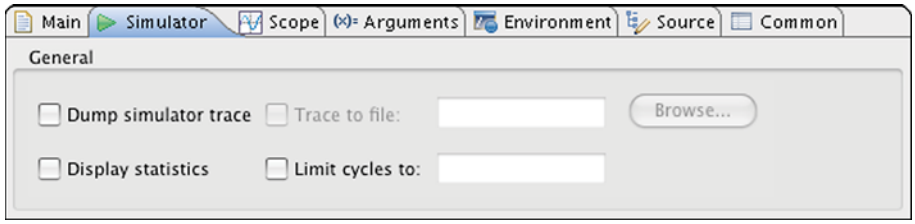

Figure 1: Simulator configuration options

XMOS © 2013, All Rights Reserved

- To output the processor instruction trace during simulation, select **Dump simulator trace**.
  By default, the instruction trace is displayed in the **Console**. To write the trace to a file instead, select **Trace to file** and enter a filename. The filename must be different from all other files in your project.
  The format of the instruction trace is described XM-000930-PC.
- To view a summary of the program's execution once the program terminates, select **Display statistics**. The summary includes the instruction count for each logical core, and the number of data and control tokens sent through the switches.
- To limit the number of cycles executed by the simulator, enter a value in the **Limit cycles to** text box. Leave blank if you want the program to run from start to finish. This is useful for simulating programs with infinite loops.

7. To save and run the configuration, click **Run**.

xTIMEcomposer loads your executable, displaying any output generated by your program in the **Console**.

xTIMEcomposer remembers the configuration last used to run your program. To run it again using the same configuration, just click the **Run** button button. To use a different configuration, click the arrow to the right of the **Run** button and select a configuration from the drop-down list.

2 Trace a signal

The simulator can output signal tracing data to a VCD file which you can visualize with the xTIMEcomposer waveform viewer.

2.1 Enable signal tracing

To enable signal tracing during simulation, follow these steps:

1. Create a simulator **Run Configuration** (see §1).
2. In the **Simulator** tab, in the **Signal Tracing** panel, select **Enable signal tracing**.
   - To trace all I/O pins, in the **System Trace Options** group, select **Pins**.
   - To trace machine state on a specific core, in the **Core Trace Options** group, click **Add** to display a set of configurable drop-down lists and checkboxes. Then select the core and machine state you wish to trace. You can trace process cycles, ports, cores, clock blocks, pads and processor instructions.
3. Click **Run**.

xTIMEcomposer loads your program into the simulator and, on termination, adds the generated VCD file to your project.

2.2 View a trace file

In the **Project Explorer**, double-click on a VCD file to open it in the **Signals** view, as shown in Figure 2.

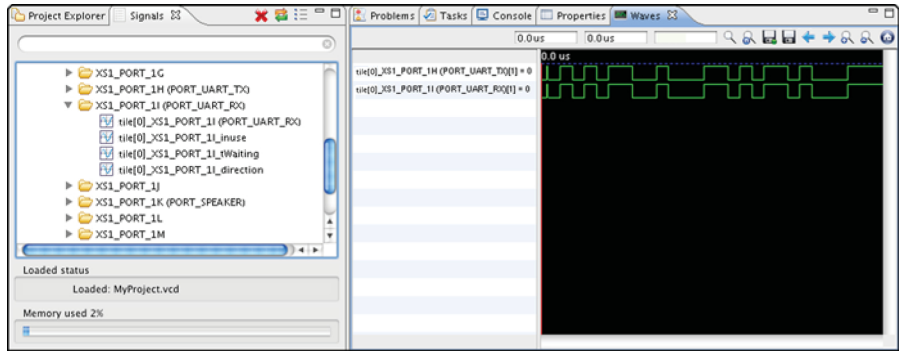

Figure 2: Signals and Waves views

In the **Signals View**, click the **plus sign** (Windows) or the **disclosure triangle** (Mac) to expand a folder and display its collection of signals or subfolders. Double-click on a signal or an entire folder to display in the **Waves** view.

To switch between a hierarchical and flat view of the signals, click the **Display** button.

2.3 View a signal

In the **Waves** view, move the cursor over a signal to view the time since the start of the simulation in the right numerical control at the top of the Waves view. If the cursor changes to a pointing finger, you can double-click to locate the output statement in the source code responsible for driving the signal. Use the **Waves** view toolbar to perform the following operations:

To view the entire waveform, click the **Zoom Fit** button.

To move between transitions of the selected signal, click the **Next** and **Previous** buttons. The output statement responsible for the transition is highlighted in the editor.

To search for a specific transition, click the **Search Transition** button to open a dialog box. Enter a value and click **Find**.

To save the configuration, click the **Write Session File** button and enter a filename for the file. Your settings are saved for use whenever you load the VCD file in the **Waves** view. Click the **Read Session File** button to load a recently saved settings file.

You can control how signals are displayed in the **Waves** view as follows:

- **Display signal values in ASCII:** Right-click on a signal in the **Waves** view to bring up a contextual menu and choose **Data Format ▶ ASCII**.
- **Add a separator between signals:** Right-click on a signal in the Waves view to bring up a menu and choose **Add Separator**.
- **Name a separator:** Right-click on a separator to bring up a menu and choose **Name Separator**. Enter a name for the separator in the **Name Separator** dialog box and click **OK**.
- **Move a separator:** Click-and-drag a separator to the desired position.

3 Set up a loopback

You can connect any two ports or pins together in your simulation, to model connections between the pins. To configure a loopback, follow these steps:

1. Create a simulator **Run Configuration** (see §1).
2. Click the **Simulator** tab to display the simulator configuration options.
3. Click the **Loopback** tab in the **Plugins** panel and select **Enable pin connections**.
4. In the **Pin Connections** panel, click **Add**. An empty loopback configuration is displayed. The loopback consists of two sets of options that you can configure for two different ports.

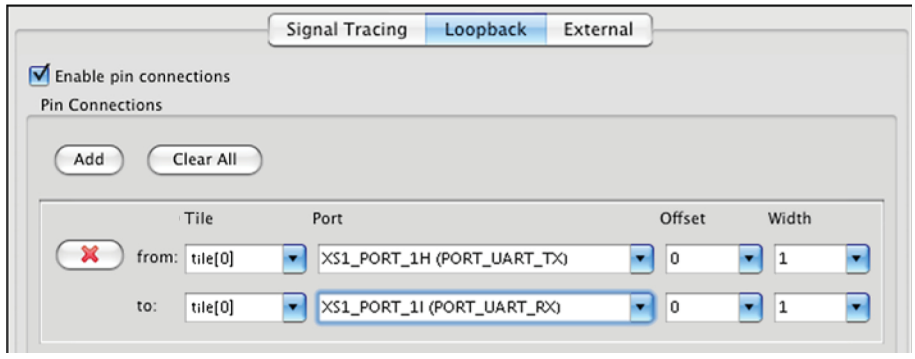

Figure 3: Setting up a loopback connection

5. In the drop-down lists for each end of the connection, select a value for **Tile** and **Port**. If you leave the tile unspecified, the list of ports is taken from the project's XN file and the tile is determined automatically. If you specify tile, the list of ports is taken from the header file `<xs1.h>`. To specify that only a subset of the pins are connected to the port, change the values for **Offset** and **Width**.

6. Click **Run**.

4 Configure a simulator plugin

You can connect the simulator to any external plugin that has been compiled on your host PC using the XMOS simulator plugin interface. To configure an external plugin, follow these steps:

1. Create a simulator **Run Configuration** (see §1).

2. Click the **Simulator** tab to display the simulator configuration options.

3. In the **Plugins** panel, click the **External** tab.

4. Click **Add** to open the plugin configuration dialog.

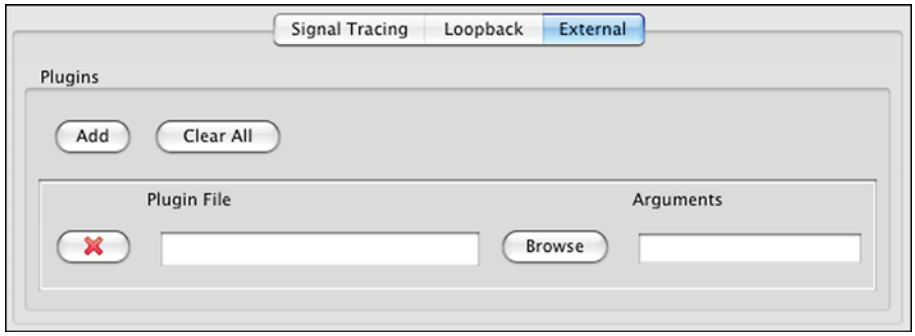

Figure 4: Setting up an external plugin

5. Select the plugin DLL and specify an optional command-line argument string.

6. Click **Run** to save your settings and run your program on the simulator with the specified plugins.

XMOS®

Copyright © 2013, All Rights Reserved.

Xmos Ltd. is the owner or licensee of this design, code, or Information (collectively, the "Information") and is providing it to you "AS IS" with no warranty of any kind, express or implied and shall have no liability in relation to its use. Xmos Ltd. makes no representation that the Information, or any particular implementation thereof, is or will be free from any claims of infringement and again, shall have no liability in relation to any such claims.