ForSyDe NoC System Generator Demo - Quickstart Manual

Setting up the system

1. Unpack the zipped files in a directory

2. Set the environment variable 'FORSYDE_PATH' to the directory

3. Create a Shortcut on your Desktop to the file '$FORSYDE_PATH/bin/Windows/ForSyDe_NSG_v2014_demo.exe'

4. Start the program

    a. The first time you start the program you will be prompted for the root directories where you have the Xilinx and Altera tools installed.

        i. The Xilinx root will have a list of directories in it named with the version number of the installation, e.g., '12.1', '14.2', etc.

        ii. The Altera root typically have the following list of directories in it, 'installer', 'ip', 'modelsim_ae',' nios2eds', and 'quartus'

    b. If you lack one of these tools, put in a dummy directory path, e.g., 'C:\'

    c. In the Demo version, only Altera is enabled, but the Xilinx path needs to be filled in anyway.

    d. Three blinking command windows appear before the Main Form starts. The first time the Main Form do not start, but a configuration file is saved in the $FORSYDE_PATH directory. The second time the Main Form should start.

5. In case the Main Form does not start the second time, your computer is slightly incompatible with our settings when we created the program. If you cannot make it run, look for the text file named 'test.txt' in the '$FORSYDE_PATH/bin/Windows' directory. It contains information how far in the start procedure that your computer managed to get. This file should be sent to johnnyob@kth.se so we can find a remedy to the problem.

6. Continue with the Work Flow below.

Work Flow

1. Load a Demo project (select the xml-file stored in one of the Example directories)

2. Inspect the Hardware View

    a. The drawing board is divided into sections. Each section represent a node in the on-chip interconnection network. The blue line represents a shared bus inside the node.

    b. Double-click on a node if you want to change its configuration

        i. Turn on and off processors using the drop down lists

        ii. Turn on and off memories using the drop down lists

            1. All processors need one memory for their programs. CPU #0 is tied to Memory #0, CPU #1 to Memory #1, etc.

        iii. Turn on and off JTAG units (needed for print statements in the software)

        iv. Turn on and off GPIOs pins and assign pins/wires to them

    c. In the Demo mode, the number of nodes is limited to four (4). (You can change the topology of the network in the NoC Settings Menu)

3. Inspect the Software View

    a. There is a list of different types of SW processes to the left. They are COMB, SMOC, Code, Connector, Input, and Output processes.

        i. COMB represent a pure combinatorial function, it is executed asynchronously as soon as it has an input.

        ii. SMOC represents a synchronous function, it is executed on the global clock tick, the Heartbeat of the system. (The Heartbeat is set in the NoC Settings Menu)

        iii. Code represents a Code Container. It is used to import software from other tools. It is currently only executing in synchronous mode. In case the <import> clause is removed from the .xml-file, the process is converted into a SMOC ditto.

        iv. Connector represents a communication channel. You connect the output of a process (on the right side) to the input of a process (on the left side). The GUI can only show three connections to a process, but you can connect more, they will all be drawn as connected to the third connection.

        v. Input represents a process that is connected to an external input channel, like an Ethernet port. Input processes are not enabled in the Demo version.

        vi. Output represents a process that is connected to an external output channel, like an Ethernet port. Output processes are not enabled in the Demo version.

    b. The Drawing board is divided into sections. Each section corresponds to a hardware node, and the subsections to the CPUs inside the nodes.

   c. Double-click on a node if you want to see the SW of that processes

      i. The SW view is divided into six editable windows.

         1. The two most important windows are the Init and Main Window - The Init window contains the 'reset' code executed when booting the system. The Main window contains the code executed every time the process is triggered.

         2. The Channel window lists the input and output channels associated with the process, its type and the c-variable associated with the channel.

         3. The window in the top right lists additional files that is needed to be able to compile the process on the target CPU.

      ii. Save the file and close (or cancel if you don't want to drop your changes).

      iii. You may from time to time have to redraw the SW View by selecting View->Redraw.

   d. Right-click on a node to get a list of transformation options in a drop-down menu

      i. Select/Move allows picks the process and let you move to another processor

      ii. Edit opens the SW menu for Combs and SMOCs, but opens the import menu for Code Containers.

      iii. Clone allows to create an unconnected duplicate of the selected process in the node. The Clone command makes a copy of the underlying C-code. Select/Move it to where it should be, and connect it to other processes.

      iv. Delete allows you to delete excess nodes. Deletion only removes the logical link in the .xml-file, the c-code is not removed.

4. Generate HW – This step sets up an image of the HW of the system in a format readable by the target technology (Xilinx or Altera).

5. Generate SW – This step creates the SW structure of the system in a form understandable by the target technology (Xilinx or Altera).

6. Remember to save the system every now and then. In particular before you run step 4 or 5 to be sure that the latest version is generated. The generate steps use the .xml-file that is stored on the file system, not the one in the Main Form. Unsaved changes are marked with an asterisk next to the system name in the system tab.

7. In case you have a SW process that contains a pure calculation (in double precision format), you may choose to apply the Calc2HW transformation (Main Form->Transformations->Calc2HW). The transformation converts the selected process into a VLIW implementation in VHDL and replaces the CPU in the node with a generated IP Block. Note, processes on the target node that are not selected for acceleration will disappear from the system – however, their links are still kept in the xml-file and can be restored by replacing the created accelerator with a processor.

8. Compile – This step is run in conjunction with the backend tool. In the demo version, the Altera tool flow is shown. Some steps are grayed out. These steps are run in the target tool.

    a. Start Quartus in the background.

        i. Create a project in Quartus. It should have the same name as the system name.

        ii. Run qsys or sopc_builder. Generate the System HW.

    b. Select the nodes for which you want to compile the SW.

        i. Generate the compilation scripts

        ii. Compile the SW for the selected Nodes

    c. Run the Synthesis in Quartus – if you do this step before step b), you have to update the .mif-files of the system before you proceed to d).

    d. Configure the FPGA by downloading the bit stream.

    e. Open one Nios II command shell for every node that contains print statements. Use the command 'nios2-terminal –I <jtag_nr>'. Note – the *jtag_nr* is for some unknown reason displaced by one from the *node_nr*. (jtag #1 is connected node #0, jtag #2 to node #1, etc). See the detailed manual.

        i. Print statements are slow. If you use them when debugging a node, you have to set the Heartbeat frequency of the system to less than 10 Hz (we recommend 1 Hz when debugging manually, and 10 Hz when redirecting the output stream to files).

    f. By pressing 'reset' button, the system restarts from the beginning and does a simultaneous boot of all processors in the system.