XAPP744 (v1.0.2) November 2, 2012

Hardware In The Loop (HIL) Simulation for the Zynq-7000 All Programmable SoC

Author: Umang Parekh

Summary

The Zynq™-7000 All Programmable SoC does not deliver a simulation model which poses a problem for designers. This application note describes a method/solution to bring the MPCore™ processor subsystem (PS) within the ISE® Design Suite Simulator (ISim) simulation environment with the help of Zynq-7000 platform Hardware In The Loop (HIL) simulation technology.

Introduction

The Zynq-7000 All Programmable SoC (AP SoC) is a new class of product from Xilinx, which combines an industry-standard ARM® dual-core Cortex™-A9 MPCore processor subsystem (PS) with Xilinx 28 nm programmable logic (PL). Traditionally, Xilinx offered the MicroBlaze™ embedded processor. Peripheral IP in the FPGA logic was created in RTL and the entire system was simulated using an RTL simulator in which the simulation RTL model for the MicroBlaze processor was provided by Xilinx.

Zynq-7000 AP SoC HIL technology is a technique that simulates, debugs, and test both the PS and the PL portions of a Zynq-7000 AP SoC design. All of the AXI-based IP connected to the PS through the master/slave general purpose (GP), high performance (HP), or auxiliary coherency port (ACP) can be simulated in ISim, while the PS is simulated in the hardware (the ZC702 board). The AXI-based PL interfaces are clocked using the testbench clock, allowing cycle accurate simulation of IP in the PL. The PS and the DDR memory operate in free-running mode. This approach is very useful for developing IP with the PS and also for IP driver development and software debugging. The software support for Zynq-7000 AP SoC HIL is released in the 14.2 version of the Xilinx ISE design tools.

Figure 1 shows an overview of the Zynq-7000 AP SoC HIL solution.

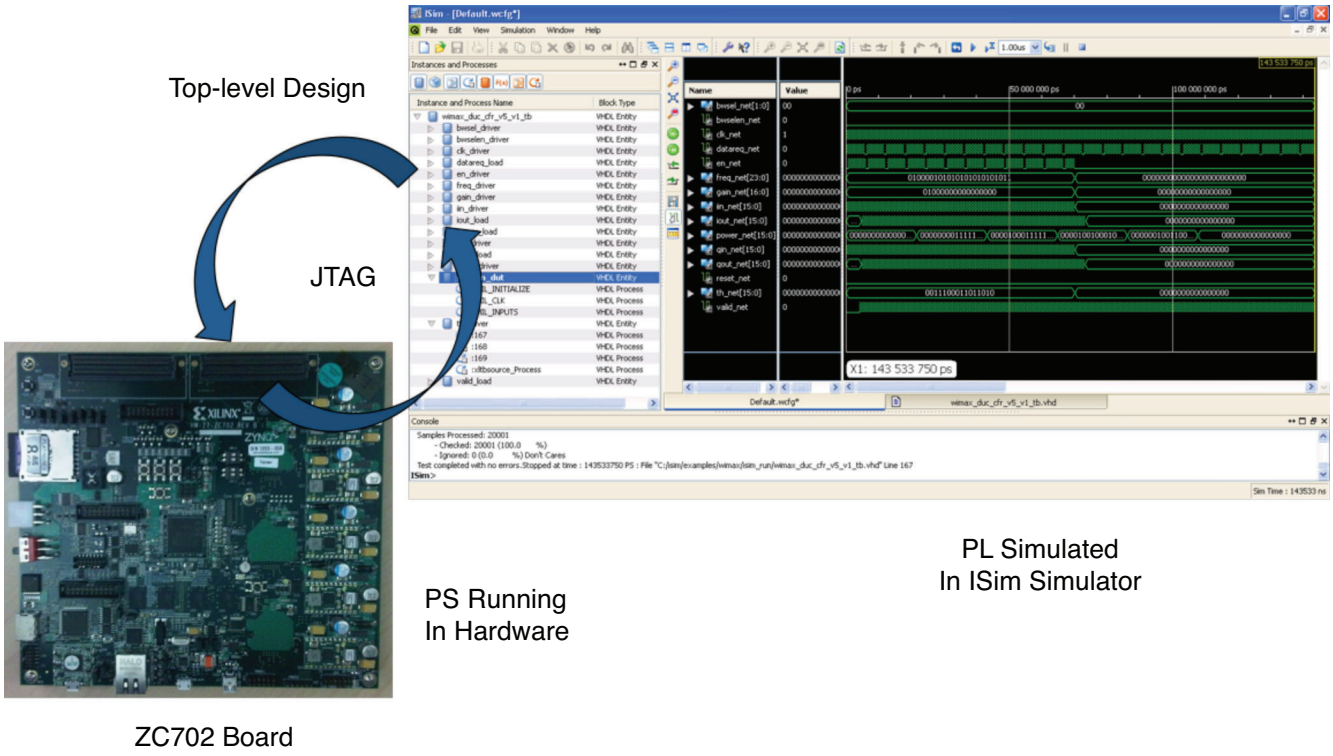


Figure 1: **Zynq-7000 AP SoC HIL Solution Overview**

Objective

The objective of this document:

- Explain how the PS in a Zynq-7000 AP SoC design can be brought into the simulation environment
- Run Zynq-7000 AP SoC HIL simulation using an example
- Guide the user to set up a system environment for Zynq-7000 AP SoC HIL simulation

Example System Overview

The concept of Zynq-7000 AP SoC HIL simulation is explained with the help of an example. Figure 2 shows a Zynq-7000 AP SoC design with the PS and a timer. The AXI timer is a slave connected to the general purpose master interface (M_AXI_GP0) from the PS via the AXI interconnect.

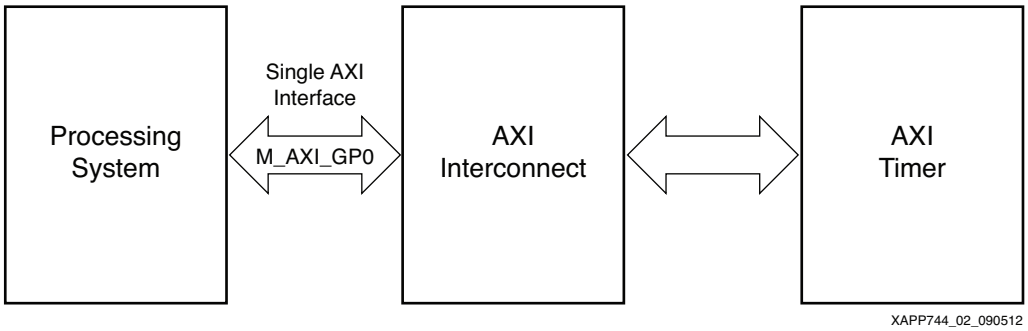


Figure 2: **Example Zynq-7000 AP S0C Design**

Figure 3 shows the system partitioning for simulation. The PS is running on the ZC702 board and the rest of the design (interconnect and the timer) is simulated using the ISim tool. The communication between the ISim tool and the ZC702 board is done via JTAG.

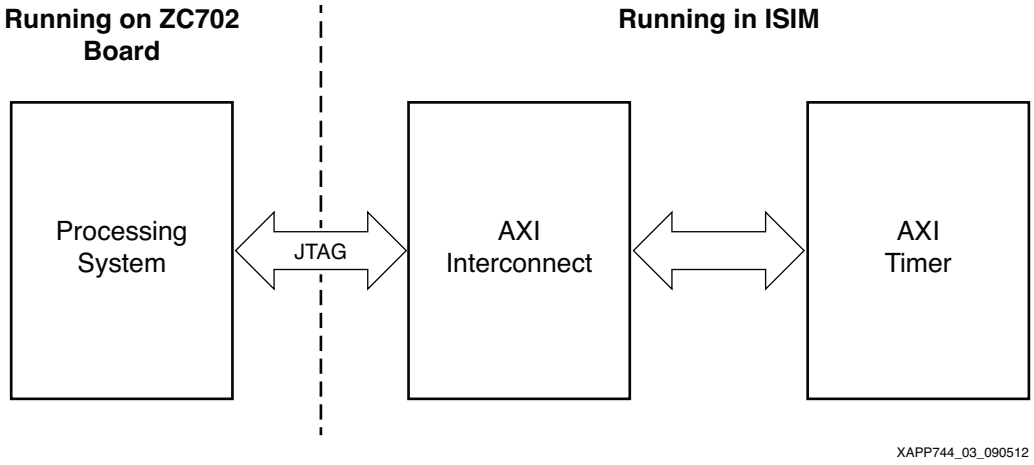


Figure 3: **System Partitioning for Simulation**

Figure 4 shows the simulation and the system clock connection. The AXI-based PL interfaces are clocked using the single-stepped clock, which is derived from the testbench running in the ISim tool. This allows cycle-accurate operation of the IP in the simulation. The PS and the DDR memory use the free running clock, derived from the board.

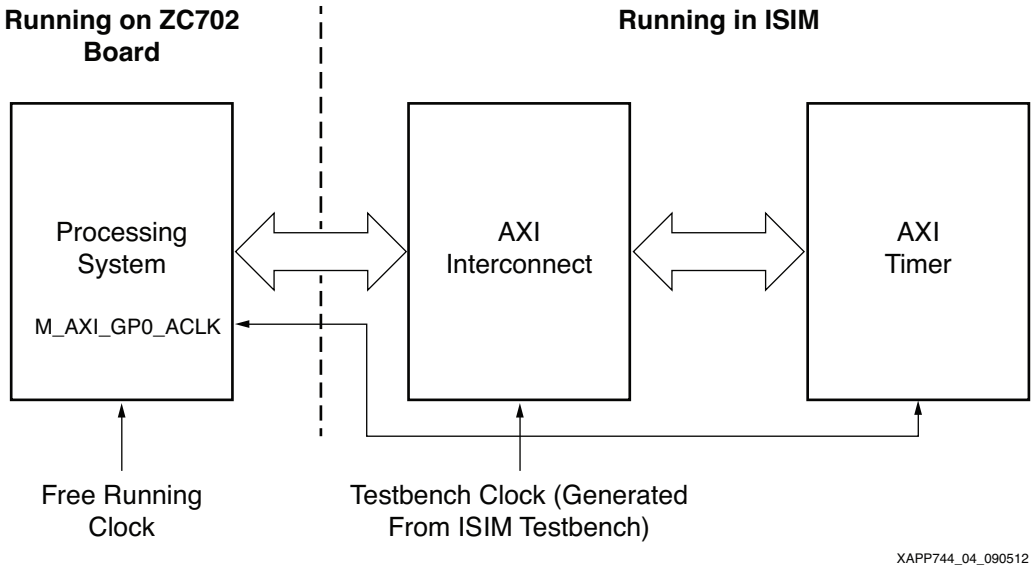


Figure 4: **System Simulation Clock Connections**

Tutorial

The tutorial can be downloaded from:

https://secure.xilinx.com/webreg/clickthrough.do?cid=197021

Unzip the file. Three additional ZIP files are extracted. For this tutorial unzip and use `zynq_HIL_timer_single_clock`.

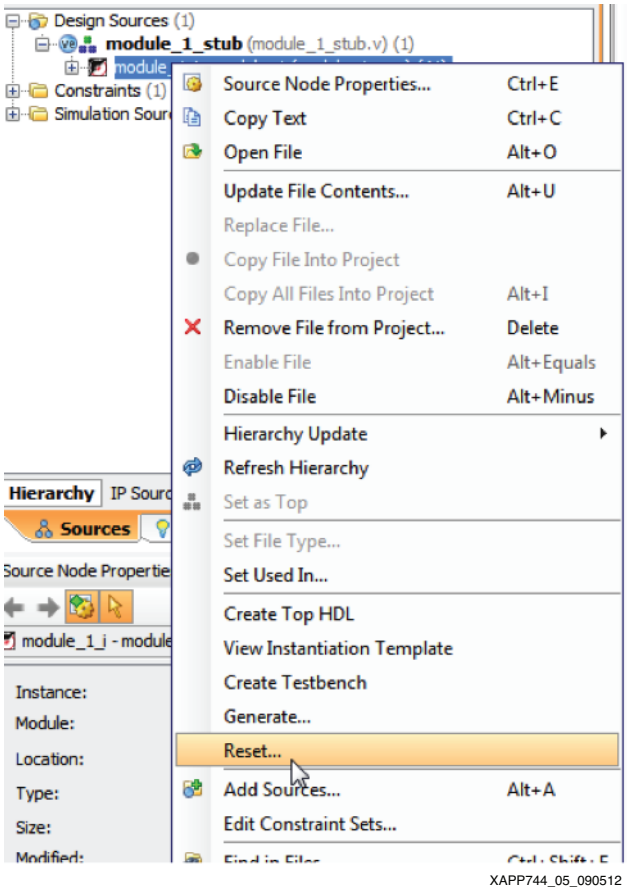
The tutorial contains a PlanAhead™ design tool project, which incorporates a Zynq-7000 AP SoC design as shown in Figure 2. This tutorial is intended for use with the System Edition of the Xilinx ISE Design Suite version 14.2 on Windows®. However, with some minor adjustments,

this tutorial can be performed easily in the Linux environment as well. This tutorial assumes some knowledge of Xilinx tools including SDK, ISim and PlanAhead.

In this tutorial, a PlanAhead design tool project is provided which contains the design (as described in Example System Overview, page 2). An AXI timer is connected to the Zynq-7000 AP SoC's PS through the AXI interconnect using the M_AXI_GP0 channel. The PS is the master and the timer is the slave. The timer is mapped to an address of `0x428000000`. Writing to the timer registers in this address range starts the timer, reads the timer value, and stops the timer.

Setting up Behavioral Simulation

1. Open the tutorial PlanAhead tool project file `zynq_hil_timer.ppr`.
2. Right click the EDK module as shown in Figure 5 and click **Reset**. Select **All**. Click **OK** as shown in Figure 6).



XAPP744_05_090512

Figure 5: **EDK Module Reset**

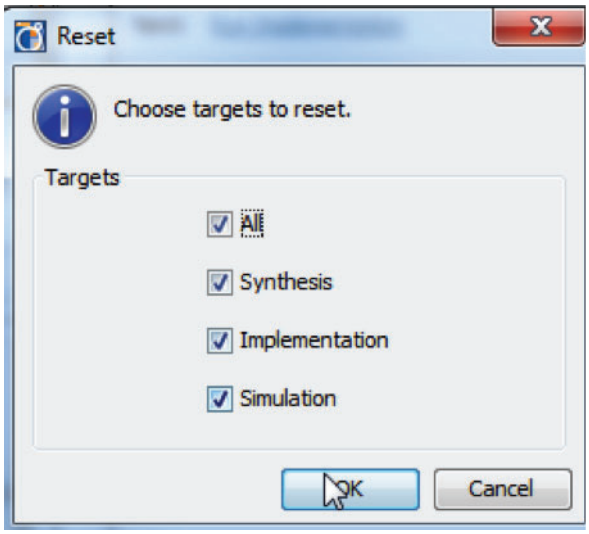


Figure 6: **Reset All Targets**

3. Optional: Double-click the EDK module. This opens the XPS tool. Notice that the same system that is explained in Example System Overview, page 2 is present. Also notice that the timer is present at address `0x42800000`.
4. Click **Run Behavioral Simulation**. Click **Launch** (see Figure 7). This causes these actions occur internally:
   - Launch XPS SimGen tool to generate the simulation sources in the embedded design
   - Issue an ISim fuse command to compile all the simulation source files for simulation
   - Launch the ISim simulator
   - Configure the PL with a pre-compiled bitstream file and set the JTAG port to share the cable

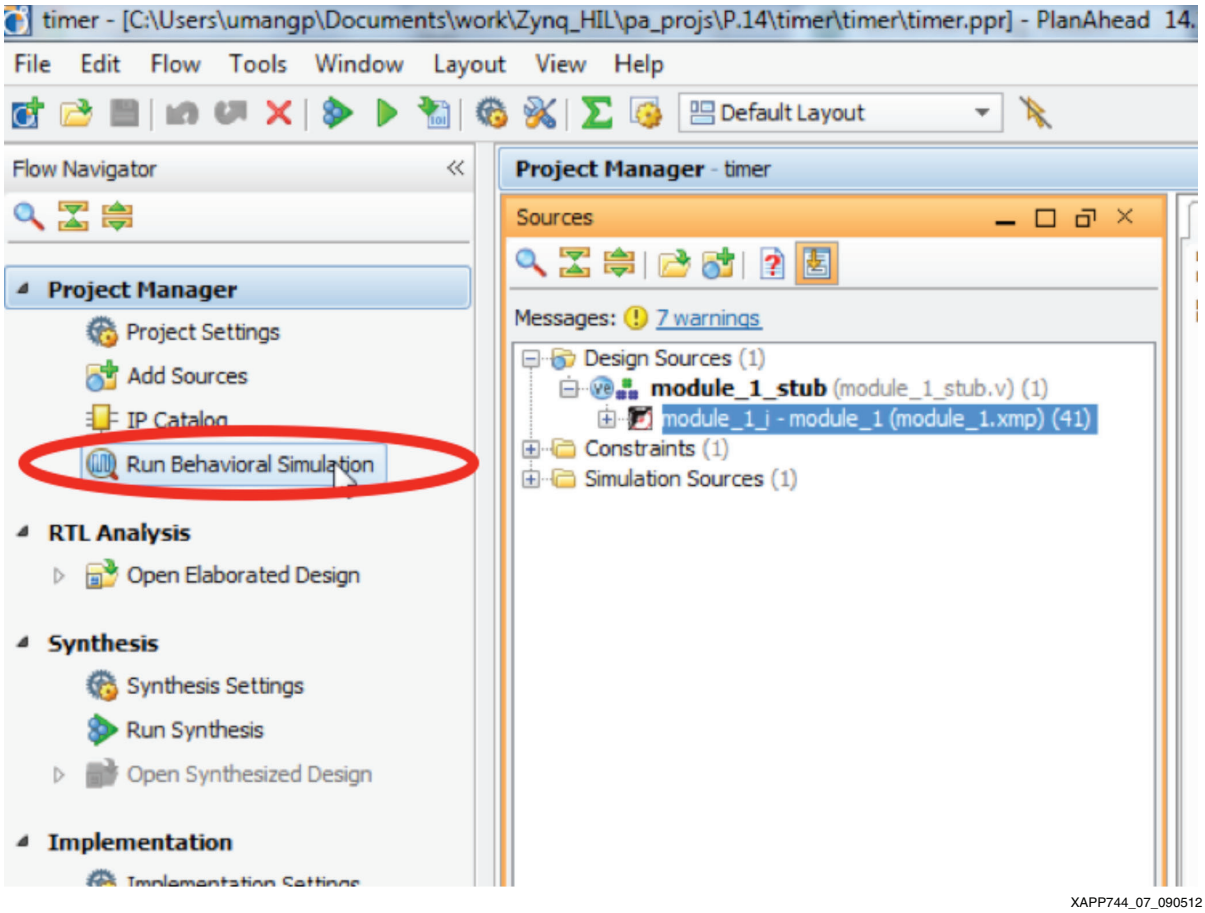
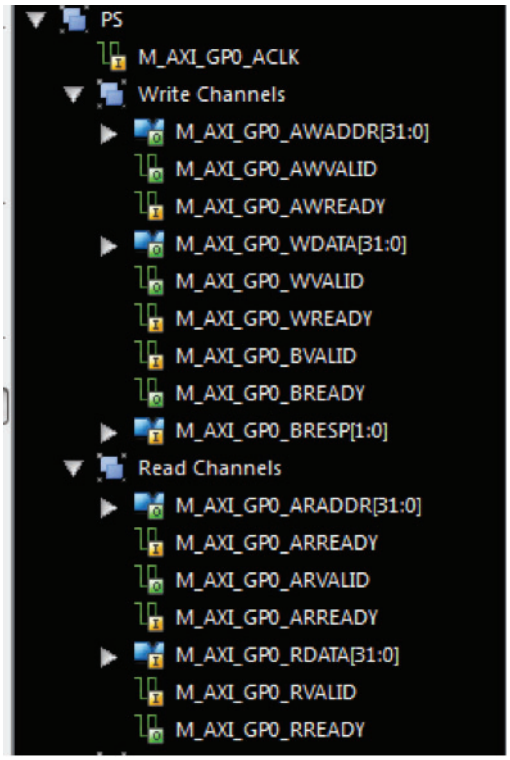


Figure 7: **Run Behavioral Simulation**

5. The ISim simulator window should open. Wait until the bitstream file download is complete. If any errors occur during this process, go to Common Issues and Solutions, page 23. In the waveform viewer, expand PS, Write Channels, and Read Channels as shown in Figure 8.

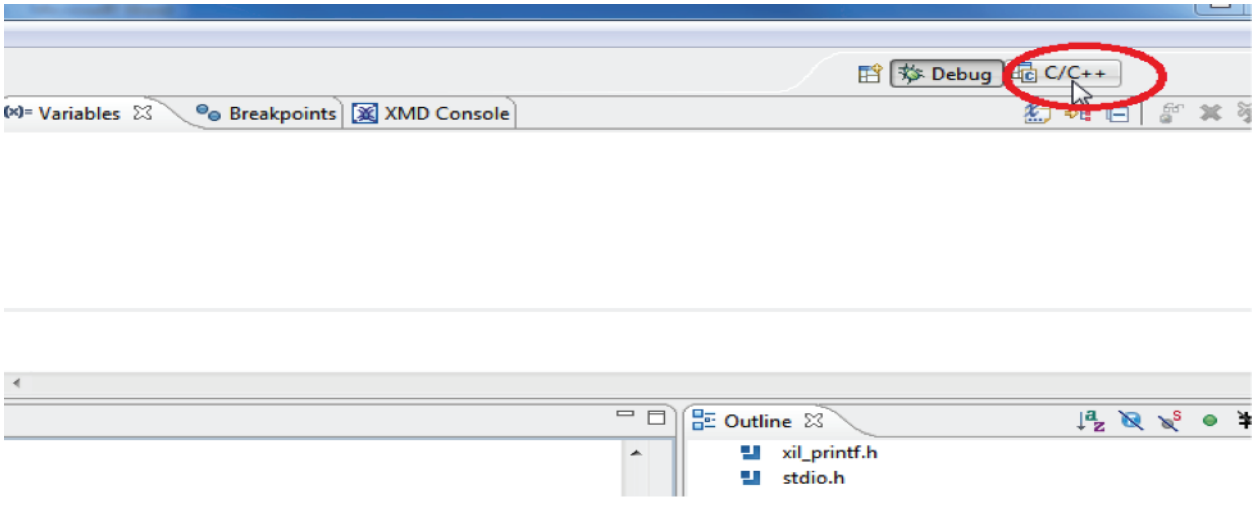


XAPP744_08_090512

Figure 8: **Waveform Viewer**

Setting up Software

1. Open the tutorial PlanAhead tool project file `zynq_hil_timer.ppr`.
2. In the PlanAhead tool project, select **File > Export > Export Hardware**
3. Select both **Export Hardware** and **Launch SDK**. This launches the SDK tool which includes a board support package (BSP) and an application named hello_world_0.
4. If currently in the Debug view, switch to the C/C++ view as shown in Figure 9.



XAPP744_09_090512

Figure 9: **C++ View**

5. Open the `helloworld.c` file as shown in Figure 10 and look through the program. The program performs these functions:
   - Initializes the platform
   - Configures the timer as an up-counter (generates a write operation on the AXI channel)
   - Starts the counter (generates a write operation on the AXI channel)
   - Continues reading the value of the counter until the value of the counter is less than `0x100` (generates a read operation on the AXI channel)
   - Stops the counter (generates a write operation on the AXI channel)
6. Right-click the Hello World application (see Figure 10).

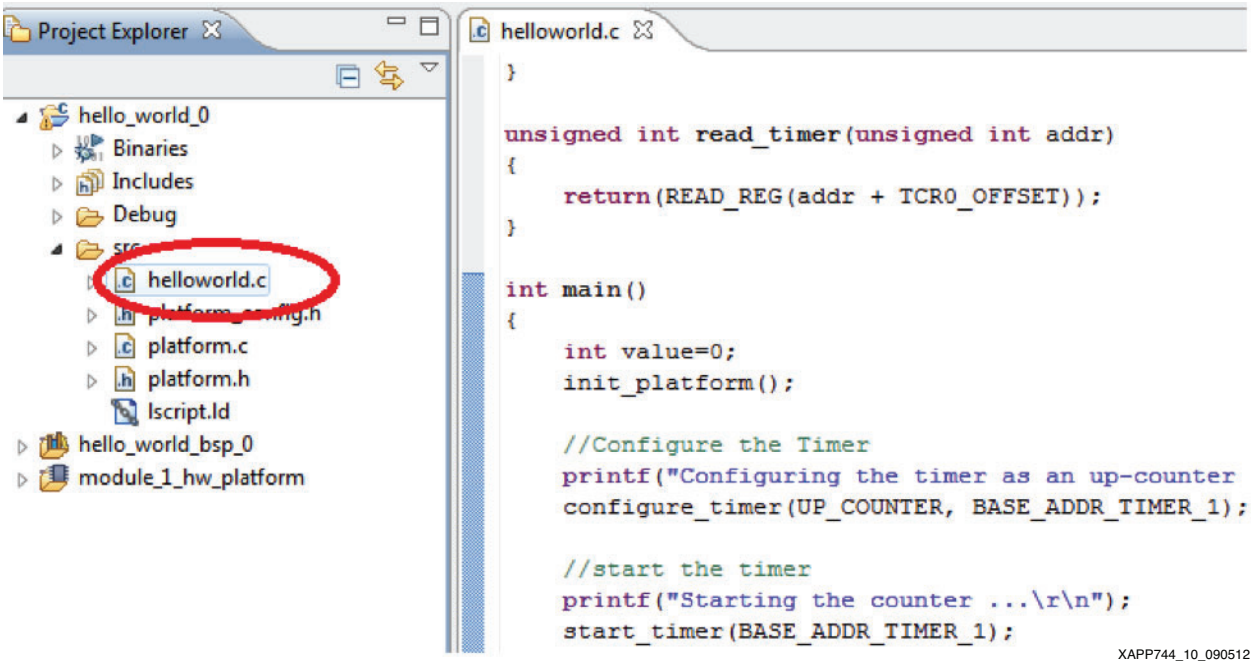


Figure 10: **Hello World Application Selection**

7. Select **Debug As > Debug Configurations** (see Figure 11).

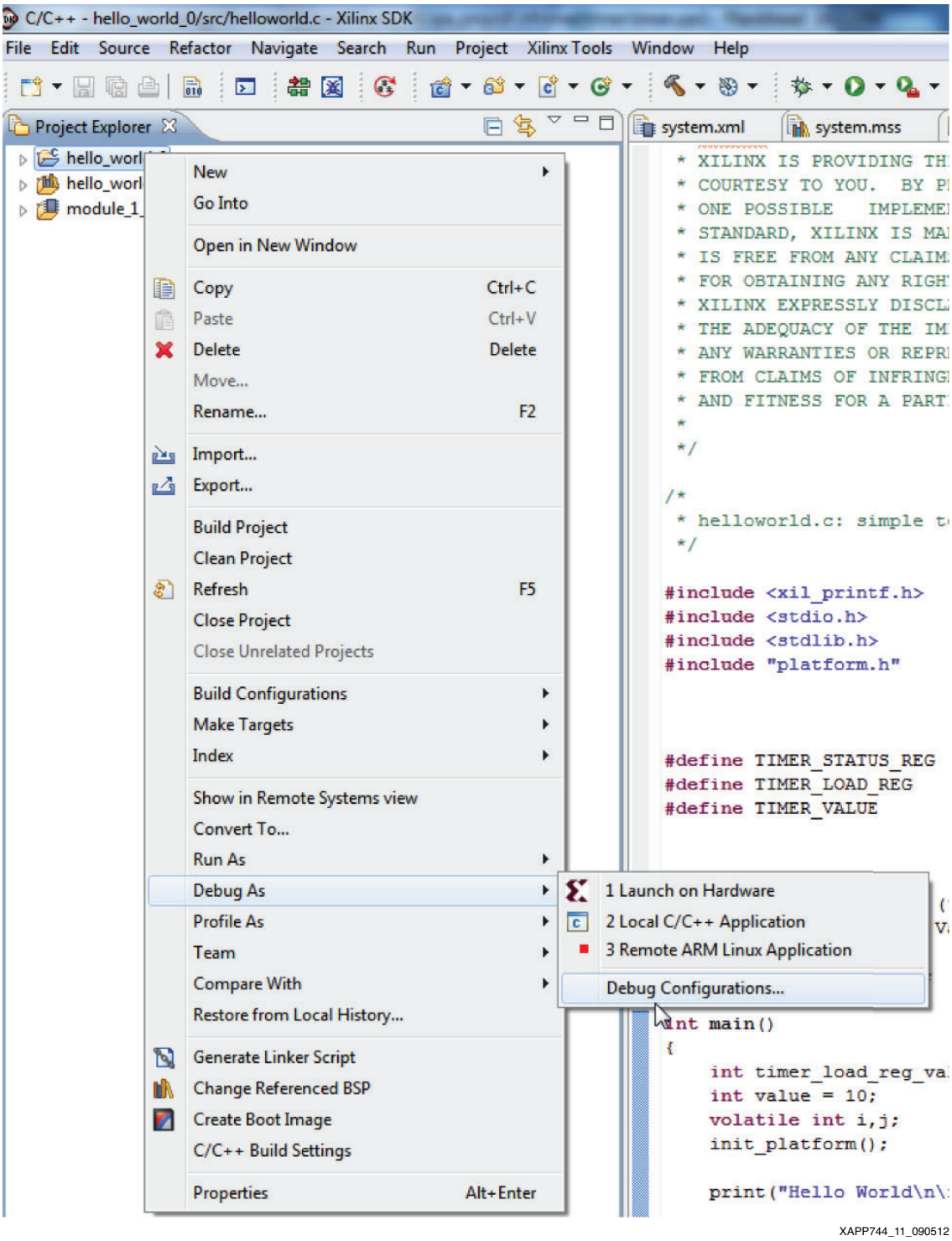


Figure 11: **Debug Configurations Example**

8. Click **Xilinx C/C++ ELF > hello_world_0 Debug**. Under the **STDIO Connection** tab, set the COM **Port** based on your workstation environment and set the **Baud Rate** to **115200**. (COM 4 is shown in Figure 12 as an example.)

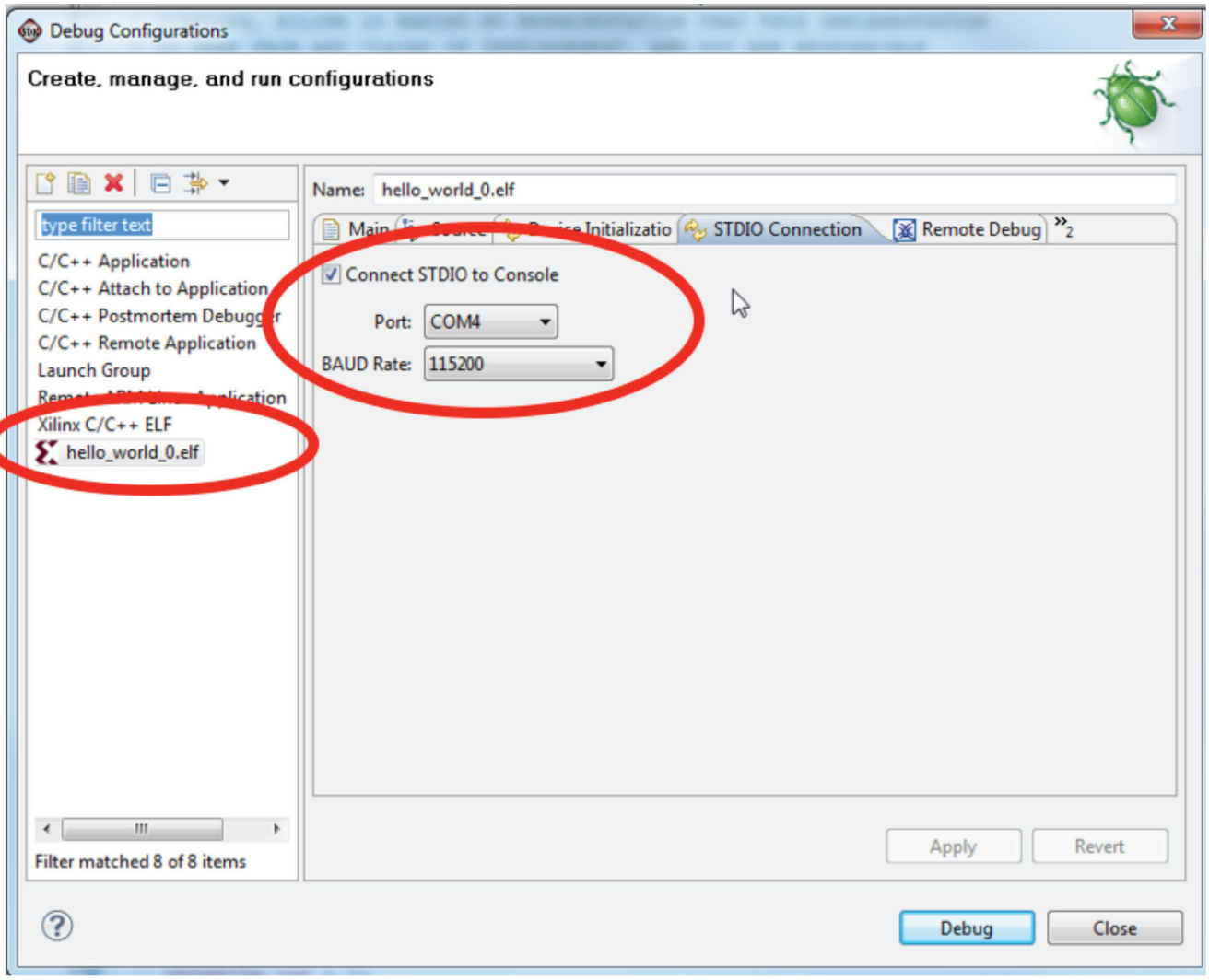
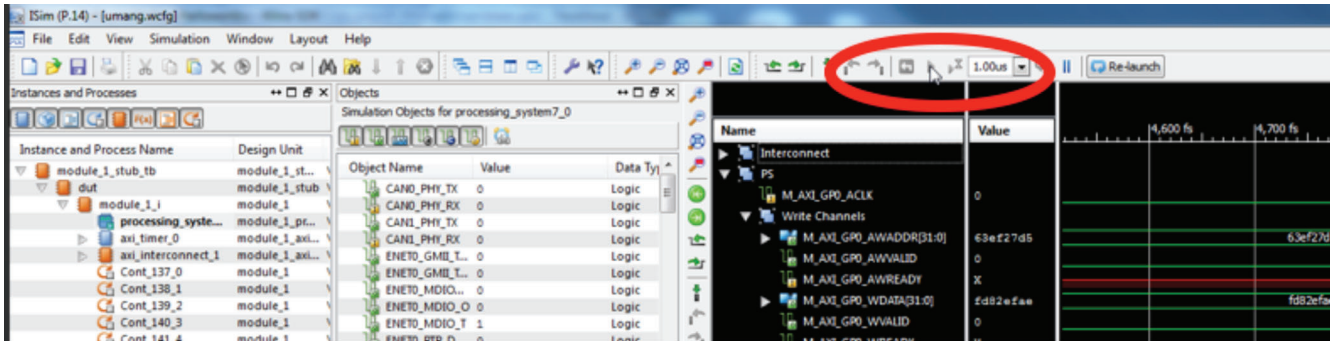


Figure 12: **COM Port Settings**

9. Click the **Device Initialization** tab. Ensure that the path to the `ps7_init.tcl` file is correct. If not, browse to the correct location on your workstation. Also, the reset type should be **Reset Processor Only**.
10. Click **Apply**, then **Debug**.
11. If any error occurs during the debug process, try performing the steps in Common Issues and Solutions, page 23.
12. The compiled code (elf file) is loaded into the PS memory and is ready to be run.

Running HIL Simulation

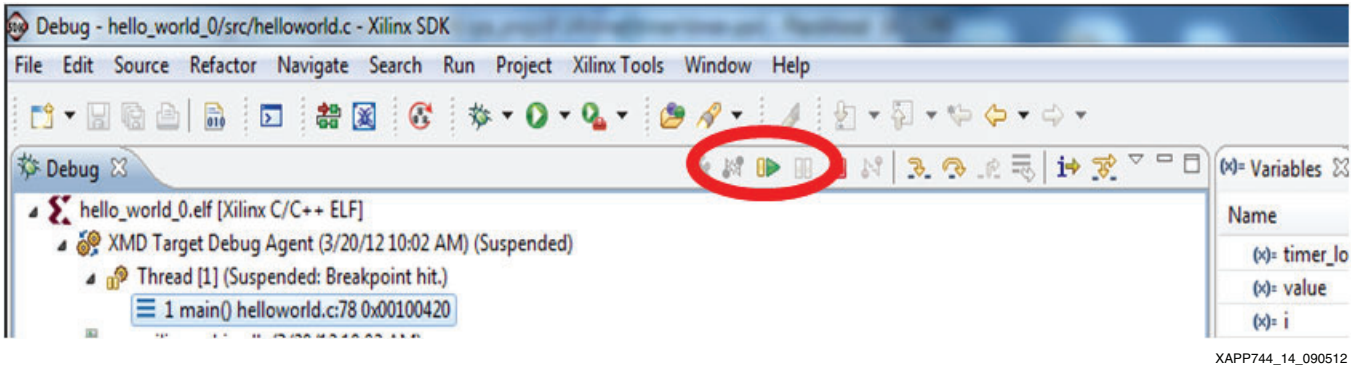
1. In the ISim window, run the testbench. Click the **Run All** button as shown in Figure 13.



XAPP744_13_090512

Figure 13: **Run All Example**

2. In the SDK tool, step through the program by clicking the **Step Over** button as shown in Figure 14.



XAPP744_14_090512

Figure 14: **Step Over Selection**

3. If the COM port settings are configured correctly, the output in the SDK Debug view console window appears as shown in Figure 15.

```
Configuring the timer as an up-counter ...
Starting the counter ...
Read value: 0x7
Read value: 0xf
Read value: 0x17
Read value: 0x1f
```

XAPP744_15_090512

Figure 15: **Debug Console Output Example**

4. Look also at the corresponding write and read transaction in the ISim window. For example, with a write operation (configuring the timer as an up-counter or starting the counter), the corresponding write operation can be seen in the ISim window by zooming into that part, as shown in Figure 16.

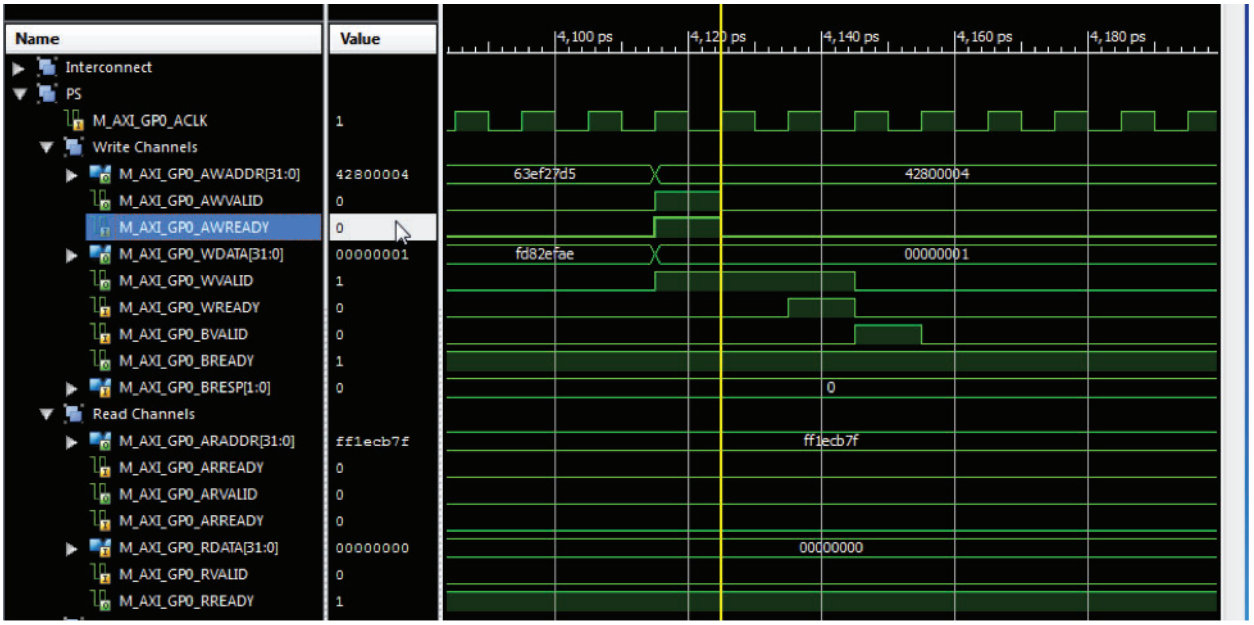


Figure 16: **ISIM Tool Window**

5. Experiment by putting breakpoints in the SDK and ISim tools to enable HW/SW debugging.
6. Once the value of the counter is more than `0x100`, the program sends a stop command to the counter which ends the program, as shown in Figure 17.

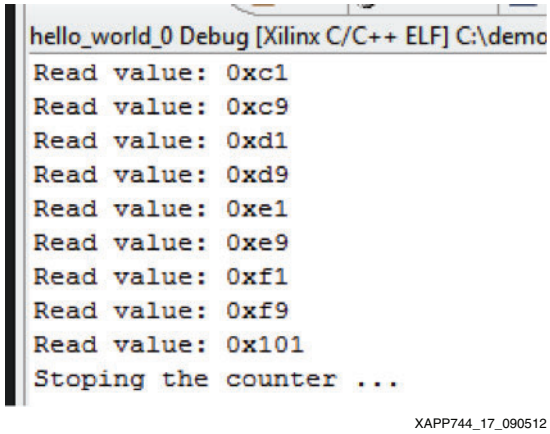


Figure 17: **End of Program Example**

Ending the Simulation

Note: This step is very important because if an XMD or ISIM HIL debug session is not closed properly, the simulation can be prevented from running again.

Follow these steps to ensure that the current Zynq-7000 AP SoC HIL simulation session has ended correctly.

1. To end the Zynq-7000 AP SoC HIL simulation session at any point, click the **stop** button in the SDK Debug view.
2. Go to the ISim tool window and pause the simulation.
3. In the SDK tool, in the Debug view, Click the name of the application being debugged. Right-click the application name and click **Terminate and Remove**. Do this for all of the applications in the Debug window as shown in Figure 18, Figure 19, and Figure 20.

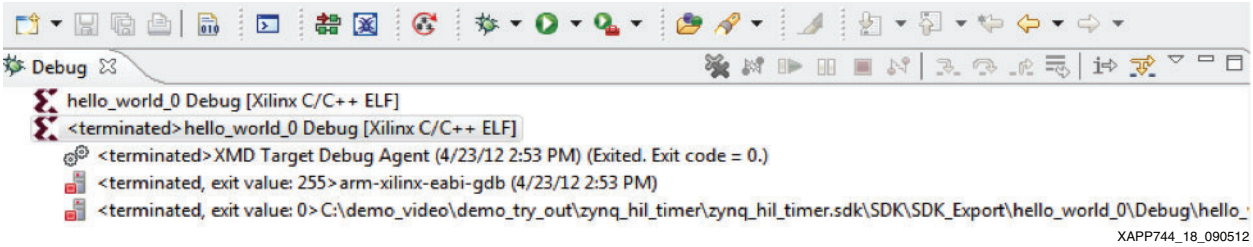


Figure 18: **Terminated Application**

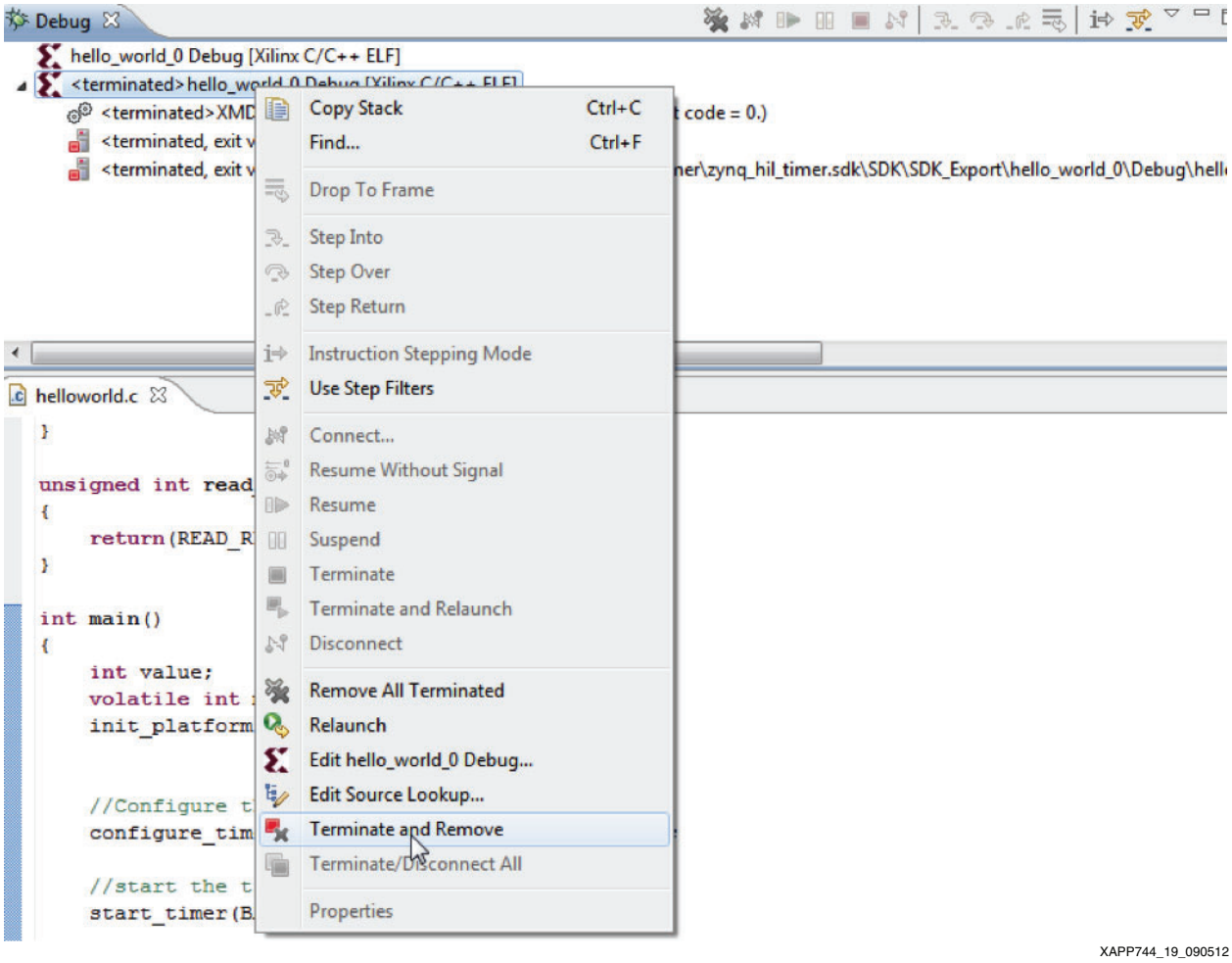


Figure 19: **Terminate and Remove Example 1**

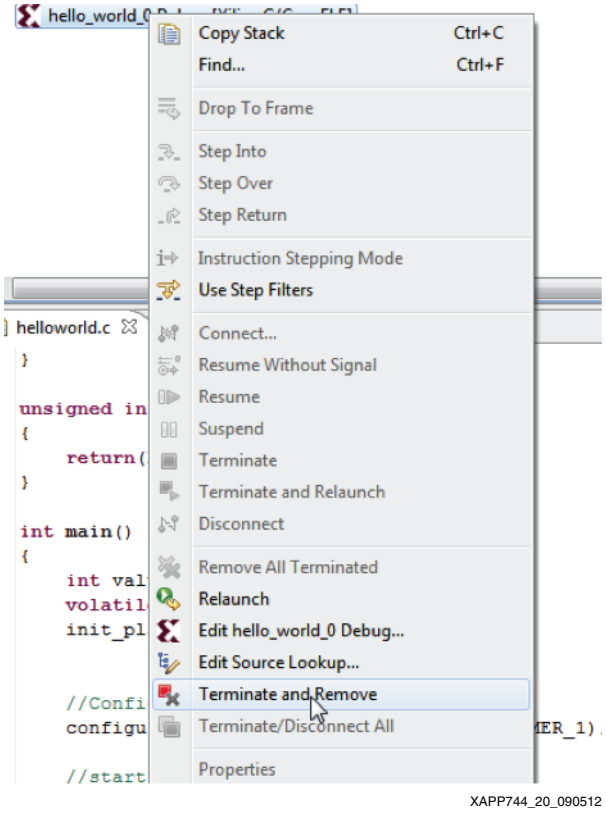


Figure 20: **Terminate and Remove Example 2**

4. Ensure that the Debug view is clean.

Re-running the Simulation

After following all of the steps in Ending the Simulation, page 12, follow these steps to re-run the simulation:

1. Power cycle the device.
2. In the ISim window TCL console, enter the **`restart`** command. Then enter **`run 0ns`**. This downloads the bitstream file.
3. In the SDK tool, launch the debug session for the software application. When the program stops at the first line of code, start the testbench in the ISim tool.
4. Run the program in the SDK tool.
5. If any issues are encountered, follow the steps in Common Issues and Solutions, page 23.

Enabling Zynq-7000 AP SoC HIL Simulation in a Project

Creating a PlanAhead Tool Project With an Embedded Source

1. Create a new PlanAhead tool project for the ZC702 board (see Figure 21).

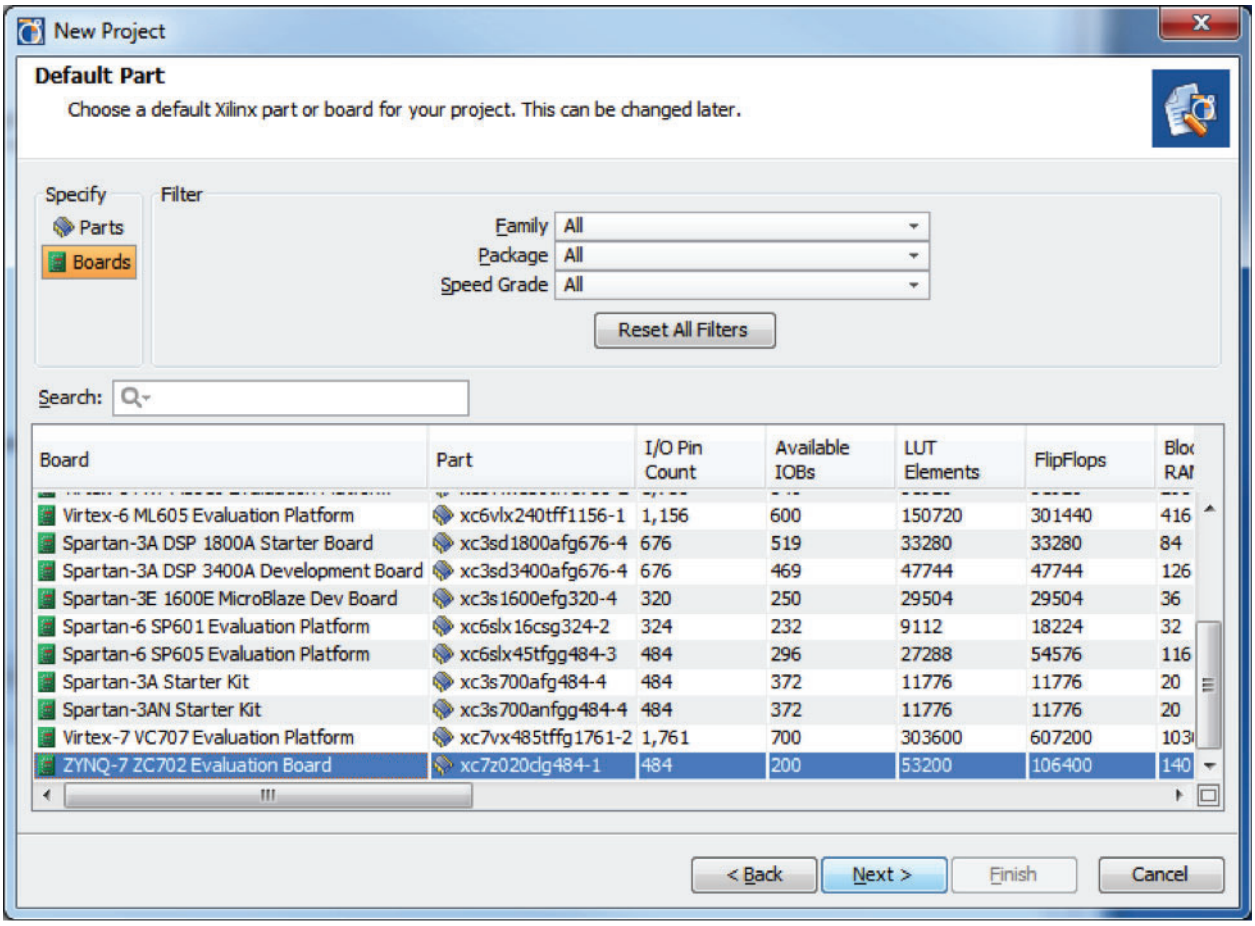


Figure 21: **New PlanAhead Project**

2. Click **Add Sources** (see Figure 22).

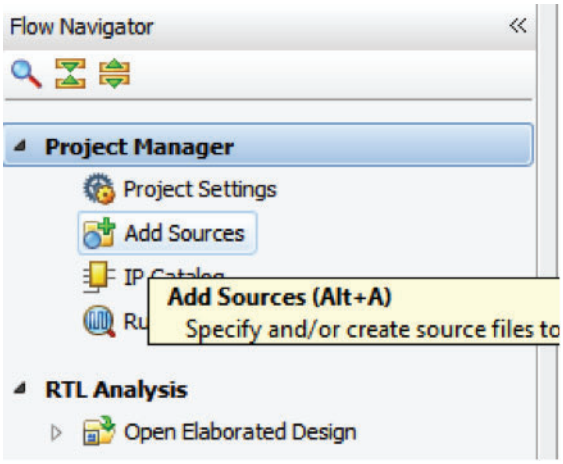


Figure 22: **Add Sources**

3. Select **Add or Create Embedded sources > Next**.
4. Click **Create Sub-Design**. Click **Ok**. Click **Finish**.
5. This starts the XPS tool. The dialog in Figure 23 appears prompting to create a Base System using the BSB Wizard. Select **Yes**.

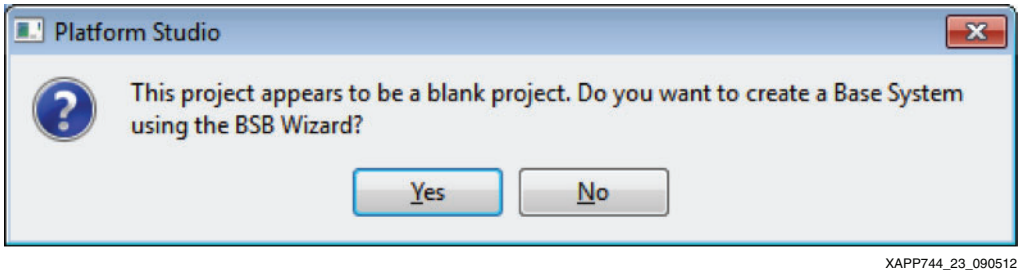


Figure 23: **BSB Wizard Dialog**

6. The New XPS Project dialog shown in Figure 24 appears. Click **OK**.

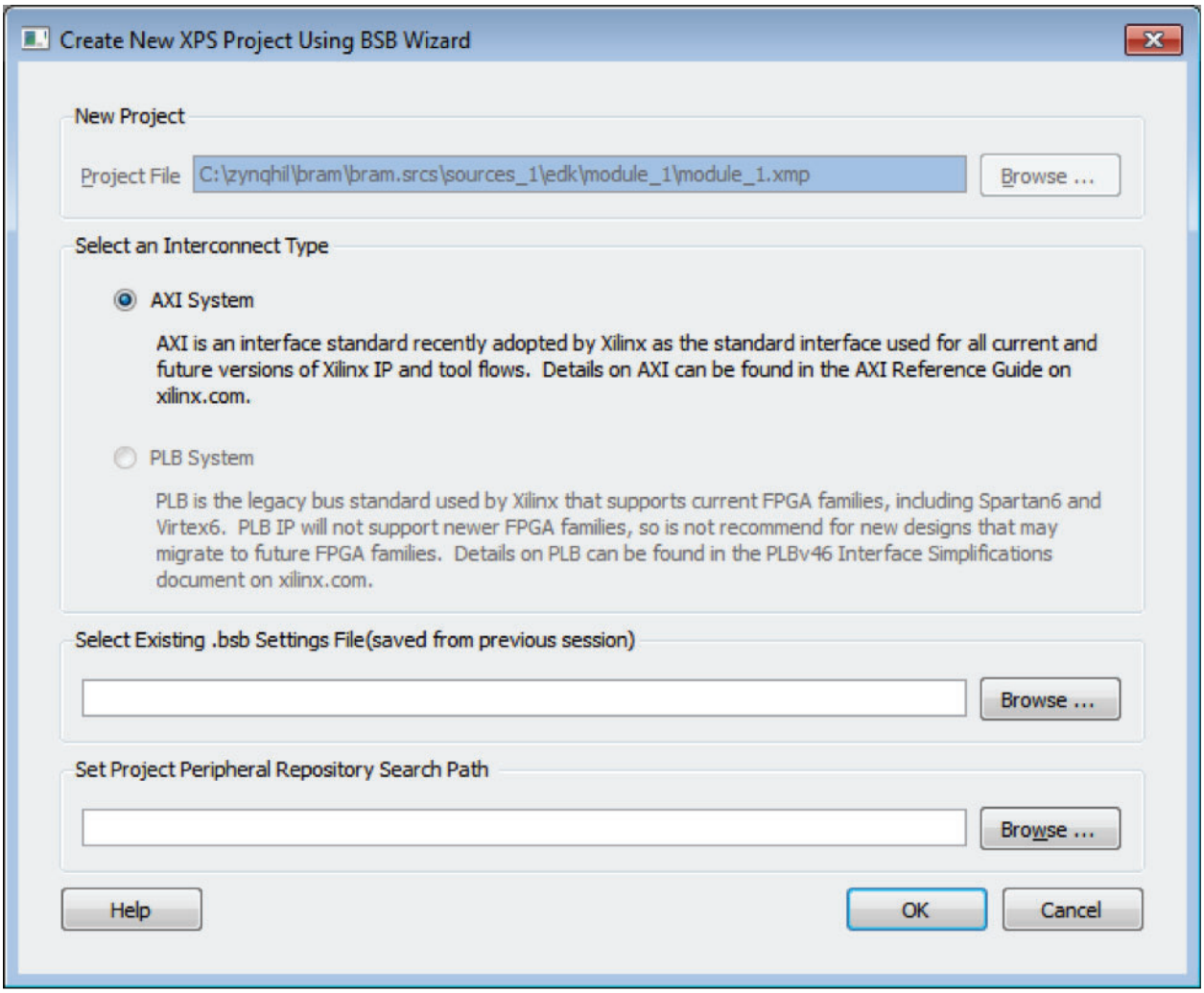
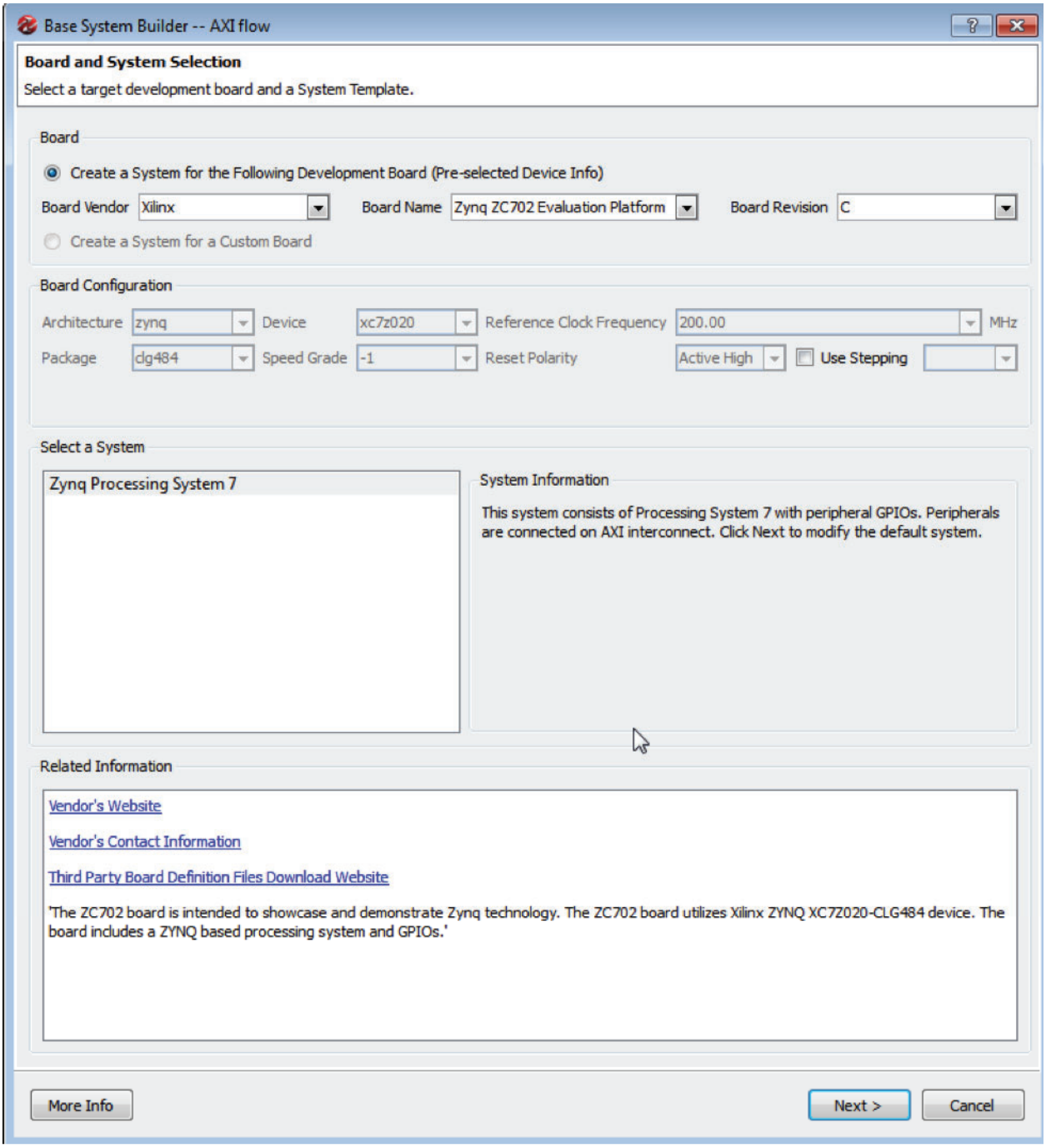


Figure 24: **New XPS Project Dialog**

7. The Base System Builder wizard, Board and System Selection page appears as shown in Figure 25. Click **Next**.



XAPP744_25_090512

Figure 25: **Board and System Selection Page of Base System Builder Wizard**

8. The Peripheral Configuration page appears as shown in Figure 26. Click **Finish**.

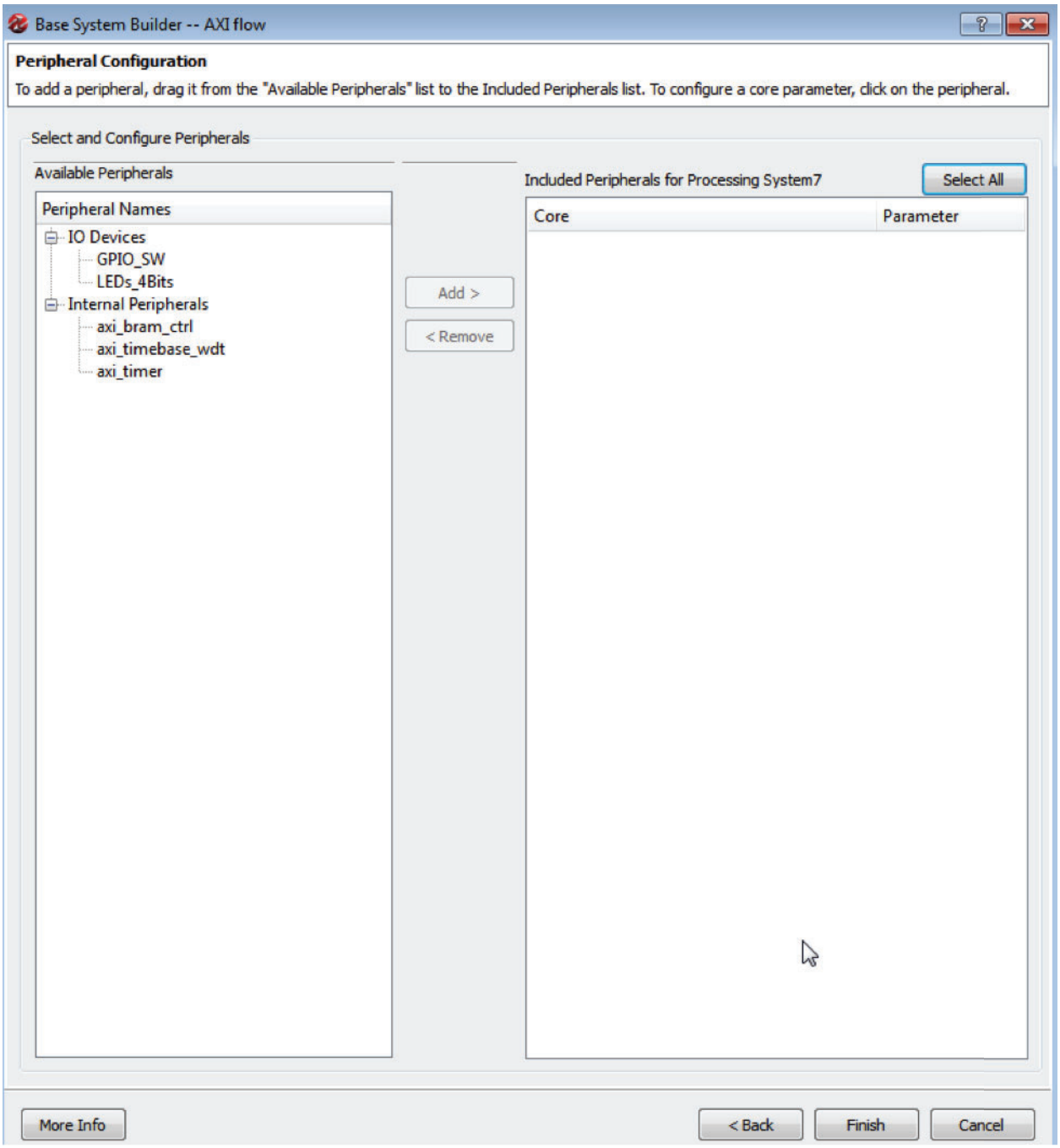


Figure 26: **Peripheral Configuration Page of Base System Builder Wizard**

9. Add the peripherals and build the embedded design.
10. After building the embedded design, Click the **Ports** tab.

11. In the XPS tool, by default, the entire design is driven by the FCLK_CLK0 signal. Right-click FCLK_CLK0 under processing_system7_0 and select **No Connection** as shown in Figure 27. The clock and reset signals are driven through the testbench in ISIM.

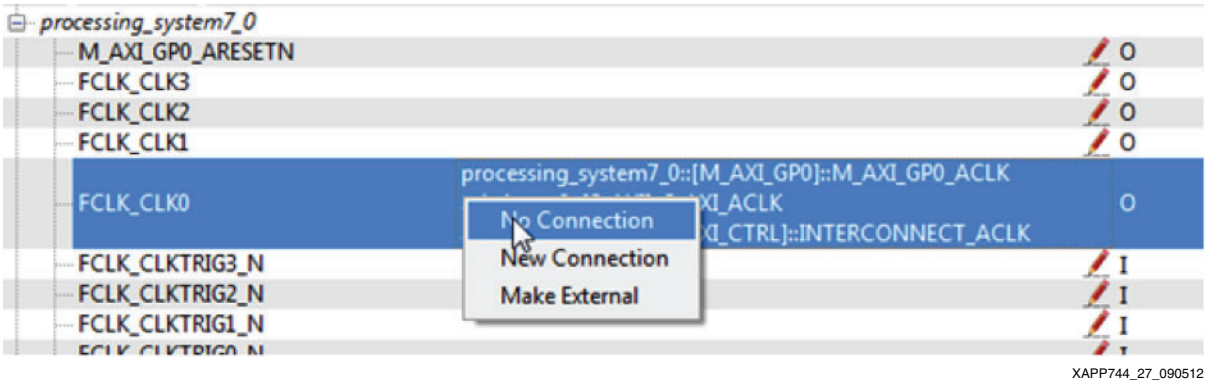


Figure 27: **Setting No Signal Connection**

12. Mark the AXI interface clock and reset signals external by right-clicking the port and selecting **Make External** as shown in Figure 28 and Figure 29. The clock and reset signals are now driven by the testbench.

Note: Currently, Zynq-7000 AP SoC HIL simulation supports only a single clock driving all instances of the AXI interface.

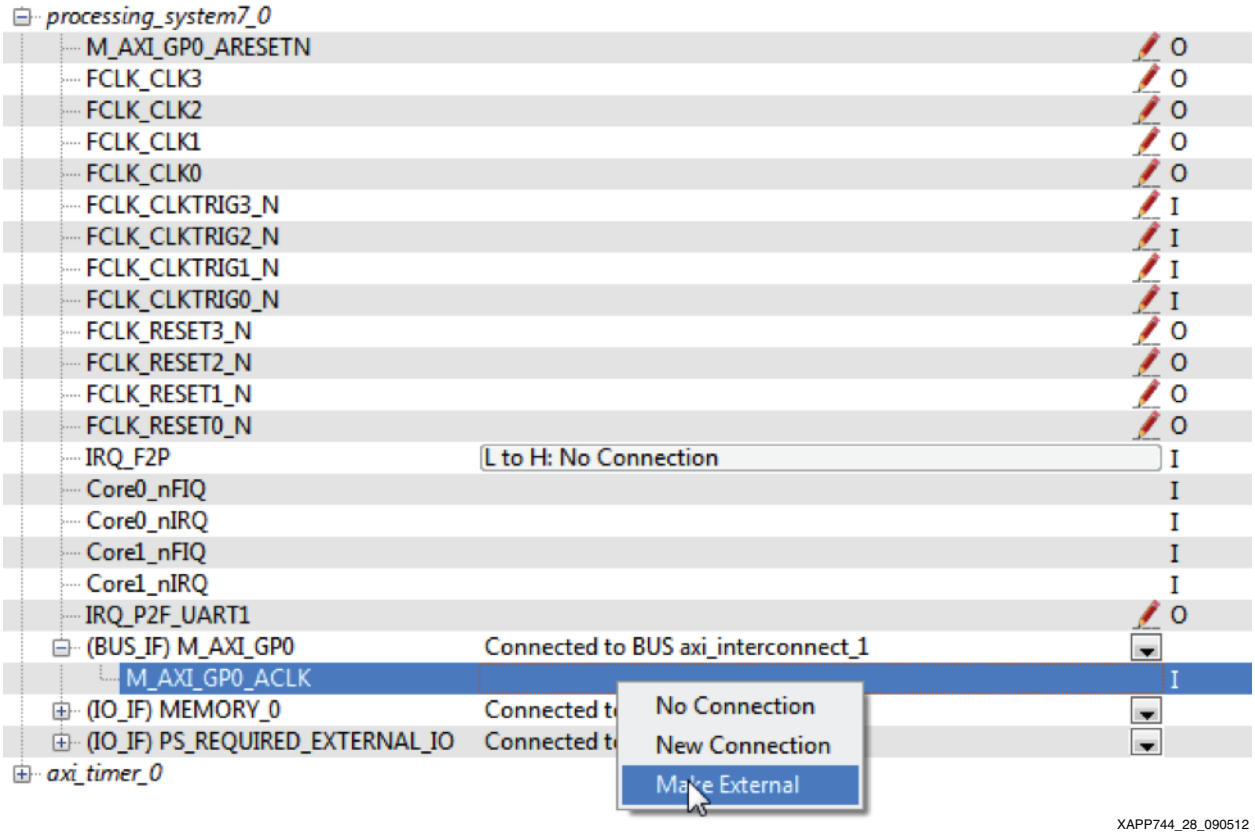


Figure 28: **Setting External Clock Signal Connection**

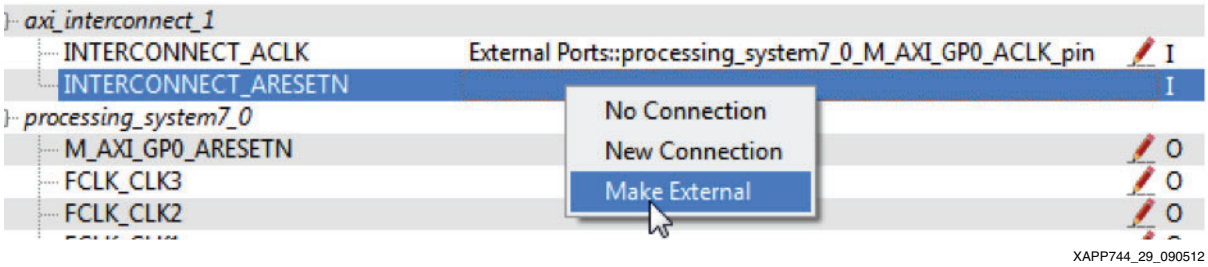


Figure 29: **Setting External Reset Signal Connection**

13. Ensure that there are no errors appearing in the Errors window. When satisfied with the design, exit the XPS tool: Select **File > Exit**. It is recommended to exit the XPS tool using only this method.

Enabling Zynq-7000 AP SoC HIL Simulation Using the PlanAhead Tool

1. Right-click the embedded source (.xmp) file name and click **Create Top HDL** as shown in Figure 30. When the Top HDL is created, ensure that the ports that were marked external in the EDK tool show up in the Top HDL. These signals are driven from the testbench.

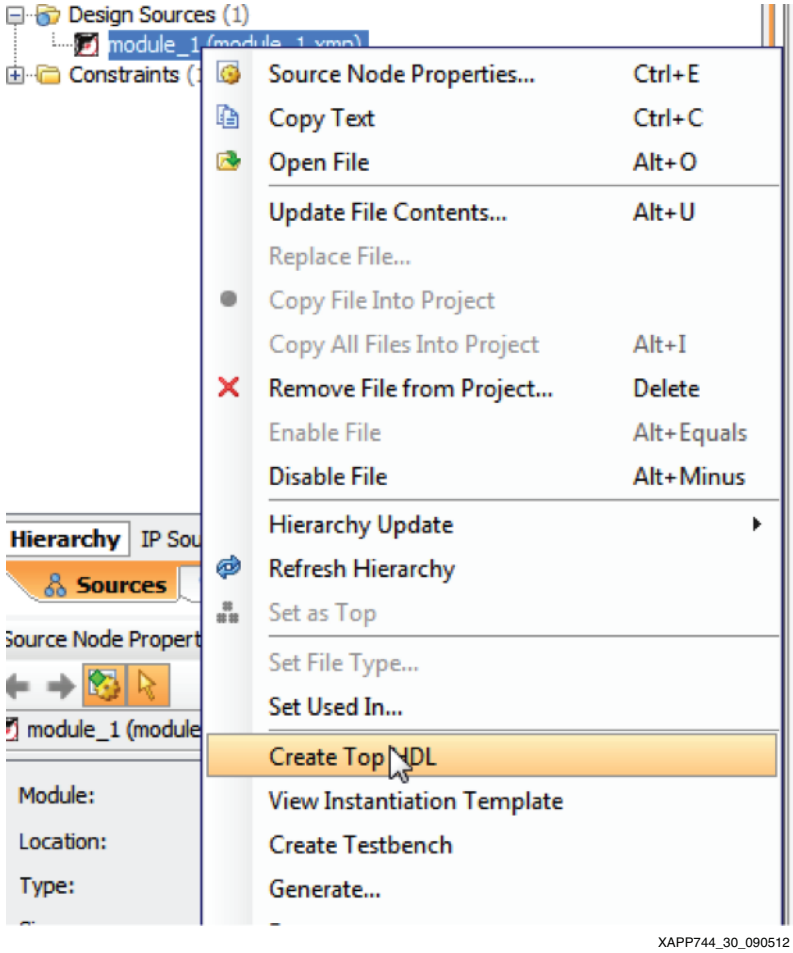


Figure 30: **Creating Top-Level HDL File**

2. Click **Add Sources > Add or Create Simulation Sources**. Use the sample testbench provided with the example project as reference or write a custom testbench. Ensure that the clock and the reset ports are driven from the embedded design by the clock and the reset produced by the testbench.
3. Click **Run Behavioral Simulation > Options**.
4. In **More Fuse Options** under the **Compilation** tab, set these options:
   - **`-hwcosim_instance <heirarchichal path>/processing_system7_0`**

     Path example:
     `/module_1_stub_tb/dut/module_1_i/processing_system7_0`

     The `-hwcosim_instance` option informs ISim that the instance is in hardware. A complete hierarchical path to the PS is needed for this option. This option can vary for different designs.
   - **`-hwcosim_clock M_AXI_GP0_ACLK`**

     This option remains unchanged for all user designs in 14.2. This option is used as a workaround for 14.2. Details about this option are beyond the scope of this document.

     Starting with 14.2.1, Zynq-7000 AP SoC HIL simulation supports multiple clocks. Therefor depending on the interface used in the design, the **`-hwcosim_clock`** option accepts all of the clocks. For example:

     -hwcosim_clock M_AXI_GP0_ACLK,M_AXI_GP1_ACLK,S_AXI_GP0_ACLK,S_AXI_GP1_ACLK,S_AXI_ACP_ACLK,S_AXI_HP0_ACLK,S_AXI_HP1_ACLK,S_AXI_HP2_ACLK,S_AXI_HP3_ACLK (If all the ports are driven by separate clocks)

     -hwcosim_clock M_AXI_GP0_ACLK,M_AXI_GP1_ACLK (If only M_AXI_GP0 and M_AXI_GP1 ports are being used and driven by separate clocks)
   - **`-hwcosim_board zc702-jtag`**

     This option is used to determine which board is being targeted. For Zynq-7000 AP SoC HIL simulation, this option cannot be changed.
   - **`-hil_zynq_ps`**

     This option is used by the hardware co-simulation compiler to distinguish between traditional ISim tool hardware cosim and Zynq-7000 AP SoC HIL simulation. For Zynq-7000 AP SoC HIL simulation, this option cannot be changed.
5. Create a tcl script with the commands shown. Use the tcl script provided with the example project for reference.

   ```
   run 0ns;
   ```
6. On the **Simulation** tab:
   a. Clear the Simulation Run Time section
   b. Under **`-tclbatch`**, give the location of the TCL script that was created in step 5.
7. Click **Launch**. This launches the ISim tool and downloads the bitstream file.

Software

Depending upon the type of IP in a design, software might be created for the design. When the executable (elf file) for the software is created, follow the steps to download and run the program in Setting up Software, page 7.

Zynq HIL Multiple Clock

The Zynq-7000 AP SoC HIL has the ability to drive different AXI ports of the PS with different clocks. Potentially, all of the AXI ports (general purpose/high performance/ACP) can be driven by separate clocks. An example/reference project is located in the zip file obtained from:

https://secure.xilinx.com/webreg/clickthrough.do?cid=197021

For this tutorial unzip and use `zynq_HIL_timer_mulitple_clock.zip`.

In this project, there are two AXI timers connected to M_AXI_GP0 and M_AXI_GP1 ports. Both the timers are driven with two separate clocks. Running Zynq-7000 AP SoC HIL on this project follows similar steps as the single clock solution.

Differences between the single clock solution and multiple clock solution is under *More Fuse Options.*

Single clock:

```
-hwcosim_clock M_AXI_GP0_ACLK
```

If all ports are driven by separate clocks:

```
-hwcosim_clock M_AXI_GP0_ACLK,M_AXI_GP1_ACLK,S_AXI_GP0_ACLK,S_AXI_GP1_ACLK,S_AXI_ACP_ACLK,S_AXI_HP0_ACLK,S_AXI_HP1_ACLK,S_AXI_HP2_ACLK,S_AXI_HP3_ACLK
```

If only M_AXI_GP0 and M_AXI_GP1 ports are used and are driven by separate clocks:

```
-hwcosim_clock M_AXI_GP0_ACLK,M_AXI_GP1_ACLK
```

Advantages

Zynq-7000 AP SoC platform HIL simulation has these advantages:

- Much faster simulation speed than RTL-level simulation.
- Pre-configured bitstream file for the PS.
- Cycle-accurate simulation of the PL portion of the design.
- Full visibility of the IP in the PL section using the ISim tool.
- Quick design turns for both hardware and software.

Limitations

Zynq-7000 AP SoC platform HIL simulation has these limitations:

- Limited only to AXI connections between the PS and the PL.
- No support for multiple clocks. All the AXI interfaces must be connected to the same input clock derived from the testbench.
- No support for interrupts.
- Only the PS is simulated on the ZC702 board. Unlike traditional hardware co-simulation, some pieces of PL cannot be placed in the hardware.
- The designer must modify the design to bring out the clock and reset in the XPS tool.

Common Issues and Solutions

Avoid XMD-Related Errors

Take these precautionary steps to ensure that the simulation runs smoothly:

1. In the SDK tool, Debug view, right-click the name of the application being debugged, and then click **Terminate and Remove**.
2. In the process explorer, make sure that there are no unwanted XMD sessions already open. This can prevent further use of the XMD tool.
3. Make sure that the Eclipse > XMD session from the SDK tool is terminated correctly in process explorer.

When The XMD Connection Fails

- Exit the SDK tool and try again.
- Exit the SDK and ISim tools and try again.
- Exit the SDK and ISim tools, power off the board and try again.
- Exit the SDK and ISim tools, power off the board, unplug the USB cable from the platform USB connector and try again.
- Power off the board, close all of the applications and try again.
- Power off the board, restart your workstation and try again.

Avoid Single Stepping in The Software

For sections of the program that do not generate an AXI transaction, single stepping the software should be fine. But, a cable locking error might occur if trying to single step (step in or Step over) over sections of code that generate AXI transactions. Instead of single stepping, try to place break-points and then resume debugging the software from that point on.

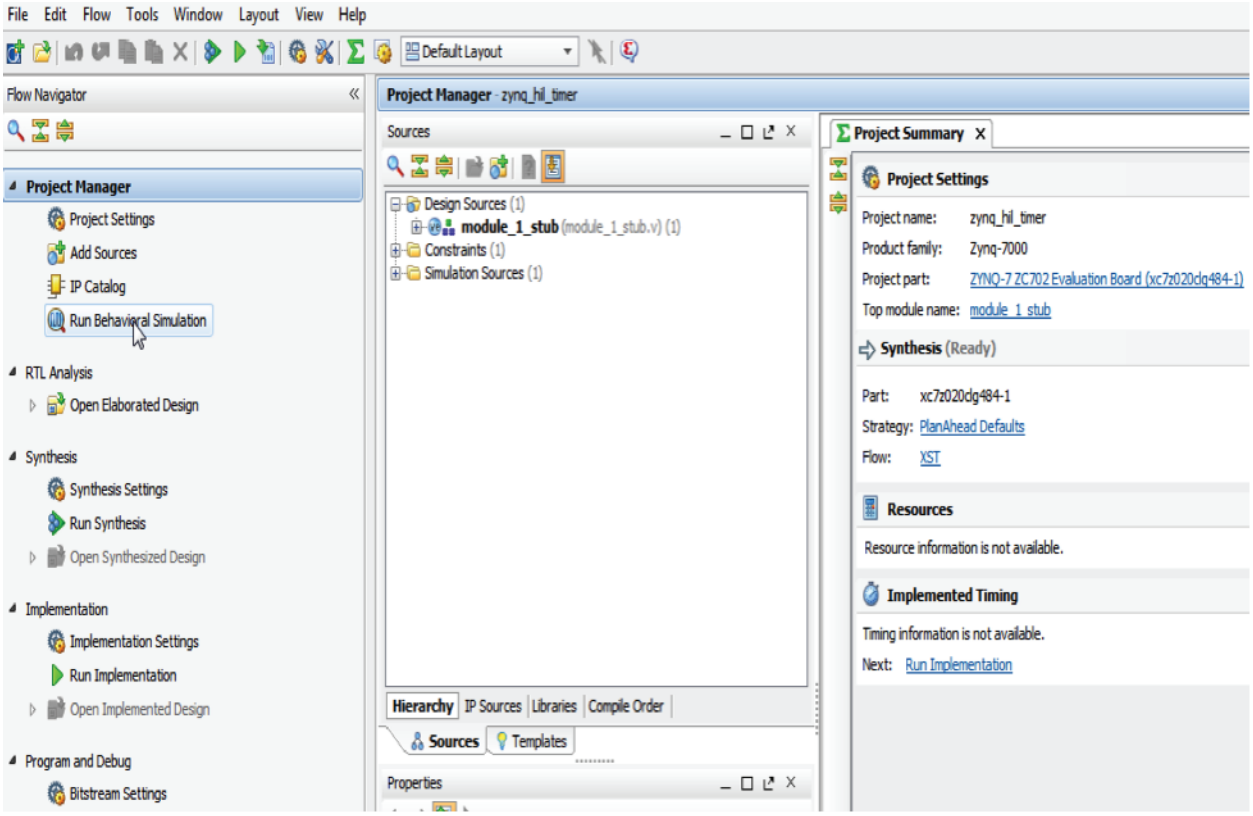
Known Issues and Workarounds

The Zynq-7000 AP SoC HIL Simulation Does Not Work on Windows 32-bit XP Computers

Note: Zynq-7000 AP SoC HIL does not work on Windows 32-bit XP computers in v14.2. A fix has been released in v14.2.1. Therefore this section is not applicable for users with v14.2.1 and above.

ISim crashes when downloading the bitstream file. This issue occurs when the Hwcosim routine attempts to download the bitstream file. The workaround is to use the iMPACT tool to download the bitstream file:

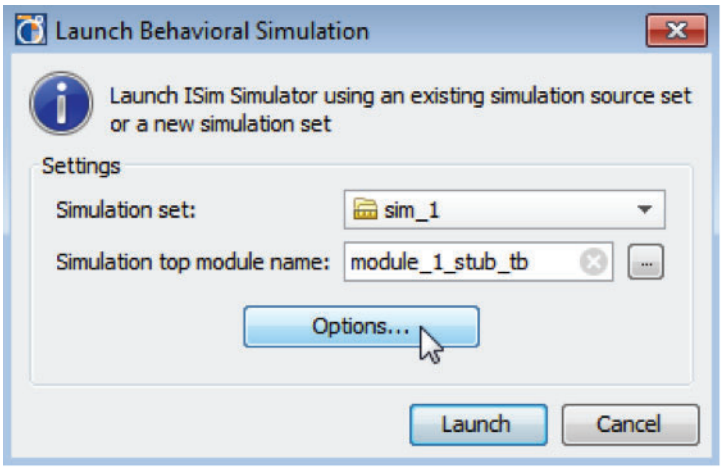
1. Open the PlanAhead tool Project.
2. Click **Run Behavioral Simulation** as shown in Figure 31.



XAPP744_31_090412

Figure 31: **Run Behavioral Simulation**

3. The Launch Behavioral Simulation dialog appears as shown in Figure 32. Click **Options**.



XAPP744_32_090512

Figure 32: **Launch Behavioral Simulation Dialog**

4. Click the **Simulation** tab as shown in Figure 33.

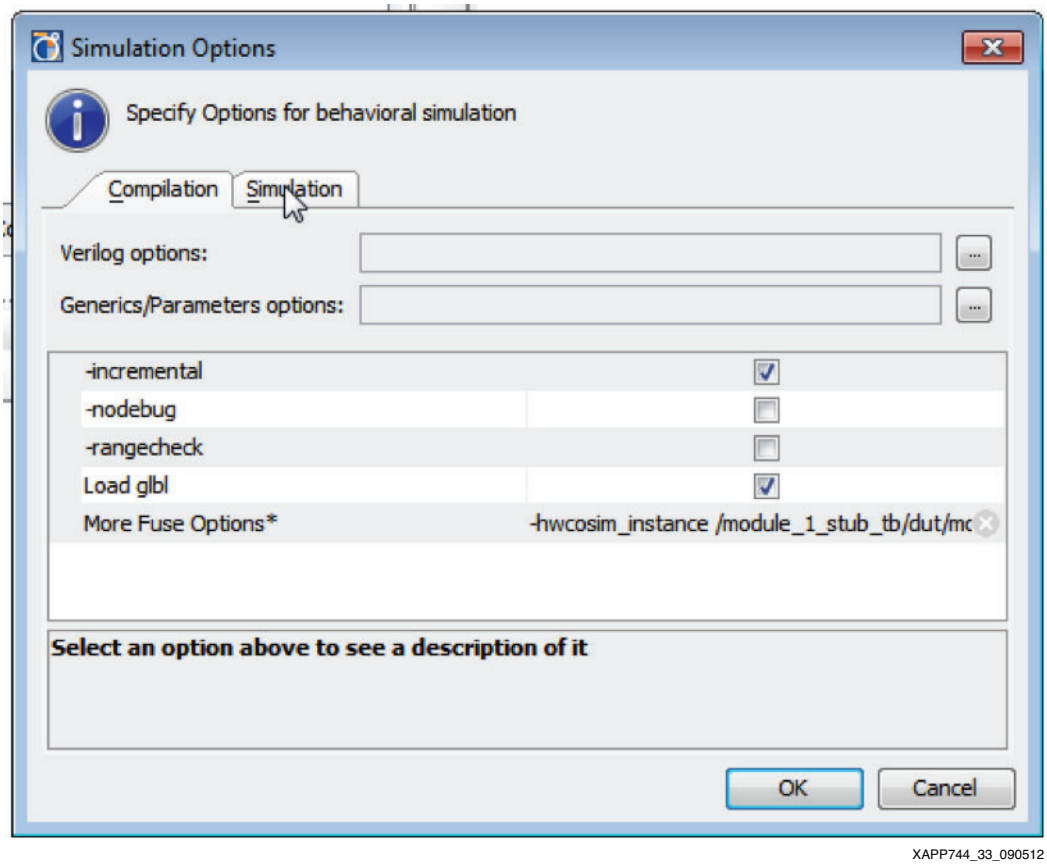


Figure 33: **Simulation Options Dialog**

5. Clear the `zynqHIL.tcl` command file name from the –tclbatch target field as shown in Figure 34.

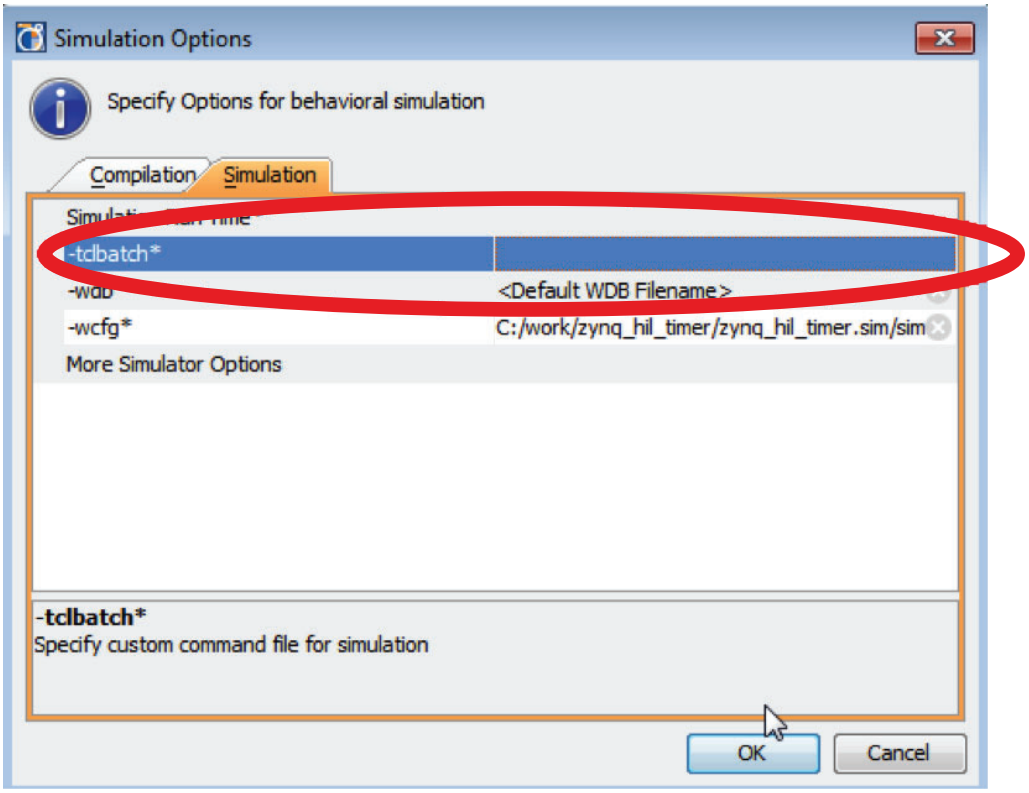


Figure 34: **Simulation Options**

6. Click **Launch**.
7. The ISim tool window opens.
8. Download the bitstream file located in `<PlanAhead Project area>\zynq_hil_timer.sim\sim_1\isim\hwcosim` using the iMPACT tool.
9. In the ISim tool tcl Console, enter these commands sequentially:

```
scope module_1_stub_tb/dut/module_1_i/processing_system7_0
hwcosim set shareCable 1
hwcosim set skipConfig 1
run 0ns
```

10. Continue with Setting up Software, page 7.

Revision History

The following table shows the revision history for this document.

Date	Version	Description of Revisions
09/27/12	1.0	Initial draft.
10/12/12	1.0.1	Fixed broken link to tutorial ZIP file.
11/02/12	1.0.2	Placed tutorial ZIP file behind click-through license.

Notice of Disclaimer