Getting Started with the EDK

Summary

This document provides an introduction to the Embedded Development Kit (EDK).

EDK Contents

The Embedded Development Kit is distributed as a single media installable CD image.

The components of the Xilinx EDK are:

- Hardware IP for the Xilinx embedded processors and its peripherals
- Embedded System Tools (EST)
- Documentation

The Xilinx EDK does not include printed documentation material. Please refer to the "Documentation" section for the included electronic documents. Also not included, but available as separate products are an FPGA development board and the Xilinx FPGA implementation tools ISE 5.1i. Please see the "Requirements" section, and the "Development Boards" section for further details.

Requirements

Several other products are required in addition to the Xilinx EDK:

- Xilinx ISE 5.1i
  - The Xilinx FPGA design implementation tools ISE 5.1i are required to implement embedded designs generated with the tools of the EDK.
  - Several EST applications from the Xilinx EDK invoke functionality delivered with tools in ISE 5.1i
  - Updates to ISE 5.1i including service packs are available at **http://support.xilinx.com/support/techsup/sw_updates**
- Patch to the Xilinx ISE tools. This patch is only required for ISE 5.1i and ISE 5.1i SP1 and is available on the EDK CD.
- Development Board
  - To test your MicroBlaze or PowerPC system on an FPGA, you must have access to a development board which contains a Xilinx FPGA and several other components as well as standard download, configuration and debug connectors

Supported Platforms

Operating Systems

The Xilinx EDK is available for the following operating system platforms:

- Windows 2000
- Solaris 2.8

Xilinx FPGA Families

The Xilinx EDK supports designing MicroBlaze embedded processor systems for several FPGA families

- Xilinx Spartan-II FPGAs (XC2S50 or larger devices)
- Xilinx Virtex/E FPGAs (XCV50 or larger devices)
- Xilinx Virtex-II FPGAs (XC2V250 or larger devices)

The Xilinx EDK also supports designing MicroBlaze and PowerPC embedded processor systems for

- Xilinx Virtex-II Pro FPGAs

Development Boards

Several development boards are available from Xilinx partners. Currently available boards include:

- Avnet Spartan-II Evaluation Kit (Part #: ADS-XLX-SP2-EVL)
- Avnet Virtex-E Evaluation Kit (Part#: ADS-XLX-VE-EVL)
- Insight Virtex-E Demo Board
- Insight Spartan-II Demo Board
- Insight Virtex-II MicroBlaze Board
- Digilent Spartan-IIE Board
- Xilinx ML300 Board
- Xilinx AFX Board

Please contact your local Avnet, Insight or any other authorized distributor to obtain any of these development boards.

Installation on Windows

This section provides a summary of the Xilinx EDK installation process on the Windows platform. Please refer to *<edk_install_dir>\doc\installation.htm* after installing the Xilinx EDK for detailed installation and setup instructions.

Registration

A software registration ID is required for installation. It can be obtained from http://www.xilinx.com/ise/embedded/EDK/register.htm. This requires logging in and

providing Software Product Information (including Product ID). The Software Registration ID will then be e-mailed to the address provided during login.

Installing Xilinx EDK

- Insert the EDK Installation CD in your PC. The installer should automatically pop up.
- If the installation process does not start on its own, open the windows explorer and double click on **setup.exe** on the CD.
- The installation process will prompt you to obtain the registration ID.
- Once you have the registration ID, please continue to install the product.
- The default EDK installation directory is ***c:\EDK*** and can be changed to any other directory.
- The ISE patch can be installed by clicking on "Install ISE Patch" link in the installer.

Note: Destination Directory can not have spaces in its name.

Installation on Solaris

This section provides a summary of the Xilinx EDK installation process on the Solaris platform. Please refer to ***<edk_install_dir>/doc/installation.htm*** after the Xilinx EDK installer has finished for detailed installation and setup instructions.

Registration

A software registration ID is required for installation. It can be obtained from http://www.xilinx.com/ise/embedded/EDK/register.htm. This requires logging in and providing Software Product Information (including Product ID). The Software Registration ID will then be e-mailed to the address provided during login.

Installing Xilinx EDK

- Insert the CD in your solaris machine.
- Change directory to the CD home.
- Execute ***install_solaris.csh*** to install the EDK on solaris
- The installer prompts for the Registration ID, which you should have obtained from web-site indicated above.
- The default directory for installation is ***${HOME}/EDK***, but can be changed during installation.
- At the end of EDK installation, the install script will prompt for installation of ISE patch. Please install this to the XILINX installation area.

Simulation Libraries

Most simulators require you to compile the HDL libraries before you can use them for design simulations. The advantages of compiling HDL libraries are speed of execution and economy of memory.

Xilinx provides an application, to specifically compile the HDL libraries for all Xilinx-supported simulators. This utility will compile the UNISIM, XilinxCoreLib and SIMPRIM libraries for all supported device architectures.

Library compilation

To compile your HDL libraries using *compxlib*, follow these steps:

1. Set the XILINX environment variable (if not already set):
   - Unix

     setenv XILINX path_to_xilinx_software
   - Windows

     Open the "System Properties / Environment Variables" dialog box and set the XILINX to path_to_XILINX_software
2. Add $XILINX / bin / sol to the PATH variable (if not already set):
   - Unix

     set path = ($XILINX / bin / platform $path)

     Note: Platform can be either sol for 32-bit Solaris, sol64 for 64-bit Solaris or nt if using Windows or Linux OS.
   - Windows

     In the "System Properties / Environment Variables" dialog box, add %XILINX% \ bin \ nt to the PATH variable.
3. Run *compxlib* with -help option to display a brief description for the available options.

   ```
   compxlib -help
   ```
4. Run *compxlib* tool using the following syntax:

   ```
   compxlib -s <simulator> -f <family[:lib],<family[:lib],...|all>
            [-l <language>]
            [-o <output_directory>]
            [-w]
            [-p <simulator_path>]
   ```

   Note: Each simulator uses certain environment variables which must be set before invoking compxlib. Consult your simulator documentation to ensure that the environment is properly set up to run your simulator.

The following is an example of a command for compiling Xilinx libraries for MTI_SE:

```
compxlib -s mti_se -f virtex -l verilog -o .
```

This command will compile all Verilog based libraries on ModelSim SE for the Virtex family in the current working directory. The compiled results will be saved in the following directories:

```
./unisim_ver
./XilinxCoreLib_v
./simprim_ver
```

Behavioral model libraries

Xilinx provides precompiled behavioral model libraries of the EDK IP for ModelSim 5.5b. The *vmap_edk_libs* program will install the libraries in a specified location. To install these libraries, you need to have the EDK and ModelSim previously set up. This installation applies for Windows and UNIX platforms.

Run the *vmap_edk_libs* program, specifying a target directory for the libraries. For example,

```
vmap_edk_libs $XILINX_EDK/edk_nd_libs
```

For a more recent version than ModelSim 5.5b, use the *-refresh* option,

```
vmap_edk_libs -refresh $XILINX_EDK/edk_nd_libs
```

The ModelSim no-debug libraries are now installed in the specified directory.

SmartModels

SmartModels are precompiled with the Xilinx 5.1i implementation tools, but they are not installed. This allows you to install the PPC405 and MGT SmartModels with additional SmartModels incorporated in the design. All SmartModels must be compiled into a common library for use by the simulator.

Windows

1. BEGIN SmartModel INSTALLATION
   a. To install the SmartModels, simply execute the *sl_admin.exe* program from the *$XILINX\smartmodel\nt\image\pcnt* directory.
2. SELECT SmartModels TO INSTALL :
   a. The *sl_admin* GUI and a pop-up "Set Library Directory" window will appear. Change the default directory from *image\pcnt* to *installed*, then click on *OK*. If the directory does not exist, the program will ask for permission to create it -- select *OK*.
   b. Next, click on the *Install* button on the left side of the *sl_admin* window. This will allow you choose the models to install.
   c. When the *Install From...* pop-up window appears, click on *Browse*, and select *$XILINX\smartmodel\nt\image* directory. Click on *OK* to select that directory.
   d. When the *Select Models to Install* window appears, select *Add All*, then click on *OK*.
   e. The *Choose Platform* window will then appear. *Wintel* should be selected for *Platforms*. For EDAV Packages, *Other* should be selected. Click on the *OK* button to install.
   f. From the *sl_admin* window, you will see *Loading: gt_swift*, and *Loading: ppc405_swift*. When the words *Install complete* appear, the installation has completed.

The SmartModels are now installed. You may exit the GUI using the *File -> Exit* menu, or you may use the GUI to perform other operations, including bringing up documentation and running checks on your newly installed library.

To properly utilize the newly compiled models, the LMC_HOME variable must be set to the image directory.

For example:

```
Set LMC_HOME=$Xilinx$\smartmodel\nt\installed
```

Solaris

1. BEGIN SmartModel INSTALLATION
   a. To install the SmartModels, simply run the *sl_admin.csh* program from the *$XILINX/smartmodel/sol/image* directory:

```
$ cd $XILINX/smartmodel/sol/image
$ sl_admin.csh
```

2. SELECT SmartModels TO INSTALL
   a. The *sl_admin* GUI and a pop-up window for *Set Library Directory* will appear. Change the default directory from *image/pcnt* to *installed*, then click on *OK*. If the directory does not exist, the program will ask for permission to create it -- select *OK*.
   b. The *sl_admin* GUI and a pop-up window for *Install From...* will appear. Select *Open* to use the default directory.
   c. Next, a *Select Models to Install* window will appear. Click on *Add All* to select all models, then select *Continue*.
   d. In the *Select Platforms for Installation* window, *Sun-4* should be selected for *Platforms*. For EDAV Packages, *Other* should be selected. Click on the *Install* button.
   e. When the words *Install complete* appear, and the status line (bottom line of the sl_admin GUI) reports *Ready*, the installation is complete.

The SmartModels are now installed. You may exit the GUI using the File ->Exit menu, or you may use the GUI to perform other operations, such as bringing up documentation and running checks on your newly installed library.

To properly utilize the newly compiled models, the LMC_HOME variable must be set to the image directory.

For example:

```
setenv LMC_HOME $XILINX/smartmodel/sol/installed
```

Third Party Tools

The Xilinx EDK requires some third party tools to be obtained and set up. This section provides some information on these tools.

WindRiver XE

The EDK is designed to work with the WindRiver XE toolset. These tools may be obtained from http://www.xilinx.com/xlnx/xil_entry2.jsp?sMode=login&group=windriver

IBM(R) CoreConnect(TM) Toolkit

The EDK is designed to integrate seamlessly with the IBM(R) CoreConnect(TM) Toolkit. This toolkit is not included with the EDK, but is required if bus functional simulation is desired.

The toolkit provides a number of features which enhance design productivity. To obtain the toolkit, a license for the IBM CoreConnect Bus Architecture must be obtained. Licensing CoreConnect provides access to a wealth of documentation, Bus Functional Models, Hardware IP and the toolkit.

Xilinx provides a Web-based licensing mechanism that allows you to obtain the CoreConnect from the Xilinx website. To license CoreConnect, use an internet browser to access: http://www.xilinx.com/ipcenter/processor_central/register_coreconnect.htm Once the request has been approved (Typically takes 24 hours), an email granting access to the protected web site will be sent out. The toolkit may be then be downloaded.

For further documentation on the CoreConnect Bus Architecture, refer to IBM's CoreConnect website: http://www.ibm.com/chips/products/coreconnect

The CoreConnect license may also be obtained directly from IBM.

There are some differences between the IBM CoreConnect and the Xilinx implementation of CoreConnect. These are described in the Embedded Processors IP HandBook. Refer to the *On-Chip Peripheral Bus V2.0 with OPB Arbiter* for differences in the OPB bus, the *Processor Local Bus (PLB) V3.4* for differences in the PLB bus, and *Device Control Register Bus (DCR) V2.9* for differences in the DCR bus.

ModelSim Setup for using SmartModels

The Xilinx Virtex-II Pro simulation flow uses Synopsys VMC models to simulate the IBM PowerPC microprocessor and Rocket I/O multi-gigabit transceiver. VMC models are simulator-independent models that are derived from the actual design and are therefore accurate evaluation models. To simulate these models, a simulator that supports the SWIFT interface must be used.

The SmartModels are included in the 5.1i Implementation Tools. You must install the SmartModels included in the 5.1i Implementation Tools in order to use them.

Although ModelSim SE supports the SWIFT interface, certain modifications must be made to the default ModelSim setup to enable this feature. The ModelSim installation directory contains an initialization file called *modelsim.ini*. In this initialization file, you may edit GUI and simulator settings so that they default to your preferences. You must edit parts of this modelsim.ini file in order for it to work properly with the Virtex-II Pro device simulation models.

MTI ModelSim SE - Windows

The following changes should be made to the modelsim.ini file located in the MODEL_TECH directory. (An alternative to making these edits is to change the MODELSIM environment variable setting in the MTI setup script so that it points to the modelsim.ini file located in the project design directory.)

1. After the lines

```
; Simulator resolution
; Set to fs, ps, ns, us, ms, or sec with optional prefix
; of 1, 10, or 100.
```

edit the statement that follows from "Resolution = ns" to "Resolution = ps".

2. After the lines

```
; Specify whether paths in simulator commands should be described
; in VHDL or Verilog format. For VHDL, PathSeparator = /
; for Verilog, PathSeparator = .
```

comment the following statement called "PathSeparator" = / by adding a ";" at the beginning of the line.

3. After the line

```
; List of dynamically loaded objects for Verilog PLI applications
```

add the following statement:

```
Veriuser = $MODEL_TECH/libswiftpli.dll
```

a. After the line

```
; Logic Modeling’s SmartModel SWIFT software (Windows NT)
```

add the following statements:

```
libsm = $MODEL_TECH/libsm.dll
libswift = $LMC_HOME/lib/pcnt.lib/libswift.dll
```

NOTE: It is important to make the changes in the order in which the commands appear in the modelsim.ini file. The simulation may not work if the order recommended above is not followed.

After you edit the modelsim.ini file, add the following environment variables:

```
set MODELSIM <path_to_modelsim.ini_script>\modelsim.ini
set XILINX <Xilinx path>
set LMC_HOME %XILINX%\smartmodel\nt\installed
set MODEL_TECH <MTI path>
set MODEL_TECH_LIBSM %MODEL_TECH%\win32\libsm.dll
set LM_LICENSE_FILE <license.dat>;%LM_LICENSE_FILE%
set path %LMC_HOME%\bin;%LMC_HOME%\lib\pcnt.lib;%MODEL_TECH%\bin;
        %XILINX%\bin\nt;%path%
```

You are responsible for changing the parameters included within <> to match the systems configuration.

If the MODELSIM environment variable is not set properly, MTI might not use this .ini file, and the simulator will not read the initialization settings required for simulation. Set up the MTI SE simulation environment by sourcing the MTI SE setup script from the terminal.

MTI ModelSim SE - Solaris

The following changes should be made to the modelsim.ini file located in the $MODEL_TECH directory. (An alternative to making these edits is to change the MODELSIM environment variable setting in the MTI setup script so that it points to the modelsim.ini file located in the project design directory.)

1. After the lines

```
; Simulator resolution
; Set to fs, ps, ns, us, ms, or sec with optional prefix
; of 1, 10, or 100.
```

edit the statement that follows from "Resolution = ns" to "Resolution = ps"

2. After the lines

```
; Specify whether paths in simulator commands should be described
; in VHDL or Verilog format. For VHDL, PathSeparator = /
; for Verilog, PathSeparator = .
```

comment the statement called PathSeparator = / by adding a ";" at the beginning of the line.

3. After the line

```
; List of dynamically loaded objects for Verilog PLI applications
```

add the following statement:

```
Veriuser = $MODEL_TECH/libswiftpli.sl
```

4. After the line

```
; Logic Modeling's SmartModel SWIFT software (Sun4 Solaris 2.x)
```

add the following statements:

```
libsm = $MODEL_TECH/libsm.sl
libswift = $LMC_HOME/lib/sun4Solaris.lib/libswift.so
```

NOTE: It is important to make the changes in the order in which the commands appear in the modelsim.ini. The simulation may not work if the order recommended above is not followed.

After you edit the modelsim.ini file, add the following environment variables to the MTI ModelSim SE setup script:

```
setenv MODELSIM <path_to_modelsim.ini_script>/modelsim.ini
setenv XILINX <Xilinx path>
setenv MODEL_TECH <MTI path>
setenv MODEL_TECH_LIBSM $MODEL_TECH/sunos5/libsm.sl
setenv LM_LICENSE_FILE <modelsim_license.dat>;$LM_LICENSE_FILE
setenv LMC_HOME ${XILINX}/smartmodel/sol/installed
setenv PATH ${LMC_HOME}/bin:${LMC_HOME}/lib/sun4Solaris.lib:
      ${MODEL_TECH}/bin:${XILINX}/bin/sol:{PATH}
```

You are responsible for changing the parameters included within <> to match the systems configuration.

If the MODELSIM environment variable is not set properly, MTI might not use this .ini file, and the simulator will not read the initialization settings required for simulation. Set up the MTI SE simulation environment by sourcing the MTI SE setup script from the terminal.

Directory Structure

The installed image of the Xilinx EDK is organized into the following directories. It is assumed that the root of the EDK image is located at <edk_install_dir>.

<edk_install_dir>/ (installed EDK root)

- **bin/** (EST application executables
- **doc/** (EST documentation)
- **data/** (Contains the default option files required by ISE tools)
- **gnu/** (EST GNU Tools)
- **hw/** (MicroBlaze processor and peripheral hardware components)
- **lib/** (libc and libm C function libraries; system function libraries)
- **xygwin/** (only on Windows platform)

The EDK installer has already set up your environment variables to include the executables of the EST tools that reside in the ***bin*** directory.

All EDK related documentation resides in the doc directory. Please see the "Documentation" section for an overview.

The ***hw*** directory contains the untailored sources of the hardware IP of the MicroBlaze processor and its peripheral components.

System program functions and other library functions reside in the ***lib*** directory.

xygwin is a portability layer that gets automatically installed on all Windows platforms.

Hardware IP Cores

The Xilinx EDK includes the following hardware IP cores:

- PLB Arbiter and Bus Structure
- OPB Arbiter and Bus Structure
- DCR Bus Structure
- OPB2PLB Bridge
- OPB2OPB Bridge
- System Reset Module
- PLB BRAM Controller
- PLB External Memory Controller (EMC) Includes support for: Flash, SRAM,ZBT, SystemACE
- OPB DDR SDRAM Controller
- OPB SDRAM Controller
- OPB BRAM Controller
- OPB External Memory Controller (EMC) Includes support for: Flash, SRAM, ZBT, SystemACE
- PLB IPIF
- OPB IPIF
- DCR Interrupt Controller (INTC)
- OPB SPI Master and Slave Bus Controller
- OPB Interrupt Controller (INTC)
- OPB UART - Lite
- OPB JTAG UART
- OPB GPIO Controller
- OPB Timer / Counter
- OPB TimeBase / WatchDog Timer

The following Ala-Carte cores are also included. These cores may be used freely in simulation, and for a limited time on chip. Licenses for these cores may be purchased from the Xilinx Web site.

- PLB 1Gb Ethernet Controller
- PLB UART-16550
- PLB UART-16450
- OPB <- PCI Full Bridge
- OPB 10/100M Ethernet Controller
- OPB 10/100M Ethernet Controller - Lite
- OPB ATM Utopia Level 2 Slave
- OPB ATM Utopia Level 2 Master
- OPB Single Channel HDLC Controller
- OPB IIC Master and Slave Bus Controller
- OPB 16550 UART Controller
- OPB 16450 UART Controller

Documentation

The Xilinx EDK documentation is organized into several individual components, all of which are accessible from the EDK documentation directory (***<edk_install_dir>/doc***):

- Release Notes Page (***readme.htm***)
- Product Home Page (***index.htm***)
- Documents Page (***documents.htm***)
- Support Page (***support.htm***)

Release Notes Page

Important information applicable only to one specific release of the EDK or amendments to the general EDK documentation are covered in the release notes.

Product Home Page

The installed product home page serves as a starting point from where the contents of the EDK can be explored.

Documents Page

The product user manuals are the most detailed documents included in the EDK. They include information about:

- *Getting Started with the Xilinx EDK*
- *Processor IP*
- *Embedded System Tools*
- *MicroBlaze Processor*
- *PowerPC Processor*

The Embedded Processors IP Guide covers all embedded processor bus interfaces, peripherals, and the creation of embedded processor systems.

The Embedded System Tools Guide covers the software development tools. It provides an overview of the EST design flow and describes the functionality and invocation options provided by every tool.

Support Page

Please refer to the information provided on the support page to learn about the technical product support available for EDK.