Getting Started with EDK

Summary

This document provides an introduction to the Xilinx 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 6.1i. 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 6.1i
  - The Xilinx FPGA design implementation tools ISE 6.1i 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 6.1i
  - Updates to ISE 6.1i including service packs are available at **http://support.xilinx.com/support/techsup/sw_updates**
- 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

Operating system specific requirements:

- Solaris
  - Bash shell

Supported Platforms

Operating Systems

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

- Windows 2000 (Service Pack 2)
- Windows XP
- Solaris 2.8 / 2.9

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 Spartan-IIE FPGAs
- Xilinx Spartan-III FPGAs
- 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)
- Digilent Spartan-IIE Board
- Memec Design Virtex-E Demo Board
- Memec Design Spartan-II Demo Board
- Memec Design Virtex-II MicroBlaze Board
- Memec Design Virtex-II Pro Board
- Xilinx ML300 Board
- Xilinx AFX Board

Please contact your local Avnet, Memec 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/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 installation of EDK 6.1 requires uninstallation of all previous versions of EDK. The installer will uninstall all the components of previous installations before proceeding with the installation of EDK 6.1. This is a requirement only on Windows.

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

Environment Variables

EDK installer creates/modifies the following variables on the users machine. These variables are added to the system settings.

- **XILINX_EDK**: Set the value of this variable to the installation direction.
- **PATH**: Prefixes the PATH environment variable with
  - *%XILINX_EDK%*\bin\sol

In addition to these variable, the registry is updated with the following entries:

- *My Computer\HKEY_CURRENT_USER\SOFTWARE\Xilinx\EDK61*
- *My Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Xilinx\EDK61*

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/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.

Environment Variables

EDK installer creates a ***setup.csh*** file in the installation directory. This file setups the environment for using the EDK tools.

- ♦ **$XILINX_EDK**: Set the value of this variable to the installation direction.
- ♦ **$PATH:** Prefixes the $PATH environment variable with
  - *$XILINX_EDK*/bin/sol
- ♦ **$LD_LIBRARY_PATH**: Prefixes the $LD_LIBRARY environment variable with
  - *$XILINX_EDK*/bin/sol

Note: While executing the script, the user has to make sure that the $XILINX_EDK/bin/sol appears before $XILINX/bin/sol in the $LD_LIBRARY_PATH variable.

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 *compxlib* to 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 (you need to have an installation of the Xilinx implementation tools):

1. Run *compxlib* with -help option to display a brief description for the available options.

```
compxlib -help
```

2. Run *compxlib* tool using the following syntax:

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

Note: Make sure you use the -p <simulator_path> option to point to the directory where the modelsim executable is, if it is not in your path.

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

```
compxlib -s mti_se -f virtex2p:u:s:c:m -l vhdl -w -o .
```

This command will compile unism, simprim and XilinxCoreLib vhdl based libraries on ModelSim SE only for the Virtex2 PRO family in the current working directory. It will also compile the vhdl smartmodels. The compiled results will be saved in the following directories:

```
./unisim
./simprim
./XilinxCoreLib
```

Note: If you need libraries for other family architectures, language or simulator, make sure you use the appripriate switches. By using "all" on each of these switches, compxlib will compile libraries for all the available options for that switch.

Behavioral model libraries

Xilinx provides a tool to compile behavioral model libraries of the EDK IP for ModelSim. The *compedklib* program will compile the libraries into a specified location. For encrypted IP, Xilinx provides precompiled libraries and will be placed by *compedklib* in the same location. To install these libraries, you need to have the EDK and ModelSim (version 5.6e or higher) already set up.

1. Run *compedklib* tool using the following syntax:

```
compedklib -X <compxlib_output_directory>
        [-lp <user_core_library]
        -o <compedklib_output_directory>
```

The following is an example of a command for compiling Xilinx libraries for MTI_SE (this command would be used in the same directory where you ran compxlib):

```
compedklib -X . -o edklib
```

This command will compile the EDK simulation libraries and the compiled results will be in a directory in the same location as Xilinx libraries:

```
./unisim
./simprim
./XilinxCoreLib
./edklib
```

Note: All Xilinx EDK IP is written in VHDL and behavioral model libraries are only available for ModelSim SE/PE using VHDL at this time.

Note: ModelSim will overwrite the modelsim.ini file when mapping the compiled libraries.

SmartModels

SmartModels represent integrated circuits and system buses as black boxes that accept input stimulus and respond with appropriate output behavior. Such behavioral models provide improved performance over gate-level models, while at the same time protect the proprietary designs created by semiconductor vendors. SmartModels connect to hardware simulators through the SWIFT interface, which is integrated with over 30 commercial simulators, including Synopsys VCS, Cadence Verilog-XL, and Model Technology ModelSim.

SmartModels are included in the Xilinx 6.1i release. The following steps outline how to set up the models.

Windows

- On Windows, open the "System Properties” dialog box by Selecting “System” in the Windows Control Panel. Select the “Advanced” tab and click on “Environment Variables”. The “Environment Variables” dialog box will appear.

  Set the variables to the following values (if not already set):

  ```
  LMC_HOME  %Xilinx%\smartmodel\nt\installed_nt
  PATH      %LMC_HOME%\bin;%LMC_HOME%\lib\pcnt.lib;%PATH%
  ```

  Note: %PATH% represents what your PATH variable had before doing the changes. Make sure you keep that.

Solaris

- Set the following variables (if not already set):

  ```
  setenv LMC_HOME $XILINX/smartmodel/sol/installed_sol
  setenv LD_LIBRARY_PATH $LMC_HOME/lib/sun4Solaris.lib:$LD_LIBRARY_PATH
  setenv PATH $LMC_HOME/bin:${PATH}
  ```

Linux

- Set the following variables (if not already set):

  ```
  setenv LMC_HOME $XILINX/smartmodel/lin/installed_lin
  setenv LD_LIBRARY_PATH $LMC_HOME/lib/x86_linux.lib:$LD_LIBRARY_PATH
  setenv PATH $LMC_HOME/bin:${PATH}
  ```

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 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 6.1i Implementation Tools.

Although ModelSim PE/SE support 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.

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. Edit the statement "Resolution = ns" and change it to "Resolution = ps".
2. Comment the following statement called "PathSeparator = /" by adding a ";" at the beginning of the line.
3. For Verilog designs enable smartmodels by searching for the variable "Veriuser" and change it to:
   - On Windows:
     ```
     Veriuser = $MODEL_TECH/libswiftpli.dll
     ```
   - On Solaris:
     ```
     Veriuser = $MODEL_TECH/libswiftpli.sl
     ```
   - On Linux:
     ```
     Veriuser = $MODEL_TECH/libswiftpli.sl
     ```
4. Search for the [lmc] section and uncomment the libsm and libswift definitions according to your operating system.

   For Example,
   - On Windows, uncomment these lines:
     ```
     libsm = $MODEL_TECH/libsm.dll
     libswift = $LMC_HOME/lib/pcnt.lib/libswift.dll
     ```
   - On Solaris, uncomment these lines
     ```
     libsm = $MODEL_TECH/libsm.sl
     libswift = $LMC_HOME/lib/sun4Solaris.lib/libswift.so
     ```
   - On Linux, uncomment these lines
     ```
     libsm = $MODEL_TECH/libsm.sl
     ```

```
    libswift = $LMC_HOME/lib/x86_linux.lib/libswift.so
```

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 some environment variables, if they are not already defined.

- On Windows, open the "System Properties” dialog box by Selecting “System” in the Windows Control Panel. Select the “Advanced” tab and click on “Environment Variables”. The “Environment Variables” dialog box will appear.

  Set the variables to the following values:

  ```
  MODELSIM  <path_to_modelsim.ini_script>\modelsim.ini
  PATH      <MTI_path>\win32;%PATH%
  ```

  Note: %PATH% represents what your PATH variable had before doing the changes. Make sure you keep that.

- On Solaris and Linux, add the following environment variables to the MTI ModelSim setup script:

  ```
  setenv MODELSIM <path_to_modelsim.ini_script>/modelsim.ini
  setenv PATH <MTI_path>/bin:${PATH}
  ```

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

Note: 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.

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)
`boards/`	(Consist of the board description files)
`doc/`	(EDK documentation)
`data/`	(Contains the default option files required by ISE tools)
`gnu/`	(EST GNU Tools)
`hw/`	(MicroBlaze processor and peripheral hardware components)
`sw/`	(Drivers, BSP’s and software services [Xilinx microkernel])
`systems/`	(Contains Reference systems [user templates in EDK 3.2])
`xygwin/`	(only on Windows platform. Emulation of unix shell.)

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.

The ***sw*** directory contains the drivers required by the IP installed in the ***hw*** directory, the board support packages. It also contains the source for various modules of the Xilinx microkernel, such as the Networking library and the file system. Software libraries and initialization files are also contained in the ***sw*** directory.

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

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`):

- Product Home Page (`index.htm`)
- What's New Page (`whatsnew.htm`)
- Installation Page (`installation.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.

Installation Page

Instructions to install, uninstall and perfom required EDK environment setup for the supported operating systems.

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.