Using the CORE Generator System

Introduction

This section on the Xilinx CORE Generator System™ and the Xilinx IP Core offerings is provided as an overview of products that relate to the facilitation of Virtex-II designs. For more detailed and complete information, consult the CORE Generator Guide, which can be accessed from the Xilinx online documentation in the Xilinx software installation, as well as on the Web under the heading of "Design Entry Tools" at: http://toolbox.xilinx.com/docsan/3_1i.

The CORE Generator System

The Xilinx CORE Generator System is the cataloging, customization, and delivery vehicle for IP cores targeted to Xilinx FPGAs. This tool is included with all Xilinx Foundation, Foundation ISE, and Alliance Series software packages. The CORE Generator provides centralized access to a catalog of ready-made IP functions ranging in complexity from simple arithmetic operators, such as adders, accumulators, and multipliers, to system-level building blocks, such as filters, transforms, and memories. Cores can be displayed alphabetically, by function, by vendor, or by type. Each core comes with its own data sheet, which documents the core's functionality in detail.

The CORE Generator User Interface has direct links to key Xilinx web support pages, such as the Xilinx IP Center and Xilinx Technical Support, making it very easy to access the latest Virtex-II IP releases and get helpful, up-to-date specifications and information on technical issues. See Figure 2-114. Links to partner IP providers are also built into the informational GUIs for the various partner-supplied AllianceCOREs described in the next section.

The use of CORE Generator IP cores in your Virtex-II designs enables you to slash your design time, and at the same time helps you to realize high levels of performance and area efficiency without any special knowledge about the Virtex-II architecture. The IP cores achieve these high levels of performance and logic density by using the Xilinx Smart-IP™ technology.

ug002_c2_068_110800

Figure 2-114: **Core Generator User Interface**

Smart-IP Technology

Smart-IP technology leverages Xilinx FPGA architectural features, such as look-up tables (LUTs), distributed RAM, segmented routing and floor planning information, as well as relative location constraints and expert logic mapping to optimize the performance of every core instance in a given Xilinx FPGA design. In the context of Virtex-II cores, Smart-IP Technology includes the use of the special high-performance Virtex-II architectural features, such as embedded 18x18 multipliers, block memory, shift register lookup tables (SRL16's), and special wide mux elements.

Smart-IP technology delivers:

- Physical layouts optimized for high performance
- Predictable high performance and efficient resource utilization
- Reduced power requirements through compact design and interconnect minimization
- Performance independent of device size
- Ability to use multiple cores without deterioration of performance
- Reduced compile time over competing architectures

CORE Generator Design Flow

A block diagram of the CORE Generator design flow is shown in Figure 2-115.

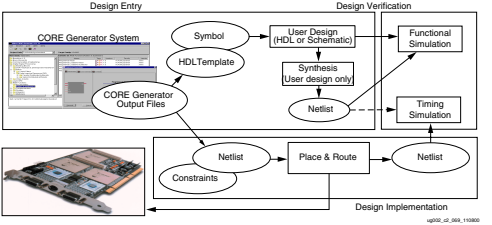

Figure 2-115: **CORE Generator Design Flow**

Note:

1. The outputs produced by the CORE Generator consist of an implementation Netlist, and optionally, a schematic Symbol and HDL template files.

Core Types

Parameterized Cores

The CORE Generator System supplies a wide assortment of parameterized IP cores that can be customized to meet specific Virtex-II design needs and size constraints. See Figure 2-116. For each parameterized core, the CORE Generator System supplies:

- A customized EDIF implementation netlist (EDN)

- A parameterized Verilog® or VHDL behavioral simulation model (.V, .VHD)
- Verilog or VHDL templates (.VEO, .VHO)
- A Foundation or Viewlogic® schematic symbol

The EDIF implementation netlist is used by the Xilinx tools to implement the core. The other design files generated depend on the Design Entry settings you specify (target CAE vendor, and design flow type—schematic or HDL).

The parameterized HDL simulation models are provided in two separate HDL simulation libraries called XilinxCoreLib, which are included as part of the Xilinx installation, one for Verilog functional simulation support, and the other for VHDL functional simulation support:

$XILINX/verilog/src/XilinxCoreLib

$XILINX/vhdl/src/XilinxCoreLib

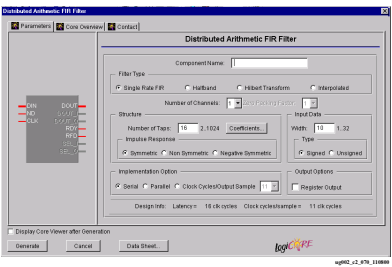

Figure 2-116: **Core Customization Window for a Parameterized Core**

If using a compiled simulator, these libraries must be precompiled before performing a functional simulation of the cores. An analyze_order file describing the required compile order of these models is included with each XilinxCoreLib library, one for Verilog (verilog_analyze_order) and one for VHDL (vhdl_analyze_order).

For an HDL design flow, Verilog and VHDL templates (.VEO and .VHO files) are also provided to facilitate the integration of the core into the design for the purposes of functional simulation, synthesis, and implementation. The HDL template file for a particular core conveys custom parameter values to the corresponding generic, parameterized behavioral model for that core in the XilinxCoreLib library. The custom parameter values are used to tailor the behavior of the customized core.

A sample VHO template is as follows;

```
-- User: Make sure that these statements appear
-- above the top-level entity declaration in your VHDL
-- design...
--LIB_TAG
Library xilinxcorelib;
```

```
Use xilinxcorelib.null_comp.all;
-- LIB_TAG_END
-- User: Make sure that this statement appears
-- in the architecture header in your VHDL design...
-- COMP_TAG
component adder8
port (
a: IN std_logic_VECTOR(7 downto 0);
b: IN std_logic_VECTOR(7 downto 0);
c: IN std_logic;
ce: IN std_logic;
ci: IN std_logic;
clr: IN std_logic;
s: OUT std_logic_VECTOR(8 downto 0));
end component;
-- COMP_TAG_END
-- User: Make sure that this statement appears
-- in the architecture body in your VHDL design,
-- substituting your own instance name where shown.
-- Do not forget to change the net names in the port map
-- to your own design's net names.-- INST_TAG

your_instance_name : adder8 port map (
a => a,
b => b,
c => c,
ce => ce,
ci => ci,
clr => clr,
s => s );
-- INST_TAG_END
-- User: Make sure that this text appears
-- within the top-level configuration body in your VHDL
-- design, for example:
--
-- configuration cfg_top of top_level is
-- for arch_name
-- Insert text here
-- end for;
-- end cfg_top;
--
-- CONF_TAG
for all : adder8 use entity
XilinxCoreLib.null(behavioral)
generic map(
signed => true,
input_width => 8);
end for;
-- CONF_TAG_END
```

Note:

1. Note that the customizing values for the simulation model of the multiplier ("signed = true" and "input_width=8") are passed as generics in the configuration declaration. For more details on the contents of the VEO and VHO template files, refer to the online CORE Generator Guide at http://toolbox.xilinx.com/docsan/3_1i/.

Schematic symbol files are generated when a schematic design flow is specified for the project.

Fixed Netlist Cores

The other type of Virtex-II core provided by the CORE Generator is the fixed netlist core. These are preset, non-parameterized designs that are shipped with the following:

- A fixed EDIF implementation netlist (as opposed to one that is customized on the fly)
- .VEO and .VHO templates
- Non-parameterized .V and .VHD behavioral simulation models
- Schematic symbol support

Examples include the v2.0 Xilinx FFTs and most AllianceCOREs.

Since the HDL behavioral models for fixed netlist cores is not parameterized, the corresponding .VEO and .VHO template files are correspondingly simpler, since they do not need to pass customizing parameter values to a library behavioral model.

Xilinx IP Solutions and the IP Center

The CORE Generator works in conjunction with the Xilinx IP Center on the world wide web to provide the latest IP and software upgrades. To make the most of this resource, Xilinx highly recommends that whenever starting a design, first do a quick search of the Xilinx IP Center (www.xilinx.com/ipcenter) to see whether there already is a ready-made core solution available.

The complete Xilinx cores library resides on the IP Center and consists of the following:

- LogiCOREs
- AllianceCOREs
- Reference Designs

When installing the CORE Generator software, the designer gains immediate access to dozens of cores supplied by the LogiCORE Program. In addition, data sheets for all AllianceCORE products and for additional, separately licensed, advanced function LogiCORE products are also available. As new and updated Virtex-II IP are released for the CORE Generator, they can be downloaded from the IP Center and added to the CORE Generator catalog.

LogiCORE Program

LogiCORE products are designed, sold, licensed, and supported by Xilinx. LogiCORE products include a wide selection of generic, parameterized functions, such as muxes, adders, multipliers, and memory cores which are bundled with the Xilinx CORE Generator software at no additional cost to licensed software customers. System-level cores, such as PCI, Reed-Solomon, ADPCM, HDLC, POS-PHY, and Color Space Converters are also available as optional, separately licensed products. Probably, the most common application of the CORE Generator is to use it to quickly generate Virtex-II block and distributed memories. A more detailed listing of available Virtex-II LogiCOREs is available in Table 2-59.

Types of IP currently offered by the Xilinx LogiCORE program include:

- Basic Elements: logic gates, registers, multiplexers, adders, multipliers.
- Communications and Networking: ADPCM modules, HDLC controllers, ATM building blocks.
- DSP and Video Image Processing: cores ranging from small building blocks (e.g., Time Skew Buffers) to larger system-level functions (e.g., FIR Filters and FFTs).
- System Logic: accumulators, adders, subtracters, complementers, multipliers, integrators, and square root generator functions; pipelined delay elements, single and dual port distributed, and block RAM, ROM, and synchronous and asynchronous FIFOs.
- Standard Bus Interfaces: PCI 64/66 (64-bit, 66 MHz), 64/33 (64-bit, 33 MHz), and 32/33 (32-bit, 33MHz) Interfaces.

AllianceCORE Program

The AllianceCORE program is a cooperative effort between Xilinx and third-party IP developers to provide additional system-level IP cores optimized for Xilinx FPGAs. To ensure a high level of quality, AllianceCORE products are implemented and verified in a Xilinx device as part of the certification process.

Xilinx develops relationships with AllianceCORE partners who can complement the Xilinx LogiCORE product offering. Where Xilinx does not offer a LogiCORE for a particular function, Xilinx partners with an AllianceCORE partner to offer that function. A large percentage of Xilinx AllianceCORE partners focus on data and telecommunication applications, as well as processor and processor peripherals designs.

Together, Xilinx and its AllianceCORE partners are able to provide an extensive library of cores to accelerate the design process. AllianceCOREs include customizable cores which can be configured to exact needs, as well as fixed netlist cores targeted toward specific applications. In many cases, partners can provide cores customized to meet the specific design needs if the primary offerings do not fit the requirements. Additionally, source code versions of the cores are often available from the partners at additional cost for those who need maximum flexibility.

The library of Xilinx and AllianceCORE IP cores allows the user to leverage the expertise of experienced designers who are well-versed in optimizing designs for Virtex-II and other Xilinx architectures. This enables the designer to obtain the highest performance and density in the target Virtex-II device with faster time to market.

Reference Designs

Xilinx also offers an assortment of Reference Designs on its corporate web pages. These designs are supplied as application notes, usually with supporting design files. They are extremely valuable to customers who are looking for guidance in designing their systems and can often be used as starting points for implementing a broad spectrum of simple-to-complex functions in Xilinx programmable logic.

Reference designs are supplied free of charge by Xilinx without technical support or warranty. For the latest list of reference design offerings for Virtex-II and other Xilinx FPGAs, refer to: http://www.xilinx.com/products/logicore/refdes.htm.

Design Reuse Tools

To facilitate the archiving and sharing of IP created by different individuals and workgroups within a company, Xilinx offers the ***IP Capture Tool***. The IP Capture Tool helps to package design modules created by individual engineers in a standardized format so that they can be cataloged and distributed using the Xilinx CORE Generator. A core can take the form of synthesizable VHDL or Verilog code, or a fixed function netlist. Once it is packaged by the IP Capture Tool and installed into the CORE Generator, the *"captured"* core can be shared with other designers within a company through an internal network. The IP Capture Tool is supplied as a separate utility through the Xilinx IP Center.

CORE Generator Summary

The CORE Generator delivers a complete catalog of IP including behavioral models, synthesis templates, and netlists with performance guaranteed by Xilinx Smart-IP technology. It is a repository for LogiCORE products from Xilinx, AllianceCORE products from Xilinx partners, and it supports Design Reuse for internally developed IP. In addition, LogiCORE products are continuously updated to add support for new Xilinx architectures, such as Virtex-II. The most current IP updates are available from the Xilinx IP Center.

Utilizing the CORE Generator library of parameterizable cores, designed by Xilinx for Xilinx FPGAs the designer can enjoy the advantages of design reuse, including faster time to market and lower cost solutions.

For more information, visit the Xilinx IP Center at: **www.xilinx.com/ipcenter.**

Virtex-II IP Cores Support

A partial listing of cores available for Virtex-II designs is listed below. For a complete catalog of Virtex-II IP, refer to the Xilinx IP Center web pages at: www.xilinx.com/ipcenter.

Table 2-59: **Virtex-II IP Cores Support**

Core Type (P=Parameterizable)	Features	Input Data Width	Output Data Width	Word Depth	Smart-IP
Basic Elements					
Comparator v2.0 or later (P)	Performs =, <>, <=, <, >=, and > comparisons; Supports unsigned or 2's complement signed inputs	1 - 64 bits	1	N/A	Y
Binary Counter v2.0 or later (P)	Up, Down, or Up/Down Counters; Count limit	Load value: 2 - 64 bits	2 – 64 bits	N/A	Y
Binary Decoder v2.0 or later (P)	Generates One-Hot output from binary coded input; 2-64 outputs	1 - 6 bits	2 – 64 outputs	N/A	Y
Two's Complementer v2.0 or later (P)		2 - 256 bits	3 - 257 bits	N/A	Y
Bit Gate v2.0 or later (P)	AND, NAND, OR, NOR, XOR, or XNOR; Input inversion mask	2 - 64 bits	1	N/A	Y
Bus Gate v2.0 or later (P)	2 - 4 input buses; AND, NAND, OR, NOR, XOR, XNOR, Buffer or Inverter	1 - 64 bits	1 – 64 bits	N/A	Y
Bit Bus Gate v2.0 or later (P)	Generates AND, NAND, OR, NOR, XOR, and XNOR gates where a bus is gated with a single control bit; Input inversion mask	2 - 64 bits	2 – 64 bits	N/A	Y
BUFE-based Multiplexer Slice v2.0 or later (P)	Use to create wide muxes	1 - 64 bits	1 – 64 bits	N/A	N
BUFT-based Multiplexer Slice v2.0 or later (P)	Use to create wide muxes	1 - 64 bits	1 – 64 bits	N/A	N
Bit Multiplexer v2.0 or later (P)	N : 1 multiplexer	2 - 256 bit-wide inputs, 1 – 8 bit Select inputs	1	N/A	Y
Bus Multiplexer v2.0 or later (P)	2 - 32 input buses	1 - 256 bits	1- 256 bits	N/A	Y
Math Functions					
Accumulator v2.0 or later (P)	Add, Subtract and Add/Subtract accumulators; Support for feedback scaling; Support for adding or subtracting a constant value	1 - 256 bits	1 - 258 bits	N/A	Y

Table 2-59: **Virtex-II IP Cores Support *(Continued)***

Core Type (P=Parameterizable)	Features	Input Data Width	Output Data Width	Word Depth	Smart-IP
Adder/Subtracter v2.0 or later (P)	Add, Subtract and Add/Subtract functions; Add or Subtract a constant value	A: 1 - 256 bits B: 1 - 256 bits	1 - 258 bits	N/A	Y
Comparator	See entry for this function under Basic Elements	-	-	-	-
Complementer	See entry under Basic Elements	-	-	-	-
Divider v2.0 or later (P)	Pipelined - new output every 1, 2, 4, or 8 cycles; Can be used to implement 1/X operation fractional or integer remainder	Dividend: 1 - 32 bits Divisor: 3 - 32 bits	Remainder: 3 - 32 bits	N/A	N
Multiplier v2.0 or later (P)	constant coefficient, reloadable constant coefficient, parallel or sequential multipliers; Specify 18x18 multiplier option to utilize Virtex-II	1 – 64 bits	1 – 128 bits	N/A	Y
Multiply Accumulator	See entry for this core under "Digital Signal Processing"	-	-	-	-
Memories and Storage Elements					
Dual Port Block Memory v2.0 or later (P)	Read and Write, Read Only (ROM), and Write Only configurations; "Read Before Write", "Read After Write", or "No Read on Write" Write Modes	1 - 256 bits	1 – 256 bits	2 - 1M words	N
Single Port Block Memory v2.0 or later (P)	Read and Write, or Read Only (ROM) configurations; "Read Before Write", "Read After Write", or "No Read on Write" Write Modes	1 - 256 bits	1 – 256 bits	2 - 1M words	N
Distributed Memory v2.0 or later (P)	SelectRAM+ based Single or Dual Port RAM, ROM, or SRL16 based RAM	1 - 256 bits	1 – 256 bits	16 - 256 words	Y
Registers, Shifters, and Pipelining Modules					
FD-based Parallel Register v2.0 or later (P)		1 - 256 bits	1 - 256 bits	N/A	Y
FD-based Shift Register v2.0 or later (P)		2 - 64 bits	2 - 64 bits	N/A	Y
LD-based Parallel Latch v2.0 or later (P)		1 - 256 bits	1 - 256 bits	N/A	Y

Table 2-59: **Virtex-II IP Cores Support *(Continued)***

Core Type (P=Parameterizable)	Features	Input Data Width	Output Data Width	Word Depth	Smart-IP
RAM-based Shift Register v2.0 or later (P)	SRL16-based shift registers	1 - 256 bits	1 – 256 bits	1 - 1024 words	Y
Communications and Networking					
8b/10b Encoder v1.0 or later (P)	supports encoding of an 8-bit byte (256 unique data words) and an additional 12 special (or "K") characters into a 10-bit symbol. Fully synchronous operation and tracking of "running disparity"	8 bits	10 bits	N/A	Y
8b/10b Decoder v2.0 or later (P)	decodes 10-bit symbols into 8-bit bytes and an accompanying "K" bit. Decoding of 268 unique transmitted characters is supported; Fully synchronous operation and tracking of "running disparity"	10 bits	8 bits	N/A	Y
Digital Signal Processing					
Distributed Arithmetic FIR Filter v2.0 or later (P)	2 - 1024 taps, 1 - 8 channels, 1 - 32 bit coefficients. Includes: - single rate, - half band, - Hilbert transform, - interpolated, - polyphase decimating, - polyphase interpolating, - half band decimating, and - half band interpolating filters	1 - 32 bits	Full precision	N/A	N
Complex FFTs v2.0 or later	1024-pt, 256-pt, 64-pt, and 16-pt forward and inverse transforms; 2's complement complex data	16-bit real and 16-bit imaginary components	16-bit real and 16-bitimaginary components	N/A	Y
Multiply Accumulator (MAC) v2.0 or later (P)	Variable, Constant Coefficient, or Dynamic Constant Coefficient; combinatorial or pipelined	A: 1 - 32 bits (unsigned), 2 - 32 bits (signed) B: 1 - 32 bits (unsigned), 2 - 32 bits (signed)	1 – 65 bits	N/A	Y

Table 2-59: **Virtex-II IP Cores Support *(Continued)***

Core Type (P=Parameterizable)	Features	Input Data Width	Output Data Width	Word Depth	Smart-IP
FIFOs					
Asynchronous FIFO v2.0 or later (P)	Block or Distributed memory implementation; Independent clock domains	1 - 256 bits		15 -65535 words	Y
Synchronous FIFO v1.0 (P)	Distributed memory implementation	1 - 256 bits		16 - 256 words	Always RPM'd

Also, LogiCOREs in the following categories are in development:

- *Digital Signal Processing* cores such as Sine-Cosine Lookup Tables and DDS;
- *Communications and Networking* cores, such as: 8b/10b Encoder and Decoder, Reed Solomon Encoder and Decoder, ADPCM and HDLC LogiCOREs;
- *Standard Bus Interface* cores, such as PCI 64/66, PCI 64/33, and PCI 32/33 LogiCOREs;
- *Video, Audio, and Image Processing* cores, such as JPEG Encoder/Decoder and RGB2YcrCb/YcrCb2RGB Color Converter LogiCOREs.