Chapter 1
Xilinx Unified Libraries

This chapter describes the Unified Libraries and the applicable device architectures for each library. It also briefly discusses the contents of the other chapters, the general naming conventions, and performance issues.

This chapter consists of the following major sections.

Overview

Xilinx maintains software libraries with thousands of functional design elements (primitives and macros) for different device architectures. New functional elements are assembled with each release of development system software. The catalog of design elements is known as the "Unified Libraries." Elements in these libraries are common to all Xilinx device architectures. This "unified" approach means that you can use your circuit design created with "unified" library elements across all current Xilinx device architectures that recognize the element you are using.

Elements that exist in multiple architectures look and function the same, but their implementations might differ to make them more efficient for a particular architecture. A separate library still exists for each architecture (or architectural group) and common symbols are duplicated in each one, which is necessary for simulation (especially board level) where timing depends on a particular architecture.

If you have active designs that were created with former Xilinx library primitives or macros, you may need to change references to the design elements that you were using to reflect the Unified Libraries' elements.

The *Libraries Guide* describes the primitive and macro logic elements available in the Unified Libraries for XC3000A, XC3000L, XC3100A, XC3100L, XC4000E, XC4000L, XC4000EX, XC4000XL, XC4000XV, XC400XLA, XC5200, XC9500, XC9500XL, Spartan, SpartanXL, and Virtex architectures. Common logic functions can be implemented with these elements and more complex functions can be built by combining macros and primitives. Several hundred design elements (primitives and macros) are available across multiple device architectures, providing a common base for programmable logic designs.

This libraries guide provides a functional selection guide, describes the design elements, and addresses attributes, constraints, and carry logic.

Applicable Architectures

Design elements for the XC3000, XC4000E, XC4000X, XC5200, XC9000, Spartan, SpartanXL, and Virtex libraries are included in the Xilinx Unified Libraries. Each library supports specific device architectures. For detailed information on the architectural families referenced below and the devices in each, refer to the current *Programmable Logic Data Book*. (For Virtex device information, refer to the Xilinx web site, http://www.xilinx.com.)

XC3000 Library

Information appearing under the title of XC3000 pertains to the XC3000A, XC3100A, XC3000L, and XC3100L families. The XC3000L and XC3100L are identical in architecture and features to the XC3000A and XC3100A, respectively, but operate at a nominal supply voltage of 3.3 V.

XC4000E Library

Information appearing under the title XC4000E pertains to the XC4000E and XC4000L families. The XC4000L is identical in architecture and features to the XC4000E but operates at a nominal supply voltage of 3.3 V.

XC4000X Library

Information appearing under the title XC4000X pertains to the XC4000EX, XC4000XL, XC4000XV, and XC4000XLA families. The XC4000XL is identical in architecture and features to the XC4000EX but operates at a nominal supply voltage of 3.3 V. The XC4000XV has identical library symbols to the XC4000EX and XC4000XL but operates at a nominal supply voltage of 2.5 V and includes additional features.

XC4000 References

Wherever *XC4000* is mentioned, the information applies to all architectures supported by the XC4000E and XC4000X libraries.

XC5200 Library

The information appearing under the title XC5200 pertains to the XC5200 family.

XC9000 Library

The information appearing under the title XC9000 pertains to the XC9500 and XC9500XL CPLD families.

Spartan Library

The information appearing under the title Spartan pertains to the Spartan family XCS* devices.

SpartanXL Library

The information appearing under the title SpartanXL pertains to the SpartanXL family XCS*XL devices.

Spartans and Spartan Series References

Wherever *Spartans* and *Spartan series* is mentioned, the information applies to all architectures supported by the Spartan and SpartanXL libraries.

Virtex Library

The information appearing under the title Virtex pertains to the Virtex family XCV* devices.

Selection Guide

The **"Selection Guide" chapter** briefly describes, then tabularly lists the logic elements that are explained in detail in the "Design Elements" sections. The tables included in this section are organized into functional categories. They list the available elements in each category along with a brief description of each element and an applicability table identifying which libraries (XC3000, XC4000E, XC4000X, XC5200, XC9000, Spartan, SpartanXL, Virtex) contain the element.

Design Elements

Design elements are organized in alphanumeric order, with all numeric suffixes in ascending order. For example, FDR precedes FDRS, and ADD4 precedes ADD8, which precedes ADD16.

The following information is provided for each library element.

- Graphic symbol
- Applicability table (with primitive versus macro identification)
- Functional description
- Truth table (when applicable)
- Topology (when applicable)
- Schematic for macros

Schematic Examples

Schematics are included for each library if the implementation differs.

Design elements with bused or multiple I/O pins (2-, 4-, 8-, 16-bit versions) typically include just one schematic—generally the 8-bit version. When only one schematic is included, implementation of the smaller and larger elements differs only in the number of sections. In cases where an 8-bit version is very large, an appropriate smaller element serves as the schematic example.

Naming Conventions

Examples of the general naming conventions for the unified library elements are shown in the following figures.

Figure 1-1Naming Conventions

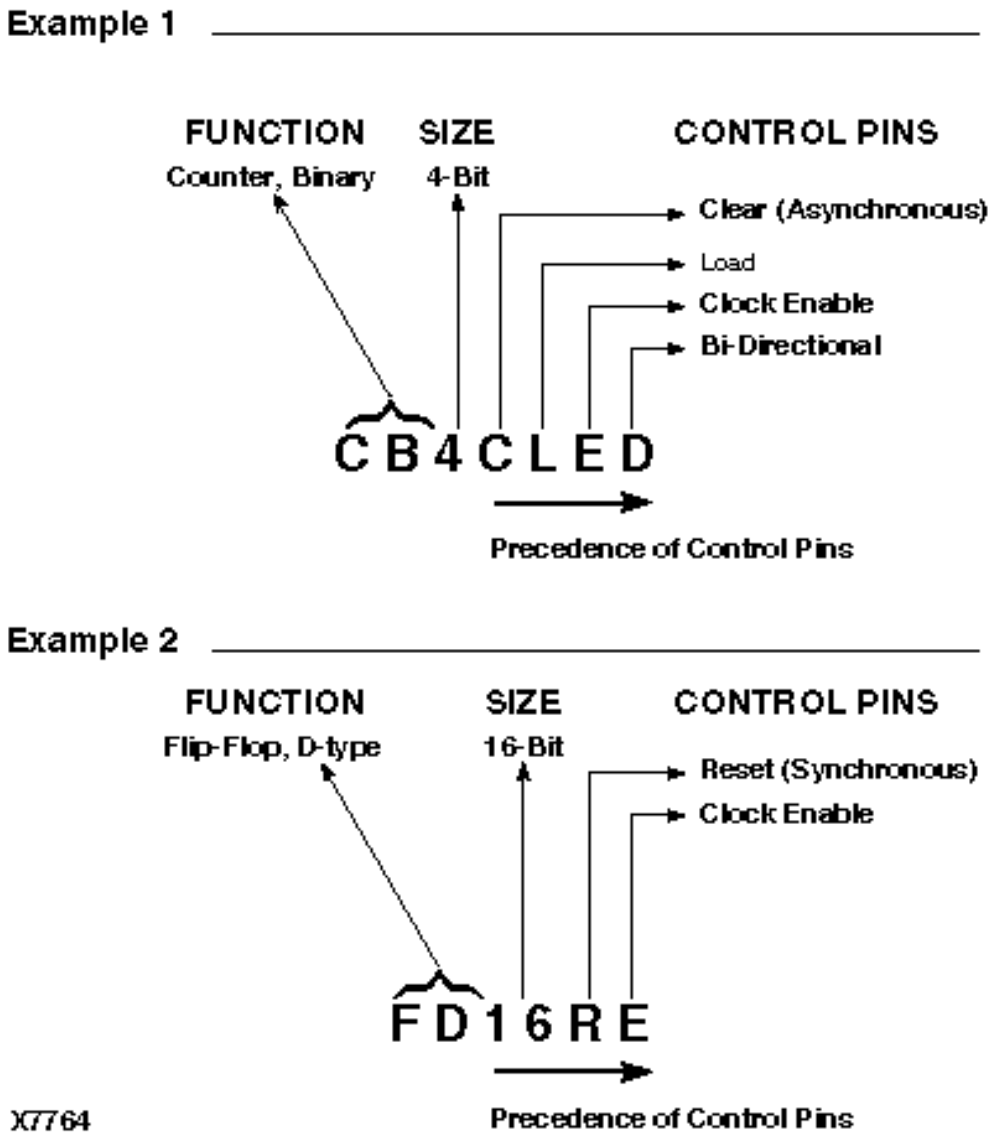
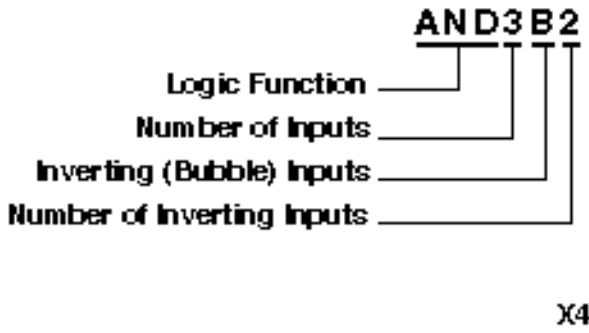


Figure 1-2Combinatorial Naming Conventions



Refer to the **"Selection Guide" chapter** for examples of functional component naming conventions.

Attributes, Constraints, and Carry Logic

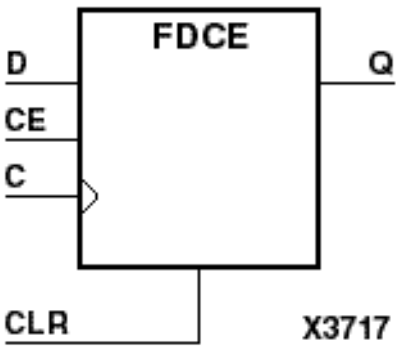
Attributes are instructions placed on symbols or nets in a schematic to indicate their placement, implementation, naming, directionality, and so forth. Constraints are a type of attribute used only to limit where an element should be placed. The **"Attributes, Constraints, and Carry Logic" chapter** provides information on all attributes and constraints.

Flip-Flop, Counter, and Register Performance

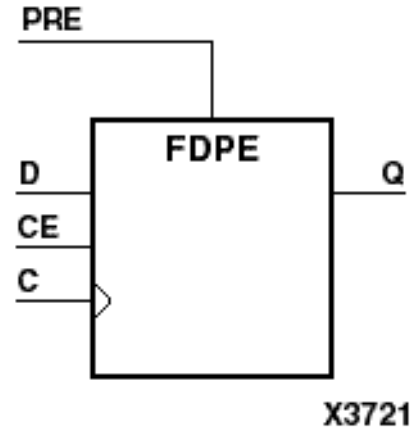
All counter, register, and storage functions are derived from the flip-flops (and latches in XC4000X and SpartanXL) available in the Configurable Logic Blocks (CLBs).

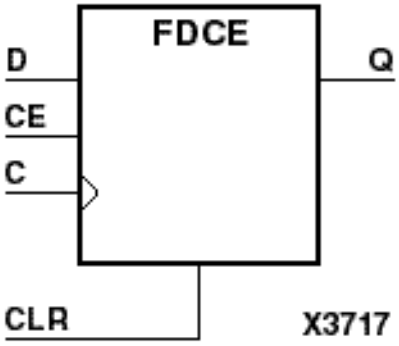
The D flip-flop is the basic building block for all architectures. Differences occur from the availability of asynchronous Clear (CLR) and Preset (PRE) inputs, and the source of the synchronous control signals, such as, Clock Enable (CE), Clock (C), Load enable (L), synchronous Reset (R), and synchronous Set (S). The basic flip-flop configuration for each architecture follows.

The XC3000 and XC5200 have a direct-connect Clock Enable input and a Clear input.

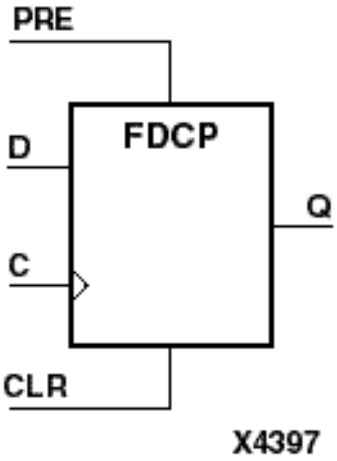


The XC4000, XC9500XL, and Spartans have a direct-connect Clock Enable input and a choice of either the Clear or the Preset inputs, but not both.

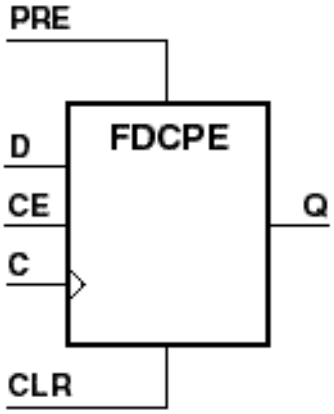


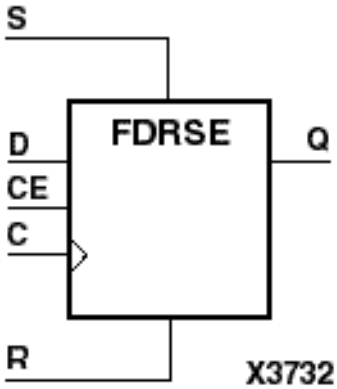


The basic XC9000 flip-flops have both Clear and Preset inputs.



Virtex has two basic flip-flop types. One has both Clear and Preset inputs and one has both asynchronous and synchronous control functions.





The asynchronous and synchronous control functions, when used, have a priority that is consistent across all devices and architectures. These inputs can be either active-High or active-Low as defined by the macro. The priority, from highest to lowest is as follows.

- Asynchronous Clear (CLR)
- Asynchronous Preset (PRE)
- Synchronous Set (S)
- Synchronous Reset (R)
- Clock Enable (CE)

Note: The asynchronous CLR and PRE inputs, by definition, have priority over all the synchronous control and clock inputs.

For FPGA families, the Clock Enable (CE) function is implemented using two different methods in the Xilinx Unified Libraries; both are shown in the **"Clock Enable Implementation Methods" figure**.

- In method 1, CE is implemented by connecting the CE pin of the macro directly to the dedicated Enable Clock (EC) pin of the internal Configurable Logic Block (CLB) flip-flop. This allows one CE per CLB. CE takes precedence over the L, S, and R inputs. All flip-flops with asynchronous clear or preset use this method.

- In method 2, CE is implemented using function generator logic. This allows two CEs per CLB. CE has the same priority as the L, S, and R inputs. All flip-flops with synchronous set or reset use this method.

The method used in a particular macro is indicated by the inclusion of asynchronous clear, asynchronous preset, synchronous set, or synchronous reset in the macro's description.

Figure 1-3Clock Enable Implementation Methods