XILINX®

ONLINE

LIBRARIES
GUIDE

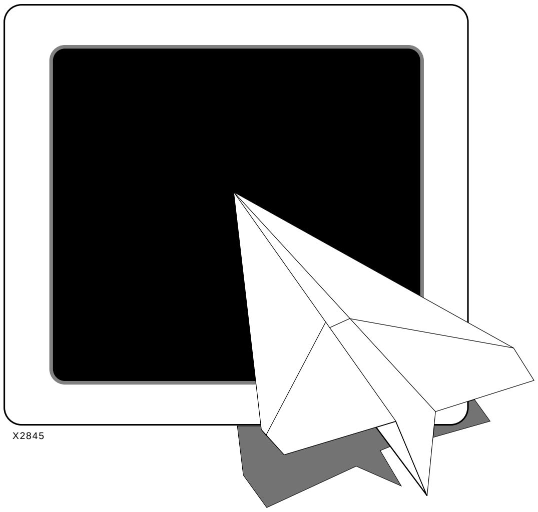
TABLE OF CONTENTS

INDEX

GO TO OTHER BOOKS

XACT™ STEP

0401410



Libraries Guide

Xilinx does not assume any liability arising out of the application or use of any product described herein; nor does it convey any license under its patents, copyrights, or maskwork rights or any rights of others. Xilinx, Inc. reserves the right to make changes, at any time, in order to improve reliability, function or design and to supply the best product possible. Xilinx, Inc. cannot assume responsibility for the use of any circuitry described herein other than circuitry entirely embodied in its products. Xilinx products are protected under at least the following U.S. patent: 5,224,056. Xilinx, Inc. does not represent that Xilinx products are free from patent infringement or from any other third-party right. Xilinx assumes no obligation to correct any errors contained herein or to advise any user of this text of any correction if such be made. Xilinx will not be liable for the accuracy or correctness of any engineering or software or assistance provided to a user.

Xilinx products are not intended for use in life support appliances, devices, or systems. Use of a Xilinx product in such applications without the written consent of the appropriate Xilinx officer is prohibited.

Preface

About This Manual

This manual describes Xilinx's XC7000 and XC9000 Libraries.

Before using this manual, you should be familiar with the operations that are common to all Xilinx's software tools: how to bring up the system, select a tool for use, specify operations, and manage design data.

Manual Contents

This libraries guide covers the following topics.

- Chapter 1, **XC7000 and XC9000 Libraries**, discusses the contents of the other chapters, general naming conventions, and performance issues.
- Chapter 2, S**election Guide**, describes then lists design elements by function that are described in detail in the "Design Elements" chapter.
- Chapter 3, **Design Elements**, provides a graphic symbol, functional description, primitive versus macro table, truth table (when applicable), topology (when applicable), and schematics for macros of the design elements.

Conventions

The following conventions are used in this manual’s syntactical statements:

`Courier font regular`
System messages or program files appear in regular Courier font.

`Courier font bold`
Literal commands that you must enter in syntax statements are in bold Courier font.

italic font
Variables that you replace in syntax statements are in italic font.

`[ ]`
Square brackets denote optional items or parameters. However, in bus specifications, such as bus [7:0], they are required.

`{ }`
Braces enclose a list of items from which you must choose one or more.

.
.
.
A vertical ellipsis indicates material that has been omitted.

. . .
A horizontal ellipsis indicates that the preceding can be repeated one or more times.

|
A vertical bar separates items in a list of choices.

↵
This symbol denotes a carriage return.

Contents

Chapter 1 Xilinx XC7000 and XC9000 Libraries

Chapter 2 Selection Guide

Chapter 3 Design Elements

Chapter 1

Xilinx XC7000 and XC9000 Libraries

Xilinx maintains software libraries with hundreds of functional design elements (primitives and macros) for different device architectures. This release contains libraries design elements for XC7000 and XC9000 CPLD architectures. Refer to the Xilinx Libraries Guide for descriptions of library elements for other Xilinx device families.

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 and common symbols are duplicated in each one, which is necessary for simulation (especially board level) where timing depends on a particular architecture.

Note: OrCAD symbols differ in appearance. They do not support busing; each input and output pin appears on the symbol. Inputs and outputs only appear on the left and right sides of symbols, respectively (none appear on the top or bottom).

Overview

This guide describes the primitive and macro logic elements available in the XC7000 and XC9000 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.

This book is organized into three parts.

- Xilinx XC7000 and XC9000 Libraries Overview
- Selection guide
- Design elements

Xilinx XC7000 and XC9000 Libraries Overview

This chapter describes the XC7000 and XC9000 Libraries, briefly discusses the contents of the other chapters, the general naming conventions, and performance issues.

Selection Guide

The “Selection Guide” briefly describes, then tabularly lists the macro logic elements that are described in detail in the “Design Elements” chapter. The tables included in this section are organized into functional categories specifying all the available elements from the XC7000 and XC9000 families.

Design Elements

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

The following information is provided for each library element.

- Graphic symbol
- Functional description
- Primitive versus macro table
- Truth table (when applicable)
- Schematic for macros

Note: Schematics are included for each architecture if the implementation differs. Also, design elements with bused or multiple I/O pins typically include just one schematic — generally the 8-bit version. (In cases where no 8-bit version exists, an appropriate smaller or larger element serves as the schematic example.)

Naming Conventions

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

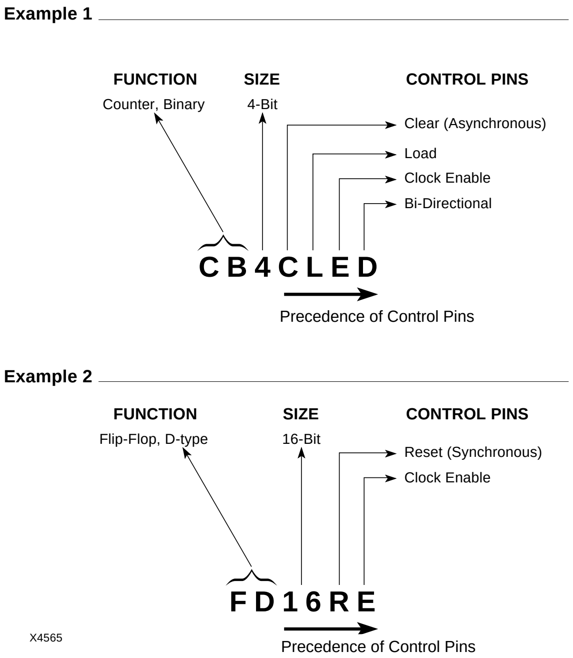


Figure 1-1 Naming Conventions

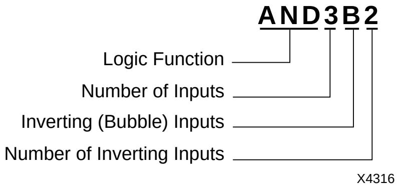


Figure 1-2 Combinatorial Naming Conventions

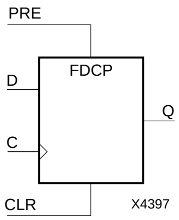
Refer to the Selection Guide for examples of functional component naming conventions.

Flip-Flop, Counter, and Register Performance

All counter, register, and storage functions are derived from the flip-flops available in the PLD architecture.

The D flip-flop is the basic storage element for all four 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 XC7000 and XC9000 flip-flops have both Clear and Preset inputs.



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)
- Load Enable (L)
- Shift Left/Right (LEFT)
- Clock Enable (CE)

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

Chapter 2

Selection Guide

The Selection Guide briefly describes, then tabularly lists the macro logic elements that are described in detail in the “Design Elements” chapter. The tables included in this section are organized into functional categories specifying all the available macros from the XC7000 and XC9000 families. The tables categorize the elements into subcategories based on similar functions. The sequence of each sub-category is based on an ascending order of complexity. The categories are as follows.

- Arithmetic functions
- Buffers
- Comparators
- Counters
- Decoders
- Encoders
- Flip-Flops
- General
- Input/output flip-flops and latches
- Input/output functions
- Latches
- Logic primitives
- Multiplexers
- Shifters

The elements from each architecture that provide the same function are listed adjacent to each other in the table, even though they might not have the same name. For particular elements, use the name specified for the architecture of interest.

There are a number of standard TTL 7400-type functions in the XC7000 and XC9000 architectures. All 7400-type functions are in alphanumeric order starting with “X,” and the numeric sequence uses ascending numbers following the “74” prefix. For example, X74_42 precedes X74_138.

Functional Categories

The following sections briefly describe, then tabularly list the Unified Libraries design element functions by category. Elements are listed in alphanumeric order according to architecture in each applicable architecture column. N/A means the element does not exist in that particular architecture.

Arithmetic Functions

There are three types of arithmetic functions: accumulators (ACC), adders (ADD), and adder / subtracters (ADSU). With an ADSU, either unsigned binary or twos-complement operations cause an overflow. If the result crosses the overflow boundary, an overflow is generated. Similarly, when the result crosses the carry-out boundary, a carry-out is generated. The following figure shows the ADSU carry-out and overflow boundaries.

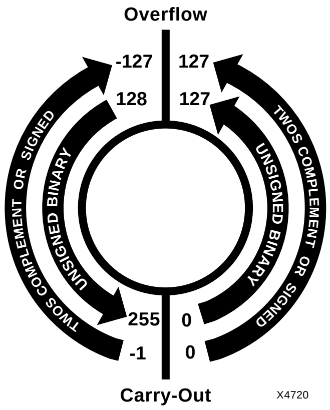


Figure 2-1 ADSU Carry-Out and Overflow Boundaries

XC7000	XC9000	Description	Page
ACC1	ACC1	1-Bit Accumulator with Carry-In, Carry-Out, and Synchronous Reset	3-1
ACC1X1	NA	1-Bit Accumulator with Carry-Out for XC7000	3-4
ACC1X2	NA	1-Bit Accumulator with Carry-In and Carry-Out for XC7000	3-7
ACC4	ACC4	4-Bit Accumulator with Carry-In, Carry-Out, and Synchronous Reset	3-10
ACC4X1	NA	4-Bit Accumulator with Carry-Out for XC7000	3-13
ACC4X2	NA	4-Bit Accumulator with Carry-In and Carry-Out for XC7000	3-15
ACC8	ACC8	8-Bit Accumulator with Carry-In, Carry-Out, and Synchronous Reset	3-17
ACC8X1	NA	8-Bit Accumulator with Carry-Out for XC7000	3-21

XC7000	XC9000	Description	Page
ACC8X2	NA	8-Bit Accumulator with Carry-In and Carry-Out for XC7000	3-24
ACC16	ACC16	16-Bit Accumulator with Carry-In, Carry-Out, and Synchronous Reset	3-27
ACC16X1	NA	16-Bit Accumulator with Carry-Out for XC7000	3-29
ACC16X2	NA	16-Bit Accumulator with Carry-In and Carry-Out for XC7000	3-31
ADD1	ADD1	1-Bit Full Adder with Carry-In and Carry-Out	3-33
ADD1X1	NA	1-Bit Adder with Carry-Out for XC7000	3-35
ADD1X2	NA	1-Bit Adder with Carry-In and Carry-Out for XC7000	3-37
ADD4	ADD4	4-Bit Cascadable Full Adder with Carry-In and Carry-Out	3-38
ADD4X1	NA	4-Bit Adder with Carry-Out for XC7000	3-41
ADD4X2	NA	4-Bit Adder with Carry-In and Carry-Out for XC7000	3-42
ADD8	ADD8	8-Bit Cascadable Full Adder with Carry-In and Carry-Out	3-43
ADD8X1	NA	8-Bit Adder with Carry-Out for XC7000	3-46
ADD8X2	NA	8-Bit Adder with Carry-In and Carry-Out for XC7000	3-48
ADD16	ADD16	16-Bit Cascadable Full Adder with Carry-In and Carry-Out	3-50
ADD16X1	NA	16-Bit Adder with Carry-Out for XC7000	3-52
ADD16X2	NA	16-Bit Adder with Carry-In and Carry-Out for XC7000	3-54
ADSU1	ADSU1	1-Bit Adder/Substracter with Carry-In and Carry-Out	3-55
ADSU1X1	NA	1-Bit Adder/Subtracter with Carry-Out for XC7000	3-58
ADSU1X2	NA	1-Bit Adder/Subtracter with Carry-In and Carry-Out for XC7000	3-60
ADSU4	ADSU4	4-Bit Cascadable Adder/Subtracter with Carry-In and Carry-Out	3-62
ADSU4X1	NA	4-Bit Adder/Subtracter with Carry-Out for XC7000	3-65
ADSU4X2	NA	4-Bit Adder/Subtracter with Carry-In and Carry-Out for XC7000	3-66
ADSU8	ADSU8	8-Bit Adder/Subtracter with Carry-In, Carry-Out, and Overflow	3-67

XC7000	XC9000	Description	Page
ADSU8X1	NA	8-Bit Adder/Subtracter with Carry-Out for XC7000	3-71
ADSU8X2	NA	8-Bit Adder/Subtracter with Carry-In and Carry-Out for XC7000	3-73
ADSU16	ADSU16	16-Bit Adder/Subtracter with Overflow	3-75
ADSU16X1	NA	16-Bit Adder/Subtracter with Carry-Out for XC7000	3-78
ADSU16X2	NA	16-Bit Adder/Subtracter with Carry-In and Carry-Out for XC7000	3-80
X74_280	X74_280	9-Bit Odd/Even Parity Generator/Checker	3-376
X74_283	X74_283	4-Bit Full Adder with Carry-In and Carry-Out	3-378

Buffers

The buffers in this section are used to generate on-chip 3-state signals, or to isolate nets in a design. The "Input/Output Functions" section later in this chapter covers off-chip interface buffers.

XC7000	XC9000	Description	Page
BUF* BUF4, BUF8, BUF16	BUF* BUF4, BUF8, BUF16	General Purpose Buffers	3-86
BUFT*, BUFT4, BUFT8, BUFT16	BUFT*, BUFT4, BUFT8, BUFT16	Internal 3-State Buffers with Active-Low Enable	3-94

*Primitive Symbol

Comparators

There are two types of comparators, identity (COMP) and magnitude (COMPM).

XC7000	XC9000	Description	Page
COMP2	COMP2	2-Bit Identity Comparator	3-170
COMP4	COMP4	4-Bit Identity Comparator	3-171
COMP8	COMP8	8-Bit Identity Comparator	3-172
COMP16	COMP16	16-Bit Identity Comparator	3-173
COMPM2	COMPM2	2-Bit Magnitude Comparator	3-174
COMPM4	COMPM4	4-Bit Magnitude Comparator	3-175
COMPM8	COMPM8	8-Bit Magnitude Comparator	3-176
NA	COMPM16	16-Bit Magnitude Comparator for XC9000	3-179
X74_L85	X74_L85	4-Bit Expandable Magnitude Comparator	3-324
X74_518	X74_518	8-Bit Identity Comparator with Active-Low Enable	3-388
X74_521	X74_521	8-Bit Identity Comparator with Active-Low Enable and Output	3-389

Counters

There are four types of counters with various synchronous and asynchronous inputs. The name of the counter defines the modulo or bit size, the counter type, and which control functions are included. The counter naming convention is shown in the following figure.

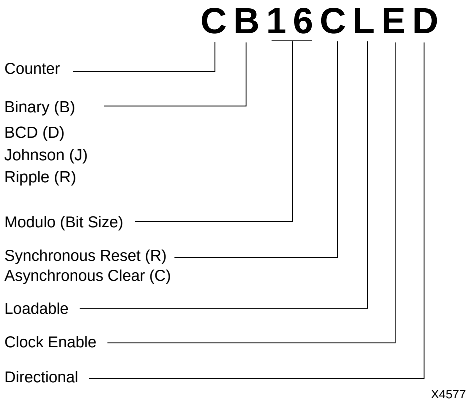


Figure 2-2 Counter Naming Convention

A carry-lookahead design accommodates large counters without extra gating. On TTL 7400-type counters with trickle clock enable (ENT), parallel clock enable (ENP), and ripple carry-out (RCO), both the ENT and ENP inputs must be High to count. ENT is propagated forward to enable RCO, which produces a High output with the approximate duration of the QA output.

XC7000	XC9000	Description	Page
CB2CE	CB2CE	2-Bit Cascadable Binary Counter with Clock Enable and Asynchronous Clear	3-96
CB2CLE	CB2CLE	2-Bit Loadable Cascadable Binary Counter with Clock Enable and Asynchronous Clear	3-98
CB2CLED	CB2CLED	2-Bit Loadable Cascadable Bidirectional Binary Counter with Clock Enable and Asynchronous Clear	3-100
CB2RE	CB2RE	2-Bit Cascadable Binary Counter with Clock Enable and Synchronous Reset	3-102
CB2RLE	CB2RLE	2-Bit Loadable Cascadable Binary Counter with Clock Enable and Synchronous Reset	3-104

XC7000	XC9000	Description	Page
CB2X1	CB2X1	2-Bit Loadable Cascadable Bidirectional Binary Counter with Asynchronous Clear	3-106
CB2X2	CB2X2	2-Bit Loadable Cascadable Bidirectional Binary Counter with Synchronous Reset	3-108
CB4CE	CB4CE	4-Bit Cascadable Binary Counter with Clock Enable and Asynchronous Clear	3-110
CB4CLE	CB4CLE	4-Bit Loadable Cascadable Binary Counter with Clock Enable and Asynchronous Clear	3-111
CB4CLED	CB4CLED	4-Bit Loadable Cascadable Bidirectional Binary Counter with Clock Enable and Asynchronous Clear	3-113
CB4RE	CB4RE	4-Bit Cascadable Binary Counter with Clock Enable and Synchronous Reset	3-116
CB4RLE	CB4RLE	4-Bit Loadable Cascadable Binary Counter with Clock Enable and Synchronous Reset	3-117
CB4X1	CB4X1	4-Bit Loadable Cascadable Bidirectional Binary Counter with Asynchronous Clear	3-119
CB4X2	CB4X2	4-Bit Loadable Cascadable Bidirectional Binary Counter with Synchronous Reset	3-122
CB8CE	CB8CE	8-Bit Cascadable Binary Counter with Clock Enable and Asynchronous Clear	3-125
CB8CLE	CB8CLE	8-Bit Loadable Cascadable Binary Counter with Clock Enable and Asynchronous Clear	3-127
CB8CLED	CB8CLED	8-Bit Loadable Cascadable Bidirectional Binary Counter with Clock Enable and Asynchronous Clear	3-130
CB8RE	CB8RE	8-Bit Cascadable Binary Counter with Clock Enable and Synchronous Reset	3-132
CB8RLE	CB8RLE	8-Bit Loadable Cascadable Binary Counter with Clock Enable and Synchronous Reset	3-134
CB8X1	CB8X1	8-Bit Loadable Cascadable Bidirectional Binary Counter with Asynchronous Clear	3-137
CB8X2	CB8X2	8-Bit Loadable Cascadable Bidirectional Binary Counter with Synchronous Reset	3-139
CB16CE	CB16CE	16-Bit Cascadable Binary Counter with Clock Enable and Asynchronous Clear	3-141
CB16CLE	CB16CLE	16-Bit Loadable Cascadable Binary Counter with Clock Enable and Asynchronous Clear	3-142

XC7000	XC9000	Description	Page
CB16CLED	CB16CLED	16-Bit Loadable Cascadable Bidirectional Binary Counter with Clock Enable and Asynchronous Clear	3-144
CB16RE	CB16RE	16-Bit Cascadable Binary Counter with Clock Enable and Synchronous Reset	3-146
CB16RLE	CB16RLE	16-Bit Loadable Cascadable Binary Counter with Clock Enable and Synchronous Reset	3-147
CB16X1	CB16X1	16-Bit Loadable Cascadable Bidirectional Binary Counter with Asynchronous Clear	3-149
CB16X2	CB16X2	16-Bit Loadable Cascadable Bidirectional Binary Counter with Synchronous Reset	3-151
CD4CE	CD4CE	4-Bit Cascadable BCD Counter with Clock Enable and Asynchronous Clear	3-153
CD4CLE	CD4CLE	4-Bit Loadable Cascadable BCD Counter with Clock Enable and Asynchronous Clear	3-155
CD4RE	CD4RE	4-Bit Cascadable BCD Counter with Clock Enable and Synchronous Reset	3-157
CD4RLE	CD4RLE	4-Bit Loadable Cascadable BCD Counter with Clock Enable and Synchronous Reset	3-159
CJ4CE	CJ4CE	4-Bit Johnson Counter with Clock Enable and Asynchronous Clear	3-162
CJ4RE	CJ4RE	4-Bit Johnson Counter with Clock Enable and Synchronous Reset	3-163
CJ5CE	CJ5CE	5-Bit Johnson Counter with Clock Enable and Asynchronous Clear	3-164
CJ5RE	CJ5RE	5-Bit Johnson Counter with Clock Enable and Synchronous Reset	3-165
CJ8CE	CJ8CE	8-Bit Johnson Counter with Clock Enable and Asynchronous Clear	3-166
CJ8RE	CJ8RE	8-Bit Johnson Counter with Clock Enable and Synchronous Reset	3-168
CR8CE	CR8CE	8-Bit Negative-Edge Binary Ripple Counter with Clock Enable and Asynchronous Clear	3-180
CR16CE	CR16CE	16-Bit Negative-Edge Binary Ripple Counter with Clock Enable and Asynchronous Clear	3-182
X74_160	X74_160	4-Bit Loadable Cascadable BCD Counter with Parallel and Trickle Enables and Asynchronous Clear	3-348

XC7000	XC9000	Description	Page
X74_161	X74_161	4-Bit Loadable Cascadable Binary Counter with Parallel and Trickle Enables and Asynchronous Clear	3-351
X74_162	X74_162	4-Bit Loadable Cascadable BCD Counter with Parallel and Trickle Enables and Synchronous Reset	3-354
X74_163	X74_163	4-Bit Loadable Cascadable Binary Counter with Parallel and Trickle Enables and Synchronous Reset	3-357
X74_168	X74_168	4-Bit Loadable Cascadable Bidirectional BCD Counter with Parallel and Trickle Enables	3-364
X74_390	X74_390	4-Bit BCD / Bi-Quinary Ripple Counter with Negative-Edge Clocks and Asynchronous Clear	3-385

Decoders

Decoder names, shown in the following figure, indicate the number of inputs and outputs and if an enable is available. Decoders with an enable can be used as multiplexers. This group includes some standard TTL 7400-type decoders whose names have an "X74" prefix.

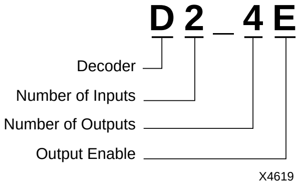


Figure 2-3 Decoder Naming Convention

XC7000	XC9000	Description	Page
D2_4E	D2_4E	2- to 4-Line Decoder / Demultiplexer with Enable	3-183
D3_8E	D3_8E	3- to 8-Line Decoder / Demultiplexer with Enable	3-184
D4_16E	D4_16E	4- to 16-Line Decoder / Demultiplexer with Enable	3-186
X74_42	X74_42	4- to 10-Line BCD-to-Decimal Decoder with Active-Low Outputs	3-322
X74_138	X74_138	3- to 8-Line Decoder / Demultiplexer with Active-Low Outputs and Three Enables	3-326
X74_139	X74_139	2- to 4-Line Decoder / Demultiplexer with Active-Low Outputs and Active-Low Enable	3-328
X74_154	X74_154	4- to 16-Line Decoder / Demultiplexer with Two Enables and Active-Low Outputs	3-342

Encoders

There are two priority encoders (ENCPR) that function like the TTL 7400-type elements they are named after. There is a 10- to 4-line BCD encoder and an 8- to 3-line binary encoder.

XC7000	XC9000	Description	Page
X74_147	X74_147	10- to 4-Line Priority Encoder with Active-Low Inputs and Outputs	3-330
X74_148	X74_148	8- to 3-Line Cascadable Priority Encoder with Active-Low Inputs and Outputs	3-332

Flip-Flops

There are three types of flip-flops (D, J-K, toggle) with various synchronous and asynchronous inputs. Some are available with inverted clock inputs and / or the ability to set in response to global set / reset rather than reset. The naming convention shown in the following figure provides a description for each flip-flop. D-type flip-flops are available in multiples of up to 16 in one macro.

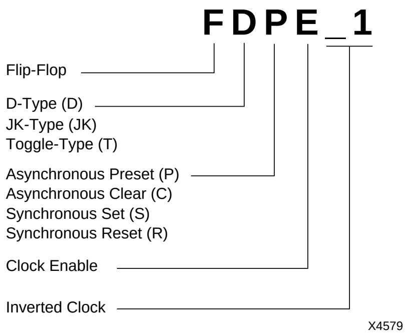


Figure 2-4 Flip-Flop Naming Convention

XC7000	XC9000	Description	Page
FD FD4, FD8, FD16	FD FD4, FD8, FD16	Single and Multiple D Flip-Flops	3-188
FD4CE	FD4CE	4-Bit Data Register with Clock Enable and Asynchronous Clear	3-190
FD4RE	FD4RE	4-Bit Data Register with Clock Enable and Synchronous Reset	3-191
FD8CE	FD8CE	8-Bit Data Register with Clock Enable and Asynchronous Clear	3-192
FD8RE	FD8RE	8-Bit Data Register with Clock Enable and Synchronous Reset	3-194
FD16CE	FD16CE	16-Bit Data Register with Clock Enable and Asynchronous Clear	3-196
FD16RE	FD16RE	16-Bit Data Register with Clock Enable and Synchronous Reset	3-197
FDC	FDC	D Flip-Flop with Asynchronous Clear	3-198

XC7000	XC9000	Description	Page
FDCE	FDCE	D Flip-Flop with Clock Enable and Asynchronous Clear	3-199
FDCP*	FDCP*	D Flip-Flop with Asynchronous Preset and Clear	3-200
FDCPE	FDCPE	D Flip-Flop with Clock Enable and Asynchronous Preset and Clear	3-201
FDP	FDP	D Flip-Flop with Asynchronous Preset	3-203
FDPE	FDPE	D Flip-Flop with Clock Enable and Asynchronous Preset	3-204
FDR	FDR	D Flip-Flop with Synchronous Reset	3-205
FDRE	FDRE	D Flip-Flop with Clock Enable and Synchronous Reset	3-206
FDRS	FDRS	D Flip-Flop with Synchronous Reset and Synchronous Set	3-208
FDRSE	FDRSE	D Flip-Flop with Synchronous Reset and Set and Clock Enable	3-209
FDS	FDS	D Flip-Flop with Synchronous Set	3-211
FDSE	FDSE	D Flip-Flop with Clock Enable and Synchronous Set	3-212
FDSR	FDSR	D Flip-Flop with Synchronous Set and Reset	3-213
FDSRE	FDSRE	D Flip-Flop with Synchronous Set and Reset and Clock Enable	3-214
FJKC	FJKC	J-K Flip-Flop with Asynchronous Clear	3-216
FJKCE	FJKCE	J-K Flip-Flop with Clock Enable and Asynchronous Clear	3-217
FJKCP	FJKCP	J-K Flip-Flop with Asynchronous Clear and Preset	3-219
FJKCPE	FJKCPE	J-K Flip-Flop with Asynchronous Clear and Preset and Clock Enable	3-221
FJKP	FJKP	J-K Flip-Flop with Asynchronous Preset	3-223
FJKPE	FJKPE	J-K Flip-Flop with Clock Enable and Asynchronous Preset	3-224
FJKRSE	FJKRSE	J-K Flip-Flop with Clock Enable and Synchronous Reset and Set	3-226
FJKSRE	FJKSRE	J-K Flip-Flop with Clock Enable and Synchronous Set and Reset	3-228
FTC	FTC	Toggle Flip-Flop with Toggle Enable and Asynchronous Clear	3-230

XC7000	XC9000	Description	Page
FTCE	FTCE	Toggle Flip-Flop with Toggle and Clock Enable and Asynchronous Clear	3-231
FTCLE	FTCLE	Toggle/Loadable Flip-Flop with Toggle and Clock Enable and Asynchronous Clear	3-232
FTCP	FTCP	Toggle Flip-Flop with Toggle Enable and Asynchronous Clear and Preset	3-234
FTCPE	FTCPE	Toggle Flip-Flop with Toggle and Clock Enable and Asynchronous Clear and Preset	3-236
FTCPLE	FTCPLE	Loadable Toggle Flip-Flop with Toggle and Clock Enable and Asynchronous Clear and Preset	3-238
FTP	FTP	Toggle Flip-Flop with Toggle Enable and Asynchronous Preset	3-240
FTPE	FTPE	Toggle Flip-Flop with Toggle and Clock Enable and Asynchronous Preset	3-241
FTPLE	FTPLE	Toggle/Loadable Flip-Flop with Toggle and Clock Enable and Asynchronous Preset	3-243
FTRSE	FTRSE	Toggle Flip-Flop with Toggle and Clock Enable and Synchronous Reset and Set	3-245
FTRSLE	FTRSLE	Toggle/Loadable Flip-Flop with Toggle and Clock Enable and Synchronous Reset and Set	3-247
FTSRE	FTSRE	Toggle Flip-Flop with Toggle and Clock Enable and Synchronous Set and Reset	3-249
FTSRLE	FTSRLE	Toggle/Loadable Flip-Flop with Toggle and Clock Enable and Synchronous Set and Reset	3-251
X74_174	X74_174	6-Bit Data Register with Active-Low Asynchronous Clear	3-367
X74_273	X74_273	8-Bit Data Register with Active-Low Asynchronous Clear	3-374
X74_377	X74_377	8-Bit Data Register with Active-Low Clock Enable	3-383

* Primitive Symbol

General

General elements include FPGA configuration functions, oscillators, boundary-scan logic, and other functions not classified in other sections.

XC7000	XC9000	Description	Page
GND*	GND*	Ground-Connection Signal Tag	3-253
TIMEGRP*	TIMEGRP*	Timing Requirement Group Definition Table	3-315
TIMESPEC*	TIMESPEC*	Schematic-Level Timing Requirement Table	3-314
VCC	VCC	VCC-Connection Signal Tag	3-317

* Primitive Symbol

Input/Output Flip-Flops and Latches

Input/output flip-flops and latches are configured in IOBs. They include flip-flops whose outputs are enabled by 3-state buffers, flip-flops that can be set upon global set/reset rather than reset, and flip-flops with inverted clock inputs. The naming convention specifies each flip-flop function and is illustrated in the following figure.

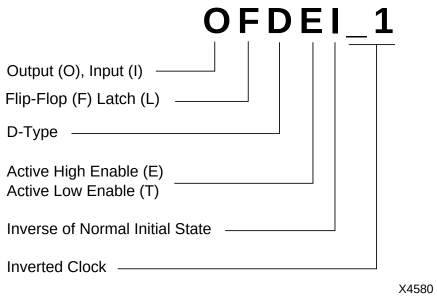


Figure 2-5 Input/Output Flip-Flop Naming Convention

XC7000	XC9000	Description	Page
IFD, IFD4, IFD8, IFD16	IFD, IFD4, IFD8, IFD16	Single- and Multiple-Input D Flip-Flops	3-255
IFDX1*, IFD4X1, IFD8X1, IFD16X1	NA NA NA NA	Input D Flip-Flops with Clock Enable for XC7000	3-258
ILD*, ILD4, ILD8, ILD16	NA NA NA NA	Input Transparent Data Latches for XC7000	3-260
OFD, OFD4, OFD8, OFD16	OFD, OFD4, OFD8, OFD16	Single- and Multiple-Output D Flip-Flops	3-281
OFDE, OFDE4, OFDE8, OFDE16	OFDE, OFDE4, OFDE8, OFDE16	D Flip-Flops with Active-High 3-State Output Buffers	3-283

XC7000	XC9000	Description	Page
OFDT, OFDT4, OFDT8, OFDT16	OFDT, OFDT4, OFDT8, OFDT16	Single and Multiple D Flip-Flops with Active-Low 3-State Output Buffers	3-285

* Primitive Symbol

Input/Output Functions

Input/Output Block (IOB) resources are configured into various I/O primitives and macros for convenience, such as, output buffers (OBUFs) and output buffers with an enable (OBUFEs). Pads used to connect the circuit to PLD device pins are also included.

XC7000	XC9000	Description	Page
BUFCE*	NA	Global Clock-Enable Input Buffer for XC7000	3-87
BUFE, BUFE4, BUFE8, BUFE16	BUFE, BUFE4, BUFE8, BUFE16	Internal 3-State Buffers with Active-High Enable	3-88
BUFFOE*	NA	Global Fast-Output-Enable (FOE) Input Buffer for XC7000	3-90
BUFG*	BUFG*	Global Clock Buffer	3-91
NA	BUFGSR*	Global Set/Reset (GSR) Input Buffer for XC9000	3-92
NA	BUFGTS*	Global 3-State Control (GTS) Input Buffer for XC9000	3-93
IBUF*, IBUF4, IBUF8, IBUF16	IBUF*, IBUF4, IBUF8, IBUF16	Single- and Multiple-Input Buffers	3-254
IOPAD*, IOPAD4, IOPAD8, IOPAD16	IOPAD*, IOPAD4, IOPAD8, IOPAD16	Single- and Multiple-Input/Output Pads	3-263
IPAD*, IPAD4, IPAD8, IPAD16	IPAD*, IPAD4, IPAD8, IPAD16	Single- and Multiple-Input Pads	3-264

XC7000	XC9000	Description	Page
OBUF*, OBUF4, OBUF8, OBUF16	OBUF*, OBUF4, OBUF8, OBUF16	Single- and Multiple-Output Buffers	3-277
OBUFE, OBUFE4, OBUFE8, OBUFE16	OBUFE, OBUFE4, OBUFE8, OBUFE16	3-State Output Buffers with Active-High Fast Output Enable	3-278
OBUFT*, OBUFT4, OBUFT8, OBUFT16	OBUFT*, OBUFT4, OBUFT8, OBUFT16	Single and Multiple 3-State Output Buffers with Active-Low Enable	3-279
OPAD*, OPAD4, OPAD8, OPAD16	OPAD*, OPAD4, OPAD8, OPAD16	Single- and Multiple-Output Pads	3-287
UPAD*	UPAD*	Unbonded I/O pad	3-316

* Primitive Symbol

Latches

Latches (LD) are available in XC7000 and XC9000 architectures.

XC7000	XC9000	Description	Page
LD LD4, LD8, LD16	LD LD4, LD8, LD16	Single and Multiple Transparent Data Latches	3-265

Logic Primitives

Combinatorial logic gates that implement the basic Boolean functions are available in XC7000 XC9000 architectures with up to five inputs in all combinations of inverted and non-inverted inputs, and with six to nine inputs non-inverted.

XC7000	XC9000	Description	Page
AND2*, AND2B1*, AND2B2*, AND3*, AND3B1*, AND3B2*, AND3B3*, AND4*, AND4B1*, AND4B2*, AND4B3*, AND4B4*, AND5*, AND5B1*, AND5B2*, AND5B3*, AND5B4*, AND5B5*, AND6*, AND7*, AND8*, AND9*	AND2*, AND2B1*, AND2B2*, AND3*, AND3B1*, AND3B2*, AND3B3*, AND4*, AND4B1*, AND4B2*, AND4B3*, AND4B4*, AND5*, AND5B1*, AND5B2*, AND5B3*, AND5B4*, AND5B5*, AND6*, AND7*, AND8*, AND9*	2- to 9-Input AND Gates with Inverted and Non-Inverted Inputs	3-82
INV*, INV4, INV8, INV16	INV*, INV4, INV8, INV16	Single and Multiple Inverters	3-262

XC7000	XC9000	Description	Page
NAND2*, NAND2B1*, NAND2B2*, NAND3*, NAND3B1*, NAND3B2*, NAND3B3*, NAND4*, NAND4B1*, NAND4B2*, NAND4B3*, NAND4B4*, NAND5*, NAND5B1*, NAND5B2*, NAND5B3*, NAND5B4*, NAND5B5*, NAND6*, NAND7*, NAND8*, NAND9*	NAND2*, NAND2B1*, NAND2B2*, NAND3*, NAND3B1*, NAND3B2*, NAND3B3*, NAND4*, NAND4B1*, NAND4B2*, NAND4B3*, NAND4B4*, NAND5*, NAND5B1*, NAND5B2*, NAND5B3*, NAND5B4*, NAND5B5*, NAND6*, NAND7*, NAND8*, NAND9*	2- to 9-Input NAND Gates with Inverted and Non-Inverted Inputs	3-275

XC7000	XC9000	Description	Page
NOR2*, NOR2B1*, NOR2B2*, NOR3*, NOR3B1*, NOR3B2*, NOR3B3*, NOR4*, NOR4B1*, NOR4B2*, NOR4B3*, NOR4B4*, NOR5*, NOR5B1*, NOR5B2*, NOR5B3*, NOR5B4*, NOR5B5*, NOR6*, NOR7*, NOR8*, NOR9*	NOR2*, NOR2B1*, NOR2B2*, NOR3*, NOR3B1*, NOR3B2*, NOR3B3*, NOR4*, NOR4B1*, NOR4B2*, NOR4B3*, NOR4B4*, NOR5*, NOR5B1*, NOR5B2*, NOR5B3*, NOR5B4*, NOR5B5*, NOR6*, NOR7*, NOR8*, NOR9*	2- to 9-Input NOR Gates with Inverted and Non-Inverted Inputs	3-276

XC7000	XC9000	Description	Page
OR2*, OR2B1*, OR2B2*, OR3*, OR3B1*, OR3B2*, OR3B3*, OR4*, OR4B1*, OR4B2*, OR4B3*, OR4B4*, OR5*, OR5B1*, OR5B2*, OR5B3*, OR5B4*, OR5B5*, OR6*, OR7*, OR8*, OR9*	OR2*, OR2B1*, OR2B2*, OR3*, OR3B1*, OR3B2*, OR3B3*, OR4*, OR4B1*, OR4B2*, OR4B3*, OR4B4*, OR5*, OR5B1*, OR5B2*, OR5B3*, OR5B4*, OR5B5*, OR6*, OR7*, OR8*, OR9*	2- to 9-Input OR Gates with Inverted and Non-Inverted Inputs	3-288
SOP3, SOP3B1A, SOP3B1B, SOP3B2A, SOP3B2B, SOP3B3, SOP4, SOP4B1, SOP4B2A, SOP4B2B, SOP4B3, SOP4B4	SOP3, SOP3B1A, SOP3B1B, SOP3B2A, SOP3B2B, SOP3B3, SOP4, SOP4B1, SOP4B2A, SOP4B2B, SOP4B3, SOP4B4	Sum of Products	3-289

XC7000	XC9000	Description	Page
XNOR2*, XNOR3*, XNOR4*, XNOR5, XNOR6, XNOR7, XNOR8, XNOR9	XNOR2*, XNOR3*, XNOR4*, XNOR5, XNOR6, XNOR7, XNOR8, XNOR9	2- to 9-Input XNOR Gates with Non-Inverted Inputs	3-318
XOR2*, XOR3*, XOR4*, XOR5, XOR6, XOR7, XOR8, XOR9	XOR2*, XOR3*, XOR4*, XOR5, XOR6, XOR7, XOR8, XOR9	2- to 9-Input XOR Gates with Non-Inverted Inputs	3-320

* Primitive Symbol

Multiplexers

The multiplexer naming convention shown in the following figure, indicates the number of inputs and outputs and if an enable is available. There are a number of TTL 7400-type multiplexers that have active-Low or inverted outputs.

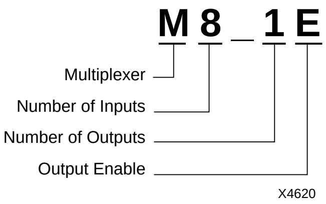


Figure 2-6 Multiplexer Naming Convention

XC7000	XC9000	Description	Page
M2_1	M2_1	2-to-1 Multiplexer	3-267
M2_1B1	M2_1B1	2-to-1 Multiplexer with D0 Inverted	3-268
M2_1B2	M2_1B2	2-to-1 Multiplexer with D0 and D1 Inverted	3-269
M2_1E	M2_1E	2-to-1 Multiplexer with Enable	3-270
M4_1E	M4_1E	4-to-1 Multiplexer with Enable	3-271
M8_1E	M8_1E	8-to-1 Multiplexer with Enable	3-272
M16_1E	M16_1E	16-to-1 Multiplexer with Enable	3-274
X74_150	X74_150	16-to-1 Multiplexer Active-Low Enab. & Output	3-334
X74_151	X74_151	8-to-1 Multiplexer with Active-Low Enable and Complementary Outputs	3-336
X74_152	X74_152	8-to-1 Multiplexer with Active-Low Output	3-338
X74_153	X74_153	Dual 4-to-1 Multiplexer with Active-Low Enables and Common Select Input	3-340
X74_157	X74_157	Quadruple 2-to-1 Multiplexer with Common Select and Active-Low Enable	3-344
X74_158	X74_158	Quadruple 2-to-1 Multiplexer with Common Select, Active-Low Enable, and Active-Low Outputs	3-346
X74_298	X74_298	Quadruple 2-Input Multiplexer with Storage and Negative-Edge Clock	3-379
X74_352	X74_352	Dual 4-to-1 Multiplexer with Active-Low Enables and Outputs	3-381

Shifters

Shift registers and barrel shifters are available in a variety of sizes and capabilities. The naming convention shown in the following figure illustrates available features for shift registers.

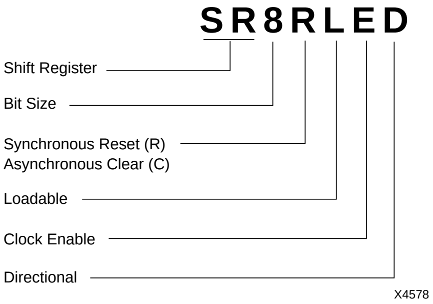


Figure 2-7 Shift Register Naming Convention

XC7000	XC9000	Description	Page
BRLSHFT4	BRLSHFT4	4-Bit Barrel Shifter	3-83
BRLSHFT8	BRLSHFT8	8-Bit Barrel Shifter	3-84
SR4CE	SR4CE	4-Bit Serial-In Parallel-Out Shift Register with Clock Enable and Asynchronous Clear	3-290
SR4CLE	SR4CLE	4-Bit Loadable Serial / Parallel-In Parallel-Out Shift Register with Clock Enable and Asynchronous Clear	3-291
SR4CLED	SR4CLED	4-Bit Shift Register with Clock Enable and Asynchronous Clear	3-292
SR4RE	SR4RE	4-Bit Serial-In Parallel-Out Shift Register with Clock Enable and Synchronous Reset	3-293
SR4RLE	SR4RLE	4-Bit Loadable Serial / Parallel-In Parallel-Out Shift Register with Clock Enable and Synchronous Reset	3-294
SR4RLED	SR4RLED	4-Bit Shift Register with Clock Enable and Synchronous Reset	3-295
SR8CE	SR8CE	8-Bit Serial-In Parallel-Out Shift Register with Clock Enable and Asynchronous Clear	3-296
SR8CLE	SR8CLE	8-Bit Loadable Serial / Parallel-In Parallel-Out Shift Register with Clock Enable and Asynchronous Clear	3-298

XC7000	XC9000	Description	Page
SR8CLED	SR8CLED	8-Bit Shift Register with Clock Enable and Asynchronous Clear	3-300
SR8RE	SR8RE	8-Bit Serial-In Parallel-Out Shift Register with Clock Enable and Synchronous Reset	3-302
SR8RLE	SR8RLE	8-Bit Loadable Serial / Parallel-In Parallel-Out Shift Register with Clock Enable and Synchronous Reset	3-304
SR8RLED	SR8RLED	8-Bit Shift Register with Clock Enable and Synchronous Reset	3-306
SR16CE	SR16CE	16-Bit Serial-In Parallel-Out Shift Register with Clock Enable and Asynchronous Clear	3-308
SR16CLE	SR16CLE	16-Bit Loadable Serial / Parallel-In Parallel-Out Shift Register with Clock Enable and Asynchronous Clear	3-309
SR16CLED	SR16CLED	16-Bit Shift Register with Clock Enable and Asynchronous Clear	3-310
SR16RE	SR16RE	16-Bit Serial-In Parallel-Out Shift Register with Clock Enable and Synchronous Reset	3-311
SR16RLE	SR16RLE	16-Bit Loadable Serial / Parallel-In Parallel-Out Shift Register with Clock Enable and Synchronous Reset	3-312
SR16RLED	SR16RLED	16-Bit Shift Register with Clock Enable and Synchronous Reset	3-313
X74_164	X74_164	8-Bit Serial-In Parallel-Out Shift Register with Active-Low Asynchronous Clear	3-360
X74_165S	X74_165S	8-Bit Loadable Serial / Parallel-In Parallel-Out Shift Register with Clock Enable	3-362
X74_194	X74_194	4-Bit Loadable Directional Serial / Parallel-In Parallel-Out Shift Register	3-369
X74_195	X74_195	4-Bit Loadable Serial / Parallel-In Parallel-Out Shift Register	3-372

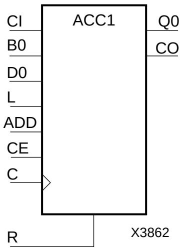
Chapter 3

Design Elements

This chapter contains design elements (primitives and macros) for the XC7000 and XC9000 architectures, organized in alphanumeric order.

ACC1

1-Bit Loadable Cascadable Accumulator with Carry-In, Carry-Out, and Synchronous Reset



XC7000	XC9000
Macro*	Macro

* not supported for XC7336 designs

ACC1 can add or subtract a 1-bit unsigned-binary word to or from the contents of a 1-bit data register and store the results in the register. The register can be loaded with a 1-bit word. The synchronous reset (R) has priority over all other inputs and, when High, causes the output to go to logic level zero. Otherwise, clock (C) transitions are ignored when clock enable (CE) is Low. The default initial state of the flip-flop is zero.

Load

When the load input (L) is High, CE is ignored and the data on the input D0 is loaded into the 1-bit register.

Add

When control inputs ADD and CE are both High, the accumulator adds a 1-bit word (B0) and carry-in (CI) to the contents of the 1-bit register. The result is stored in the register and appears on output Q0 during the Low-to-High clock transition. The carry-out (CO) is not registered synchronously with the data output. CO allows cascading of ACC1s by connecting CO of one stage to CI of the next stage. In add mode, CO acts as a carry-out, and CO and CI are active-High.

Subtract

When ADD is Low and CE is High, the 1-bit word B0 and CI are subtracted from the contents of the register. The result is stored in the register and appears on output Q0 during the Low-to-High clock transition. The carry-out (CO) is not registered synchronously with the data output. CO allows cascading of ACC1s by connecting CO of one stage to CI of the next stage. In subtract mode, CO acts as a borrow, and CO and CI are active-Low.

For the XC7000 CPLD architecture, the CO output is not valid during load (L=High), during reset (R=High), or while CE is inactive (Low). Also, the CI and CO pins are not implemented using the XC7000 arithmetic carry path and should not be used to cascade accumulators. Refer to “ACC1X1” and “ACC1X2” for descriptions of cascadable XC7000 accumulators.

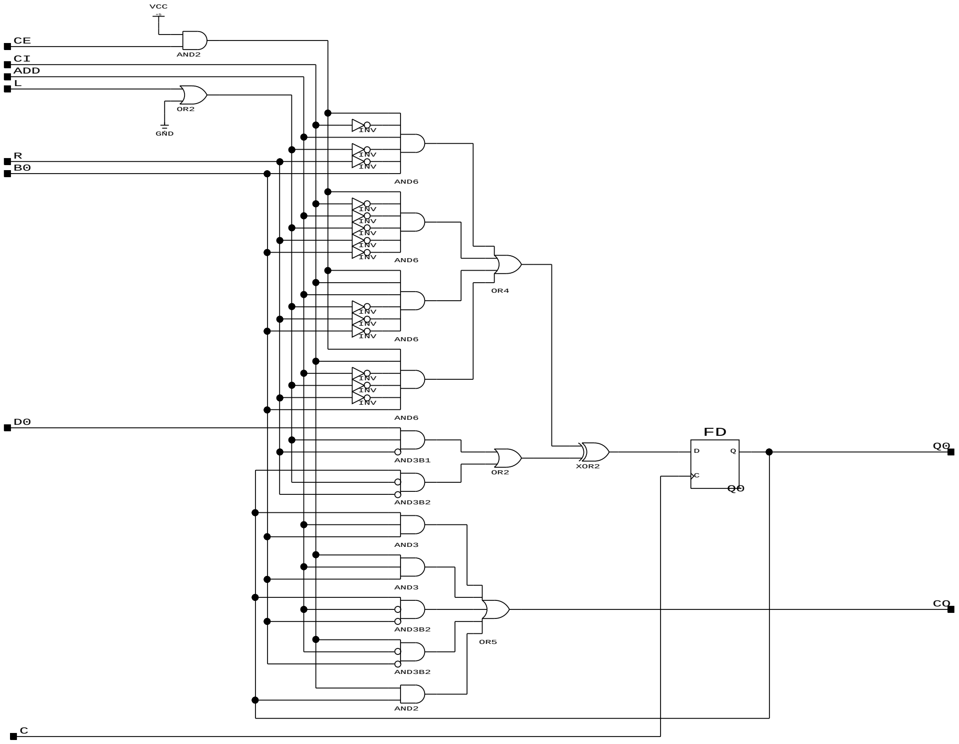
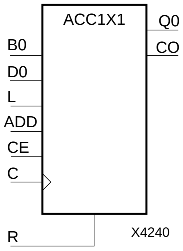


Figure 3-1 ACC1 Implementation (XC7000 and XC9000)

ACC1X1

1-Bit Loadable Cascadable Accumulator with Carry-Out for XC7000



XC7000	XC9000
Macro*	NA

*not supported for XC7336 designs

ACC1X1 is implemented using the XC7000 arithmetic carry-logic chain for high-speed ripple-carry addition. ACC1X1 is a low-order adder component that can be used as a stand-alone or cascaded with high-order accumulators through its CO output. ACC1X1 adds or subtracts a 1-bit binary word (B0) to or from the contents of a 1-bit data register and stores the results in the register. The register can be loaded with a 1-bit word. When the load input (L) is High, CE is ignored and the data on input D0 is loaded into the 1-bit register. The synchronous reset (R) has priority over all other inputs and, when High, causes all outputs to go to logic level zero. When reset (R) and load (L) are inactive, clock (C) transitions are ignored when clock enable (CE) is Low.

Add

When control inputs ADD and CE are both High, the accumulator adds a 1-bit word (B0) to the contents of the 1-bit register. The result is stored in the register and appears on output Q0 during the Low-to-High clock transition. In add mode, CO acts as a carry-out and is active-High.

Subtract

When ADD is Low and CE is High, the 1-bit word B0 is subtracted from the contents of the register. The result is stored in the register and appears on output Q0 during the Low-to-High clock transition. In subtract mode, CO acts as a borrow and is active-Low.

The CO output is passed into the XC7000 carry chain, and therefore can only be connected to the CI input of another XC7000-specific arithmetic component. To generate a carry-out signal for general-purpose logic, connect an ADD1X2 to the CO output of the accumulator and tie its A and B inputs to GND; the S output becomes the carry-out. If a carry-in is required from general-purpose logic, use an ACC1X2 for the least-significant accumulator and connect an ADD1X1 to its CI input. Then connect your carry-in signal to both the A and B inputs of the ADD1X1 (the S output is not used) to generate a carry into the carry chain for the first bit of the accumulator.

The default initial state of the flip-flop is zero.

Inputs							Outputs	
R	**L**	**CE**	**B0**	**D0**	**C**	**ADD**	**Q0**	**CO**
1	X	X	X	X	↑	X	0	0
0	1	X	X	D0	↑	X	d	0
0	0	0	X	X	X	X	No Chg	0
0	0	1	B0	X	↑	1	q+b	CO
0	0	1	B0	X	↑	0	q-b	CO

d, q, b = state of referenced input one set-up time prior to active clock transition

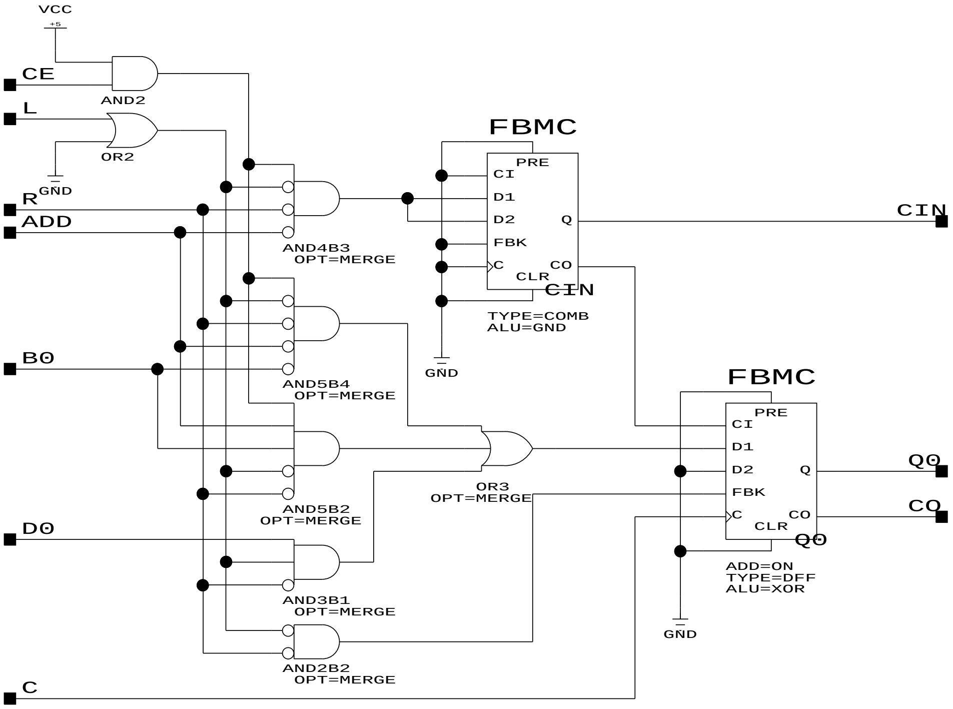
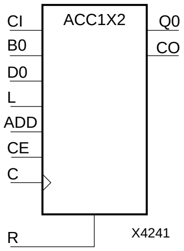


Figure 3-2 ACC1X1 Implementation (XC7000)

ACC1X2

1-Bit Loadable Cascadable Accumulator with Carry-In and Carry-Out for XC7000



XC7000	XC9000
Macro*	NA

* not supported for XC7336 designs

ACC1X2 is implemented using the XC7000 arithmetic carry-logic chain for high-speed ripple-carry addition. ACC1X2 is a high-order adder component cascaded to lower-order accumulators through its CI input. ACC1X2 adds or subtracts a 1-bit binary word (B0) to or from the contents of a 1-bit data register and stores the results in the register. The register can be loaded with a 1-bit word. When the load input (L) is High, CE is ignored and the data on input D0 is loaded into the 1-bit register. The synchronous reset (R) has priority over all other inputs and, when High, causes all outputs to go to logic level zero. When reset (R) and load (L) are inactive, clock (C) transitions are ignored when clock enable (CE) is Low.

Add

When control inputs ADD and CE are both High, the accumulator adds a 1-bit word (B0) and carry-in (CI) to the contents of the 1-bit register. The result is stored in the register and appears on output Q0 during the Low-to-High clock transition. In add mode, CO acts as a carry-out, and CO and CI are active-High.

Subtract

When ADD is Low and CE is High, the 1-bit word B0 and CI are subtracted from the contents of the register. The result is stored in the register and appears on output Q0 during the Low-to-High clock transition. In subtract mode, CO acts as a borrow, and CO and CI are active-Low.

The CI input is taken from the XC7000 carry chain, and therefore, must only be connected to the CO output of another XC7000-specific arithmetic component. The CO output is passed into the XC7000 carry chain, and can only be connected to the CI input of another

XC7000-specific arithmetic component. To generate a carry-out signal for general-purpose logic, connect an ADD1X2 to the CO output of the accumulator and tie its A and B inputs to GND; the S output becomes the carry-out.

The default initial state of the flip-flop is zero.

Inputs								Outputs	
R	**L**	**CE**	**B0**	**D0**	**CI**	**C**	**ADD**	**Q0**	**CO**
1	X	X	X	X	X	↑	X	0	0
0	1	X	X	D0	X	↑	X	d	0
0	0	0	X	X	X	X	X	No Chg	0
0	0	1	B0	X	CI	↑	1	q+b+ci	CO
0	0	1	B0	X	CI	↑	0	q-b-ci	CO

d, q, b, ci = state of referenced input one set-up time prior to active clock transition

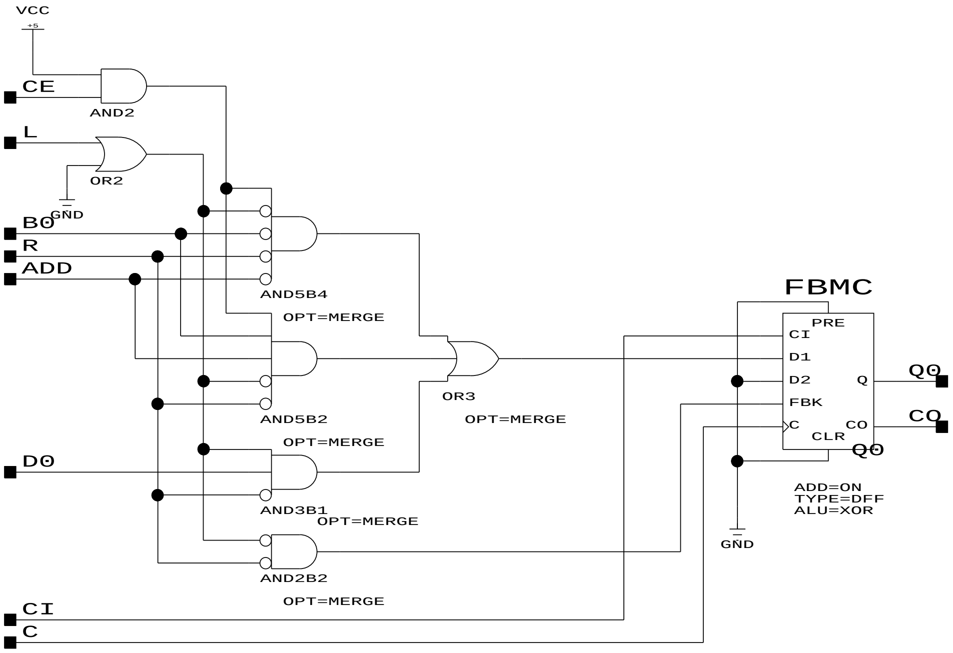
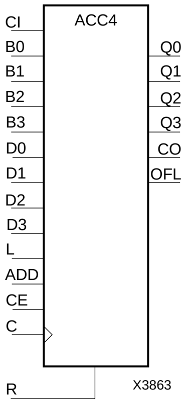


Figure 3-3 ACC1X2 Implementation (XC7000)

ACC4

4-Bit Loadable Cascadable Accumulator with Carry-In, Carry-Out, and Synchronous Reset



XC7000	XC9000
Macro*	Macro

* not supported for XC7336 designs

ACC4 can add or subtract a 4-bit unsigned-binary or twos-complement word to or from the contents of a 4-bit data register and store the results in the register. The register can be loaded with a 4-bit word. The synchronous reset (R) has priority over all other inputs, and when High, causes all outputs to go to logic level zero. Otherwisae, clock (C) transitions are ignored when clock enable (CE) is Low. The default initial state of all flip-flops is zero.

Load

When the load input (L) is High, CE is ignored and the data on inputs D3 – D0 is loaded into the 4-bit register.

Unsigned Binary Versus Twos-Complement

ACC4 can operate on either 4-bit unsigned binary numbers or 4-bit twos-complement numbers. If the inputs are interpreted as unsigned binary, the result can be interpreted as unsigned binary. If the inputs are interpreted as twos complement, the output can be interpreted as twos complement. The only functional difference between an unsigned binary operation and a twos-complement operation is how they determine when “overflow” occurs. Unsigned binary uses CO, while twos-complement uses OFL to determine when “overflow” occurs.

For the XC7000 CPLD architecture, the CO output is not valid during load (L=High), during reset (R=High), or while CE is inactive (Low). Also, the CI and CO pins are not implemented using the XC7000 arithmetic carry path and should not be used to cascade accumulators. Refer to “ACC1X1” and “ACC1X2” for descriptions of cascadable XC7000 accumulators. The OFL output is not provided on the ACC4 symbol in XC7000.

Unsigned Binary Operation

For unsigned binary operation, the ACC4 can represent numbers between 0 and 15, inclusive. In add mode, CO is active (High) when the sum exceeds the bounds of the adder / subtracter. In subtract mode, CO is an active-Low borrow-out and goes Low when the difference exceeds the bounds. The carry-out (CO) is not registered synchronously with the data outputs. CO always reflects the accumulation of inputs B3 – B0 and the contents of the register, which allows cascading of ACC4s by connecting CO of one stage to CI of the next stage. An unsigned binary “overflow” that is always active-High can be generated by gating the ADD signal and CO as follows.

```
unsigned overflow = CO XOR ADD
```

OFL should be ignored in unsigned binary operation.

Twos-Complement Operation

For twos-complement operation, ACC4 can represent numbers between -8 and +7, inclusive. If an addition or subtraction operation result exceeds this range, the OFL output goes High. The overflow (OFL) is not registered synchronously with the data outputs. OFL always reflects the accumulation of inputs B3 – B0 and the contents of the register, which allows cascading of ACC4s by connecting OFL of one stage to CI of the next stage.

CO should be ignored in twos-complement operation.

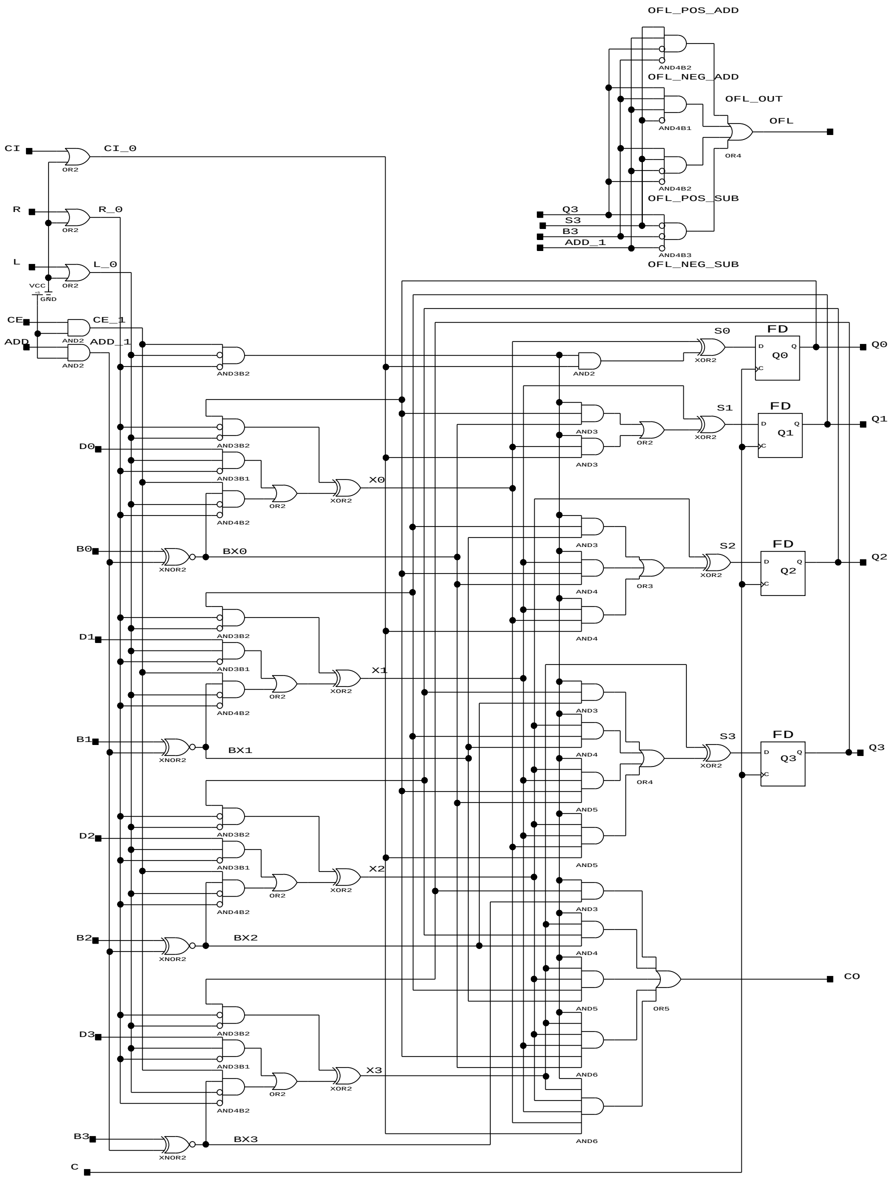
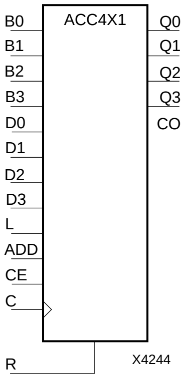


Figure 3-4   ACC4 Implementation (XC9000)

ACC4X1

4-Bit Loadable Cascadable Accumulator with Carry-Out for XC7000



XC7000	XC9000
Macro*	NA

* not supported for XC7336 designs

ACC4X1 is implemented using the XC7000 arithmetic carry-logic chain for high-speed ripple-carry addition. ACC4X1 is a low-order adder component, which can be used as a stand-alone or cascaded with high-order accumulators through its CO output. ACC4X1 adds or subtracts a 4-bit binary word (B3 – B0) to or from the contents of a 4-bit data register and stores the results in the register. The register can be loaded with a 4-bit word. When the load input (L) is High, CE is ignored and the data on inputs D3 – D0 is loaded into the 4-bit register. The synchronous reset (R) has priority over all other inputs and, when High, causes all outputs to go to logic level zero. When reset (R) and load (L) are inactive, clock (C) transitions are ignored when clock enable (CE) is Low.

Add

When control inputs ADD and CE are both High, the accumulator adds a 4-bit word (B3 – B0) to the contents of the 4-bit register. The result is stored in the register and appears on outputs Q3 – Q0 during the Low-to-High clock transition. In add mode, CO acts as a carry-out and is active-High.

Subtract

When ADD is Low and CE is High, the 4-bit word B3 – B0 is subtracted from the contents of the register. The result is stored in the register and appears on outputs Q3 – Q0 during the Low-to-High clock transition. In subtract mode, CO acts as a borrow and is active-Low.

The CO output is passed into the XC7000 carry chain, and therefore can only be connected to the CI input of another XC7000-specific arithmetic component. To generate a carry-out signal for general-

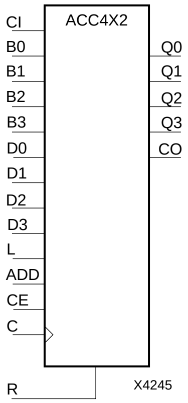
purpose logic, connect an ADD1X2 to the CO output of the accumulator and tie its A and B inputs to GND; the S output becomes the carry-out. If a carry-in is required from general-purpose logic, use an ACC4X2 for the least-significant accumulator and connect an ADD1X1 to its CI input. Then connect your carry-in signal to both the A and B inputs of the ADD1X1 (the S output is not used) to generate a carry into the carry chain for the first bit of the accumulator.

The default initial state of all flip-flops is zero.

Refer to “ACC1X1” for truth table derivation.

ACC4X2

4-Bit Loadable Cascadable Accumulator with Carry-In and Carry-Out for XC7000



XC7000	XC9000
Macro*	NA

* not supported for XC7336 designs

ACC4X2 is implemented using the XC7000 arithmetic carry-logic chain for high-speed ripple-carry addition. ACC4X2 is a high-order adder component cascaded to lower-order accumulators through its CI input. ACC4X2 adds or subtracts a 4-bit binary word (B3 – B0) to or from the contents of a 4-bit data register and stores the results in the register. The register can be loaded with a 4-bit word. When the load input (L) is High, CE is ignored and the data on inputs D3 – D0 is loaded into the 4-bit register. The synchronous reset (R) has priority over all other inputs and, when High, causes all outputs to go to logic level zero. When reset (R) and load (L) are inactive, clock (C) transitions are ignored when clock enable (CE) is Low.

Add

When control inputs ADD and CE are both High, the accumulator adds a 4-bit word (B3 – B0) and carry-in (CI) to the contents of the 4-bit register. The result is stored in the register and appears on outputs Q3 – Q0 during the Low-to-High clock transition. In add mode, CO acts as a carry-out, and CO and CI are active-High.

Subtract

When ADD is Low and CE is High, the 4-bit word B3 – B0 and CI are subtracted from the contents of the register. The result is stored in the register and appears on outputs Q3 – Q0 during the Low-to-High clock transition. In subtract mode, CO acts as a borrow, and CO and CI are active-Low.

The CI input is taken from the XC7000 carry chain, and therefore must only be connected to the CO output of another XC7000-specific arithmetic component. The CO output is passed into the XC7000 carry chain and can only be connected to the CI input of another

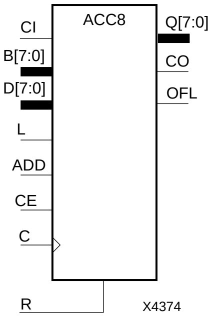
XC7000-specific arithmetic component. To generate a carry-out signal for general-purpose logic, connect an ADD1X2 to the CO output of the accumulator and tie its A and B inputs to GND; the S output becomes the carry-out.

The default initial state of all flip-flops is zero.

Refer to “ACC1X2” for truth table derivation.

ACC8

8-Bit Loadable Cascadable Accumulator with Carry-In, Carry-Out, and Synchronous Reset



XC7000	XC9000
Macro*	Macro

* not supported for XC7336 designs

ACC8 can add or subtract an 8-bit unsigned-binary or twos- complement word to or from the contents of an 8-bit data register and store the results in the register. The register can be loaded with an 8-bit word. The synchronous reset (R) has priority over all other inputs, and when High, causes all outputs to go to logic level zero. Clock (C) transitions are ignored when clock enable (CE) is Low. The default initial state of all flip-flops is zero.

Load

When the load input (L) is High, CE is ignored and the data on inputs D7 – D0 is loaded into the 8-bit register.

Unsigned Binary Versus Twos-Complement

ACC8 can operate on either 8-bit unsigned binary numbers or 8-bit twos-complement numbers. If the inputs are interpreted as unsigned binary, the result can be interpreted as unsigned binary. If the inputs are interpreted as twos complement, the output can be interpreted as twos complement. The only functional difference between an unsigned binary operation and a twos-complement operation is how they determine when “overflow” occurs. Unsigned binary uses CO, while twos-complement uses OFL to determine when “overflow” occurs.

For the XC7000 CPLD architecture, the CO output is not valid during load (L=High), during reset (R=High), or while CE is inactive (Low). Also, the CI and CO pins are not implemented using the XC7000 arithmetic carry path and should not be used to cascade accumulators. Refer to “ACC8X1” and “ACC8X2” for descriptions of cascadable XC7000 accumulators. The OFL output is not provided on the ACC8 symbol in XC7000.

Unsigned Binary Operation

For unsigned binary operation, ACC8 can represent numbers between 0 and 255, inclusive. In add mode, CO is active (High) when the sum exceeds the bounds of the adder / subtracter. In subtract mode, CO is an active-Low borrow-out and goes Low when the difference exceeds the bounds. The carry-out (CO) is not registered synchronously with the data outputs. CO always reflects the accumulation of inputs B7 – B0 and the contents of the register, which allows cascading of ACC8s by connecting CO of one stage to CI of the next stage. An unsigned binary “overflow” that is always active-High can be generated by gating the ADD signal and CO as follows.

```
unsigned overflow = CO XOR ADD
```

OFL should be ignored in unsigned binary operation.

Twos-Complement Operation

For twos-complement operation, ACC8 can represent numbers between -128 and +127, inclusive. If an addition or subtraction operation result exceeds this range, the OFL output goes High. The overflow (OFL) is not registered synchronously with the data outputs. OFL always reflects the accumulation of inputs B7 – B0 and the contents of the register, which allows cascading of ACC8s by connecting OFL of one stage to CI of the next stage. CO should be ignored in twos-complement operation.

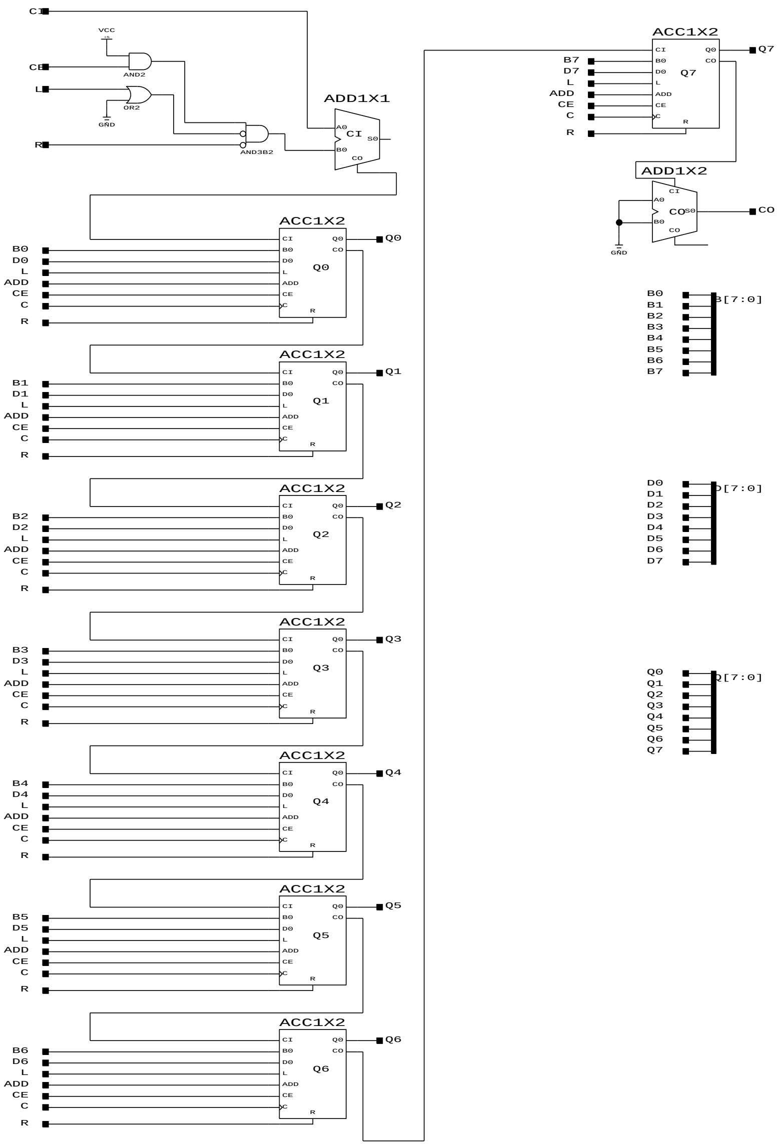


Figure 3-5 ACC8 Implementation (XC7000)

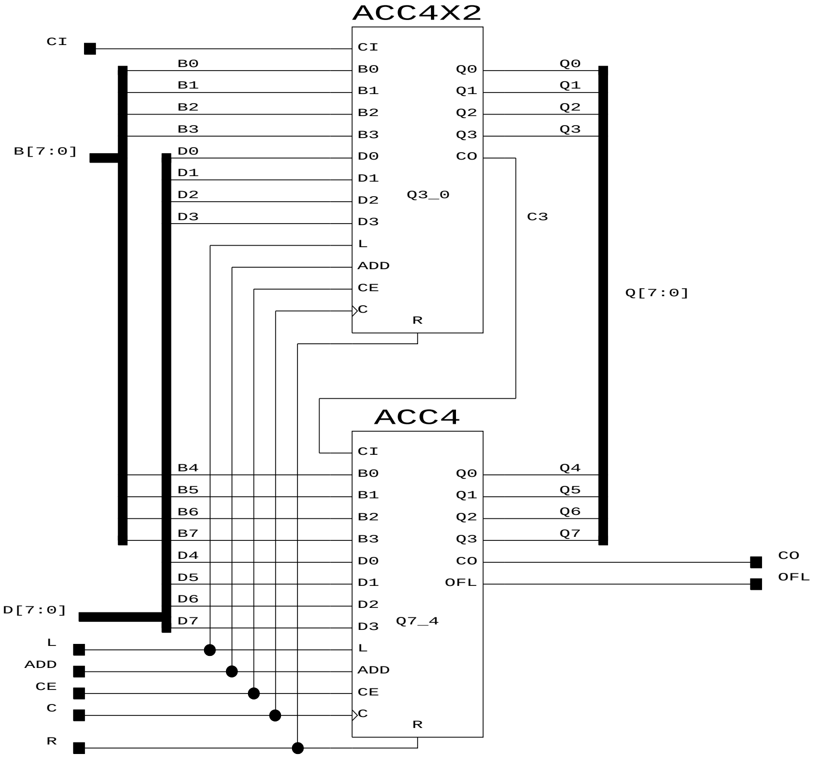
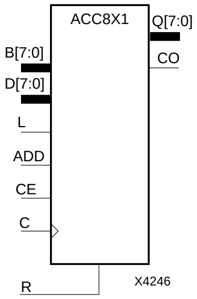


Figure 3-6 ACC8 Implementation (XC9000)

Note: The ACC4X2 schematic (not shown for XC9000) is the same as ACC4 without the OFL output logic.

ACC8X1

8-Bit Loadable Cascadable Accumulator with Carry-Out for XC7000



XC7000	XC9000
Macro*	NA

* not supported for XC7336 designs

ACC8X1 is implemented using the XC7000 arithmetic carry-logic chain for high-speed ripple-carry addition. ACC8X1 is a low-order adder component, which can be used as a stand-alone or cascaded with high-order accumulators through its CO output. ACC8X1 adds or subtracts an 8-bit binary word (B7 – B0) to or from the contents of an 8-bit data register and stores the results in the register. The register can be loaded with an 8-bit word. When the load input (L) is High, CE is ignored and the data on inputs D7 – D0 is loaded into the 8-bit register. The synchronous reset (R) has priority over all other inputs and, when High, causes all outputs to go to logic level zero. When reset (R) and load (L) are inactive, clock (C) transitions are ignored when clock enable (CE) is Low.

Add

When control inputs ADD and CE are both High, the accumulator adds an 8-bit word (B7 – B0) to the contents of the 8-bit register. The result is stored in the register and appears on outputs Q7 – Q0 during the Low-to-High clock transition. In add mode, CO acts as a carry-out and is active-High.

Subtract

When ADD is Low and CE is High, the 8-bit word B7 – B0 is subtracted from the contents of the register. The result is stored in the register and appears on outputs Q7 – Q0 during the Low-to-High clock transition. In subtract mode, CO acts as a borrow and is active-Low.

The CO output is passed into the XC7000 carry chain, and therefore can only be connected to the CI input of another XC7000-specific arithmetic component. To generate a carry-out signal for general-

purpose logic, connect an ADD1X2 to the CO output of the accumulator and tie its A and B inputs to GND; the S output becomes the carry-out. If a carry-in is required from general-purpose logic, use an ACC8X2 for the least-significant accumulator and connect an ADD1X1 to its CI input. Then connect your carry-in signal to both the A and B inputs of the ADD1X1 (the S output is not used) to generate a carry into the carry chain for the first bit of the accumulator.

The default initial state of all flip-flops is zero.

Refer to “ACC1X1” for truth table derivation.

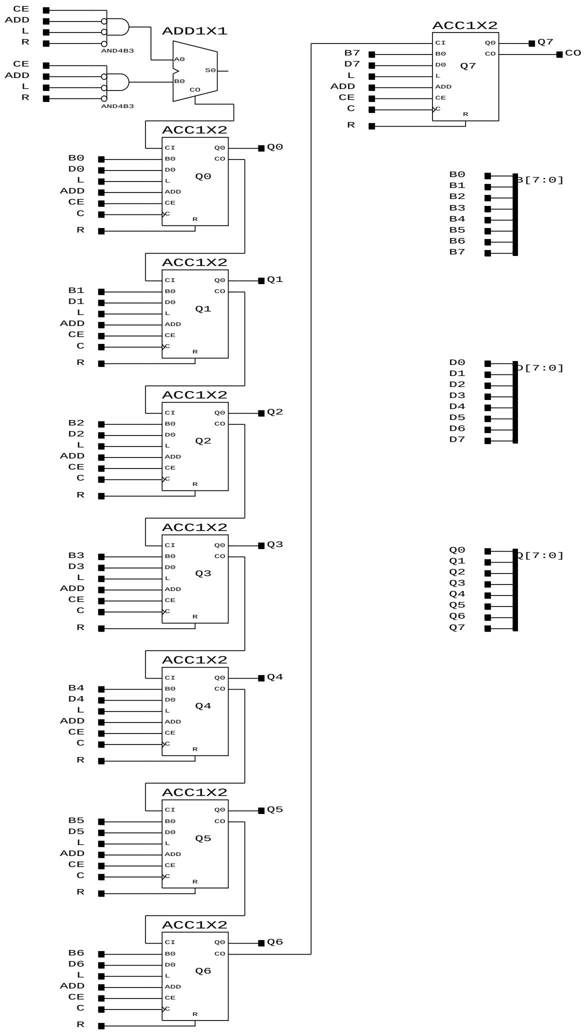
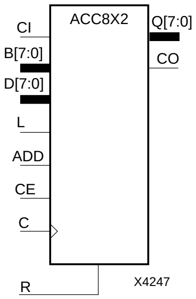


Figure 3-7 ACC8X1 Implementation (XC7000)

ACC8X2

8-Bit Loadable Cascadable Accumulator with Carry-In and Carry-Out for XC7000



XC7000	XC9000
Macro*	NA

* not supported for XC7336 designs

ACC8X2 is implemented using the XC7000 arithmetic carry-logic chain for high-speed ripple-carry addition. ACC8X2 is a high-order adder component cascaded to lower-order accumulators though its CI input. ACC8X2 adds or subtracts an 8-bit binary word (B7 – B0) to or from the contents of an 8-bit data register and stores the results in the register. The register can be loaded with an 8-bit word. When the load input (L) is High, CE is ignored and the data on inputs D7 – D0 is loaded into the 8-bit register. The synchronous reset (R) has priority over all other inputs and, when High, causes all outputs to go to logic level zero. When reset (R) and load (L) are inactive, clock (C) transitions are ignored when clock enable (CE) is Low.

Add

When control inputs ADD and CE are both High, the accumulator adds an 8-bit word (B7 – B0) and carry-in (CI) to the contents of the 8-bit register. The result is stored in the register and appears on outputs Q7 – Q0 during the Low-to-High clock transition. In add mode, CO acts as a carry-out, and CO and CI are active-High.

Subtract

When ADD is Low and CE is High, the 8-bit word B7 – B0 and CI are subtracted from the contents of the register. The result is stored in the register and appears on outputs Q7 – Q0 during the Low-to-High clock transition. In subtract mode, CO acts as a borrow, and CO and CI are active-Low.

The CI input is taken from the XC7000 carry chain, and therefore must only be connected to the CO output of another XC7000-specific arithmetic component. The CO output is passed into the XC7000 carry chain, and therefore can only be connected to the CI input of another XC7000-specific arithmetic component. To generate a carry-out signal for general-purpose logic, connect an ADD1X2 to the CO output of the accumulator and tie its A and B inputs to GND; the S output becomes the carry-out.

The default initial state of all flip-flops is zero.

Refer to “ACC1X2” for truth table derivation.

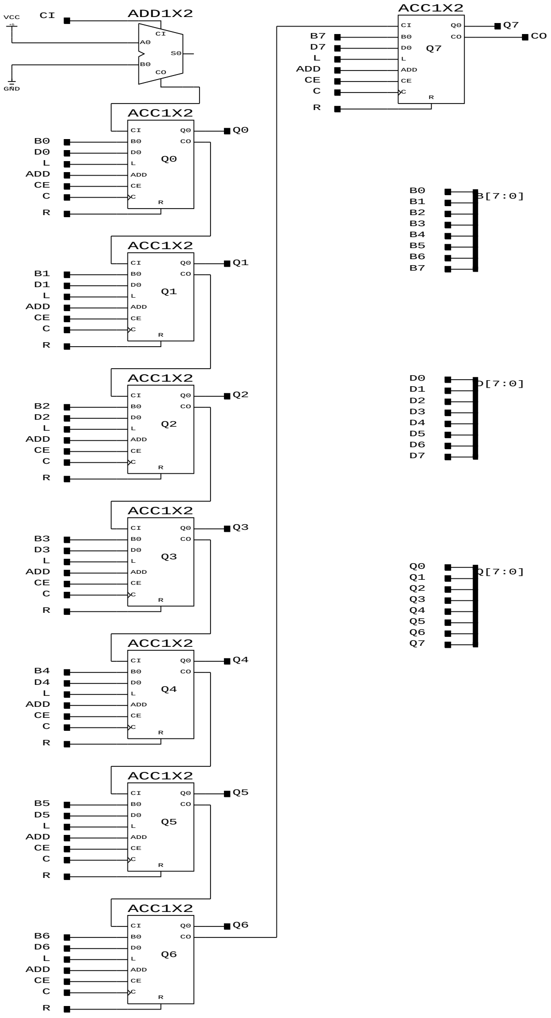
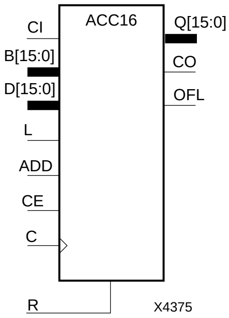


Figure 3-8 ACC8X2 Implementation (XC7000)

ACC16

16-Bit Loadable Cascadable Accumulator with Carry-In, Carry-Out, and Synchronous Reset



XC7000	XC9000
Macro*	Macro

* not supported for XC7336 designs

ACC16 can add or subtract a 16-bit unsigned-binary or twos-complement word to or from the contents of a 16-bit data register and store the results in the register. The register can be loaded with a 16-bit word. The synchronous reset (R) has priority over all other inputs, and when High, causes all outputs to go to logic level zero. Clock (C) transitions are ignored when clock enable (CE) is Low. The default initial state of all flip-flops is zero.

Load

When the load input (L) is High, CE is ignored and the data on inputs D15 – D0 is loaded into the 16-bit register.

Unsigned Binary Versus Twos-Complement

ACC16 can operate on either 16-bit unsigned binary numbers or 16-bit twos-complement numbers. If the inputs are interpreted as unsigned binary, the result can be interpreted as unsigned binary. If the inputs are interpreted as twos complement, the output can be interpreted as twos complement. The only functional difference between an unsigned binary operation and a twos-complement operation is how they determine when “overflow” occurs. Unsigned binary uses CO, while twos-complement uses OFL to determine when “overflow” occurs.

For the XC7000 CPLD architecture, the CO output is not valid during load (L=High), during reset (R=High), or while CE is inactive (Low). Also, the CI and CO pins are not implemented using the XC7000 arithmetic carry path and should not be used to cascade accumulators. Refer to “ACC8X1” and “ACC8X2” for descriptions of cascadable XC7000 accumulators. The OFL output is not provided on the ACC16 symbol in XC7000.

Unsigned Binary Operation

For unsigned binary operation, ACC16 can represent numbers between 0 and 65535, inclusive. In add mode, CO is active (High) when the sum exceeds the bounds of the adder/subtracter. In subtract mode, CO is an active-Low borrow-out and goes Low when the difference exceeds the bounds. The carry-out (CO) is not registered synchronously with the data outputs. CO always reflects the accumulation of inputs B15 – B0 and the contents of the register, which allows cascading of ACC16s by connecting CO of one stage to CI of the next stage. An unsigned binary “overflow” that is always active-High can be generated by gating the ADD signal and CO as follows.

```
unsigned overflow = CO XOR ADD
```

OFL should be ignored in unsigned binary operation.

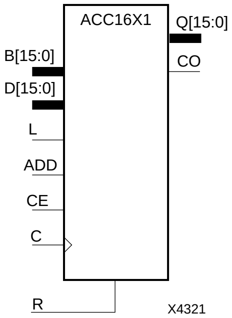
Twos-Complement Operation

For twos-complement operation, ACC16 can represent numbers between -32768 and +32767, inclusive. If an addition or subtraction operation result exceeds this range, the OFL output goes High. The overflow (OFL) is not registered synchronously with the data outputs. OFL always reflects the accumulation of inputs B15 – B0 and the contents of the register, which allows cascading of ACC16s by connecting OFL of one stage to CI of the next stage.

CO should be ignored in twos-complement operation.

ACC16X1

16-Bit Loadable Cascadable Accumulator with Carry-Out for XC7000



XC7000	XC9000
Macro*	NA

* not supported for XC7336 designs

ACC16X1 is implemented using the XC7000 arithmetic carry-logic chain for high-speed ripple-carry addition. ACC16X1 is a low-order adder component, which can be used as a stand-alone or cascaded with high-order accumulators though its CO output. ACC16X1 adds or subtracts a 16-bit binary word (B15 – B0) to or from the contents of a 16-bit data register and stores the results in the register. The register can be loaded with a 16-bit word. When the load input (L) is High, CE is ignored and the data on inputs D15 – D0 is loaded into the 16-bit register. The synchronous reset (R) has priority over all other inputs and, when High, causes all outputs to go to logic level zero. When reset (R) and load (L) are inactive, clock (C) transitions are ignored when clock enable (CE) is Low.

Add

When control inputs ADD and CE are both High, the accumulator adds a 16-bit word (B15 – B0) to the contents of the 16-bit register. The result is stored in the register and appears on outputs Q15 – Q0 during the Low-to-High clock transition. In add mode, CO acts as a carry-out and is active-High.

Subtract

When ADD is Low and CE is High, the 16-bit word B15 – B0 is subtracted from the contents of the register. The result is stored in the register and appears on outputs Q15 – Q0 during the Low-to-High clock transition. In subtract mode, CO acts as a borrow and is active-Low.

The CO output is passed into the XC7000 carry chain, and therefore can only be connected to the CI input of another XC7000-specific arithmetic component. To generate a carry-out signal for general-

purpose logic, connect an ADD1X2 to the CO output of the accumulator and tie its A and B inputs to GND; the S output becomes the carry-out. If a carry-in is required from general-purpose logic, use an ACC16X2 for the least-significant accumulator and connect an ADD1X1 to its CI input. Then connect your carry-in signal to both the A and B inputs of the ADD1X1 (the S output is not used) to generate a carry into the carry chain for the first bit of the accumulator.

The default initial state of all flip-flops is zero.

Refer to “ACC1X1” for truth table derivation.

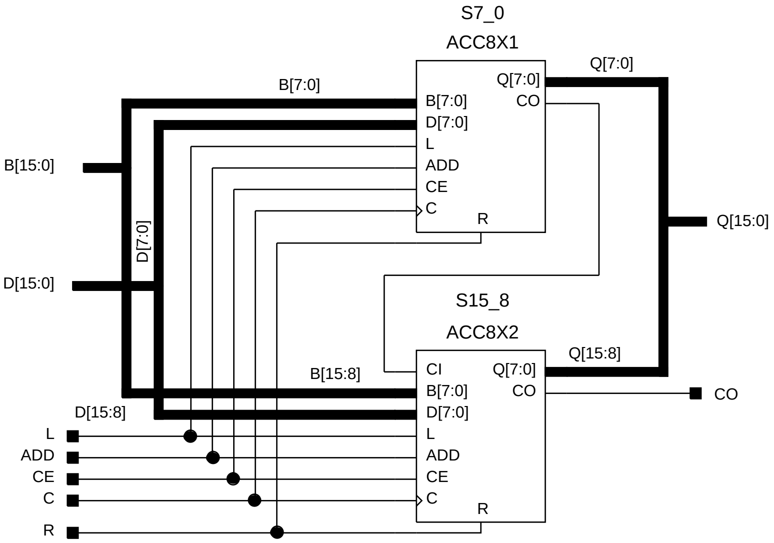
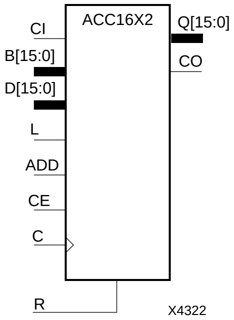


Figure 3-9 ACC16X1 Implementation (XC7000)

ACC16X2

16-Bit Loadable Cascadable Accumulator with Carry-In and Carry-Out for XC7000



XC7000	XC9000
Macro*	NA

* not supported for XC7336 designs

ACC16X2 is implemented using the XC7000 arithmetic carry-logic chain for high-speed ripple-carry addition. ACC16X2 is a high-order adder component cascaded to lower-order accumulators through its CI input. ACC16X2 adds or subtracts a 16-bit binary word (B15 – B0) to or from the contents of a 16-bit data register and stores the results in the register. The register can be loaded with a 16-bit word. When the load input (L) is High, CE is ignored and the data on inputs D15 – D0 is loaded into the 16-bit register. The synchronous reset (R) has priority over all other inputs and, when High, causes all outputs to go to logic level zero. When reset (R) and load (L) are inactive, clock (C) transitions are ignored when clock enable (CE) is Low.

Add

When control inputs ADD and CE are both High, the accumulator adds a 16-bit word (B15 – B0) and carry-in (CI) to the contents of the 16-bit register. The result is stored in the register and appears on outputs Q15 – Q0 during the Low-to-High clock transition. In add mode, CO acts as a carry-out, and CO and CI are active-High.

Subtract

When ADD is Low and CE is High, the 16-bit word B15 – B0 and CI are subtracted from the contents of the register. The result is stored in the register and appears on outputs Q15 – Q0 during the Low-to-High clock transition. In subtract mode, CO acts as a borrow, and CO and CI are active-Low.

The CI input is taken from the XC7000 carry chain, and therefore must only be connected to the CO output of another XC7000-specific arithmetic component. The CO output is passed into the XC7000 carry chain, and therefore can only be connected to the CI input of another XC7000-specific arithmetic component. To generate a carry-out signal for general-purpose logic, connect an ADD1X2 to the CO output of the accumulator and tie its A and B inputs to GND; the S output becomes the carry-out.

The default initial state of all flip-flops is zero.

Refer to “ACC1X2” for truth table derivation.

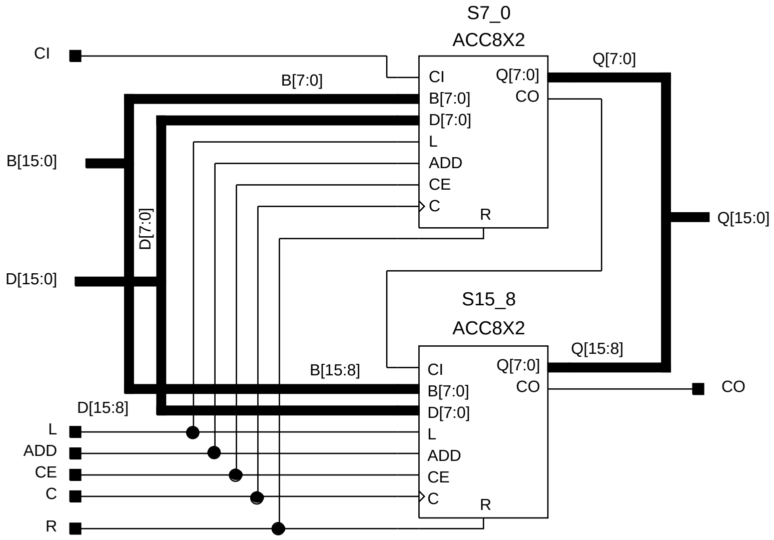
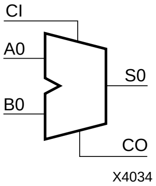


Figure 3-10 ACC16X2 Implementation (XC7000)

ADD1

1-Bit Full Adder with Carry-In and Carry-Out



XC7000	XC9000
Macro*	Macro

* not supported for XC7336 designs

ADD1, a cascadable 1-bit full adder with carry-in and carry-out, adds two 1-bit words (A and B) and a carry-in (CI), producing a binary sum (S0) output and a carry-out (CO). For XC7000 cascadable adders, refer to “ADD1X1” and “ADD1X2.”

Inputs			Outputs	
A0	B0	CI	S0	CO
0	0	0	0	0
1	0	0	1	0
0	1	0	1	0
1	1	0	0	1
0	0	1	1	0
1	0	1	0	1
0	1	1	0	1
1	1	1	1	1

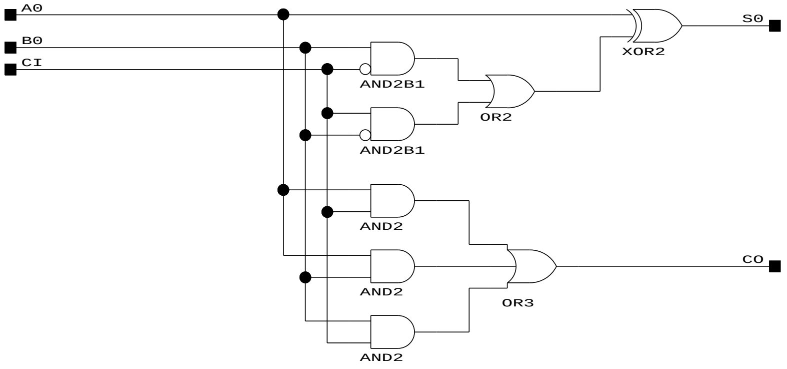
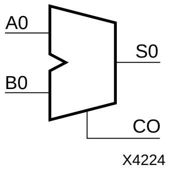


Figure 3-11   ADD1 Implementation (XC7000 and XC9000)

ADD1X1

1-Bit Cascadable Full Adder with Carry-Out for XC7000



XC7000	XC9000
Macro*	NA

* not supported for XC7336 designs

ADD1X1 is implemented using the XC7000 arithmetic carry-logic chain for high-speed ripple-carry addition. ADD1X1 is a low-order adder component, which can be used as a stand-alone or cascaded with high-order adders through its CO output. ADD1X2 adds two words (A0 and B0) and produces a sum output (ADD1X2) and carry-out (CO).

The CO output is passed into the XC7000 carry chain, and therefore can only be connected to the CI input of another XC7000-specific arithmetic component. To generate a carry-out for general-purpose logic, use an adder (or cascaded adders) with one extra bit and tie the most-significant A and B inputs to GND; the most-significant S output becomes the carry-out. If a carry-in is required from general-purpose logic, extend the length of the adder by one additional bit and connect the carry-in signal to both the least-significant A and B inputs (the least-significant S output is not used) to generate a carry into the carry chain for the second bit of the adder.

Refer to “ADD1” for truth table derivation.

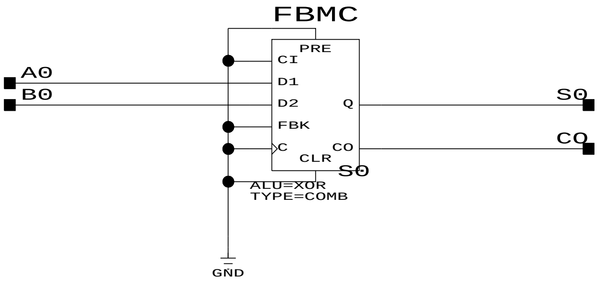
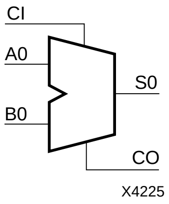


Figure 3-12 ADD1X1 Implementation (XC7000)

ADD1X2

1-Bit Cascadable Full Adder with Carry-In and Carry-Out for XC7000



XC7000	XC9000
Macro*	NA

* not supported for XC7336 designs

ADD1X2 is implemented using the XC7000 arithmetic carry-logic chain for high-speed ripple-carry addition. ADD1X2 is a high-order adder component cascaded to lower-order adders through its CI input. ADD1X2 adds two words (A0 and B0) and a carry-in (CI), producing a sum output (S0) and carry-out (CO).

The CI input is taken from the XC7000 carry chain, and therefore must only be connected to the CO output of another XC7000-specific arithmetic component. The CO output is passed into the XC7000 carry chain, and therefore can only be connected to the CI input of another XC7000-specific arithmetic component. To generate a carry-out for general-purpose logic, use an adder (or cascaded adders) with one extra bit and tie the most-significant A and B inputs to GND; the most-significant S output becomes the carry-out.

Refer to “ADD1” for truth table derivation.

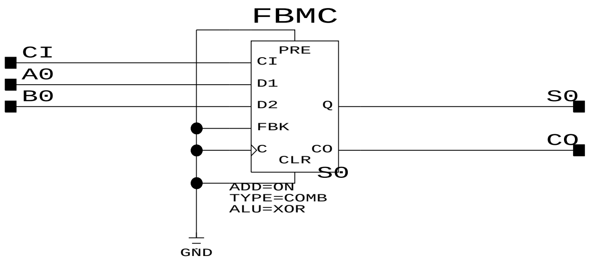
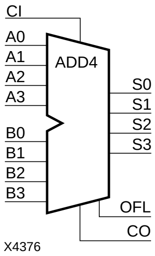


Figure 3-13 ADD1X2 Implementation (XC7000)

ADD4

4-Bit Cascadable Full Adder with Carry-In, Carry-Out, and Overflow



XC7000	XC9000
Macro*	Macro

* not supported for XC7336 designs

ADD4 adds two words (A3 – A0 and B3 – B0) and a carry-in (CI), producing a sum output (S3 – S0) and carry-out (CO) or overflow (OFL). For XC7000 cascadable adders, refer to “ADD4X1” and “ADD4X2.” The ADD4 CI and CO pins do not use the XC7000 carry chain.

Unsigned Binary Versus Twos Complement

ADD4 can operate on either 4-bit unsigned binary numbers or 4-bit twos-complement numbers. If the inputs are interpreted as unsigned binary, the result can be interpreted as unsigned binary. If the inputs are interpreted as twos complement, the output can be interpreted as twos complement. The only functional difference between an unsigned binary operation and a twos-complement operation is how they determine when “overflow” occurs. Unsigned binary uses CO, while twos-complement uses OFL to determine when “overflow” occurs.

Unsigned Binary Operation

For unsigned binary operation, ADD4 can represent numbers between 0 and 15, inclusive. CO is active (High) when the sum exceeds the bounds of the adder.

OFL is ignored in unsigned binary operation.

Twos-Complement Operation

For twos-complement operation, ADD4 can represent numbers between -8 and +7, inclusive. OFL is active (High) when the sum exceeds the bounds of the adder.

CO is ignored in twos-complement operation.

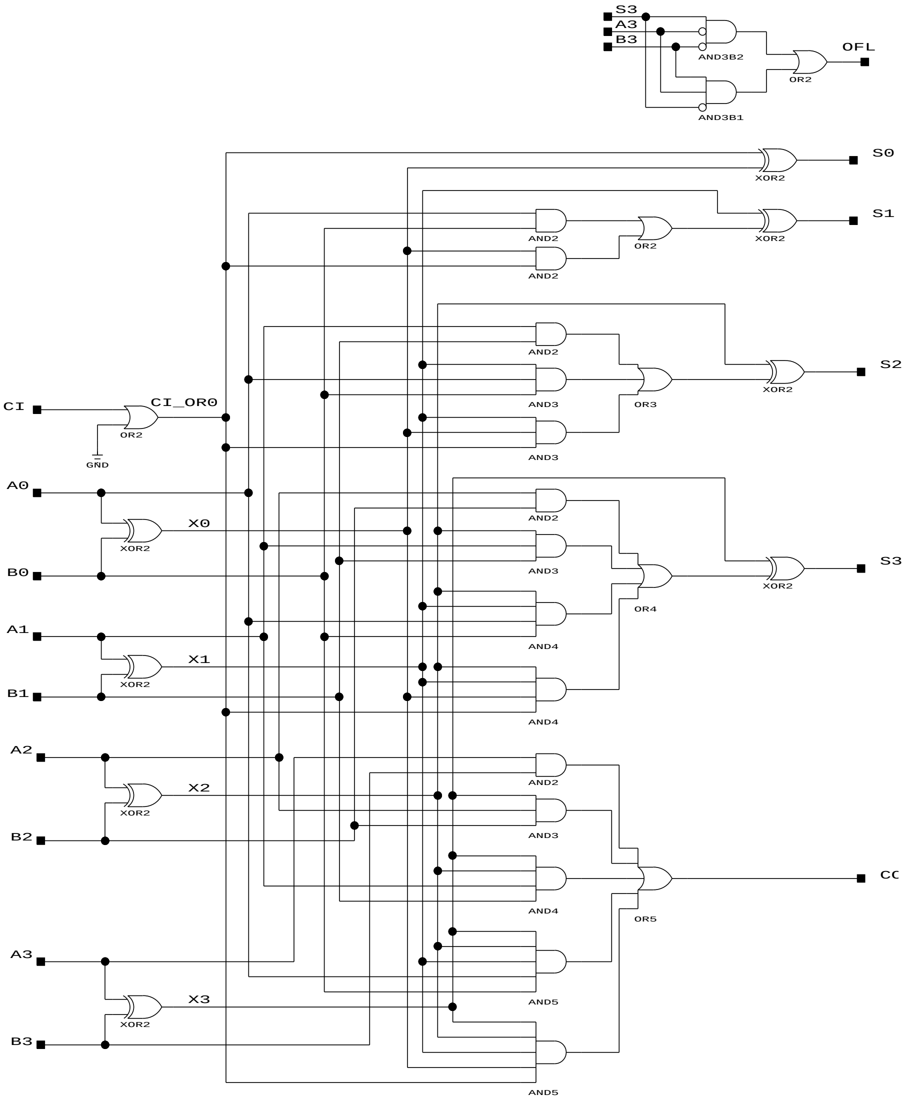
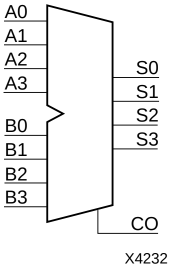


Figure 3-14 ADD4 Implementation (XC9000)

ADD4X1

4-Bit Cascadable Full Adder with Carry-Out for XC7000



XC7000	XC9000
Macro*	NA

* not supported for XC7336 designs

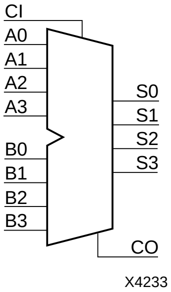
ADD4X1 is implemented using the XC7000 arithmetic carry-logic chain for high-speed ripple-carry addition. ADD4X1 is a low-order adder component, which can be used as a stand-alone or cascaded with high-order adders through its CO output. ADD4X2 adds two words (A3 – A0 and B3 – B0), producing a sum output (S3 – S0) and carry-out (CO).

The CO output is passed into the XC7000 carry chain, and therefore can only be connected to the CI input of another XC7000-specific arithmetic component. To generate a carry-out for general-purpose logic, use an adder (or cascaded adders) with one extra bit and tie the most-significant A and B inputs to GND; the most-significant S output becomes the carry-out. If a carry-in is required from general-purpose logic, extend the length of the adder by one additional bit and connect the carry-in signal to both the least-significant A and B inputs (the least-significant S output is not used) to generate a carry into the carry chain for the second bit of the adder.

Refer to “ADD1” for truth table derivation.

ADD4X2

4-Bit Cascadable Full Adder with Carry-In and Carry-Out for XC7000



XC7000	XC9000
Macro*	NA

* not supported for XC7336 designs

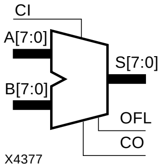
ADD4X2 is implemented using the XC7000 arithmetic carry-logic chain for high-speed ripple-carry addition. ADD4X2 is a high-order adder component cascaded to lower-order adders through its CI input. ADD4X2 adds two words (A3 – A0 and B3 – B0) and a carry-in (CI), producing a sum output (S3 – S0) and carry-out (CO).

The CI input is taken from the XC7000 carry chain, and therefore must only be connected to the CO output of another XC7000-specific arithmetic component. The CO output is passed into the XC7000 carry chain, and therefore can only be connected to the CI input of another XC7000-specific arithmetic component. To generate a carry-out for general-purpose logic, use an adder (or cascaded adders) with one extra bit and tie the most-significant A and B inputs to GND; the most-significant S output becomes the carry-out.

Refer to “ADD1” for truth table derivation.

ADD8

8-Bit Cascadable Full Adder with Carry-In, Carry-Out, and Overflow



XC7000	XC9000
Macro*	Macro

* not supported for XC7336 designs

ADD8 adds two words (A7 – A0 and B7 – B0) and a carry-in (CI), producing a sum output (S7 – S0) and carry-out (CO) or overflow (OFL). For XC7000 cascadable adders, refer to “ADD8X1” and “ADD8X2.” The ADD8 CI and CO pins do not use the XC7000 carry chain.

Unsigned Binary Versus Twos-Complement

ADD8 can operate on either 8-bit unsigned binary numbers or 8-bit twos-complement numbers. If the inputs are interpreted as unsigned binary, the result can be interpreted as unsigned binary. If the inputs are interpreted as twos complement, the output can be interpreted as twos complement. The only functional difference between an unsigned binary operation and a twos-complement operation is how they determine when “overflow” occurs. Unsigned binary uses CO, while twos-complement uses OFL to determine when “overflow” occurs.

Unsigned Binary Operation

For unsigned binary operation, ADD8 can represent numbers between 0 and 255, inclusive. CO is active (High) when the sum exceeds the bounds of the adder.

OFL is ignored in unsigned binary operation.

Twos-Complement Operation

For twos-complement operation, ADD8 can represent numbers between -128 and +127, inclusive. OFL is active (High) when the sum exceeds the bounds of the adder.

CO is ignored in twos-complement operation.

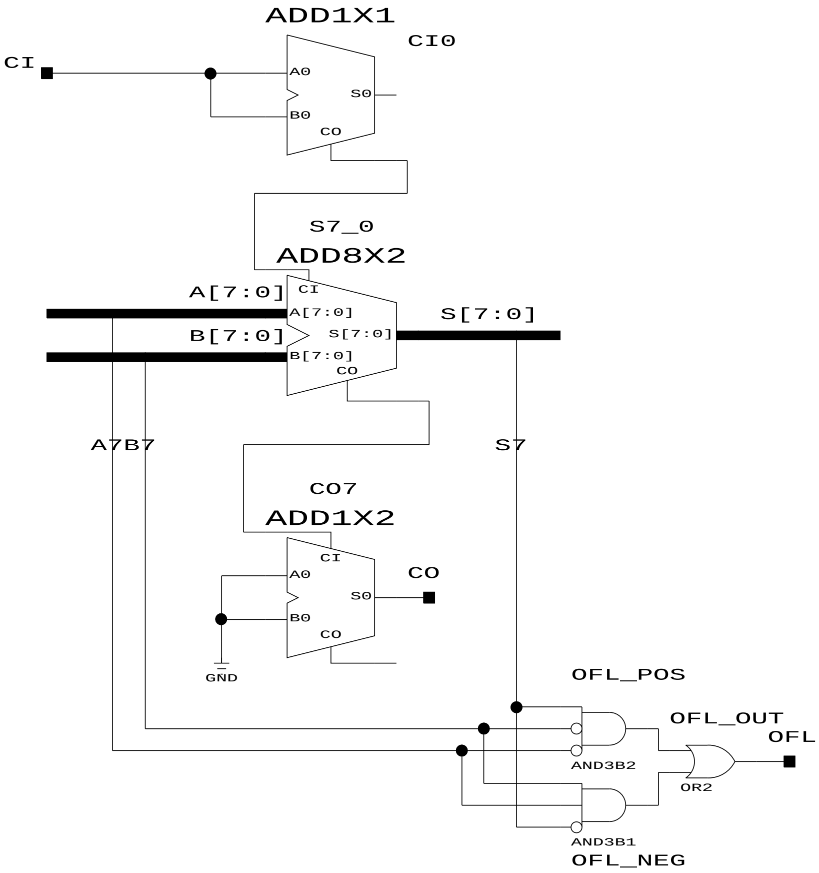


Figure 3-15 ADD8 Implementation (XC7000)

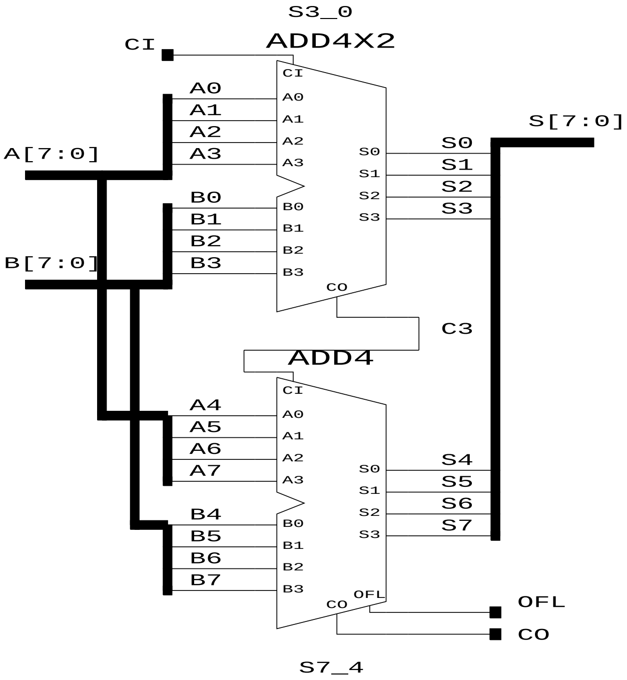
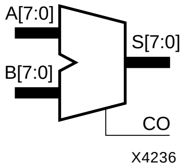


Figure 3-16 ADD8 Implementation (XC9000)

Note: The ADD4X2 schematic (not shown for XC9000) is the same as ADD4 without the OFL output logic.

ADD8X1

8-Bit Loadable Cascadable Full Adder with Carry-Out for XC7000



XC7000	XC9000
Macro*	NA

* not supported for XC7336 designs

ADD8X1 is implemented using the XC7000 arithmetic carry-logic chain for high-speed ripple-carry addition. ADD8X1 is a low-order adder component that can be used stand-alone or cascaded with high-order adders through its CO output. ADD8X2 adds two words (A7 – A0 and B7 – B0), producing a sum output (S7 – S0) and carry-out (CO). The CO output passes into the XC7000 carry chain, and therefore can only be connected to the CI input of another XC7000-specific arithmetic component. To generate a carry-out for general-purpose logic, use an adder (or cascaded adders) with one extra bit and tie the most-significant A and B inputs to GND; the most-significant S output becomes the carry-out. If a carry-in is required from general-purpose logic, extend the length of the adder by one additional bit and connect the carry-in signal to both the least-significant A and B inputs (the least-significant S output is not used). This procedure generates a carry into the carry chain for the second bit of the adder.

Refer to “ADD1” for truth table derivation.

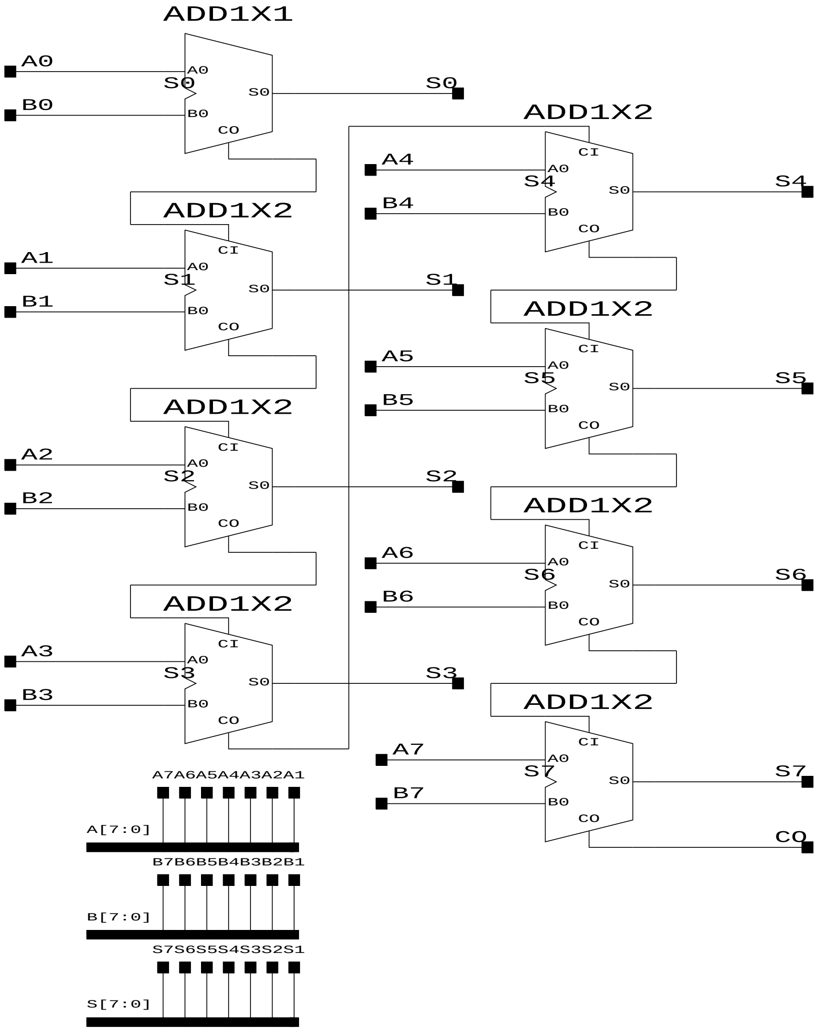
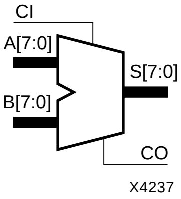


Figure 3-17 ADD8X1 Implementation (XC7000)

ADD8X2

8-Bit Cascadable Full Adder with Carry-In and Carry-Out for XC7000



XC7000	XC9000
Macro*	NA

* not supported for XC7336 designs

ADD8X2 is implemented using the XC7000 arithmetic carry-logic chain for high-speed ripple-carry addition. ADD8X2 is a high-order adder component cascaded to lower-order adders through its CI input. ADD8X2 adds two words (A7 – A0 and B7 – B0) and a carry-in (CI), producing a sum output (S7 – S0) and carry-out (CO).

The CI input is taken from the XC7000 carry chain, and therefore must only be connected to the CO output of another XC7000-specific arithmetic component. The CO output is passed into the XC7000 carry chain, and therefore can only be connected to the CI input of another XC7000-specific arithmetic component. To generate a carry-out for general-purpose logic, use an adder (or cascaded adders) with one extra bit and tie the most-significant A and B inputs to GND; the most-significant S output becomes the carry-out.

Refer to “ADD1” for truth table derivation.

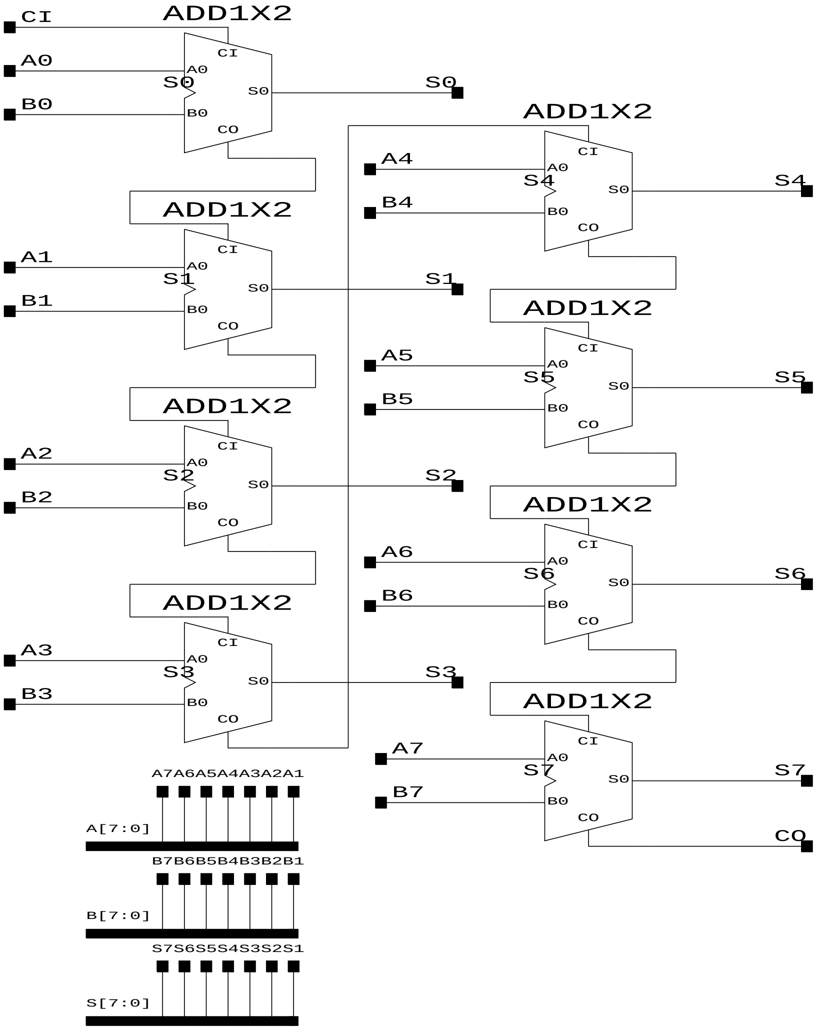
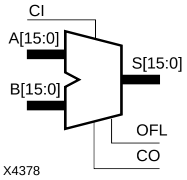


Figure 3-18 ADD8X2 Implementation (XC7000)

ADD16

16-Bit Cascadable Full Adder with Carry-In, Carry-Out, and Overflow



XC7000	XC9000
Macro*	Macro

* not supported for XC7336 designs

ADD16 adds two words (A15 – A0 and B15 – B0) and a carry-in (CI), producing a sum output (S15 – S0) and carry-out (CO) or overflow (OFL). For XC7000 cascadable adders, refer to “ADD16X1” and “ADD16X2.” The ADD16 CI and CO pins do not use the XC7000 carry chain.

Unsigned Binary Versus Twos-Complement

ADD16 can operate on either 16-bit unsigned binary numbers or 16-bit twos-complement numbers. If the inputs are interpreted as unsigned binary, the result can be interpreted as unsigned binary. If the inputs are interpreted as twos complement, the output can be interpreted as twos complement. The only functional difference between an unsigned binary operation and a twos-complement operation is how they determine when “overflow” occurs. Unsigned binary uses CO, while twos-complement uses OFL to determine when “overflow” occurs.

Unsigned Binary Operation

For unsigned binary operation, ADD16 can represent numbers between 0 and 65535, inclusive. CO is active (High) when the sum exceeds the bounds of the adder.

OFL is ignored in unsigned binary operation.

Twos-Complement Operation

For twos-complement operation, ADD16 can represent numbers between -32768 and +32767, inclusive. OFL is active (High) when the sum exceeds the bounds of the adder.

CO is ignored in twos-complement operation.

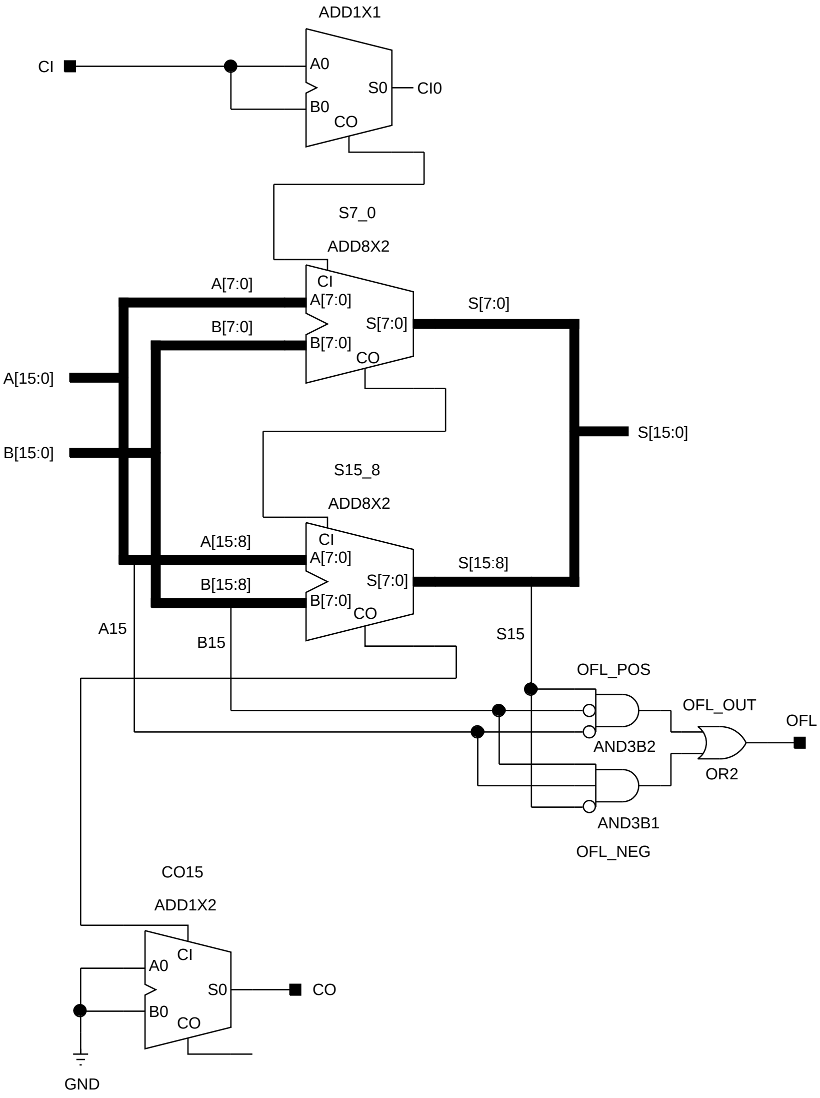
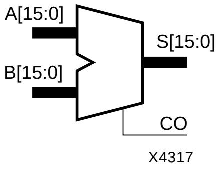


Figure 3-19 ADD16 Implementation (XC7000 and XC9000)

ADD16X1

16-Bit Cascadable Full Adder with Carry-Out for XC7000



XC7000	XC9000
Macro*	NA

* not supported for XC7336 designs

ADD16X1 is implemented using the XC7000 arithmetic carry-logic chain for high-speed ripple-carry addition. ADD16X1 is a low-order adder component, which can be used as a stand-alone or cascaded with high-order adders through its CO output. ADD16X2 adds two words (A15 – A0 and B15 – B0), producing a sum output (S15 – S0) and carry-out (CO).

The CO output is passed into the XC7000 carry chain, and therefore can only be connected to the CI input of another XC7000-specific arithmetic component. To generate a carry-out for general-purpose logic, use an adder (or cascaded adders) with one extra bit and tie the most-significant A and B inputs to GND; the most-significant S output becomes the carry-out. If a carry-in is required from general-purpose logic, extend the length of the adder by one additional bit and connect the carry-in signal to both the least-significant A and B inputs (the least-significant S output is not used) to generate a carry into the carry chain for the second bit of the adder.

Refer to “ADD1” for truth table derivation.

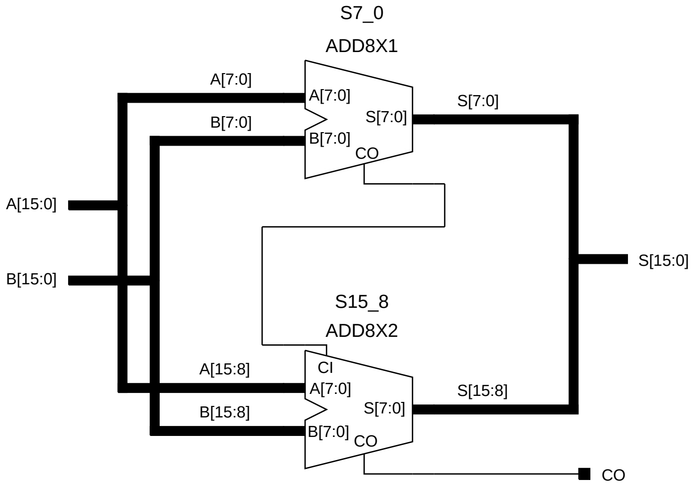
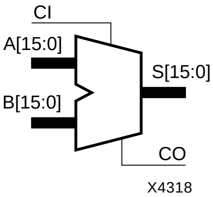


Figure 3-20 ADD16X1 Implementation (XC7000)

ADD16X2

16-Bit Cascadable Full Adder with Carry-In and Carry-Out for XC7000



XC7000	XC9000
Macro*	NA

* not supported for XC7336 designs

ADD16X2 is implemented using the XC7000 arithmetic carry-logic chain for high-speed ripple-carry addition. ADD16X2 is a high-order adder component cascaded to lower-order adders through its CI input. ADD16X2 adds two words (A15 – A0 and B15 – B0) and a carry-in (CI), producing a sum output (S15 – S0) and carry-out (CO).

The CI input is taken from the XC7000 carry chain, and therefore must only be connected to the CO output of another XC7000-specific arithmetic component. The CO output is passed into the XC7000 carry chain, and therefore can only be connected to the CI input of another XC7000-specific arithmetic component. To generate a carry-out for general-purpose logic, use an adder (or cascaded adders) with one extra bit and tie the most-significant A and B inputs to GND; the most-significant S output becomes the carry-out. Refer to “ADD1” for truth table derivation.

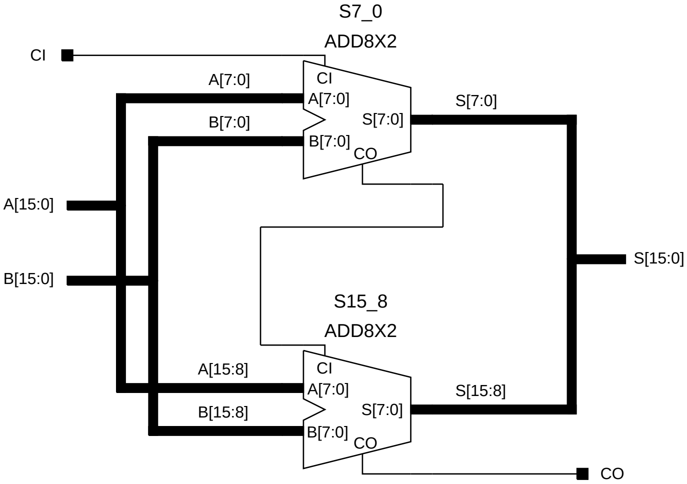
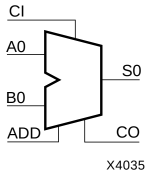


Figure 3-21 ADD16X2 Implementation (XC7000)

ADSU1

1-Bit Cascadable Adder/Subtracter with Carry-In and Carry-Out



XC7000	XC9000
Macro*	Macro

* not supported for XC7336 designs

When the ADD input is High, two 1-bit words (A0 and B0) are added with a carry-in (CI), producing a 1-bit output (S0) and a carry-out (CO). When the ADD input is Low, B0 is subtracted from A0, producing a result (S0) and borrow (CO). In add mode, CO represents a carry-out, and CO and CI are active-High. In subtract mode, CO represents a borrow, and CO and CI are active-Low. Refer to "ADSU1X1" and "ADSU1X2" for cascadable XC7000 symbols.

Inputs			Outputs	
A0	B0	CI	S0	CO
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

Add Function, ADD=1

Inputs			Outputs	
A0	**B0**	**CI**	**S0**	**CO**
0	0	0	1	0
0	1	0	0	0
1	0	0	0	1
1	1	0	1	0
0	0	1	0	1
0	1	1	1	0
1	0	1	1	1
1	1	1	0	1

Subtract Function, ADD=0

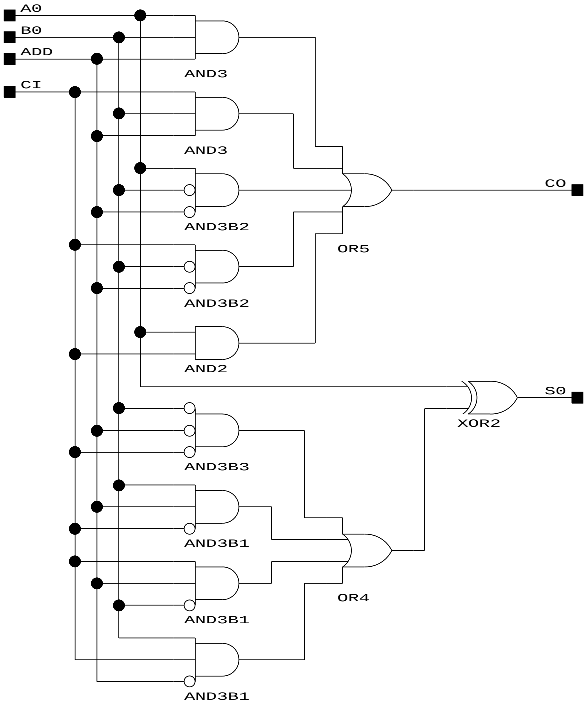
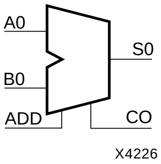


Figure 3-22 ADSU1 Implementation (XC7000 and XC9000)

ADSU1X1

1-Bit Cascadable Adder/Subtracter with Carry-Out for XC7000



XC7000	XC9000
Macro*	NA

* not supported for XC7336 designs

ADSU1X1 is implemented using the XC7000 arithmetic carry-logic chain for high-speed ripple-carry addition. ADSU1X1 is a low-order adder component, which can be used as a stand-alone or cascaded with high-order adders through its CO output. When the ADD input is High, two 1-bit words (A0 and B0) are added, producing and a 1-bit output (S0) and carry-out (CO). When the ADD input is Low, B0 is subtracted from A0, producing a result (S0) and borrow (CO). In add mode, CO represents a carry-out and is active-High. In subtract mode, CO represents a borrow and is active-Low.

The CO output is passed into the XC7000 carry chain, and therefore can only be connected to the CI input of another XC7000-specific arithmetic component. To generate a carry-out for general-purpose logic, connect an ADD1X2 to the CO output of the adder / subtracter and tie its A and B inputs to GND; the S output becomes the carry-out. If a carry-in is required from general-purpose logic, use an ADSU1X2 for the least-significant adder / subtracter and connect an ADD1X1 to its CI input. Connect your carry-in signal to both the A and B inputs of the ADD1X1 (the S output is not used) to generate a carry into the carry chain for the first bit of the adder / subtracter.

Refer to “ADSU1” for truth table derivation.

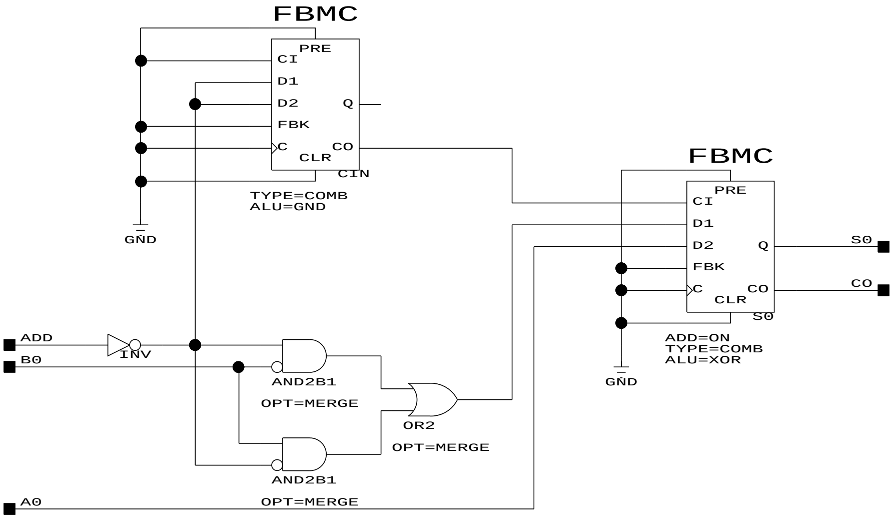
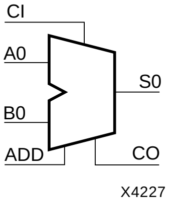


Figure 3-23   ADSU1X1 Implementation (XC7000)

ADSU1X2

1-Bit Cascadable Adder/Subtracter with Carry-In and Carry-Out for XC7000



XC7000	XC9000
Macro*	NA

* not supported for XC7336 designs

ADSU1X2 is implemented using the XC7000 arithmetic carry-logic chain for high-speed ripple-carry addition. ADSU1X2 is a high-order adder component cascaded to lower-order adders through its CI input. When the ADD input is High, two 1-bit words (A0 and B0) are added with a carry-in (CI), producing a 1-bit output (S0) and carry-out (CO). When the ADD input is Low, B0 is subtracted from A0, producing a result (S0) and borrow (CO). In add mode, CO represents a carry-out, and CO and CI are active-High. In subtract mode, CO represents a borrow, and CO and CI are active-Low.

The CI input is taken from the XC7000 carry chain, and therefore must only be connected to the CO output of another XC7000-specific arithmetic component. The CO output is passed into the XC7000 carry chain, and therefore can only be connected to the CI input of another XC7000-specific arithmetic component. To generate a carry-out signal for general-purpose logic, connect an ADD1X2 to the CO output of the adder/subtracter and tie its A and B inputs to GND; the S output becomes the carry-out.

Refer to “ADSU1” for truth table derivation.

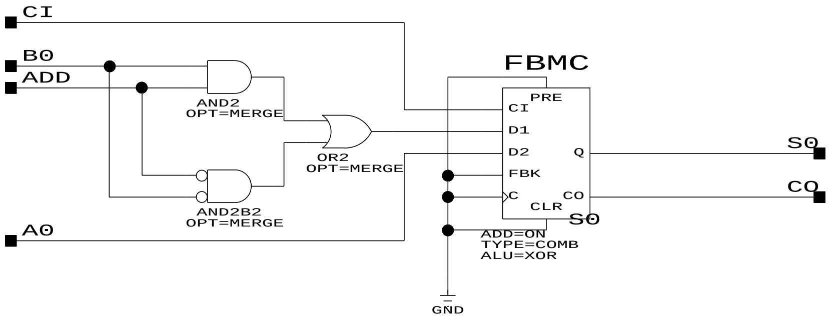
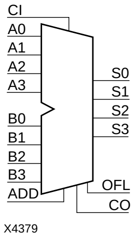


Figure 3-24 ADSU1X2 Implementation (XC7000)

ADSU4

4-Bit Cascadable Adder/Subtracter with Carry-In, Carry-Out, and Overflow



XC7000	XC9000
Macro*	Macro

* not supported for XC7336 designs

When the ADD input is High, two 4-bit words (A3 – A0 and B3 – B0) are added with a carry-in (CI), producing a 4-bit sum (S3 – S0) and carry-out (CO) or overflow (OFL). When the ADD input is Low, B3 – B0 is subtracted from A3 – A0, producing a 4-bit difference (S3 – S0) and CO or OFL. In add mode, CO and CI are active-High. In subtract mode, CO and CI are active-Low. For cascadable XC7000 symbols, refer to "ADSU4X1" and "ADSU4X2." ADSU4 CI and CO pins do not use the XC7000 carry chain.

Unsigned Binary Versus Twos-Complement

ADSU4 can operate on either 4-bit unsigned binary numbers or 4-bit twos-complement numbers. If the inputs are interpreted as unsigned binary, the result can be interpreted as unsigned binary. If the inputs are interpreted as twos complement, the output can be interpreted as twos complement. The only functional difference between an unsigned binary operation and a twos-complement operation is how they determine when "overflow" occurs. Unsigned binary uses CO, while twos-complement uses OFL to determine when "overflow" occurs.

With adder/subtracters, either unsigned binary or twos-complement operations cause an overflow. If the result crosses the overflow boundary, an overflow is generated. Similarly, when the result crosses the carry-out boundary, a carry-out is generated. The following figure shows the ADSU carry-out and overflow boundaries.

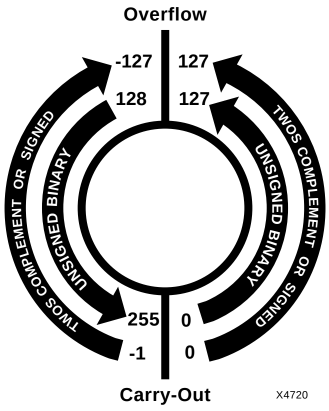


Figure 3-25 ADSU Carry-Out and Overflow Boundaries

Unsigned Binary Operation

For unsigned binary operation, ADSU4 can represent numbers between 0 and 15, inclusive. In add mode, CO is active (High) when the sum exceeds the bounds of the adder/subtracter. In subtract mode, CO is an active-Low borrow-out and goes Low when the difference exceeds the bounds.

An unsigned binary “overflow” that is always active-High can be generated by gating the ADD signal and CO as follows.

```
unsigned overflow = CO XOR ADD
```

OFL is ignored in unsigned binary operation.

Twos-Complement Operation

For twos-complement operation, ADSU4 can represent numbers between -8 and +7, inclusive. If an addition or subtraction operation result exceeds this range, the OFL output goes High.

CO is ignored in twos-complement operation.

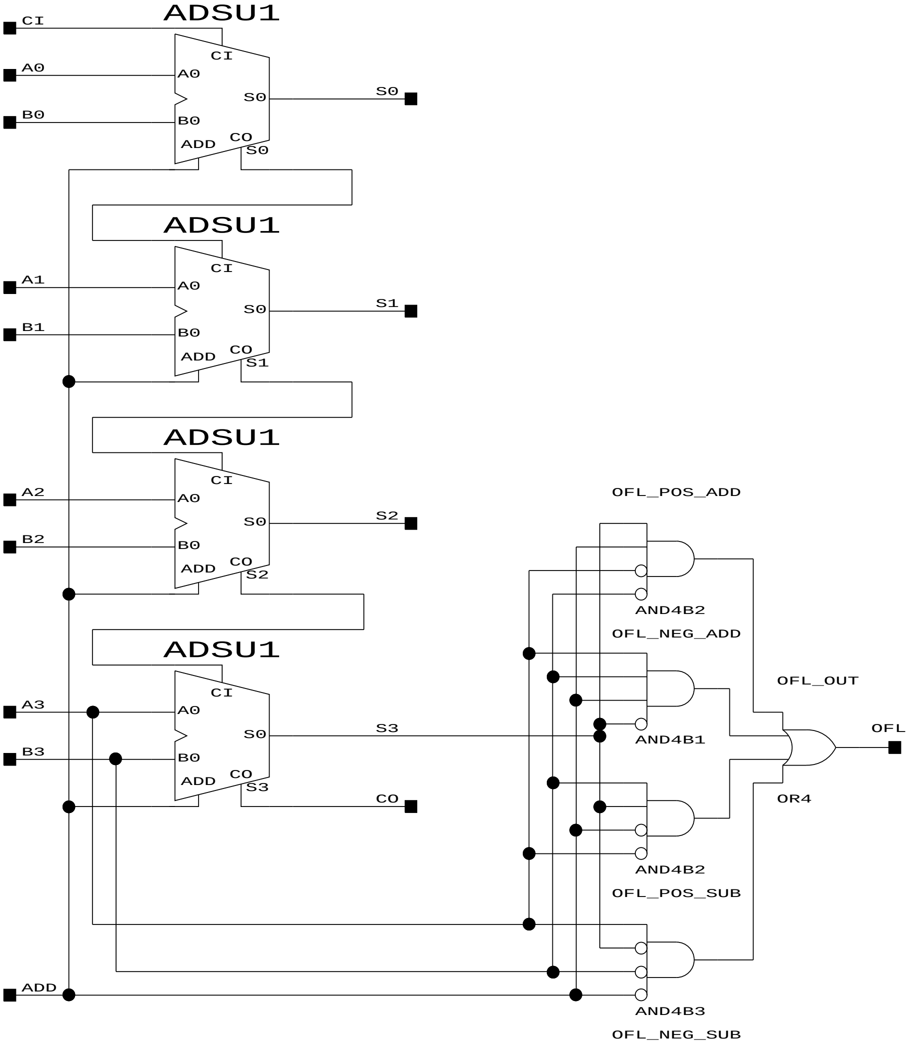
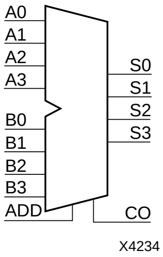


Figure 3-26 ADSU4 Implementation (XC9000)

ADSU4X1

4-Bit Cascadable Adder/Subtracter with Carry-Out for XC7000



XC7000	XC9000
Macro*	NA

* not supported for XC7336 designs

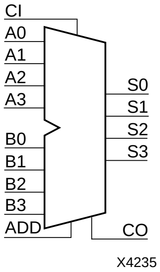
ADSU4X1 is implemented using the XC7000 arithmetic carry-logic chain for high-speed ripple-carry addition. ADSU4X1 is a low-order adder component, which can be used as a stand-alone or cascaded with high-order adders through its CO output. When the ADD input is High, two 4-bit words (A3 – A0 and B3 – B0) are added, producing a 4-bit output (S3 – S0) and carry-out (CO). When the ADD input is Low, B3 – B0 is subtracted from A3 – A0, producing a result (S3 – S0) and borrow (CO). In add mode, CO represents a carry-out and is active-High. In subtract mode, CO represents a borrow and is active-Low.

The CO output is passed into the XC7000 carry chain, and therefore can only be connected to the CI input of another XC7000-specific arithmetic component. To generate a carry-out for general-purpose logic, connect an ADD1X2 to the CO output of the adder/subtracter and tie its A and B inputs to GND; the S output becomes the carry-out. If a carry-in is required from general-purpose logic, use an ADSU4X2 for the least-significant adder/subtracter and connect an ADD1X1 to its CI input. Connect your carry-in signal to both the A and B inputs of the ADD1X1 (the S output is not used) to generate a carry into the carry chain for the first bit of the adder/subtracter.

Refer to “ADSU1” for truth table derivation.

ADSU4X2

4-Bit Cascadable Adder/Subtracter with Carry-In and Carry-Out for XC7000



XC7000	XC9000
Macro*	NA

* not supported for XC7336 designs

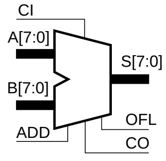
ADSU4X2 is implemented using the XC7000 arithmetic carry-logic chain for high-speed ripple-carry addition. ADSU4X2 is a high-order adder component cascaded to lower-order adders through its CI input. When the ADD input is High, two 4-bit words (A3 – A0 and B3 – B0) are added with a carry-in (CI), producing a 4-bit output (S3 – S0) and carry-out (CO). When the ADD input is Low, B3 – B0 is subtracted from A3 – A0, producing a result (S3 – S0) and borrow (CO). In add mode, CO represents a carry-out, and CO and CI are active-High. In subtract mode, CO represents a borrow, and CO and CI are active-Low.

The CI input is taken from the XC7000 carry chain, and therefore must only be connected to the CO output of another XC7000-specific arithmetic component. The CO output is passed into the XC7000 carry chain, and therefore can only be connected to the CI input of another XC7000-specific arithmetic component. To generate a carry-out signal for general-purpose logic, connect an ADD1X2 to the CO output of the adder / subtracter and tie its A and B inputs to GND; the S output becomes the carry-out.

Refer to “ADSU1” for truth table derivation.

ADSU8

8-Bit Cascadable Adder/Subtracter with Carry-In, Carry-Out, and Overflow



X4380

XC7000	XC9000
Macro*	Macro

* not supported for XC7336 designs

When the ADD input is High, two 8-bit words (A7 – A0 and B7 – B0) are added with a carry-in (CI), producing an 8-bit sum (S7 – S0) and carry-out (CO) or overflow (OFL). When the ADD input is Low, B7 – B0 is subtracted from A7 – A0, producing an 8-bit difference (S7 – S0) and CO or OFL. In add mode, CO and CI are active-High. In subtract mode, CO and CI are active-Low. OFL is active-High in add and subtract modes. For cascadable XC7000 symbols, refer to “ADSU8X1” and “ADSU8X2.” ADSU8 CI and CO pins do not use the XC7000 carry chain.

Unsigned Binary Versus Twos-Complement

ADSU8 can operate on either 8-bit unsigned binary numbers or 8-bit twos-complement numbers. If the inputs are interpreted as unsigned binary, the result can be interpreted as unsigned binary. If the inputs are interpreted as twos complement, the output can be interpreted as twos complement. The only functional difference between an unsigned binary operation and a twos-complement operation is how they determine when “overflow” occurs. Unsigned binary uses CO, while twos-complement uses OFL to determine when “overflow” occurs.

With adder/subtracters, either unsigned binary or twos-complement operations cause an overflow. If the result crosses the overflow boundary, an overflow is generated. Similarly, when the result crosses the carry-out boundary, a carry-out is generated. The following figure shows the ADSU carry-out and overflow boundaries.

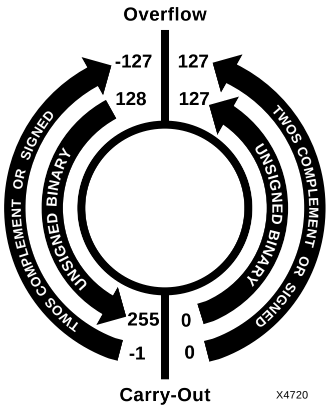


Figure 3-27 ADSU Carry-Out and Overflow Boundaries

Unsigned Binary Operation

For unsigned binary operation, ADSU8 can represent numbers between 0 and 255, inclusive. In add mode, CO is active (High) when the sum exceeds the bounds of the adder/subtracter. In subtract mode, CO is an active-Low borrow-out and goes Low when the difference exceeds the bounds.

An unsigned binary “overflow” that is always active-High can be generated by gating the ADD signal and CO as follows.

```
   unsigned overflow = CO XOR ADD
```

OFL is ignored in unsigned binary operation.

Twos-Complement Operation

For twos-complement operation, ADSU8 can represent numbers between -128 and +127, inclusive. If an addition or subtraction operation result exceeds this range, the OFL output goes High.

CO is ignored in twos complement operation.

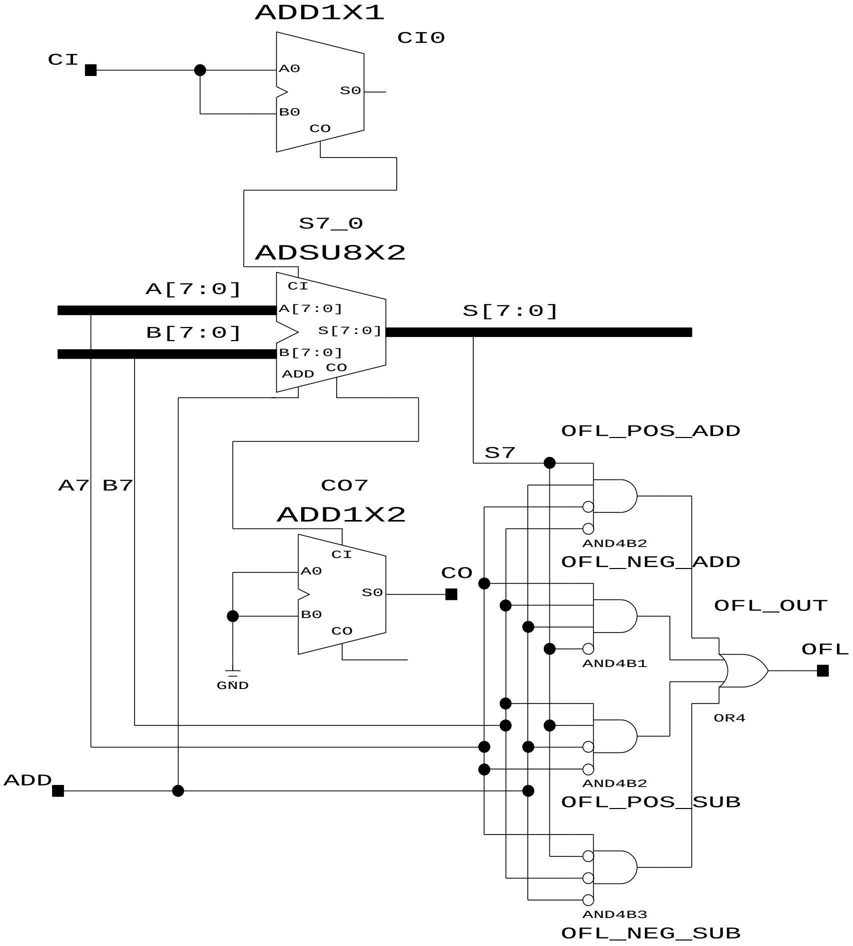


Figure 3-28 ADSU8 Implementation (XC7000)

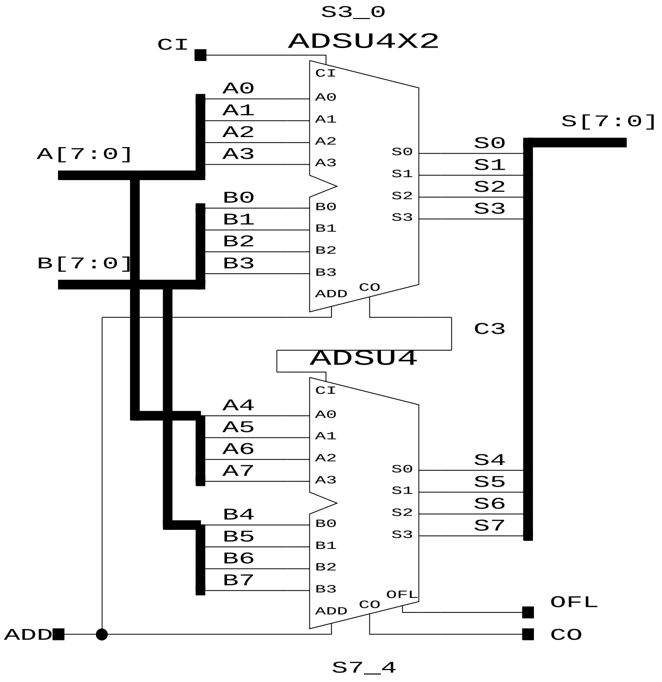
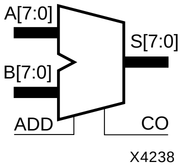


Figure 3-29 ADSU8 Implementation (XC9000)

Note: The ADSU4X2 schematic (not shown for XC9000) is the same as ADSU4 without the OFL output logic.

ADSU8X1

8-Bit Cascadable Adder/Subtracter with Carry-Out for XC7000



XC7000	XC9000
Macro*	NA

* not supported for XC7336 designs

ADSU8X1 is implemented using the XC7000 arithmetic carry-logic chain for high-speed ripple-carry addition. ADSU8X1 is a low-order adder component, which can be used as a stand-alone or cascaded with high-order adders through its CO output. When the ADD input is High, two 8-bit words (A7 – A0 and B7 – B0) are added, producing an 8-bit output (S7 – S0) and carry-out (CO). When the ADD input is Low, B7 – B0 is subtracted from A7 – A0, producing a result (S7 – S0) and borrow (CO). In add mode, CO represents a carry-out and is active-High. In subtract mode, CO represents a borrow and is active-Low.

The CO output is passed into the XC7000 carry chain, and therefore can only be connected to the CI input of another XC7000-specific arithmetic component. To generate a carry-out for general-purpose logic, connect an ADD1X2 to the CO output of the adder / subtracter and tie its A and B inputs to GND; the S output becomes the carry-out. If a carry-in is required from general-purpose logic, use an ADSU8X2 for the least-significant adder / subtracter and connect an ADD1X1 to its CI input. Connect your carry-in signal to both the A and B inputs of the ADD1X1 (the S output is not used) to generate a carry into the carry chain for the first bit of the adder / subtracter.

Refer to “ADSU1” for truth table derivation.

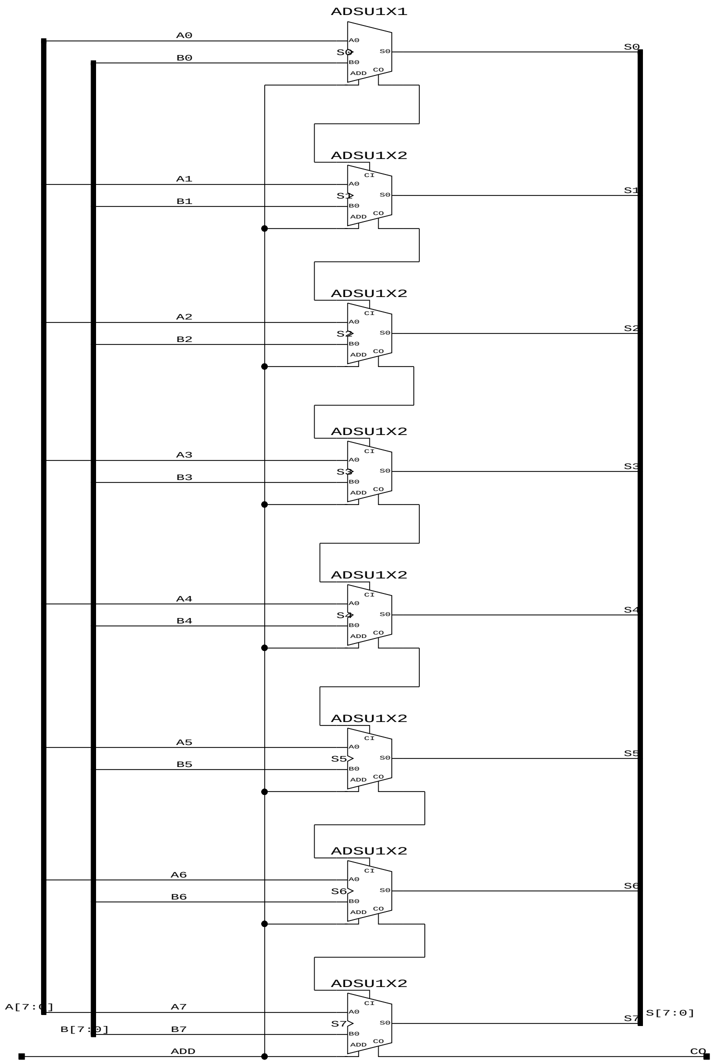
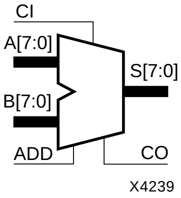


Figure 3-30 ADSU8X1 Implementation (XC7000)

ADSU8X2

8-Bit Cascadable Adder/Subtracter with Carry-In and Carry-Out for XC7000



XC7000	XC9000
Macro*	NA

* not supported for XC7336 designs

ADSU8X2 is implemented using the XC7000 arithmetic carry-logic chain for high-speed ripple-carry addition. ADSU8X2 is a high-order adder component cascaded to lower-order adders through its CI input. When the ADD input is High, two 8-bit words (A7 – A0 and B7 – B0) are added with a carry-in (CI), producing an 8-bit output (S7 – S0) and carry-out (CO). When the ADD input is Low, B7 – B0 is subtracted from A7 – A0, producing a result (S7 – S0) and borrow (CO). In add mode, CO represents a carry-out, and CO and CI are active-High. In subtract mode, CO represents a borrow, and CO and CI are active-Low.

The CI input is taken from the XC7000 carry chain, and therefore must only be connected to the CO output of another XC7000-specific arithmetic component. The CO output is passed into the XC7000 carry chain, and therefore can only be connected to the CI input of another XC7000-specific arithmetic component. To generate a carry-out signal for general-purpose logic, connect an ADD1X2 to the CO output of the adder / subtracter and tie its A and B inputs to GND; the S output becomes the carry-out.

Refer to “ADSU1” for truth table derivation.

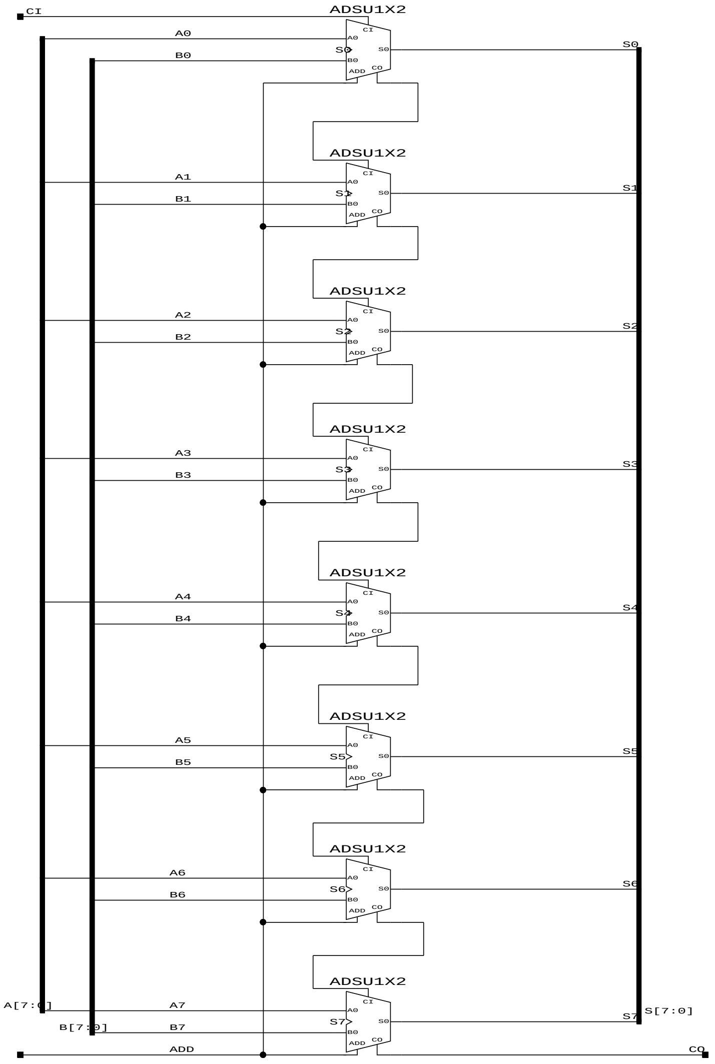
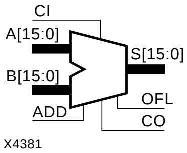


Figure 3-31 ADSU8X2 Implementation (XC7000)

ADSU16

16-Bit Cascadable Adder/Subtracter with Carry-In, Carry-Out, and Overflow



XC7000	XC9000
Macro*	Macro

* not supported for XC7336 designs

When the ADD input is High, two 16-bit words (A15 – A0 and B15 – B0) are added with a carry-in (CI), producing a 16-bit sum (S15 – S0) and carry-out (CO) or overflow (OFL). When the ADD input is Low, B15 – B0 is subtracted from A15 – A0, producing a 16-bit difference (S15 – S0) and CO or OFL. In add mode, CO and CI are active-High. In subtract mode, CO and CI are active-Low. OFL is active-High in add and subtract modes. For cascadable XC7000 symbols, refer to “ADSU16X1” and “ADSU16X2.” ADSU16 CI and CO pins do not use the XC7000 carry chain.

Unsigned Binary Versus Twos-Complement

ADSU16 can operate on either 16-bit unsigned binary numbers or 16-bit twos-complement numbers. If the inputs are interpreted as unsigned binary, the result can be interpreted as unsigned binary. If the inputs are interpreted as twos complement, the output can be interpreted as twos complement. The only functional difference between an unsigned binary operation and a twos-complement operation is how they determine when “overflow” occurs. Unsigned binary uses CO, while twos-complement uses OFL to determine when “overflow” occurs.

With adder/subtracters, either unsigned binary or twos-complement operations cause an overflow. If the result crosses the overflow boundary, an overflow is generated. Similarly, when the result crosses the carry-out boundary, a carry-out is generated. The following figure shows the ADSU carry-out and overflow boundaries.

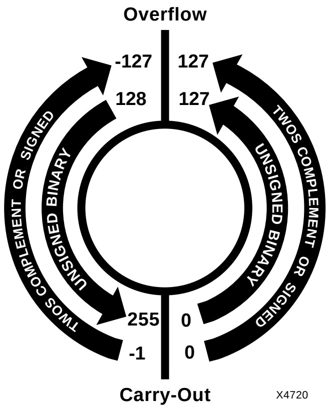


Figure 3-32 ADSU Carry-Out and Overflow Boundaries

Unsigned Binary Operation

For unsigned binary operation, ADSU16 can represent numbers between 0 and 65535, inclusive. In add mode, CO is active (High) when the sum exceeds the bounds of the adder/subtracter. In subtract mode, CO is an active-Low borrow-out and goes Low when the difference exceeds the bounds.

An unsigned binary “overflow” that is always active-High can be generated by gating the ADD signal and CO as follows.

```
   unsigned overflow = CO XOR ADD
```

OFL is ignored in unsigned binary operation.

Twos-Complement Operation

For twos-complement operation, ADSU16 can represent numbers between -32768 and +32767, inclusive. If an addition or subtraction operation result exceeds this range, the OFL output goes High.

CO is ignored in twos-complement operation.

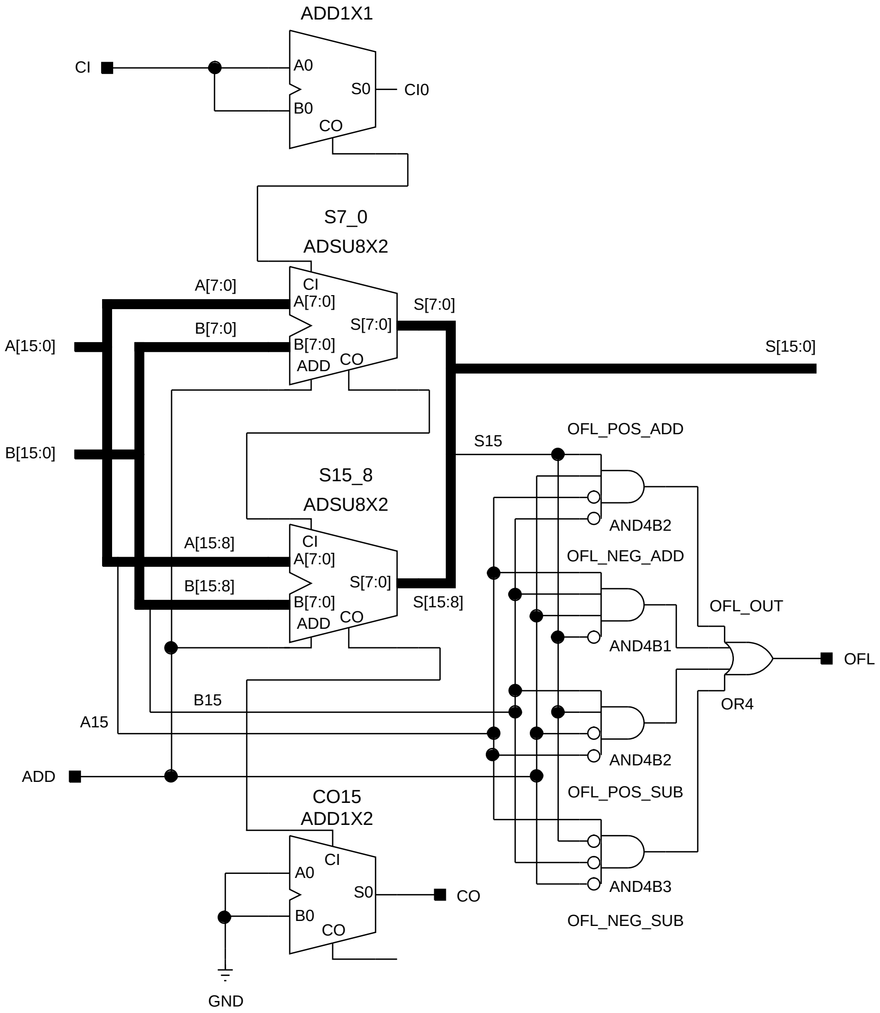
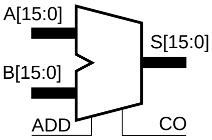


Figure 3-33 ADSU16 Implementation (XC7000)

ADSU16X1

16-Bit Cascadable Adder/Subtracter with Carry-Out for XC7000



X4319

XC7000	XC9000
Macro*	NA

* not supported for XC7336 designs

ADSU16X1 is implemented using the XC7000 arithmetic carry-logic chain for high-speed ripple-carry addition. ADSU16X1 is a low-order adder component, which can be used as a stand-alone or cascaded with high-order adders through its CO output. When the ADD input is High, two 16-bit words (A15 – A0 and B15 – B0) are added, producing a 16-bit output (S15 – S0) and carry-out (CO). When the ADD input is Low, B15 – B0 is subtracted from A15 – A0, producing a result (S15 – S0) and borrow (CO). In add mode, CO represents a carry-out and is active-High. In subtract mode, CO represents a borrow and is active-Low.

The CO output is passed into the XC7000 carry chain, and therefore can only be connected to the CI input of another XC7000-specific arithmetic component. To generate a carry-out for general-purpose logic, connect an ADD1X2 to the CO output of the adder / subtracter and tie its A and B inputs to GND; the S output becomes the carry-out. If a carry-in is required from general-purpose logic, use an ADSU16X2 for the least-significant adder / subtracter and connect an ADD1X1 to its CI input. Connect your carry-in signal to both the A and B inputs of the ADD1X1 (the S output is not used) to generate a carry into the carry chain for the first bit of the adder / subtracter.

Refer to “ADSU1” for truth table derivation.

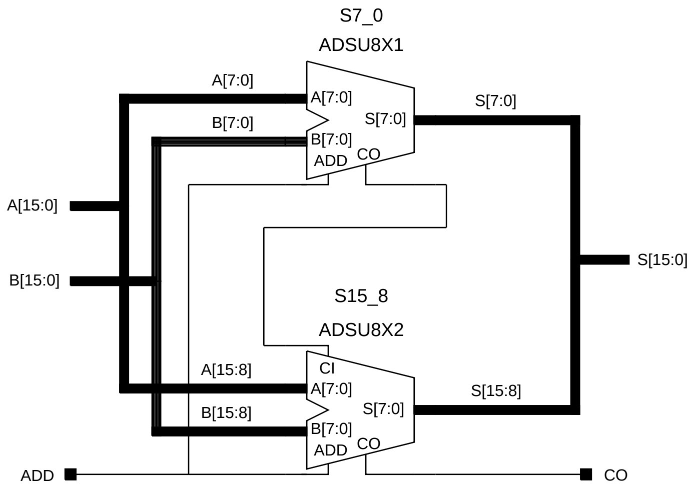
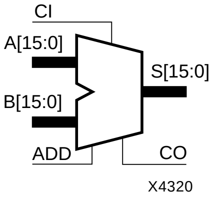


Figure 3-34 ADSU16X1 Implementation (XC7000)

ADSU16X2

16-Bit Cascadable Adder/Subtracter with Carry-In and Carry-Out for XC7000



XC7000	XC9000
Macro*	NA

* not supported for XC7336 designs

ADSU16X2 is implemented using the XC7000 arithmetic carry-logic chain for high-speed ripple-carry addition. ADSU16X2 is a high-order adder component cascaded to lower-order adders through its CI input. When the ADD input is High, two 16-bit words (A15 – A0 and B15 – B0) are added with a carry-in (CI), producing a 16-bit output (S15 – S0) and carry-out (CO). When the ADD input is Low, B15 – B0 is subtracted from A15 – A0, producing a result (S15 – S0) and borrow (CO). In add mode, CO represents a carry-out, and CO and CI are active-High. In subtract mode, CO represents a borrow, and CO and CI are active-Low.

The CI input is taken from the XC7000 carry chain, and therefore must only be connected to the CO output of another XC7000-specific arithmetic component. The CO output is passed into the XC7000 carry chain, and therefore can only be connected to the CI input of another XC7000-specific arithmetic component. To generate a carry-out signal for general-purpose logic, connect an ADD1X2 to the CO output of the adder / subtracter and tie its A and B inputs to GND; the S output becomes the carry-out.

Refer to “ADSU1” for truth table derivation.

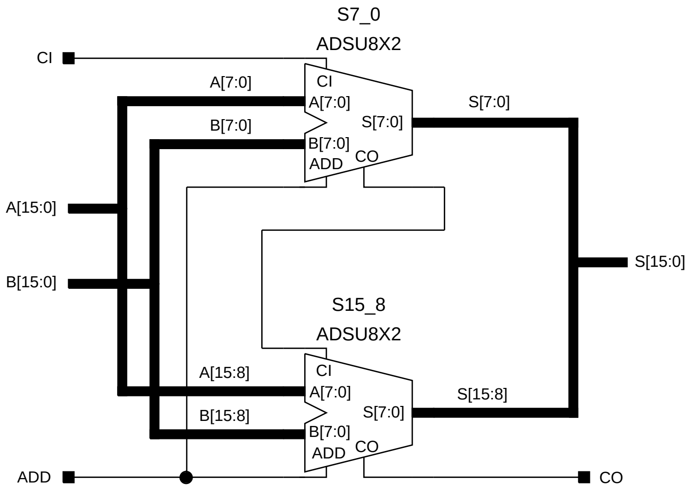


Figure 3-35 ADSU16X2 Implementation (XC7000)

AND

2- to 9-Input AND Gates with Inverted and Non-Inverted Inputs

Name	XC7000	XC9000
AND2 – AND4B4	Primitive	Primitive
AND5 – AND5B5	Primitive	Primitive
AND6, AND7, AND8, AND9	Primitive	Primitive

AND functions of up to five inputs are available in any combination of inverting and non-inverting inputs. AND functions of six to nine inputs are available with only non-inverting inputs. To make some or all inputs inverting, use external inverters. Available AND gates are shown in the following figure.

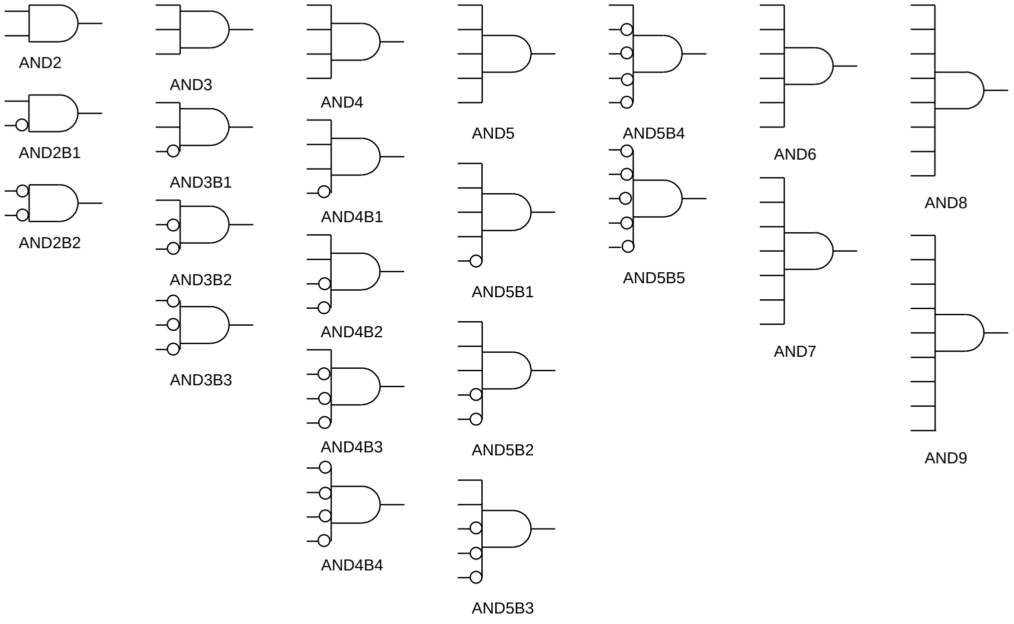
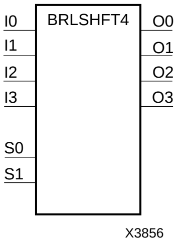


Figure 3-36 AND Gate Representations

BRLSHFT4

4-Bit Barrel Shifter



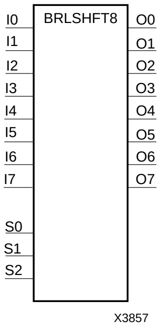
XC7000	XC9000
Macro	Macro

BRLSHFT4, a 4-bit barrel shifter, can rotate four inputs (I3 – I0) up to four places. The control inputs (S1 and S0) determine the number of positions, from one to four that the data is rotated. The four outputs (O3 – O0) reflect the shifted data inputs.

Inputs						Outputs			
S1	**S0**	**I0**	**I1**	**I2**	**I3**	**O0**	**O1**	**O2**	**O3**
0	0	a	b	c	d	a	b	c	d
0	1	a	b	c	d	b	c	d	a
1	0	a	b	c	d	c	d	a	b
1	1	a	b	c	d	d	a	b	c

BRLSHFT8

8-Bit Barrel Shifter



XC7000	XC9000
Macro	Macro

BRLSHFT8, an 8-bit barrel shifter, can rotate the eight inputs, I7 – I0, up to eight places. The control inputs (S2 – S0) determine the number of positions, from one to eight that the data is rotated. The eight outputs (O7 – O0) reflect the shifted data inputs.

Inputs											Output							
S2	**S1**	**S0**	**I0**	**I1**	**I2**	**I3**	**I4**	**I5**	**I6**	**I7**	**O0**	**O1**	**O2**	**O3**	**O4**	**O5**	**O6**	**O7**
0	0	0	a	b	c	d	e	f	g	h	a	b	c	d	e	f	g	h
0	0	1	a	b	c	d	e	f	g	h	b	c	d	e	f	g	h	a
0	1	0	a	b	c	d	e	f	g	h	c	d	e	f	g	h	a	b
0	1	1	a	b	c	d	e	f	g	h	d	e	f	g	h	a	b	c
1	0	0	a	b	c	d	e	f	g	h	e	f	g	h	a	b	c	d
1	0	1	a	b	c	d	e	f	g	h	f	g	h	a	b	c	d	e
1	1	0	a	b	c	d	e	f	g	h	g	h	a	b	c	d	e	f
1	1	1	a	b	c	d	e	f	g	h	h	a	b	c	d	e	f	g

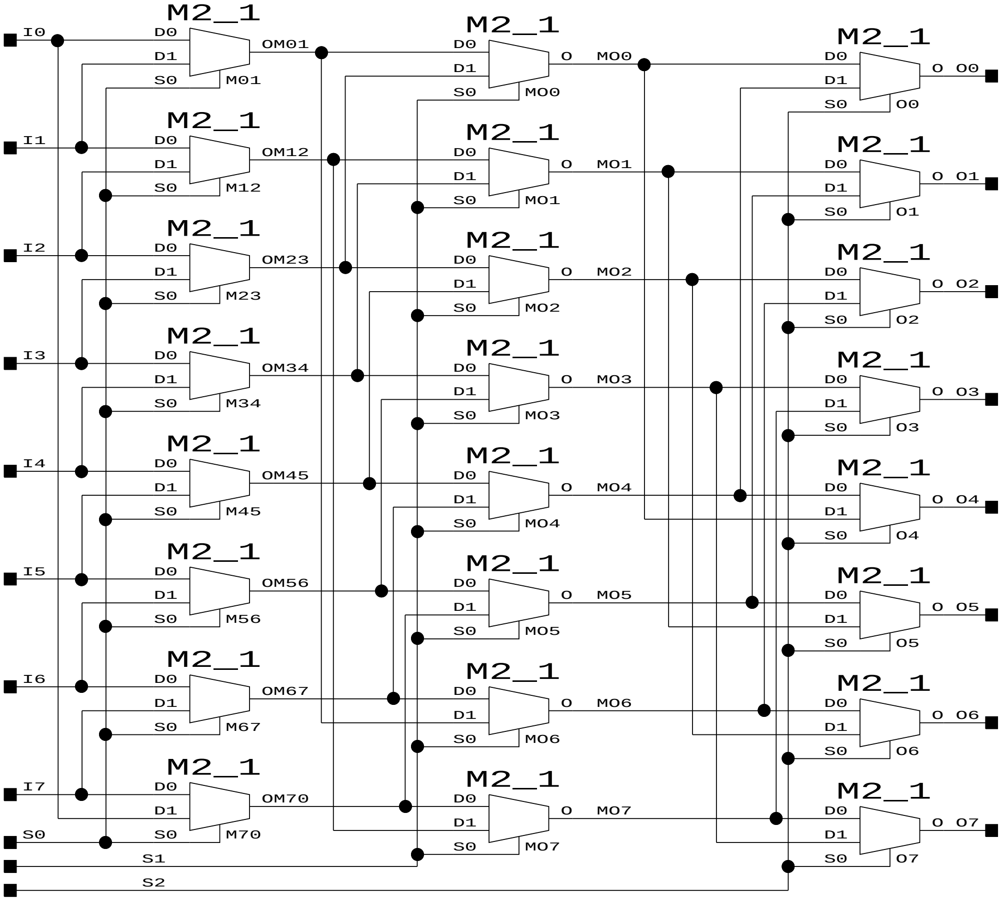
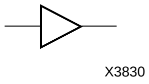


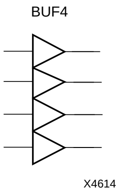
Figure 3-37 BRLSHFT8 Implementation (XC7000 and XC9000)

BUF, BUF4, BUF8, and BUF16

General-Purpose Buffers

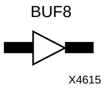


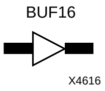
Name	XC7000	XC9000
BUF	Primitive	Primitive
BUF4, BUF8, BUF16	Macro	Macro



BUF is a general purpose, non-inverting buffer.

BUF is usually removed, unless you inhibit optimization by applying the OPT=OFF attribute to the BUF symbol.





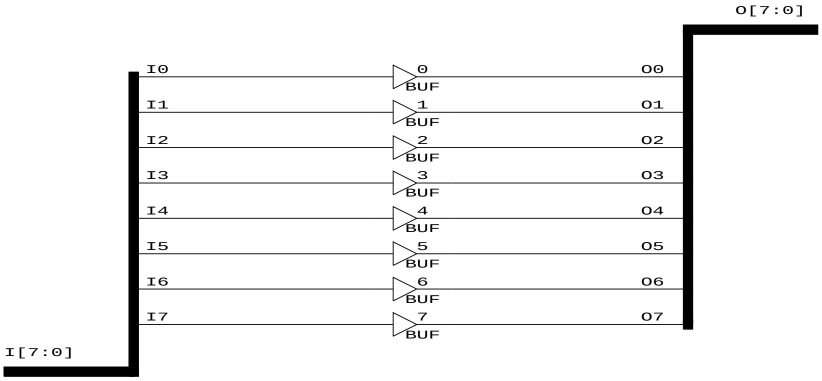
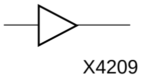


Figure 3-38 BUF8 Implementation (XC7000 and XC9000)

BUFCE

Global Clock-Enable Buffer for XC7000



XC7000	XC9000
Primitive*	NA

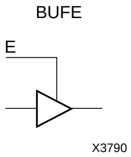
* not supported for XC7336 designs

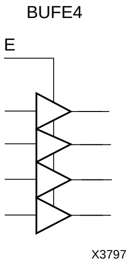
BUFCE, an XC7000-specific global buffer, distributes global clock-enable signals throughout the input-pad registers of an XC7000 device. Global clock-enable pins are available on most XC7300 series devices; consult device data sheets for applicability.

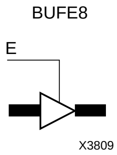
BUFCE always acts as an input buffer. To use it in a schematic, connect the input of the BUFCE symbol to an IPAD or an IOPAD that represents the clock-enable signal source. Clock-enable signals generated on-chip must be passed through an OBUF-type buffer before they are connected to a BUFCE. The output of a BUFCE can only be connected to the CE input of an XC7000-specific input-pad register symbol, IFDX1. Each BUFCE can drive any number of IFDX1 registers in a design. The CE input of IFDX1 is active-Low and cannot be inverted.

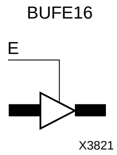
BUFE, BUFE4, BUFE8, and BUFE16

Internal 3-State Buffers









XC7000	XC9000
Macro	Macro

BUFE, BUFE4, BUFE8, and BUFE16 are single or multiple 3-state buffers with inputs I, I3 – I0, I7 – I0, and so forth; outputs O, O3 – O0, O7 – O0, and so forth; and active-High output enable (E). When E is High, data on the inputs of the buffers is transferred to the corresponding outputs. When E is Low, the output is high impedance (Z state or off).

The outputs of separate BUFE symbols can be tied together to form a bus or a multiplexer. Make sure that only one E is High at one time.

Inputs		Outputs
E	I	O
0	X	Z
1	1	1
1	0	0

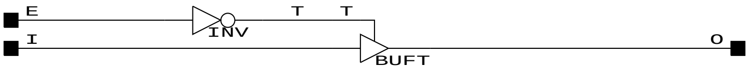


Figure 3-39 BUFE Implementation (XC7000 and XC9000)

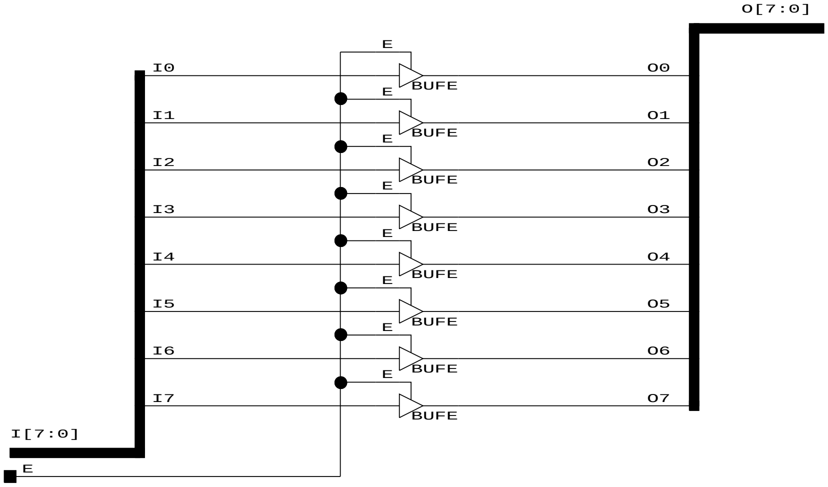
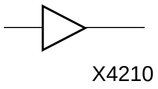


Figure 3-40 BUFE8 Implementation (XC7000 and XC9000)

BUFFOE

Global Fast Output Enable Buffer for XC7000



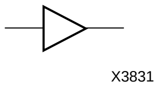
XC7000	XC9000
Primitive	NA

BUFFOE, an XC7000-specific global buffer, distributes global output-enable signals throughout the output pad drivers of an XC7000 device. Global Fast Output Enable (FOE) pins are available on XC7000 architecture devices; consult device data sheets for availability.

BUFFOE always acts as an input buffer. To use it in a schematic, connect the input of the BUFFOE symbol to an IPAD or an IOPAD representing the FOE signal source. FOE signals generated on-chip must be passed through an OBUF-type buffer before they are connected to the BUFFOE. The output of a BUFFOE can only connect to the E input of a 3-state output buffer symbol, like OBUFE. Each BUFFOE can control any number of output buffers in a design. The BUFFOE can only be used to produce an active-High output enable; it cannot be inverted.

BUFG

Global Clock Buffer



XC7000	XC9000
Primitive	Primitive

BUFG, an architecture-independent global buffer, distributes high fan-out clock signals throughout a PLD device. The Xilinx implementation software converts each BUFG to an appropriate type of global buffer for the target PLD device.

For XC7000 and XC9000 designs, consult the device date sheet for the number of available global pins.

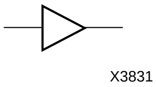
To use a BUFG in a schematic, connect the input of the BUFG symbol to the clock source. Depending on the target PLD family, the clock source can be an external PAD symbol, an IBUF symbol, or internal logic. For a negative-edge clock input, insert an INV (inverter) symbol between the BUFG output and the clock input. The inversion is implemented at the Configurable Logic Block (CLB) or Input Output Block (IOB) clock pin.

For XC7000 designs, BUFG always acts as an input buffer. Connect the input of BUFG to an IPAD or an IOPAD that represents the FastCLK signal source. FastCLK signals generated on-chip must be passed through an OBUF-type buffer before connecting to BUFG. Each BUFG can drive any number of register clocks (or ILD latch-enable inputs) in a design. However, BUFG signals cannot be used for any other logic functions in the same design. All clock inputs driven by BUFG are active-High and cannot be inverted.

For XC9000 designs, the output of a BUFG may also be used as an ordinary input signal to other logic elsewhere in the design.

BUFGSR

Global Set/Reset Input Buffer for XC9000



XC7000	XC9000
NA	Primitive

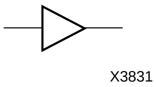
BUFGSR, an XC9000-specific global buffer, distributes global set / reset signals throughout selected flip-flops of an XC9000 device. Global Set / Reset (GSR) control pins are available on XC9000 devices; consult device data sheets for availability.

BUFGSR always acts as an input buffer. To use it in a schematic, connect the input of the BUFGSR symbol to an IPAD or an IOPAD representing the GSR signal source. GSR signals generated on-chip must be passed through an OBUF-type buffer before they are connected to BUFGSR.

For global set / reset control, the output of BUFGSR normally connects to the CLR or PRE input of a flip-flop symbol, like FDCP, or any registered symbol with asynchronous clear or preset. The global set / reset control signal may pass through an inverter to perform an active-low set / reset. The output of BUFGSR may also be used as an ordinary input signal to other logic elsewhere in the design. Each BUFGSR can control any number of flip-flops in a design.

BUFGTS

Global Three-State Input Buffer for XC9000



XC7000	XC9000
NA	Primitive

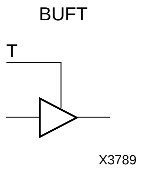
BUFGTS, an XC9000-specific global buffer, distributes global output-enable signals throughout the output pad drivers of an XC9000 device. Global Three-State (GTS) control pins are available on XC9000 devices; consult device data sheets for availability.

BUFGTS always acts as an input buffer. To use it in a schematic, connect the input of the BUFGTS symbol to an IPAD or an IOPAD representing the GTS signal source. GTS signals generated on-chip must be passed through an OBUF-type buffer before they are connected to BUFGTS.

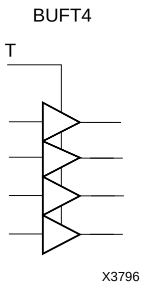
For global 3-state control, the output of BUFGTS normally connects to the E input of a 3-state output buffer symbol, OBUFE. The global 3-state control signal may pass through an inverter or control an OBUFT symbol to perform an active-low output-enable. The same 3-state control signal may even be used both inverted and non-inverted to enable alternate groups of device outputs. The output of BUFGTS may also be used as an ordinary input signal to other logic elsewhere in the design. Each BUFGTS can control any number of output buffers in a design.

BUFT, BUFT4, BUFT8, and BUFT16

Internal 3-State Buffers

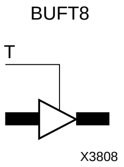


Name	XC7000	XC9000
BUFT	Primitive	Primitive
BUFT4, BUFT8, BUFT16	Macro	Macro

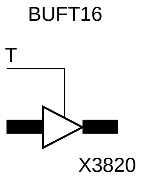


BUFT, BUFT4, BUFT8, and BUFT16 are single or multiple 3-state buffers with inputs I, I3 – I0, I7 – I0, and so forth; outputs O, O3 – O0, O7 – O0, and so forth; and active-Low output enable (T). When T is Low, data on the inputs of the buffers is transferred to the corresponding outputs. When T is High, the output is high impedance (Z state or off).

The outputs of separate BUFT symbols can be tied together to form a bus or a multiplexer. Make sure that only one T is Low at one time. Pull-up resistors can be used to establish a High logic level if all BUFT elements are off. Pull-up resistors are always assumed for CPLD designs.



Inputs		Outputs
T	I	O
1	X	Z
0	1	1
0	0	0



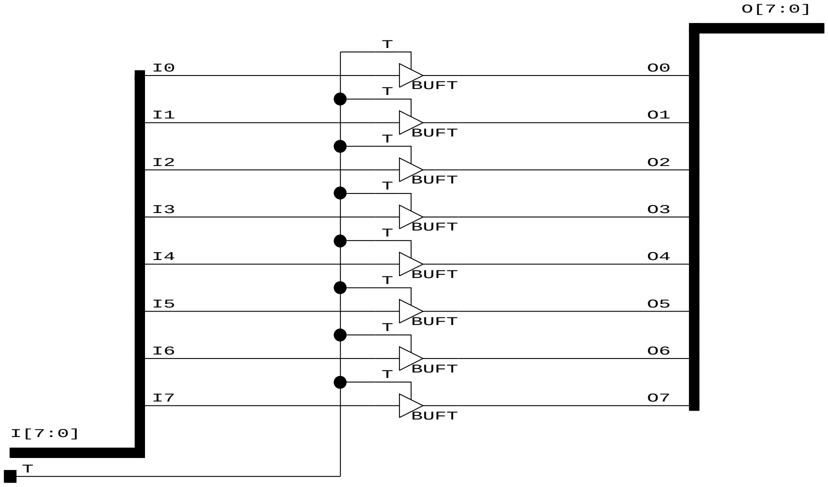
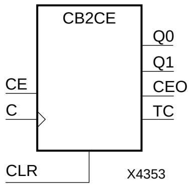


Figure 3-41 BUFT8 Implementation (XC7000 and XC9000)

CB2CE

2-Bit Cascadable Binary Counter with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

CB2CE is a 2-stage, 2-bit, synchronous, clearable, cascadable binary counter. The asynchronous clear (CLR) is the highest priority input. When CLR is High, all other inputs are ignored and data (Q1 – Q0) and terminal count (TC) outputs go to logic level zero, independent of clock transitions. The outputs (Q1 – Q0) increment when the clock enable input (CE) is High during the Low-to-High clock (C) transition. The counter ignores clock transitions when CE is Low. The TC output is High when both Q outputs are High.

Larger counters are created by connecting the count enable out (CEO) output of the first stage to the CE input of the next stage and connecting the C and CLR inputs in parallel. CEO is active (High) when TC and CE are High. The maximum length of the counter is determined by the accumulated CE-to-TC propagation delays versus the clock period. The clock period must be greater than n(tCE-TC), where “n” is the number of stages and “tCE-TC” is the CE-to-TC propagation delay of each stage.

The default initial state of all flip-flops is zero. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

Inputs			Outputs		
CLR	CE	C	Q1 – Q0	TC	CEO
1	X	X	0	0	0
0	0	X	No Chg	No Chg	0
0	1	↑	Inc	TC	CEO

TC = (Q1•Q0); CEO = (TC•CE)

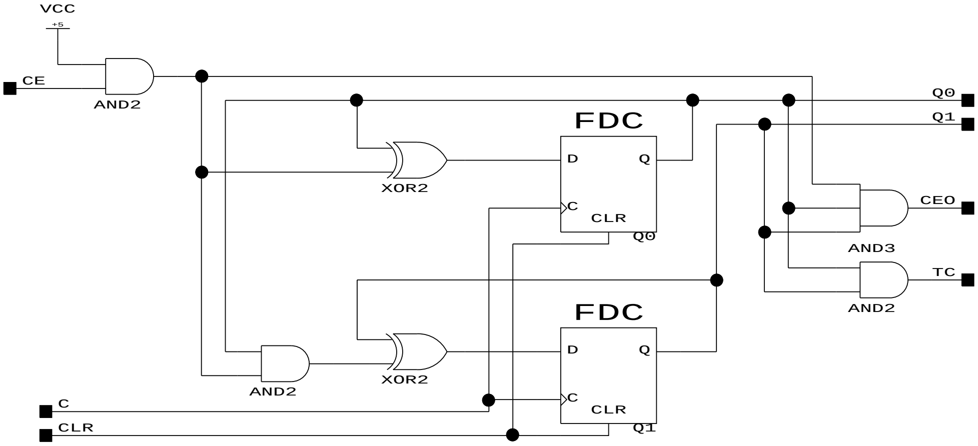
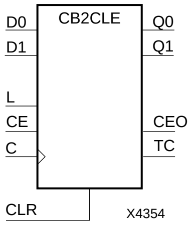


Figure 3-42   CB2CE Implementation (XC7000 and XC9000)

CB2CLE

2-Bit Loadable Cascadable Binary Counter with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

CB2CLE is a 2-stage, 2-bit, synchronous, loadable, clearable, cascadable binary counter. The asynchronous clear (CLR) is the highest priority input. When CLR is High, all other inputs are ignored and data (Q1 – Q0) and terminal count (TC) outputs go to logic level zero on the Low-to-High clock (C) transition. The data on the D1 – D0 inputs is loaded into the counter when the load enable input (L) is High during the Low-to-High clock transition, independent of the state of clock enable (CE). The outputs (Q1 – Q0) increment when CE is High during the Low-to-High clock transition. The counter ignores clock transitions when CE is Low. The TC output is High when both Q outputs are High.

Larger counters are created by connecting the count enable out (CEO) output of the first stage to the CE input of the next stage and connecting the C, L, and CLR inputs in parallel. CEO is active (High) when TC and CE are High. The maximum length of the counter is determined by the accumulated CE-to-TC propagation delays versus the clock period. The clock period must be greater than n(tCE-TC), where “n” is the number of stages and “tCE-TC” is the CE-to-TC propagation delay of each stage.

The default initial state of all flip-flops is zero. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

Inputs					Outputs		
CLR	**L**	**CE**	**C**	**D1 – D0**	**Q1 – Q0**	**TC**	**CEO**
1	X	X	X	X	0	0	0
0	1	X	↑	D	d1 – d0	TC	CEO
0	0	0	X	X	No Chg	No Chg	0
0	0	1	↑	X	Inc	TC	CEO

TC = (Q1•Q0)

CEO = (TC•CE)

dn = state of referenced input one set-up time prior to active clock transition

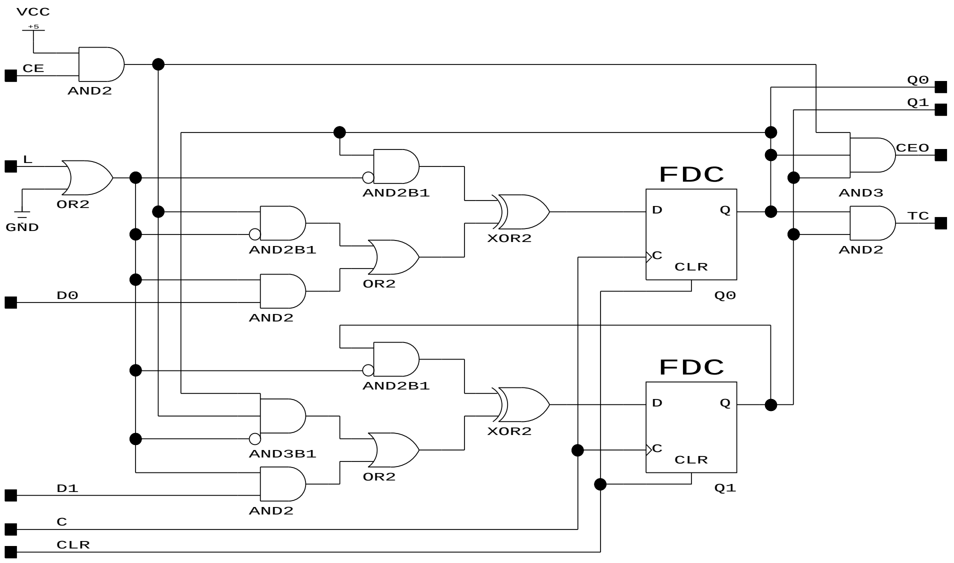
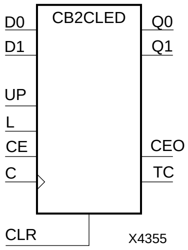


Figure 3-43 CB2CLE Implementation (XC7000 and XC9000)

CB2CLED

2-Bit Loadable Cascadable Bidirectional Binary Counter with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

CB2CLED is a 2-stage, 2-bit, synchronous, loadable, clearable, cascadable, bidirectional binary counter. The asynchronous clear (CLR) is the highest priority input. When CLR is High, all other inputs are ignored and data (Q1 – Q0) and terminal count (TC) outputs go to logic level zero, independent of clock transitions. The data on the D1 – D0 inputs is loaded into the counter when the load enable input (L) is High during the Low-to-High clock (C) transition, independent of the state of clock enable (CE). The outputs (Q1 – Q0) decrement when CE is High and UP is Low during the Low-to-High clock transition. The outputs (Q1 – Q0) increment when CE and UP are High. The counter ignores clock transitions when CE is Low.

For counting up, the TC output is High when all Q outputs and UP are High. For counting down, the TC output is High when all Q outputs and UP are Low. To cascade counters, the clock enable out (CEO) output of each counter is connected to the CE pin of the next stage. The clock, UP, L, and CLR inputs are connected in parallel. CEO is active (High) when TC and CE are High. The maximum length of the counter is determined by the accumulated CE-to-TC propagation delays versus the clock period. The clock period must be greater than n(tCE-TC), where “n” is the number of stages and “tCE-TC” is the CE-to-TC propagation delay of each stage. For XC7000 and XC9000 designs, refer to “CB2X1” for high-performance cascadable, bidirectional counters.

The default initial state of all flip-flops is zero. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

Inputs						**Outputs**		
CLR	**L**	**CE**	**C**	**UP**	**D1 – D0**	**Q1 – Q0**	**TC**	**CEO**
1	X	X	X	X	X	0	0	0
0	1	X	↑	X	D	d1 – d0	TC	CEO
0	0	0	X	X	X	No Chg	No Chg	0
0	0	1	↑	1	X	Inc	TC	CEO
0	0	1	↑	0	X	Dec	TC	CEO

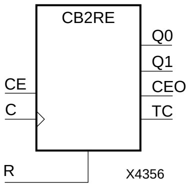
$TC = (Q1 \bullet Q0 \bullet UP) + (\overline{Q1} \bullet \overline{Q0} \bullet \overline{UP})$

$CEO = (TC \bullet CE)$

dn = state of referenced input one set-up time prior to active clock transition

CB2RE

2-Bit Cascadable Binary Counter with Clock Enable and Synchronous Reset



XC7000	XC9000
Macro	Macro

CB2RE is a 2-stage, 2-bit, synchronous, resettable, cascadable binary counter. The synchronous reset (R) is the highest priority input. When R is High, all other inputs are ignored and data (Q1 – Q0) and terminal count (TC) outputs go to logic level zero, independent of clock transitions. The outputs (Q1 – Q0) increment when the clock enable input (CE) is High during the Low-to-High clock (C) transition. The counter ignores clock transitions when CE is Low. The TC output is High when both Q outputs are High.

Larger counters are created by connecting the count enable out (CEO) output of the first stage to the CE input of the next stage and connecting the C and R inputs in parallel. CEO is active (High) when TC and CE are High. The maximum length of the counter is determined by the accumulated CE-to-TC propagation delays versus the clock period. The clock period must be greater than n(tCE-TC), where “n” is the number of stages and “tCE-TC” is the CE-to-TC propagation delay of each stage.

The default initial state of all flip-flops is zero. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

Inputs				Outputs	
R	CE	C	Q1 – Q0	TC	CEO
1	X	↑	0	0	0
0	0	X	No Chg	No Chg	0
0	1	↑	Inc	TC	CEO

TC = (Q1•Q0); CEO = (TC•CE)

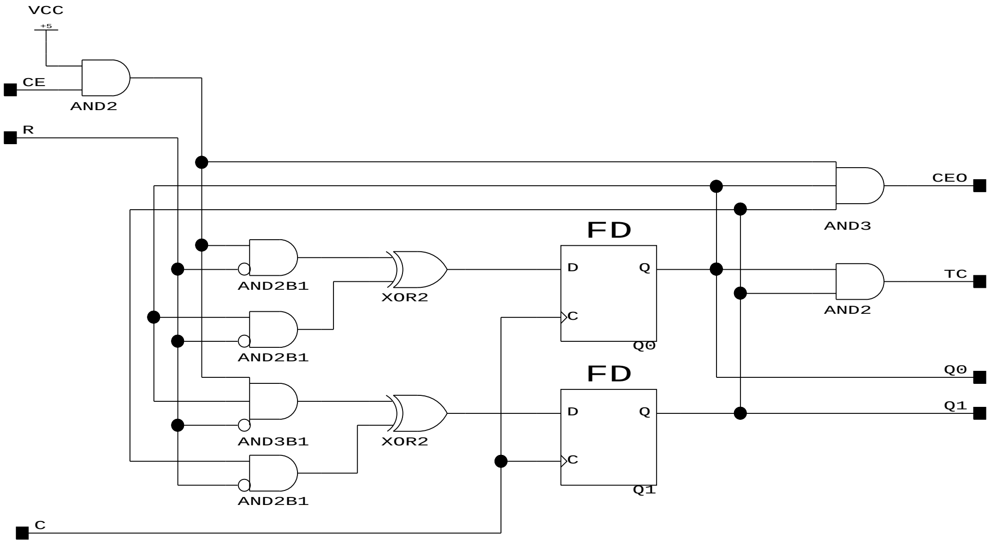
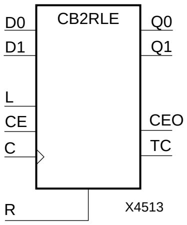


Figure 3-44 CB2RE Implementation (XC7000 and XC9000)

CB2RLE

2-Bit Loadable Cascadable Binary Counter with Clock Enable and Synchronous Reset



XC7000	XC9000
Macro	Macro

CB2RLE is a 2-stage, 2-bit, synchronous, loadable, resettable, cascadable binary counter. The synchronous reset (R) is the highest priority input. The synchronous R, when High, overrides all other inputs and resets the Q1 – Q0, terminal count (TC), and clock enable out (CEO) outputs to Low on the Low-to-High clock (C) transition.

The data on the D1 – D0 inputs is loaded into the counter when the load enable input (L) is High during the Low-to-High clock (C) transition, independent of the state of CE. The outputs (Q1 – Q0) increment when CE is High during the Low-to-High clock transition. The counter ignores clock transitions when CE is Low. The TC output is High when all Q outputs are High. The CEO output is High when all Q outputs and CE are High to allow direct cascading of counters. Larger counters are created by connecting the CEO output of the first stage to the CE input of the next stage and by connecting the C, L, and R inputs in parallel. The maximum length of the counter is determined by the accumulated CE-to-CEO propagation delays versus the clock period.

The default initial state of all flip-flops is zero. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

Inputs					Outputs		
R	**L**	**CE**	**C**	**D1 – D0**	**Q1 – Q0**	**TC**	**CEO**
1	X	X	↑	X	0	0	0
0	1	X	↑	D	d1 – d0	TC	CEO
0	0	0	X	X	No Chg	No Chg	0
0	0	1	↑	X	Inc	TC	CEO

TC = Q1•Q0

CEO = TC•CE

dn = state of referenced input one set-up time prior to active clock transition

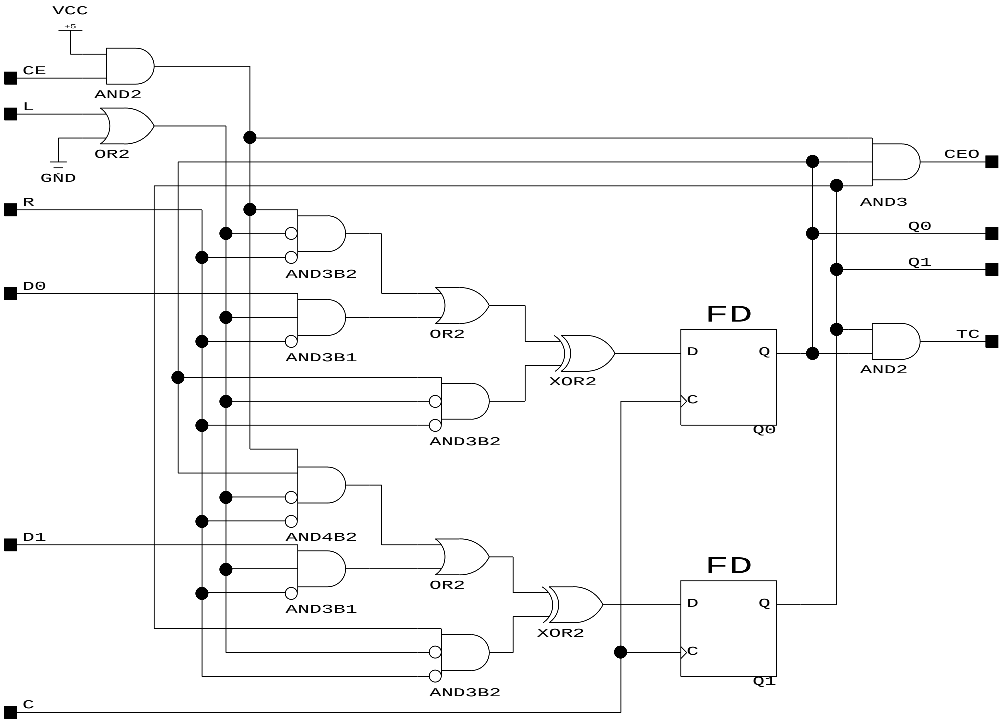
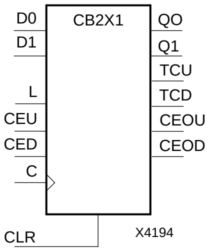


Figure 3-45 CB2RLE Implementation (XC7000 and XC9000)

CB2X1

2-Bit Loadable Cascadable Bidirectional Binary Counter with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

CB2X1 is a 2-stage, 2-bit, synchronous, loadable, clearable, bidirectional binary counter. CB2X1 has separate count-enable inputs and synchronous terminal-count outputs for up and down directions to support high-speed cascading in the XC7000 and XC9000 CPLD architectures.

The asynchronous clear (CLR) is the highest priority input. When CLR is High, all other inputs are ignored, data outputs (Q1 – Q0) go to logic level zero, and terminal count outputs TCU and TCD go to zero and one, respectively, independent of clock transitions. The data on the D1 – D0 inputs loads into the counter on the Low-to-High clock (C) transition when the load enable input (L) is High, independent of the CE inputs.

The outputs (Q1 – Q0) increment when CEU is High, provided CLR and L are Low, during the Low-to-High clock transition. The outputs (Q1 – Q0) decrement when CED is High, provided CLR and L are Low. The counter ignores clock transitions when CEU and CED are Low. Both CEU and CED should not be High during the same clock transition; the CEOU and CEOD outputs might not function properly for cascading when CEU and CED are both High.

For counting up, the CEOU output is High when all Q outputs and CEU are High. For counting down, the CEOD output is High when all Q outputs are Low and CED is High. To cascade counters, the CEOU and CEOD outputs of each counter are connected directly to the CEU and CED inputs, respectively, of the next stage. The clock, L and CLR inputs are connected in parallel.

In Xilinx CPLD devices, the maximum clocking frequency of these counter components is unaffected by the number of cascaded stages for all counting and loading functions. The TCU terminal count output is High when all Q outputs are High, regardless of CEU. The

TCD output is High when all Q outputs are Low, regardless of CED.

When cascading counters, the final terminal count signals can be produced by AND-ing all the TCU outputs (for the up direction) and all the TCD outputs (for the down direction). The default initial state of all flip-flops is zero.

Inputs						Outputs				
CLR	**L**	**CEU**	**CED**	**C**	**D1 – D0**	**Q1 – Q0**	**TCU**	**TCD**	**CEOU**	**CEOD**
1	X	X	X	X	X	0	0	1	0	CEOD
0	1	X	X	↑	D	d1 – d0	TCU	TCD	CEOU	CEOD
0	0	0	0	X	X	No Chg	No Chg	No Chg	0	0
0	0	1	0	↑	X	Inc	TCU	TCD	CEOU	0
0	0	0	1	↑	X	Dec	TCU	TCD	0	CEOD
0	0	1	1	↑	X	Inc	TCU	TCD	Invalid	Invalid

TCU = Q1•Q0

TCD = $\overline{\text{Q1}}$•$\overline{\text{Q0}}$

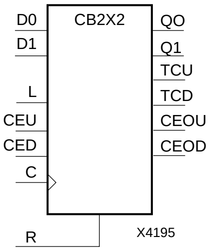
CEOU = TCU•CEU

CEOD = TCD•CED

dn = state of referenced input one set-up time prior to active clock transition

CB2X2

2-Bit Loadable Cascadable Bidirectional Binary Counter with Clock Enable and Synchronous Reset



XC7000	XC9000
Macro	Macro

CB2X2 is a 2-stage, 2-bit, synchronous, loadable, resettable, bidirectional binary counter. CB2X2 has separate count-enable inputs and synchronous terminal-count outputs for up and down directions to support high-speed cascading in the XC7000 and XC9000 CPLD architectures.

The synchronous reset (R) is the highest priority input. When R is High, all other inputs are ignored, data outputs (Q1 – Q0) go to logic level zero, and terminal count outputs TCU and TCD go to zero and one, respectively, on the Low-to-High clock (C) transition. The data on the D1 – D0 inputs loads into the counter on the Low-to-High clock (C) transition when the load enable input (L) is High, independent of the CE inputs.

The outputs (Q1 – Q0) increment when CEU is High, provided R and L are Low during the Low-to-High clock transition. The outputs (Q1 – Q0) decrement when CED is High, provided R and L are Low. The counter ignores clock transitions when CEU and CED are Low. Both CEU and CED should not be High during the same clock transition; the CEOU and CEOD outputs might not function properly for cascading when CEU and CED are both High.

For counting up, the CEOU output is High when all Q outputs and CEU are High. For counting down, the CEOD output is High when all Q outputs are Low and CED is High. To cascade counters, the CEOU and CEOD outputs of each counter are connected directly to the CEU and CED inputs, respectively, of the next stage. The C, L, and R inputs are connected in parallel.

In Xilinx CPLD devices, the maximum clocking frequency of these counter components is unaffected by the number of cascaded stages for all counting and loading functions. The TCU terminal count output is High when all Q outputs are High, regardless of CEU. The

TCD output is High when all Q outputs are Low, regardless of CED.

When cascading counters, the final terminal count signals can be produced by AND-ing all the TCU outputs (for the up direction) and all the TCD outputs (for the down direction). The default initial state of all flip-flops is zero.

Inputs						Outputs				
R	**L**	**CEU**	**CED**	**C**	**D1 – D0**	**Q1 – Q0**	**TCU**	**TCD**	**CEOU**	**CEOD**
1	X	X	X	↑	X	0	0	1	0	CEOD
0	1	X	X	↑	D	d1 – d0	TCU	TCD	CEOU	CEOD
0	0	0	0	X	X	No Chg	No Chg	No Chg	0	0
0	0	1	0	↑	X	Inc	TCU	TCD	CEOU	0
0	0	0	1	↑	X	Dec	TCU	TCD	0	CEOD
0	0	1	1	↑	X	Inc	TCU	TCD	Invalid	Invalid

$\mathrm{TCU} = \mathrm{Q1} \bullet \mathrm{Q0}$

$\mathrm{TCD} = \overline{\mathrm{Q1}} \bullet \overline{\mathrm{Q0}}$

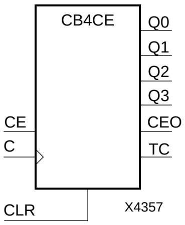
$\mathrm{CEOU} = \mathrm{TCU} \bullet \mathrm{CEU}$

$\mathrm{CEOD} = \mathrm{TCD} \bullet \mathrm{CED}$

dn = state of referenced input one set-up time prior to active clock transition

CB4CE

4-Bit Cascadable Binary Counter with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

CB4CE is a 4-stage, 4-bit, synchronous, clearable, cascadable binary counter. The asynchronous clear (CLR) is the highest priority input. When CLR is High, all other inputs are ignored and data (Q3 – Q0) and terminal count (TC) outputs go to logic level zero, independent of clock transitions. The outputs (Q3 – Q0) increment when the clock enable input (CE) is High during the Low-to-High clock (C) transition. The counter ignores clock transitions when CE is Low. The TC output is High when all Q outputs are High.

Larger counters are created by connecting the count enable out (CEO) output of the first stage to the CE input of the next stage and connecting the C and CLR inputs in parallel. CEO is active (High) when TC and CE are High. The maximum length of the counter is determined by the accumulated CE-to-TC propagation delays versus the clock period. The clock period must be greater than n(tCE-TC), where “n” is the number of stages and “tCE-TC” is the CE-to-TC propagation delay of each stage.

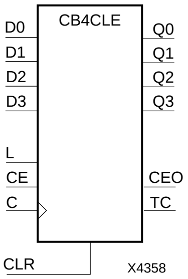
The default initial state of all flip-flops is zero. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

Inputs			Outputs		
CLR	CE	C	Q3 – Q0	TC	CEO
1	X	X	0	0	0
0	0	X	No Chg	No Chg	0
0	1	↑	Inc	TC	CEO

TC = (Q3 • Q2 • Q1 • Q0); CEO = (TC • CE)

CB4CLE

4-Bit Loadable Cascadable Binary Counter with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

CB4CLE is a 4-stage, 4-bit, synchronous, loadable, clearable, cascadable binary counter. The asynchronous clear (CLR) is the highest priority input. When CLR is High, all other inputs are ignored and data (Q3 – Q0) and terminal count (TC) outputs go to logic level zero, independent of clock transitions. The data on the D3 – D0 inputs is loaded into the counter when the load enable input (L) is High during the Low-to-High clock (C) transition, independent of the state of clock enable (CE). The outputs (Q3 – Q0) increment when CE is High during the Low-to-High clock transition. The counter ignores clock transitions when CE is Low. The TC output is High when all Q outputs are High.

Larger counters are created by connecting the count enable out (CEO) output of the first stage to the CE input of the next stage and connecting the C, L, and CLR inputs in parallel. CEO is active (High) when TC and CE are High. The maximum length of the counter is determined by the accumulated CE-to-TC propagation delays versus the clock period. The clock period must be greater than n(tCE-TC), where “n” is the number of stages and “tCE-TC” is the CE-to-TC propagation delay of each stage.

The default initial state of all flip-flops is zero. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

Inputs					Outputs		
CLR	L	CE	C	D3 – D0	Q3 – Q0	TC	CEO
1	X	X	X	X	0	0	0
0	1	X	↑	D	d3 – d0	TC	CEO
0	0	0	X	X	No Chg	No Chg	0
0	0	1	↑	X	Inc	TC	CEO

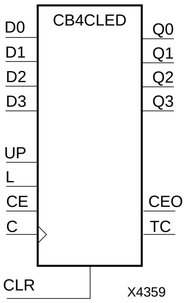
TC = (Q3•Q2•Q1•Q0)

CEO = (TC•CE)

dn = state of referenced input one set-up time prior to active clock transition

CB4CLED

4-Bit Loadable Cascadable Bidirectional Binary Counter with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

CB4CLED is a 4-stage, 4-bit, synchronous, loadable, clearable, cascadable, bidirectional binary counter. The asynchronous clear (CLR) is the highest priority input. When CLR is High, all other inputs are ignored and data (Q3 – Q0) and terminal count (TC) outputs go to logic level zero, independent of clock transitions. The data on the D3 – D0 inputs is loaded into the counter when the load enable input (L) is High during the Low-to-High clock (C) transition, independent of the state of clock enable (CE). The outputs (Q3 – Q0) decrement when CE is High and UP is Low during the Low-to-High clock transition. The outputs (Q3 – Q0) increment when CE and UP are High. The counter ignores clock transitions when CE is Low.

For counting up, the TC output is High when all Q outputs and UP are High. For counting down, the TC output is High when all Q outputs and UP are Low. To cascade counters, the count enable out (CEO) output of each counter is connected to the CE pin of the next stage. The clock, UP, L, and CLR inputs are connected in parallel. CEO is active (High) when TC and CE are High. The maximum length of the counter is determined by the accumulated CE-to-TC propagation delays versus the clock period. The clock period must be greater than n(tCE-TC), where “n” is the number of stages and “tCE-TC” is the CE-to-TC propagation delay of each stage. For XC7000 and XC9000 designs, refer to “CB4X1” for high-performance cascadable, bidirectional counters.

The default initial state of all flip-flops is zero. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

Inputs						Outputs		
CLR	**L**	**CE**	**C**	**UP**	**D3 – D0**	**Q3 – Q0**	**TC**	**CEO**
1	X	X	X	X	X	0	0	0
0	1	X	↑	X	D	d3 – d0	TC	CEO
0	0	0	X	X	X	No Chg	No Chg	0
0	0	1	↑	1	X	Inc	TC	CEO
0	0	1	↑	0	X	Dec	TC	CEO

$TC = (Q3\bullet Q2\bullet Q1\bullet Q0\bullet UP) + (\overline{Q3}\bullet\overline{Q2}\bullet\overline{Q1}\bullet\overline{Q0}\bullet\overline{UP})$

$CEO = (TC\bullet CE)$

dn = state of referenced clock one set-up time prior to active clock transition

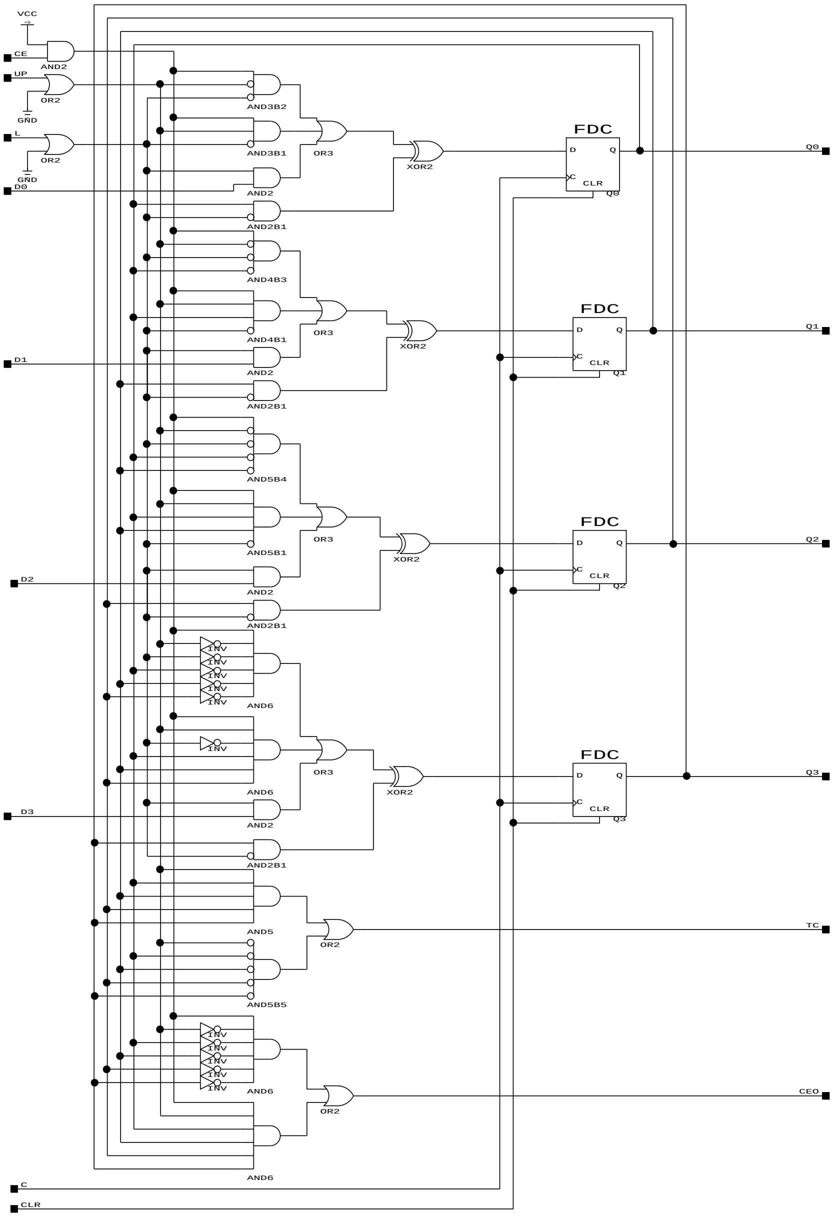
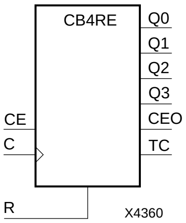


Figure 3-46 CB4CLED Implementation (XC7000 and XC9000)

CB4RE

4-Bit Cascadable Binary Counter with Clock Enable and Synchronous Reset



XC7000	XC9000
Macro	Macro

CB4CE is a 4-stage, 4-bit, synchronous, resettable, cascadable binary counter. The synchronous reset (R) is the highest priority input. When R is High, all other inputs are ignored and data (Q3 – Q0) and terminal count (TC) outputs go to logic level zero on the Low-to-High clock (C) transition. The outputs (Q3 – Q0) increment when the clock enable input (CE) is High during the Low-to-High clock transition. The counter ignores clock transitions when CE is Low. The TC output is High when all Q outputs are High.

Larger counters are created by connecting the count enable out (CEO) output of the first stage to the CE input of the next stage and connecting the C and R inputs in parallel. CEO is active (High) when TC and CE are High. The maximum length of the counter is determined by the accumulated CE-to-TC propagation delays versus the clock period. The clock period must be greater than n(tCE-TC), where "n" is the number of stages and "tCE-TC" is the CE-to-TC propagation delay of each stage.

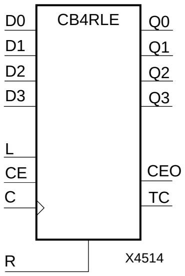
The default initial state of all flip-flops is zero. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

Inputs			Outputs		
R	CE	C	Q3 – Q0	TC	CEO
1	X	↑	0	0	0
0	0	X	No Chg	No Chg	0
0	1	↑	Inc	TC	CEO

TC = (Q3•Q2•Q2•Q0); CEO = (TC•CE)

CB4RLE

4-Bit Loadable Cascadable Binary Counter with Clock Enable and Synchronous Reset



XC7000	XC9000
Macro	Macro

CB4RLE is a 4-stage, 4-bit, synchronous, loadable, resettable, cascadable binary counter. The synchronous reset (R) is the highest priority input. The synchronous R, when High, overrides all other inputs and resets the Q3 – Q0, TC, and CEO outputs to Low on the Low-to-High clock (C) transition. The data on the D3 – D0 inputs is loaded into the counter when the load enable input (L) is High during the Low-to-High clock (C) transition, independent of the state of CE. The outputs (Q3 – Q0) increment when CE is High during the Low-to-High clock transition. The counter ignores clock transitions when CE is Low. The TC output is High when all Q outputs are High. The CEO output is High when all Q outputs and CE are High to allow direct cascading of counters.

Larger counters are created by connecting the CEO output of the first stage to the CE input of the next stage and by connecting the C, L, and R inputs in parallel. The maximum length of the counter is determined by the accumulated CE-to-CEO propagation delays versus the clock period.

The default initial state of all flip-flops is zero. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

Inputs					Outputs		
R	**L**	**CE**	**C**	**D3 – D0**	**Q3 – Q0**	**TC**	**CEO**
1	X	X	↑	X	0	0	0
0	1	X	↑	D	d3 – d0	TC	CEO
0	0	0	X	X	No Chg	No Chg	0
0	0	1	↑	X	Inc	TC	TC

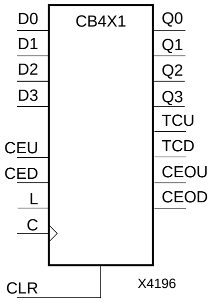
TC = Q3•Q2•Q1•Q0

CEO = TC•CE

dn = state of referenced input one set-up time prior to active clock transition

CB4X1

4-Bit Loadable Cascadable Bidirectional Binary Counter with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

CB4X1 is a 4-stage, 4-bit, synchronous, loadable, clearable, bidirectional binary counter. CB4X1 has separate count-enable inputs and synchronous terminal-count outputs for up and down directions, to support high-speed cascading in the XC7000 and XC9000 CPLD architectures.

The asynchronous clear (CLR) is the highest priority input. When CLR is High, all other inputs are ignored, data outputs (Q3 – Q0) go to logic level zero, and terminal count outputs TCU and TCD go to zero and one, respectively, independent of clock transitions. The data on the D3 – D0 inputs loads into the counter on the Low-to-High clock (C) transition when the load enable input (L) is High, independent of the CE inputs.

The outputs (Q3 – Q0) increment when CEU is High, provided CLR and L are Low, during the Low-to-High clock transition. The outputs (Q3 – Q0) decrement when CED is High, provided CLR and L are Low. The counter ignores clock transitions when CEU and CED are Low. Both CEU and CED should not be High during the same clock transition; the CEOU and CEOD outputs might not function properly for cascading when CEU and CED are both High.

For counting up, the CEOU output is High when all Q outputs and CEU are High. For counting down, the CEOD output is High when all Q outputs are Low and CED is High. To cascade counters, the CEOU and CEOD outputs of each counter are connected directly to the CEU and CED inputs, respectively, of the next stage. The clock, L, and CLR inputs are connected in parallel.

In Xilinx CPLD devices, the maximum clocking frequency of these counter components is unaffected by the number of cascaded stages for all counting and loading functions. The TCU terminal count output is High when all Q outputs are High, regardless of CEU. The

TCD output is High when all Q outputs are Low, regardless of CED.

When cascading counters, the final terminal count signals can be produced by AND-ing all the TCU outputs (for the up direction) and all the TCD outputs (for the down direction). The default initial state of all flip-flops is zero.

Inputs						Outputs				
CLR	**L**	**CEU**	**CED**	**C**	**D3 – D0**	**Q3 – Q0**	**TCU**	**TCD**	**CEOU**	**CEOD**
1	X	X	X	X	X	0	0	1	0	CEOD
0	1	X	X	↑	D	d3 – d0	TCU	TCD	CEOU	CEOD
0	0	0	0	X	X	No Chg	No Chg	No Chg	0	0
0	0	1	0	↑	X	Inc	TCU	TCD	CEOU	0
0	0	0	1	↑	X	Dec	TCU	TCD	0	CEOD
0	0	1	1	↑	X	Inc	TCU	TCD	Invalid	Invalid

$TCU = Q3 \bullet Q2 \bullet Q1 \bullet Q0$

$TCD = \overline{Q3} \bullet \overline{Q2} \bullet \overline{Q1} \bullet \overline{Q0}$

$CEOU = TCU \bullet CEU$

$CEOD = TCD \bullet CED$

dn = state of referenced input one set-up time prior to active clock transition

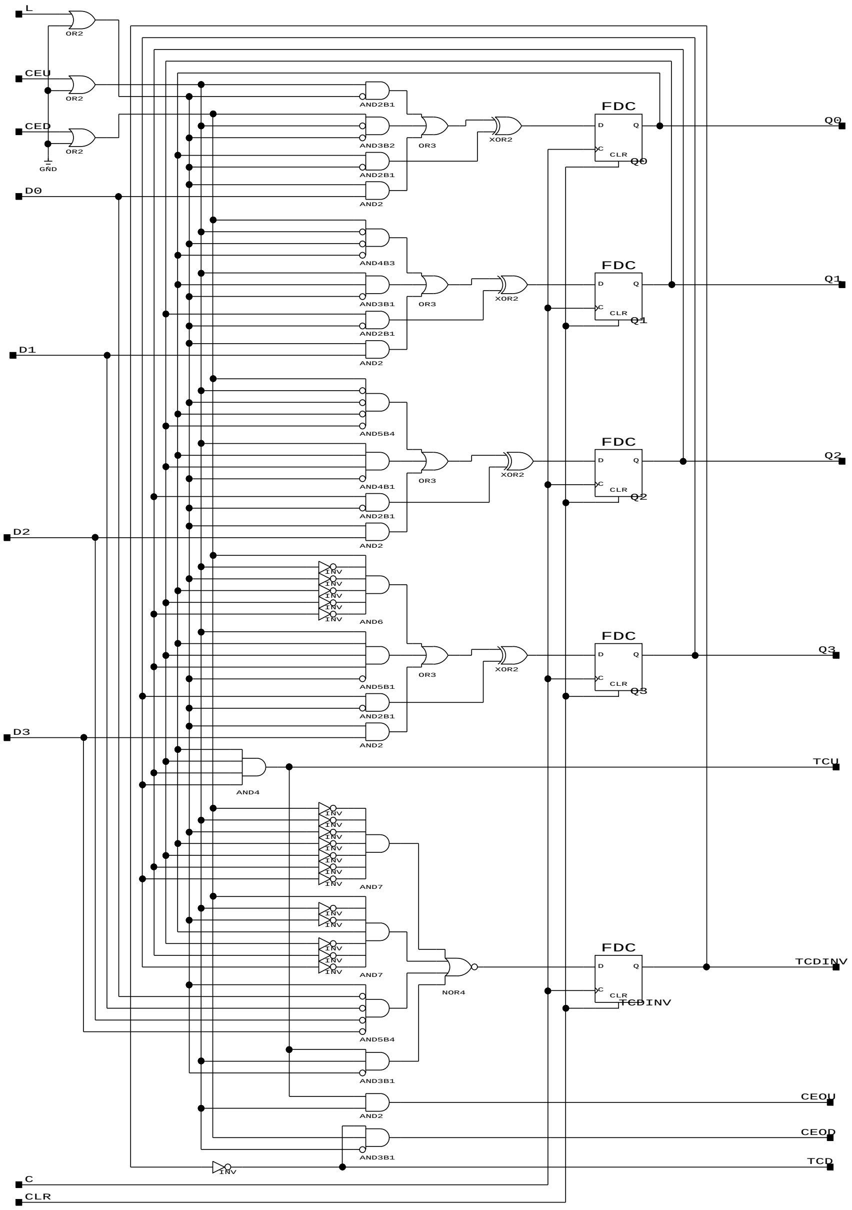
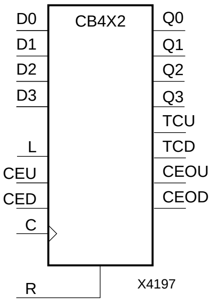


Figure 3-47 CB4X1 Implementation (XC7000 and XC9000)

CB4X2

4-Bit Loadable Cascadable Bidirectional Binary Counter with Clock Enable and Synchronous Reset



XC7000	XC9000
Macro	Macro

CB4X2 is a 4-stage, 4-bit, synchronous, loadable, resettable, bidirectional binary counter. CB4X2 has separate count-enable inputs and synchronous terminal-count outputs for up and down directions to support high-speed cascading in the XC7000 and XC9000 CPLD architectures.

The synchronous reset (R) is the highest priority input. When R is High, all other inputs are ignored, data outputs (Q3 – Q0) go to logic level zero, and terminal count outputs TCU and TCD go to zero and one, respectively, on the Low-to-High clock (C) transition. The data on the D3 – D0 inputs loads into the counter on the Low-to-High clock (C) transition when the load enable input (L) is High, independent of the CE inputs.

The outputs (Q3 – Q0) increment when CEU is High, provided R and L are Low, during the Low-to-High clock transition. The outputs (Q3 – Q0) decrement when CED is High, provided R and L are Low. The counter ignores clock transitions when CEU and CED are Low. Both CEU and CED should not be High during the same clock transition; the CEOU and CEOD outputs might not function properly for cascading when CEU and CED are both High.

For counting up, the CEOU output is High when all Q outputs and CEU are High. For counting down, the CEOD output is High when all Q outputs are Low and CED is High. To cascade counters, the CEOU and CEOD outputs of each counter are connected directly to the CEU and CED inputs, respectively, of the next stage. The C, L, and R inputs are connected in parallel.

In Xilinx CPLD devices, the maximum clocking frequency of these counter components is unaffected by the number of cascaded stages for all counting and loading functions. The TCU terminal count output is High when all Q outputs are High, regardless of CEU. The

TCD output is High when all Q outputs are Low, regardless of CED.

When cascading counters, the final terminal count signals can be produced by AND-ing all the TCU outputs (for the up direction) and all the TCD outputs (for the down direction). The default initial state of all flip-flops is zero.

Inputs						**Outputs**				
R	**L**	**CEU**	**CED**	**C**	**D3 – D0**	**Q3 – Q0**	**TCU**	**TCD**	**CEOU**	**CEOD**
1	X	X	X	↑	X	0	0	1	0	CED
0	1	X	X	↑	D	d3 – d0	TCU	TCD	CEOU	CEOD
0	0	0	0	X	X	No Chg	No Chg	No Chg	0	0
0	0	1	0	↑	X	Inc	TCU	TCD	TCU	0
0	0	0	1	↑	X	Dec	TCU	TCD	0	TCD
0	0	1	1	↑	X	Inc	TCU	TCD	Invalid	Invalid

$TCU = Q3 \bullet Q2 \bullet Q1 \bullet Q0$

$TCD = \overline{Q3} \bullet \overline{Q2} \bullet \overline{Q1} \bullet \overline{Q0}$

$CEOU = TCU \bullet CEU$

$CEOD = TCD \bullet CED$

dn = state of referenced input one set-up time prior to active clock transition

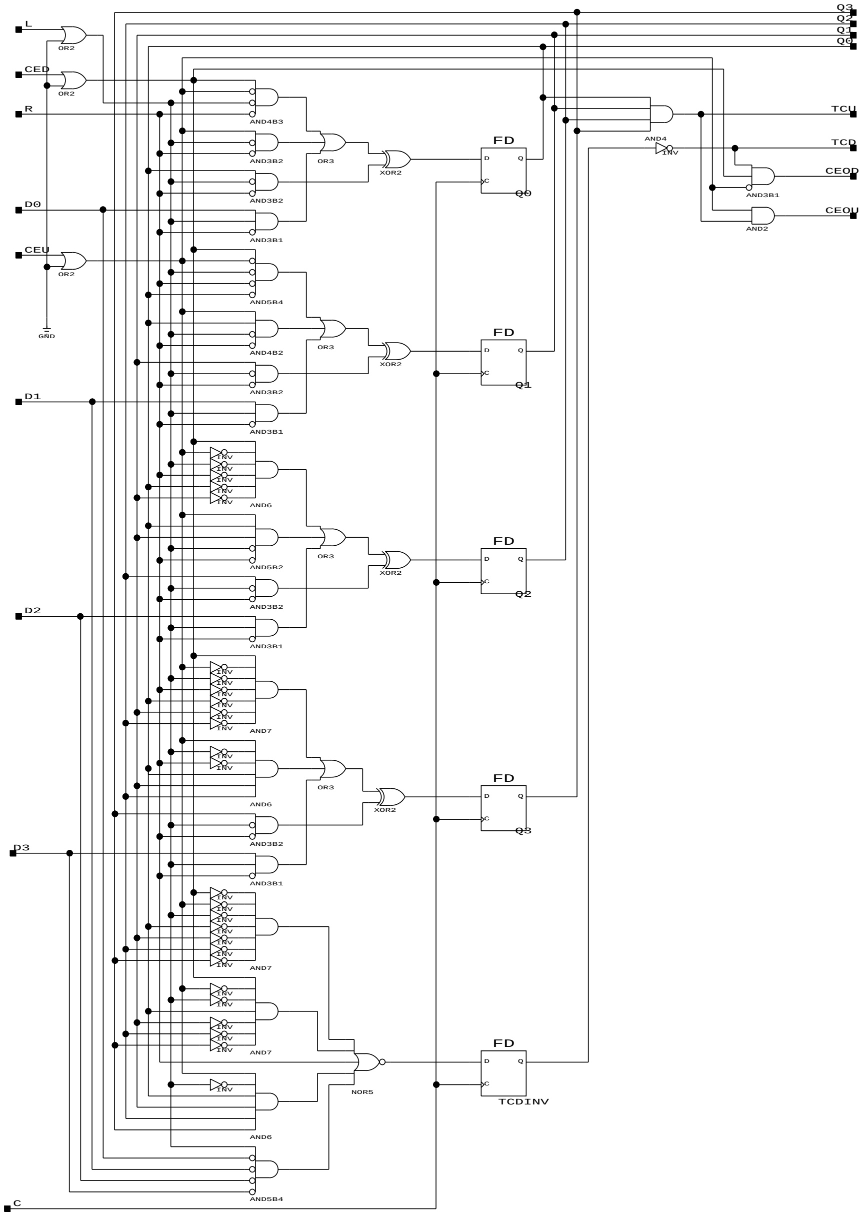
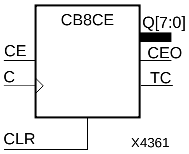


Figure 3-48 CB4X2 Implementation (XC7000 and XC9000)

CB8CE

8-Bit Cascadable Binary Counter with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

CB8CE is an 8-stage, 8-bit, synchronous, clearable, cascadable binary counter. The asynchronous clear (CLR) is the highest priority input. When CLR is High, all other inputs are ignored and data (Q7 – Q0) and terminal count (TC) outputs go to logic level zero, independent of clock transitions. The outputs (Q7 – Q0) increment when the clock enable input (CE) is High during the Low-to-High clock (C) transition. The counter ignores clock transitions when CE is Low. The TC output is High when all Q outputs are High.

Larger counters are created by connecting the count enable out (CEO) output of the first stage to the CE input of the next stage and connecting the C and CLR inputs in parallel. CEO is active (High) when TC and CE are High. The maximum length of the counter is determined by the accumulated CE-to-TC propagation delays versus the clock period. The clock period must be greater than n(tCE-TC), where “n” is the number of stages and “tCE-TC” is the CE-to-TC propagation delay of each stage.

The default initial state of all flip-flops is zero. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

Inputs			Outputs		
CLR	CE	C	Q7 – Q0	TC	CEO
1	X	X	0	0	0
0	0	X	No Chg	No Chg	0
0	1	↑	Inc	TC	CEO

TC = (Q7 • Q6 • Q5 • Q4 • ... • Q0); CEO = (TC • CE)

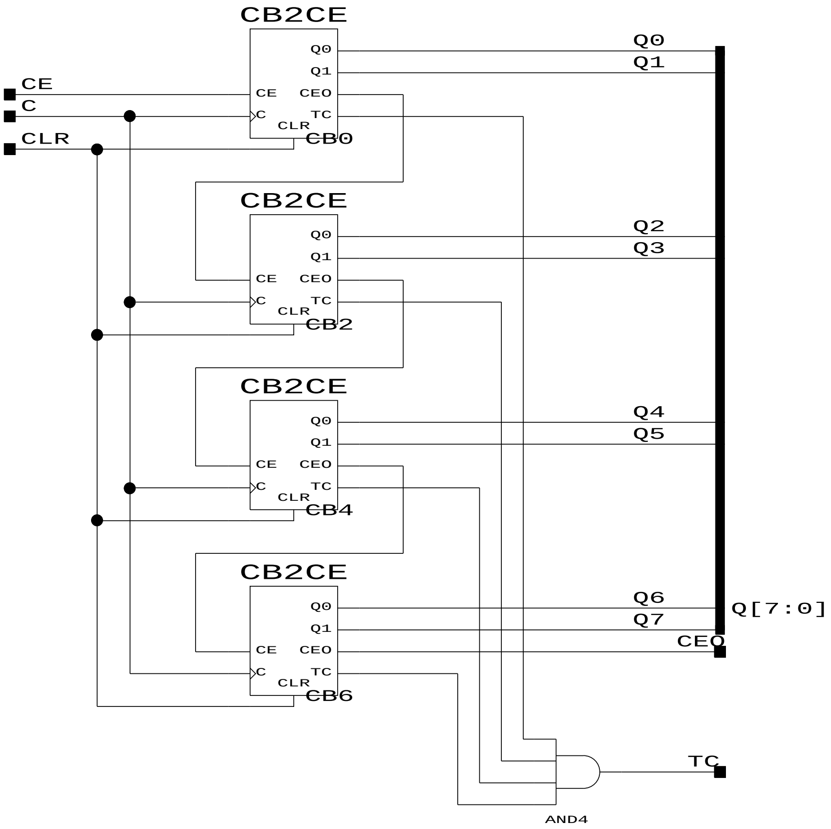
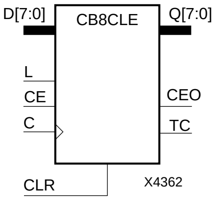


Figure 3-49 CB8CE Implementation (XC7000 and XC9000)

CB8CLE

8-Bit Loadable Cascadable Binary Counter with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

CB8CLE is an 8-stage, 8-bit, synchronous, loadable, clearable, cascadable binary counter. The asynchronous clear (CLR) is the highest priority input. When CLR is High, all other inputs are ignored and data (Q7 – Q0) and terminal count (TC) outputs go to logic level zero, independent of clock transitions. The data on the D7 – D0 inputs is loaded into the counter when the load enable input (L) is High during the Low-to-High clock (C) transition, independent of the state of clock enable (CE). The outputs (Q7 – Q0) increment when CE is High during the Low-to-High clock transition. The counter ignores clock transitions when CE is Low. The TC output is High when all Q outputs are High.

Larger counters are created by connecting the count enable out (CEO) output of the first stage to the CE input of the next stage and by connecting the C, L, and CLR inputs in parallel. CEO is active (High) when TC and CE are High. The maximum length of the counter is determined by the accumulated CE-to-TC propagation delays versus the clock period. The clock period must be greater than n(tCE-TC), where “n” is the number of stages and “tCE-TC” is the CE-to-TC propagation delay of each stage.

The default initial state of all flip-flops is zero. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

Inputs					Outputs		
CLR	**L**	**CE**	**C**	**D7 – D0**	**Q7 – Q0**	**TC**	**CEO**
1	X	X	X	X	0	0	0
0	1	X	↑	D	d7 – d0	TC	CEO
0	0	0	X	X	No Chg	No Chg	0
0	0	1	↑	X	Inc	TC	CEO

TC = (Q7•Q6•Q5•Q4•...•Q0)

CEO = (TC•CE)

dn = state of referenced input one set-up time prior to active clock transition

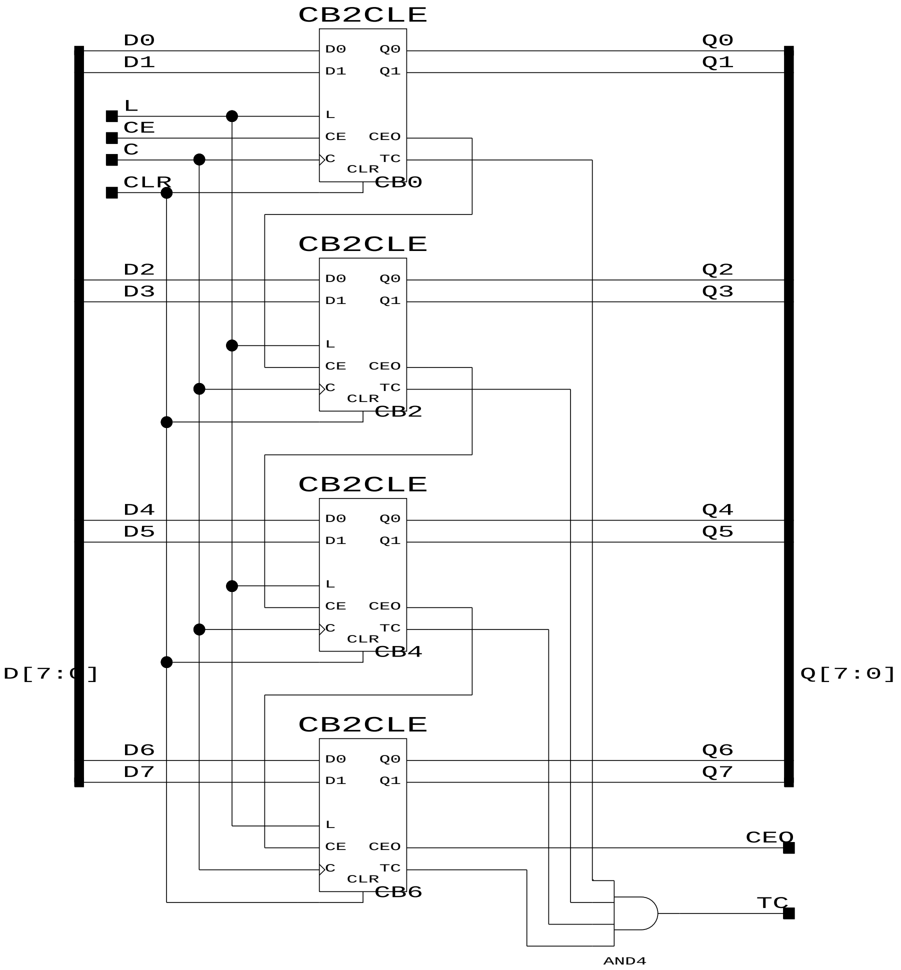
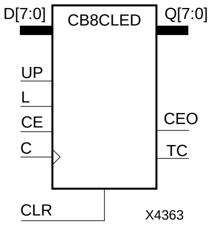


Figure 3-50 CB8CLE Implementation (XC7000 and XC9000)

CB8CLED

8-Bit Loadable Cascadable Bidirectional Binary Counter with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

CB8CLED is an 8-stage, 8-bit, synchronous, loadable, clearable, cascadable, bidirectional binary counter. The asynchronous clear (CLR) is the highest priority input. When CLR is High, all other inputs are ignored and data (Q7 – Q0) and terminal count (TC) outputs go to logic level zero, independent of clock transitions. The data on the D7 – D0 inputs is loaded into the counter when the load enable input (L) is High during the Low-to-High clock (C) transition, independent of the state of CE. The outputs (Q7 – Q0) decrement when CE is High and UP is Low during the Low-to-High clock transition. The outputs (Q7 – Q0) increment when CE and UP are High. The counter ignores clock transitions when CE is Low.

For counting up, the TC output is High when all Q outputs and UP are High. For counting down, the TC output is High when all Q outputs and UP are Low. To cascade counters, the count enable out (CEO) output of each counter is connected to the CE pin of the next stage. The clock, UP, L, and CLR inputs are connected in parallel. CEO is active (High) when TC and CE are High. The maximum length of the counter is determined by the accumulated CE-to-TC propagation delays versus the clock period. The clock period must be greater than n(tCE-TC), where “n” is the number of stages and “tCE-TC” is the CE-to-TC propagation delay of each stage. For XC7000 and XC9000 designs, refer to “CB8X1” for high-performance cascadable, bidirectional counters.

The default initial state of all flip-flops is zero. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

Inputs						Outputs		
CLR	**L**	**CE**	**C**	**UP**	**D7 – D0**	**Q7 – Q0**	**TC**	**CEO**
1	X	X	X	X	X	0	0	0
0	1	X	↑	X	D	d7 – d0	TC	CEO
0	0	0	X	X	X	No Chg	No Chg	0
0	0	1	↑	1	X	Inc	TC	CEO
0	0	1	↑	0	X	Dec	TC	CEO

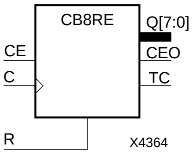
$TC = (Q7 \bullet Q6 \bullet Q5 \bullet \ldots \bullet Q0 \bullet UP) + (\overline{Q7} \bullet \overline{Q6} \bullet \overline{Q5} \bullet \ldots \bullet \overline{Q0} \bullet \overline{UP})$

$CEO = (TC \bullet CE)$

dn = state of referenced clock one set-up time prior to active clock transition

CB8RE

8-Bit Cascadable Binary Counter with Clock Enable and Synchronous Reset



XC7000	XC9000
Macro	Macro

CB8CE is an 8-stage, 8-bit, synchronous, resettable, cascadable binary counter. The synchronous reset (R) is the highest priority input. When R is High, all other inputs are ignored and data (Q7 – Q0) and terminal count (TC) outputs go to logic level zero on the Low-to-High clock (C) transition. The outputs (Q7 – Q0) increment when the clock enable input (CE) is High during the Low-to-High clock transition. The counter ignores clock transitions when CE is Low. The TC output is High when all Q outputs are High.

Larger counters are created by connecting the count enable out (CEO) output of the first stage to the CE input of the next stage and connecting the C and R inputs in parallel. CEO is active (High) when TC and CE are High. The maximum length of the counter is determined by the accumulated CE-to-TC propagation delays versus the clock period. The clock period must be greater than n(tCE-TC), where “n” is the number of stages and “tCE-TC” is the CE-to-TC propagation delay of each stage.

The default initial state of all flip-flops is zero. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

Inputs			Outputs		
R	CE	C	Q7 – Q0	TC	CEO
1	X	↑	0	0	0
0	0	X	No Chg	No Chg	0
0	1	↑	Inc	TC	CEO

TC = (Q7•Q6•Q5•...•Q0); CEO = (TC•CE)

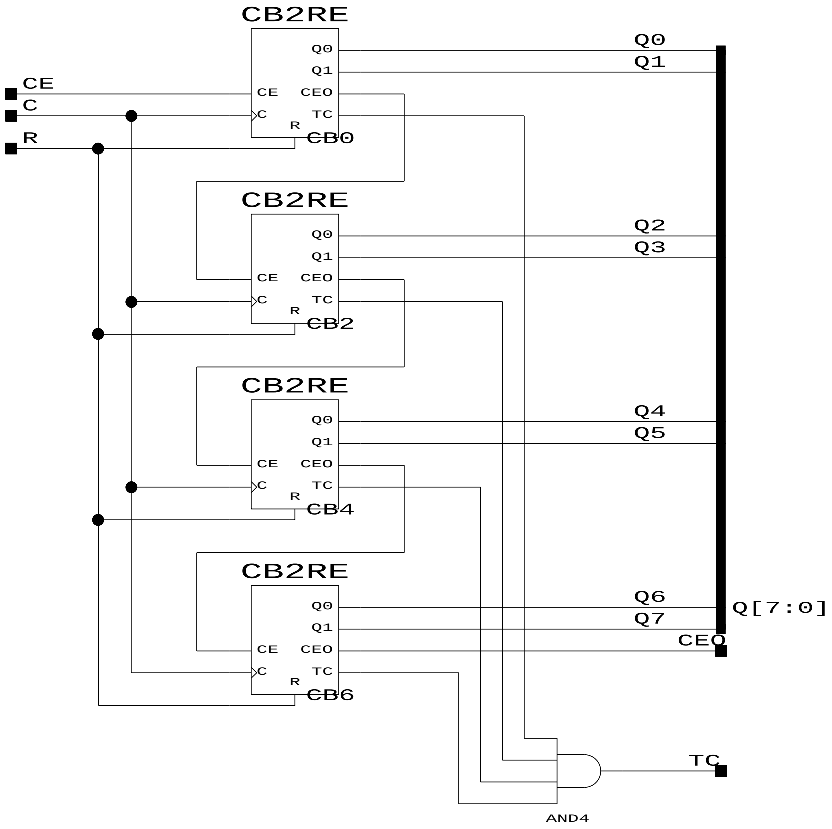
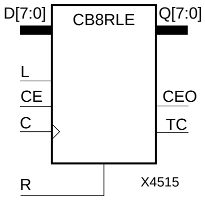


Figure 3-51 CB8RE Implementation (XC7000 and XC9000)

CB8RLE

8-Bit Loadable Cascadable Binary Counter with Clock Enable and Synchronous Reset



XC7000	XC9000
Macro	Macro

CB8RLE is an 8-stage, 8-bit, synchronous, loadable, resettable, cascadable binary counter. The synchronous reset (R) is the highest priority input. The synchronous R, when High, overrides all other inputs and resets the Q7 – Q0, TC, and CEO outputs to Low on the Low-to-High clock (C) transition. The data on the D7 – D0 inputs is loaded into the counter when the load enable input (L) is High during the Low-to-High clock (C) transition, independent of the state of CE.

The outputs (Q7 – Q0) increment when CE is High during the Low-to-High clock transition. The counter ignores clock transitions when CE is Low. The TC output is High when all Q outputs are High. The CEO output is High when all Q outputs and CE are High, to allow direct cascading of counters.

Larger counters are created by connecting the CEO output of the first stage to the CE input of the next stage and connecting the C, L, and R inputs in parallel. The maximum length of the counter is determined by the accumulated CE-to-CEO propagation delays versus the clock period.

The default initial state of all flip-flops is zero. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

Inputs					Outputs		
R	**L**	**CE**	**C**	**D7 – D0**	**Q7 – Q0**	**TC**	**CEO**
1	X	X	↑	X	0	0	0
0	1	X	↑	D	d7 – d0	TC	CEO
0	0	0	X	X	No Chg	No Chg	0
0	0	1	↑	X	Inc	TC	CEO

TC = Q7•Q6•…•Q0

CEO = TC•CE

dn = state of referenced input one set-up time prior to active clock transition

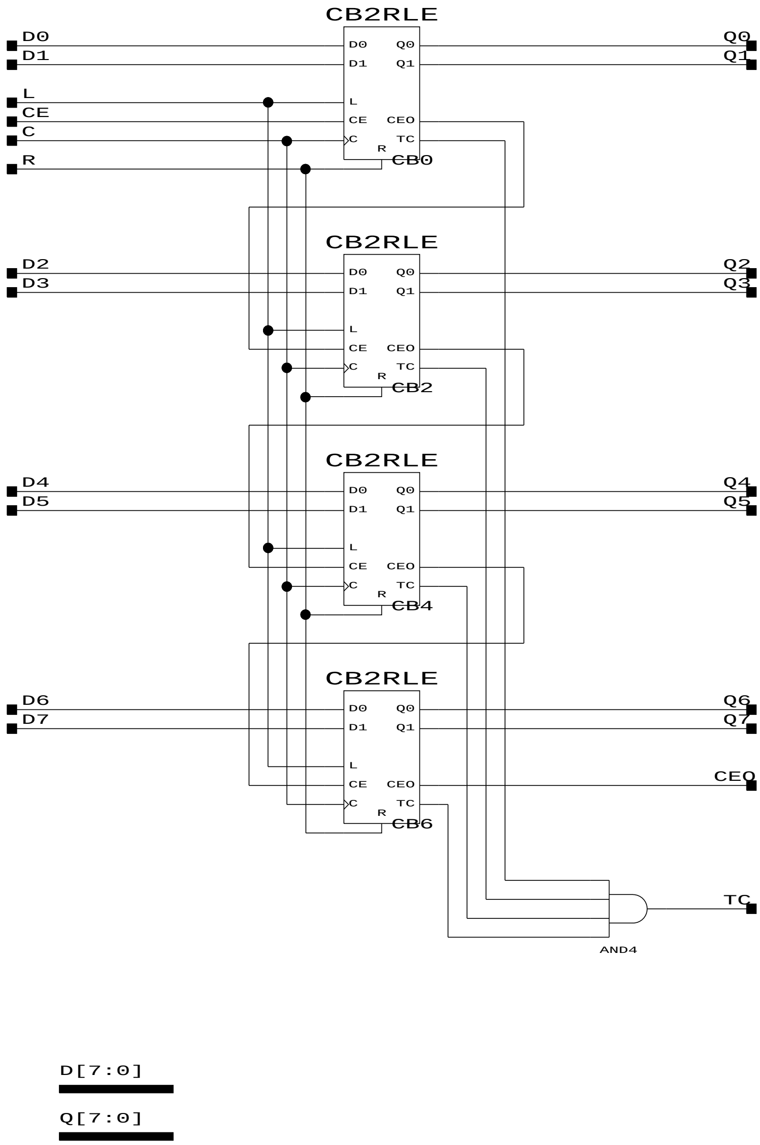
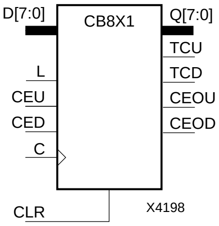


Figure 3-52 CB8RLE Implementation (XC7000 and XC9000)

CB8X1

8-Bit Loadable Cascadable Bidirectional Binary Counter with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

CB8X1 is an 8-stage, 8-bit, synchronous, loadable, clearable, bidirectional binary counter. CB8X1 has separate count-enable inputs and synchronous terminal-count outputs for up and down directions to support high-speed cascading in the XC7000 and XC9000 CPLD architectures.

The asynchronous clear (CLR) is the highest priority input. When CLR is High, all other inputs are ignored, data outputs (Q7 – Q0) go to logic level zero, and terminal count outputs TCU and TCD go to zero and one, respectively, independent of clock transitions. The data on the D7 – D0 inputs loads into the counter on the Low-to-High clock (C) transition when the load enable input (L) is High, independent of the CE inputs.

The outputs (Q7 – Q0) increment when CEU is High, provided CLR and L are Low, during the Low-to-High clock transition. The outputs (Q7 – Q0) decrement when CED is High, provided CLR and L are Low. The counter ignores clock transitions when CEU and CED are Low. Both CEU and CED should not be High during the same clock transition; the CEOU and CEOD outputs might not function properly for cascading when CEU and CED are both High.

For counting up, the CEOU output is High when all Q outputs and CEU are High. For counting down, the CEOD output is High when all Q outputs are Low and CED is High. To cascade counters, the CEOU and CEOD outputs of each counter are connected directly to the CEU and CED inputs, respectively, of the next stage. The C, L, and CLR inputs are connected in parallel.

In Xilinx CPLD devices, the maximum clocking frequency of these counter components is unaffected by the number of cascaded stages for all counting and loading functions. The TCU terminal count output is High when all Q outputs are High, regardless of CEU. The

TCD output is High when all Q outputs are Low, regardless of CED. When cascading counters, the final terminal count signals can be produced by AND-ing all the TCU outputs (for the up direction) and all the TCD outputs (for the down direction). The default initial state of all flip-flops is zero.

Inputs						Outputs				
CLR	**L**	**CEU**	**CED**	**C**	**D7 – D0**	**Q7 – Q0**	**TCU**	**TCD**	**CEOU**	**CEOD**
1	X	X	X	X	X	0	0	1	0	CEOD
0	1	X	X	↑	D	d7 – d0	TCU	TCD	CEOU	CEOD
0	0	0	0	X	X	No Chg	No Chg	No Chg	0	0
0	0	1	0	↑	X	Inc	TCU	TCD	CEOU	0
0	0	0	1	↑	X	Dec	TCU	TCD	0	CEOD
0	0	1	1	↑	X	Inc	TCU	TCD	Invalid	Invalid

$TCU = Q7 \bullet Q6 \bullet Q5 \bullet \ldots Q0$

$TCD = \overline{Q7} \bullet \overline{Q6} \bullet \overline{Q5} \bullet \ldots \overline{Q0}$

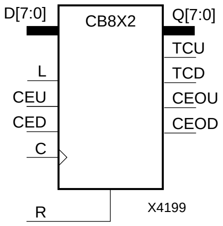
$CEOU = TCU \bullet CEU$

$CEOD = TCD \bullet CED$

dn = state of referenced input one set-up time prior to active clock transition

CB8X2

8-Bit Loadable Cascadable Bidirectional Binary Counter with Clock Enable and Synchronous Reset



XC7000	XC9000
Macro	Macro

CB8X2 is an 8-stage, 8-bit, synchronous, loadable, resettable, bidirectional binary counter. CB8X2 has separate count-enable inputs and synchronous terminal-count outputs for up and down directions to support high-speed cascading in the XC7000 and XC9000 CPLD architectures.

The synchronous reset (R) is the highest priority input. When R is High, all other inputs are ignored, data outputs (Q7 – Q0) go to logic level zero, and terminal count outputs TCU and TCD go to zero and one, respectively, on the Low-to-High clock (C) transition. The data on the D7 – D0 inputs loads into the counter on the Low-to-High clock (C) transition when the load enable input (L) is High, independent of the CE inputs.

The outputs (Q7 – Q0) increment when CEU is High, provided R and L are Low, during the Low-to-High clock transition. The outputs (Q7 – Q0) decrement when CED is High, provided R and L are Low. The counter ignores clock transitions when CEU and CED are Low. Both CEU and CED should not be High during the same clock transition; the CEOU and CEOD outputs might not function properly for cascading when CEU and CED are both High.

For counting up, the CEOU output is High when all Q outputs and CEU are High. For counting down, the CEOD output is High when all Q outputs are Low and CED is High. To cascade counters, the CEOU and CEOD outputs of each counter are connected directly to the CEU and CED inputs, respectively, of the next stage. The C, L, and R inputs are connected in parallel.

In Xilinx CPLD devices, the maximum clocking frequency of these counter components is unaffected by the number of cascaded stages for all counting and loading functions. The TCU terminal count output is High when all Q outputs are High, regardless of CEU. The

TCD output is High when all Q outputs are Low, regardless of CED.

When cascading counters, the final terminal count signals can be produced by AND-ing all the TCU outputs (for the up direction) and all the TCD outputs (for the down direction). The default initial state of all flip-flops is zero.

Inputs						**Outputs**				
R	**L**	**CEU**	**CED**	**C**	**D7 – D0**	**Q7 – Q0**	**TCU**	**TCD**	**CEOU**	**CEOD**
1	X	X	X	↑	X	0	0	1	0	CEOD
0	1	X	X	↑	D	d7 – d0	TCU	TCD	CEOU	CEOD
0	0	0	0	X	X	No Chg	No Chg	No Chg	0	0
0	0	1	0	↑	X	Inc	TCU	TCD	CEOU	0
0	0	0	1	↑	X	Dec	TCU	TCD	0	CEOD
0	0	1	1	↑	X	Inc	TCU	TCD	Invalid	Invalid

TCU = $Q7 \bullet Q6 \bullet Q5 \bullet ...Q0$

TCD = $\overline{Q7} \bullet \overline{Q6} \bullet \overline{Q5} \bullet ...\overline{Q0}$

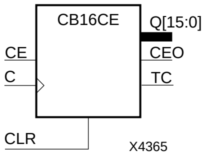
CEOU = $TCU \bullet CEU$

CEOD = $TCD \bullet CED$

dn = state of referenced input one set-up time prior to active clock transition

CB16CE

16-Bit Cascadable Binary Counter with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

CB16CE is a 16-stage, 16-bit, synchronous, clearable, cascadable binary counter. The asynchronous clear (CLR) is the highest priority input. When CLR is High, all other inputs are ignored and data (Q15 – Q0) and terminal count (TC) outputs go to logic level zero, independent of clock transitions. The outputs (Q15 – Q0) increment when the clock enable input (CE) is High during the Low-to-High clock (C) transition. The counter ignores clock transitions when CE is Low. The TC output is High when all Q outputs are High.

Larger counters are created by connecting the count enable out (CEO) output of the first stage to the CE input of the next stage and connecting the C and CLR inputs in parallel. CEO is active (High) when TC is High and CE is High. The maximum length of the counter is determined by the accumulated CE-to-TC propagation delays versus the clock period. The clock period must be greater than n(tCE-TC), where “n” is the number of stages and “tCE-TC” is the CE-to-TC propagation delay of each stage.

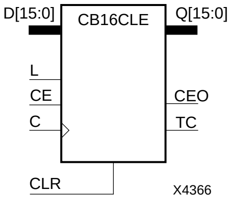
The default initial state of all flip-flops is zero. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

Inputs			Outputs		
CLR	CE	C	Q15 – Q0	TC	CEO
1	X	X	0	0	0
0	0	X	No Chg	No Chg	0
0	1	↑	Inc	TC	CEO

TC = (Q15•Q14•Q13•Q12...•Q0); CEO = (TC•CE)

CB16CLE

16-Bit Loadable Cascadable Binary Counter with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

CB16CLE is a 16-stage, 16-bit, synchronous, loadable, clearable, cascadable binary counter. The asynchronous clear (CLR) is the highest priority input. When CLR is High, all other inputs are ignored and data (Q15 – Q0) and terminal count (TC) outputs go to logic level zero, independent of clock transitions. The data on the D15 – D0 inputs is loaded into the counter when the load enable input (L) is High during the Low-to-High clock (C) transition, independent of the state of clock enable (CE). The outputs (Q15 – Q0) increment when CE is High during the Low-to-High clock transition. The counter ignores clock transitions when CE is Low. The TC output is High when all Q outputs are High.

Larger counters are created by connecting the count enable out (CEO) output of the first stage to the CE input of the next stage and connecting the C, L, and CLR inputs in parallel. CEO is active (High) when TC and CE are High. The maximum length of the counter is determined by the accumulated CE-to-TC propagation delays versus the clock period. The clock period must be greater than n(tCE-TC), where “n” is the number of stages and “tCE-TC” is the CE-to-TC propagation delay of each stage.

The default initial state of all flip-flops is zero. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

Inputs					Outputs		
CLR	**L**	**CE**	**C**	**D15 – D0**	**Q15 – Q0**	**TC**	**CEO**
1	X	X	X	X	0	0	0
0	1	X	↑	D	d15 – d0	TC	CEO
0	0	0	X	X	No Chg	No Chg	0
0	0	1	↑	X	Inc	TC	CEO

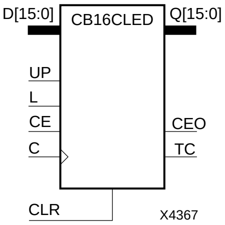
TC = (Q15•Q14•Q13•Q12...•Q0)

CEO = (TC•CE)

dn = state of referenced input one set-up time prior to active clock transition

CB16CLED

16-Bit Loadable Cascadable Bidirectional Binary Counter with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

CB16CLED is a 16-stage, 16-bit, synchronous, loadable, clearable, cascadable, bidirectional binary counter. The asynchronous clear (CLR) is the highest priority input. When CLR is High, all other inputs are ignored and data (Q15 – Q0) and terminal count (TC) outputs go to logic level zero, independent of clock transitions. The data on the D15 – D0 inputs is loaded into the counter when the load enable input (L) is High during the Low-to-High clock (C) transition, independent of the state of clock enable (CE). The outputs (Q15 – Q0) decrement when CE is High and UP is Low during the Low-to-High clock transition. The outputs (Q15 – Q0) increment when CE and UP are High. The counter ignores clock transitions when CE is Low.

For counting up, the TC output is High when all Q outputs and UP are High. For counting down, the TC output is High when all Q outputs and UP are Low. To cascade counters, the count enable out (CEO) output of each counter is connected to the CE pin of the next stage. The clock, UP, L, and CLR inputs are connected in parallel. CEO is active (High) when TC and CE are High. The maximum length of the counter is determined by the accumulated CE-to-TC propagation delays versus the clock period. The clock period must be greater than n(tCE-TC), where “n” is the number of stages and “tCE-TC” is the CE-to-TC propagation delay of each stage. For XC7000 and XC9000 designs, refer to “CB16X1” for high-performance cascadable, bidirectional counters.

The default initial state of all flip-flops is zero. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

Inputs						Outputs		
CLR	**L**	**CE**	**C**	**UP**	**D15 – D0**	**Q15 – Q0**	**TC**	**CEO**
1	X	X	X	X	X	0	0	0
0	1	X	↑	X	D	d15 – d0	TC	CEO
0	0	0	X	X	X	No Chg	No Chg	0
0	0	1	↑	1	X	Inc	TC	CEO
0	0	1	↑	0	X	Dec	TC	CEO

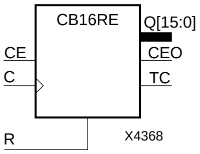
$TC = (Q15 \bullet Q14 \bullet Q13 \ldots \bullet Q0 \bullet UP) + (\overline{Q15} \bullet \overline{Q14} \bullet \overline{Q13} \ldots \bullet \overline{Q0} \bullet \overline{UP})$

$CEO = (TC \bullet CE)$

dn = state of referenced clock one set-up time prior to active clock transition

CB16RE

16-Bit Cascadable Binary Counter with Clock Enable and Synchronous Reset



XC7000	XC9000
Macro	Macro

CB16RE is a 16-stage, 16-bit, synchronous, resettable, cascadable binary counter. The synchronous reset (R) is the highest priority input. When R is High, all other inputs are ignored and data (Q15 – Q0) and terminal count (TC) outputs go to logic level zero on the Low-to-High clock (C) transition. The outputs (Q15 – Q0) increment when the clock enable input (CE) is High during the Low-to-High clock transition. The counter ignores clock transitions when CE is Low. The TC output is High when all Q outputs are High.

Larger counters are created by connecting the count enable out (CEO) output of the first stage to the CE input of the next stage and connecting the C and R inputs in parallel. CEO is active (High) when TC and CE are High. The maximum length of the counter is determined by the accumulated CE-to-TC propagation delays versus the clock period. The clock period must be greater than n(tCE-TC), where “n” is the number of stages and “tCE-TC” is the CE-to-TC propagation delay of each stage.

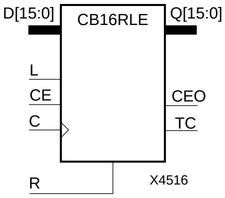
The default initial state of all flip-flops is zero. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

Inputs			Outputs		
R	CE	C	Q15 – Q0	TC	CEO
1	X	↑	0	0	0
0	0	X	No Chg	No Chg	0
0	1	↑	Inc	TC	CEO

TC = (Q15•Q14•Q13...•Q0); CEO = (TC•CE)

CB16RLE

16-Bit Loadable Cascadable Binary Counter with Clock Enable and Synchronous Reset



XC7000	XC9000
Macro	Macro

CB16RLE is a 16-stage, 16-bit, synchronous, loadable, resettable, cascadable binary counter. The synchronous reset (R) is the highest priority input.

The synchronous R, when High, overrides all other inputs and resets the Q15 – Q0, TC, and CEO outputs to Low on the Low-to-High clock (C) transition. The data on the D15 – D0 inputs is loaded into the counter when the load enable input (L) is High during the Low-to-High clock (C) transition, independent of the state of CE. The outputs (Q15 – Q0) increment when CE is High during the Low-to-High clock transition. The counter ignores clock transitions when CE is Low.

The TC output is High when all Q outputs are High. The CEO output is High when all Q outputs and CE are High, to allow direct cascading of counters. Larger counters are created by connecting the CEO output of the first stage to the CE input of the next stage and connecting the C, L, and R inputs in parallel. The maximum length of the counter is determined by the accumulated CE-to-CEO propagation delays versus the clock period.

The default initial state of all flip-flops is zero. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

Inputs					Outputs		
R	L	CE	C	D15 – D0	Q15 – Q0	TC	CEO
1	X	X	↑	X	0	0	0
0	1	X	↑	D	d15 – d0	TC	CEO
0	0	0	X	X	No Chg	No Chg	0
0	0	1	↑	X	Inc	TC	CEO

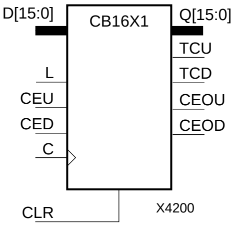
TC = Q15•Q14•...•Q0

CEO = TC•CE

dn = state of referenced input one set-up time prior to clock transition

CB16X1

16-Bit Loadable Cascadable Bidirectional Binary Counter with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

CB16X1 is a 16-stage, 16-bit, synchronous, loadable, clearable, bidirectional binary counter. CB16X1 has separate count-enable inputs and synchronous terminal-count outputs for up and down directions to support high-speed cascading in the XC7000 and XC9000 CPLD architectures.

The asynchronous clear (CLR) is the highest priority input. When CLR is High, all other inputs are ignored, data outputs (Q15 – Q0) go to logic level zero, and terminal count outputs TCU and TCD go to zero and one, respectively, independent of clock transitions. The data on the D15 – D0 inputs loads into the counter on the Low-to-High clock (C) transition, when the load enable input (L) is High, independent of the CE inputs.

The outputs (Q15 – Q0) increment when CEU is High, provided CLR and L are Low, during the Low-to-High clock transition. The outputs (Q15 – Q0) decrement when CED is High, provided CLR and L are Low. The counter ignores clock transitions when CEU and CED are Low. Both CEU and CED should not be High during the same clock transition; the CEOU and CEOD outputs might not function properly for cascading when CEU and CED are both High.

For counting up, the CEOU output is High when all Q outputs and CEU are High. For counting down, the CEOD output is High when all Q outputs are Low and CED is High. To cascade counters, the CEOU and CEOD outputs of each counter are connected directly to the CEU and CED inputs, respectively, of the next stage. The clock, L, and CLR inputs are connected in parallel.

In Xilinx CPLD devices, the maximum clocking frequency of these counter components is unaffected by the number of cascaded stages for all counting and loading functions. The TCU terminal count output is High when all Q outputs are High, regardless of CEU. The

TCD output is High when all Q outputs are Low, regardless of CED.

When cascading counters, the final terminal count signals can be produced by AND-ing all the TCU outputs (for the up direction) and all the TCD outputs (for the down direction). The default initial state of all flip-flops is zero.

Inputs						**Outputs**				
CLR	**L**	**CEU**	**CED**	**C**	**D15 – D0**	**Q15 – Q0**	**TCU**	**TCD**	**CEOU**	**CEOD**
1	X	X	X	X	X	0	0	1	0	CEOD
0	1	X	X	↑	D	d15 – d0	TCU	TCD	CEOU	CEOD
0	0	0	0	X	X	No Chg	No Chg	No Chg	0	0
0	0	1	0	↑	X	Inc	TCU	TCD	CEOU	0
0	0	0	1	↑	X	Dec	TCU	TCD	0	CEOD
0	0	1	1	↑	X	Inc	TCU	TCD	Invalid	Invalid

TCU = $Q15 \bullet Q14 \bullet Q13 \bullet \ldots \bullet Q0$

TCD = $\overline{Q15} \bullet \overline{Q14} \bullet \overline{Q13} \bullet \ldots \bullet \overline{Q0}$

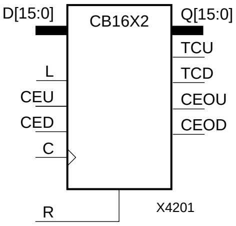
CEOU = $TCU \bullet CEU$

CEOD = $TCD \bullet CED$

dn = state of referenced input one set-up time prior to active clock transition

CB16X2

16-Bit Loadable Cascadable Bidirectional Binary Counter with Clock Enable and Synchronous Reset



XC7000	XC9000
Macro	Macro

CB16X2 is a 16-stage, 16-bit, synchronous, loadable, resettable, bidirectional binary counter. CB16X2 has separate count-enable inputs and synchronous terminal-count outputs for up and down directions, to support high-speed cascading in the CPLD architecture.

The synchronous reset (R) is the highest priority input. When R is High, all other inputs are ignored, data outputs (Q15 – Q0) go to logic level zero, and terminal count outputs TCU and TCD go to zero and one, respectively, on the Low-to-High clock (C) transition. The data on the D15 – D0 inputs loads into the counter on the Low-to-High clock (C) transition when the load enable input (L) is High, independent of the CE inputs. The outputs (Q15 – Q0) increment when CEU is High, provided R and L are Low, during the Low-to-High clock transition. The outputs (Q15 – Q0) decrement when CED is High, provided R and L are Low.

The counter ignores clock transitions when CEU and CED are Low. Both CEU and CED should not be High during the same clock transition; the CEOU and CEOD outputs might not function properly for cascading when CEU and CED are both High. For counting up, the CEOU output is High when all Q outputs and CEU are High. For counting down, the CEOD output is High when all Q outputs are Low and CED is High. To cascade counters, the CEOU and CEOD outputs of each counter are connected directly to the CEU and CED inputs, respectively, of the next stage. The C, L, and R inputs are connected in parallel.

In Xilinx CPLD devices, the maximum clocking frequency of these counter components is unaffected by the number of cascaded stages for all counting and loading functions. The TCU terminal count output is High when all Q outputs are High, regardless of CEU. The TCD output is High when all Q outputs are Low, regardless of CED.

When cascading counters, the final terminal count signals can be produced by AND wiring all the TCU outputs (for the up direction) and all the TCD outputs (for the down direction). The TCU, CEOU, and CEOD outputs are produced by optimizable AND gates within the component, resulting in zero propagation from the CEU and CED inputs and from the Q outputs, provided all connections from each such output remain on-chip. Otherwise, a macrocell buffer delay is introduced. The default initial state of all flip-flops is zero. The clock (C) input can be driven by either the FastCLK global net (represented by a BUFG symbol), an ordinary input, or other on-chip logic.

Inputs						**Outputs**				
R	**L**	**CEU**	**CED**	**C**	**D15 – D0**	**Q15 – Q0**	**TCU**	**TCD**	**CEOU**	**CEOD**
1	X	X	X	↑	X	0	0	1	0	CEOD
0	1	X	X	↑	D	d15 – d0	TCU	TCD	CEOU	CEOD
0	0	0	0	X	X	No Chg	No Chg	No Chg	0	0
0	0	1	0	↑	X	Inc	TCU	TCD	CEOU	0
0	0	0	1	↑	X	Dec	TCU	TCD	0	CEOD
0	0	1	1	↑	X	Inc	TCU	TCD	Invalid	Invalid

$TCU = Q15 \cdot Q14 \cdot Q13 \cdot \ldots \cdot Q0$

$TCD = \overline{Q15} \cdot \overline{Q14} \cdot \overline{Q13} \cdot \ldots \cdot \overline{Q0}$

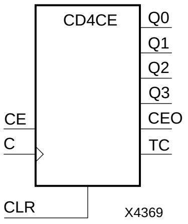
$CEOU = TCU \cdot CEU$

$CEOD = TCD \cdot CED$

dn = state of referenced input one set-up time prior to active clock transition.

CD4CE

4-Bit Cascadable BCD Counter with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

CD4CE is a 4-stage, 4-bit, synchronous, clearable, cascadable binary-coded-decimal (BCD) counter. The asynchronous clear input (CLR) is the highest priority input. When CLR is High, all other inputs are ignored and data (Q3 – Q0) and terminal count (TC) outputs go to logic level zero, independent of clock transitions. The outputs (Q3 – Q0) increment when clock enable (CE) is High during the Low-to-High clock (C) transition. The counter ignores clock transitions when CE is Low. The TC output is High when Q3 and Q0 are High and Q2 and Q1 are Low.

Larger counters are created by connecting the count enable out (CEO) output of the first stage to the CE input of the next stage and connecting the CLR and clock inputs in parallel. CEO is active (High) when TC and CE are High. The maximum length of the counter is determined by the accumulated CE-to-TC propagation delays versus the clock period. The clock period must be greater than n(tCE-TC), where “n” is the number of stages and “tCE-TC” is the CE-to-TC propagation delay of each stage.

The default initial state of all flip-flops is zero. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

Inputs			Outputs					
CLR	**CE**	**C**	**Q3**	**Q2**	**Q1**	**Q0**	**TC**	**CEO**
1	X	X	0	0	0	0	0	0
0	1	↑	--------Increment--------				TC	CEO
0	0	X	--------No Change--------				TC	0
0	1	X	1	0	0	1	1	1

TC = $(Q3 \cdot \overline{Q2} \cdot \overline{Q1} \cdot Q0)$

CEO = $(TC \cdot CE)$

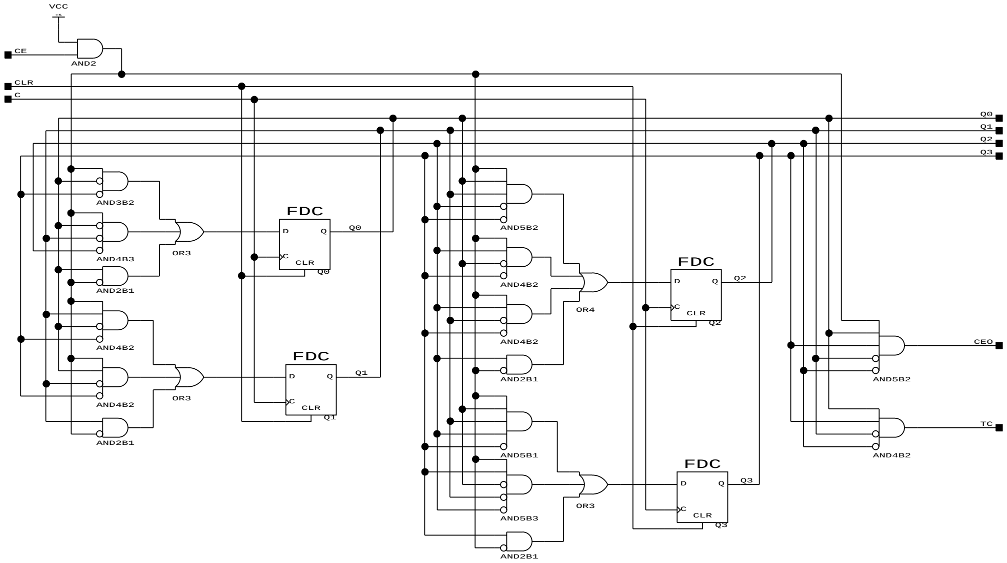
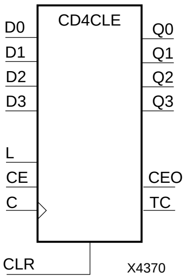


Figure 3-53 CD4CE Implementation (XC7000 and XC9000)

CD4CLE

4-Bit Loadable Cascadable BCD Counter with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

CD4CLE is a 4-stage, 4-bit, synchronous, loadable, clearable, binary-coded-decimal (BCD) counter. The asynchronous clear input (CLR) is the highest priority input. When CLR is High, all other inputs are ignored and the data (Q3 – Q0) and terminal count (TC) outputs go to logic level zero, independent of clock transitions. The data on the D3 – D0 inputs is loaded into the counter when the load enable input (L) is High during the Low-to-High clock (C) transition. The outputs (Q3 – Q0) increment when clock enable input (CE) is High during the Low- to-High clock transition. The counter ignores clock transitions when CE is Low. The TC output is High when Q3 and Q0 are High and Q2 and Q1 are Low.

Larger counters are created by connecting the count enable out (CEO) output of the first stage to the CE input of the next stage and connecting the CLR, L, and C inputs in parallel. CEO is active (High) when TC and CE are High. The maximum length of the counter is determined by the accumulated CE-to-TC propagation delays versus the clock period. The clock period must be greater than n(tCE-TC), where “n” is the number of stages and “tCE-TC” is the CE-to-TC propagation delay of each stage.

The default initial state of all flip-flops is zero. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

Inputs					Outputs					
CLR	L	CE	D3 – D0	C	Q3	Q2	Q1	Q0	TC	CEO
1	X	X	X	X	0	0	0	0	0	0
0	1	X	D3 – D0	↑	d3	d2	d1	d0	TC	CEO
0	0	1	X	↑	---------Increment----------				TC	CEO
0	0	0	X	X	--------No Change---------				TC	0
0	0	1	X	X	1	0	0	1	1	1

TC = $(Q3 \bullet \overline{Q2} \bullet \overline{Q1} \bullet Q0)$

CEO = $(TC \bullet CE)$

dn = state of referenced input one set-up time prior to active clock transition

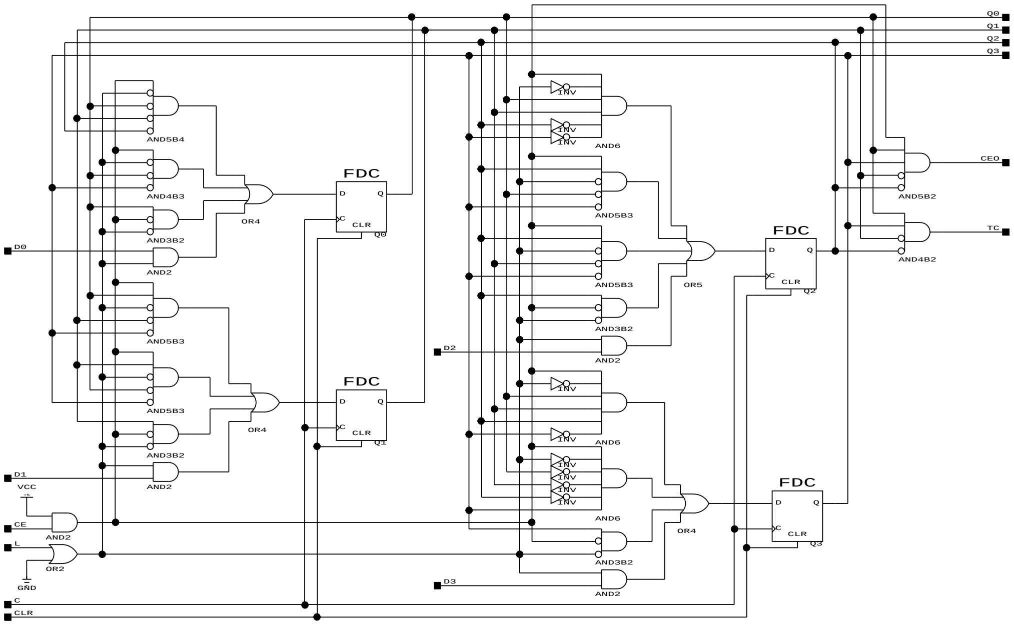
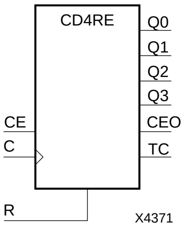


Figure 3-54 CD4CLE Implementation (XC7000 and XC9000)

CD4RE

4-Bit Cascadable BCD Counter with Clock Enable and Synchronous Reset



XC7000	XC9000
Macro	Macro

CD4RE is a 4-stage, 4-bit, synchronous, resettable, cascadable binary-coded-decimal (BCD) counter. The synchronous reset input (R) is the highest priority input. When R is High, all other inputs are ignored and (Q3 – Q0) and terminal count (TC) outputs go to logic level zero on the Low-to-High clock (C) transition. The outputs (Q3 – Q0) increment when the clock enable input (CE) is High during the Low-to-High clock transition. The counter ignores clock transitions when CE is Low. The TC output is High when Q3 and Q0 are High and Q2 and Q1 are Low.

Larger counters are created by connecting the count enable out (CEO) output of the first stage to the CE input of the next stage and connecting the R and clock inputs in parallel. CEO is active (High) when TC and CE are High. The maximum length of the counter is determined by the accumulated CE-to-TC propagation delays versus the clock period. The clock period must be greater than n(tCE-TC) where “n” is the number of stages and “tCE-TC” is the CE-to-TC propagation delay of each stage.

The default initial state of all flip-flops is zero. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

Inputs			Outputs					
R	**CE**	**C**	**Q3**	**Q2**	**Q1**	**Q0**	**TC**	**CEO**
1	X	↑	0	0	0	0	0	0
0	1	↑	---------Increment--------				TC	CEO
0	0	X	--------No Change--------				TC	0
0	1	X	1	0	0	1	1	1

TC = $(Q3 \bullet \overline{Q2} \bullet \overline{Q1} \bullet Q0)$

CEO = $(TC \bullet CE)$

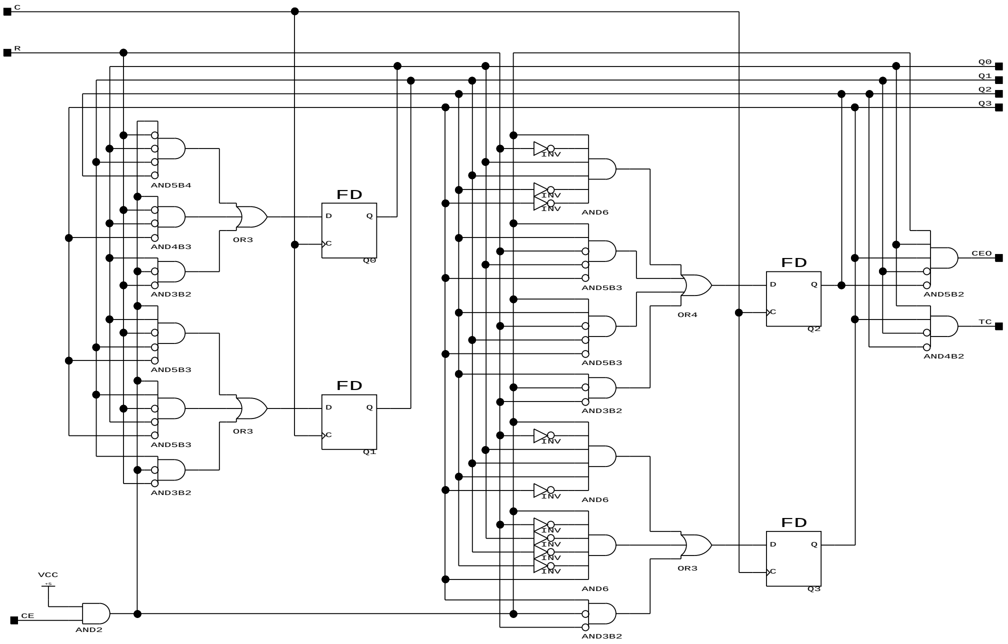
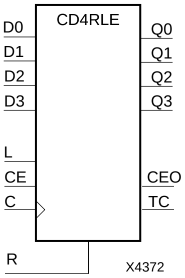


Figure 3-55 CD4RE Implementation (XC7000 and XC9000)

CD4RLE

4-Bit Loadable Cascadable BCD Counter with Clock Enable and Synchronous Reset



XC7000	XC9000
Macro	Macro

CD4RLE is a 4-stage, 4-bit, synchronous, loadable, resettable, binary-coded-decimal (BCD) counter. The synchronous reset input (R) is the highest priority input. When R is High, all other inputs are ignored and the data (Q3 – Q0) and terminal count (TC) outputs go to logic level zero on the Low-to-High clock transitions. The data on the D3 – D0 inputs is loaded into the counter when the load enable input (L) is High during the Low-to-High clock (C) transition. The outputs (Q3 – Q0) increment when the clock enable input (CE) is High during the Low-to-High clock transition. The counter ignores clock transitions when CE is Low. The TC output is High when Q3 and Q0 are High and Q2 and Q1 are Low.

Larger counters are created by connecting the count enable out (CEO) output of the first stage to the CE input of the next stage and connecting the R, L, and C inputs in parallel. CEO is active (High) when TC and CE are High. The maximum length of the counter is determined by the accumulated CE-to-TC propagation delays versus the clock period. The clock period must be greater than n(tCE-TC), where “n” is the number of stages and “tCE-TC” is the CE-to-TC propagation delay of each stage.

The default initial state of all flip-flops is zero. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

Inputs					Outputs					
R	**L**	**CE**	**D3 – D0**	**C**	**Q3**	**Q2**	**Q1**	**Q0**	**TC**	**CEO**
1	X	X	X	↑	0	0	0	0	0	0
0	1	X	D3 – D0	↑	d3	d2	d1	d0	TC	CEO
0	0	1	X	↑	----------Increment---------				TC	CEO
0	0	0	X	X	--------No Change---------				TC	0
0	0	1	X	X	1	0	0	1	1	1

TC = $(Q3 \bullet \overline{Q2} \bullet \overline{Q1} \bullet Q0)$

CEO = $(TC \bullet CE)$

dn = state of referenced input one set-up time prior to active clock transition

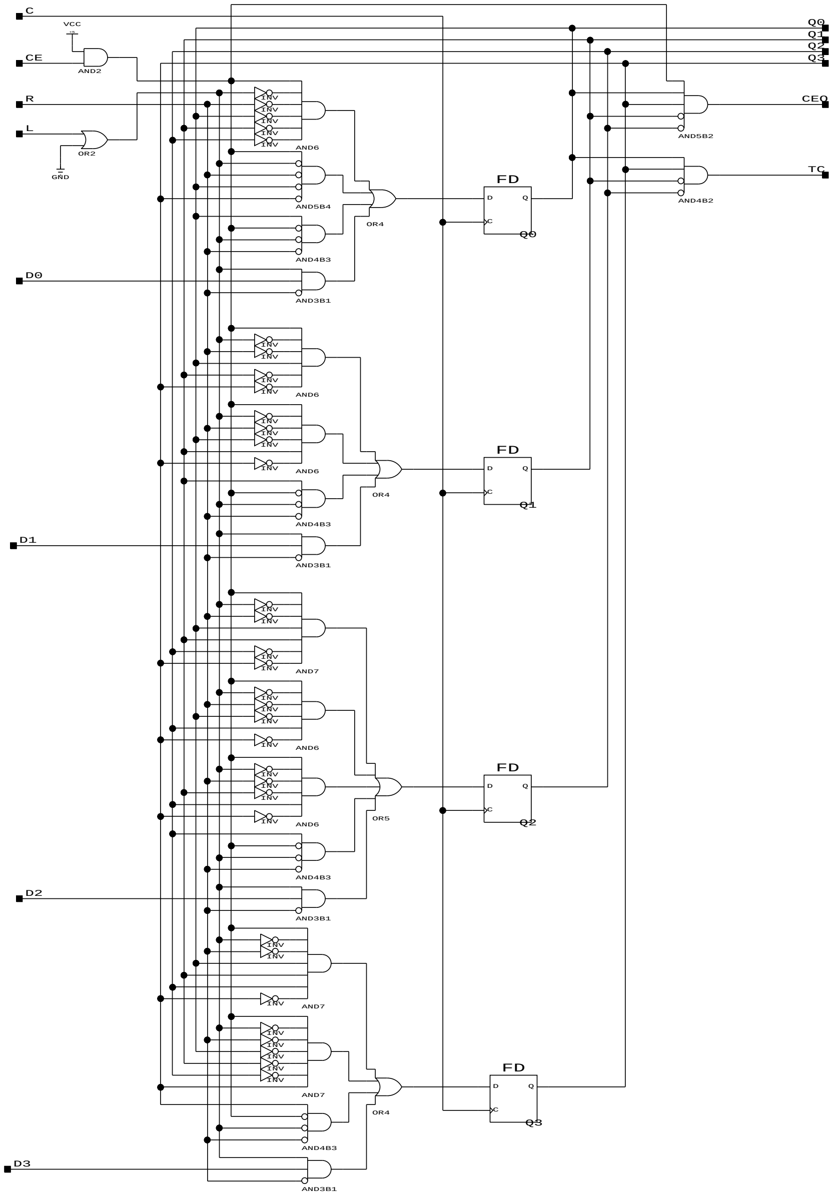
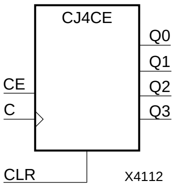


Figure 3-56 CD4RLE Implementation (XC7000 and XC9000)

CJ4CE

4-Bit Johnson Counter with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

CJ4CE is a clearable Johnson / shift counter. The asynchronous clear (CLR) input, when High, overrides all other inputs and causes the data outputs (Q3 – Q0) to go to logic level zero, independent of clock (C) transitions. The counter increments (shifts Q0 to Q1, Q1 to Q2,and so forth) when the clock enable input (CE) is High during the Low-to-High clock transition. Clock transitions are ignored when CE is Low. The Q3 output is inverted and fed back to input Q0 to provide continuous counting operation.

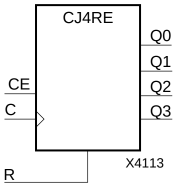
The default initial state of all flip-flops is zero.

Inputs			Outputs			
CLR	**CE**	**C**	**Q0**	**Q1**	**Q2**	**Q3**
1	X	X	0	0	0	0
0	0	X	--------No Change--------			
0	1	↑	$\overline{q3}$	q0	q1	q2

qn = state of referenced output one set-up time prior to active clock transition

CJ4RE

4-Bit Johnson Counter with Clock Enable and Synchronous Reset



XC7000	XC9000
Macro	Macro

CJ4RE is a resettable Johnson / shift counter. The synchronous reset (R) input, when High, overrides all other inputs causes the data outputs (Q3 – Q0) to go to logic level zero during the Low-to-High clock (C) transition. The counter increments (shifts Q0 to Q1, Q1 to Q2, and so forth) when the clock enable input (CE) is High during the Low-to-High clock transition. Clock transitions are ignored when CE is Low. The Q3 output is inverted and fed back to input Q0 to provide continuous counting operation.

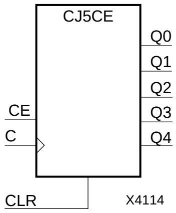
The default initial state of all flip-flops is zero.

Inputs			Outputs			
R	**CE**	**C**	**Q0**	**Q1**	**Q2**	**Q3**
1	X	↑	0	0	0	0
0	0	X	--------No Change--------			
0	1	↑	$\overline{q3}$	q0	q1	q2

qn = state of referenced output one set-up time prior to active clock transition

CJ5CE

5-Bit Johnson Counter with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

CJ5CE is a clearable Johnson / shift counter. The asynchronous clear (CLR) input, when High, overrides all other inputs and causes the data outputs (Q4 – Q0) to go to logic level zero, independent of clock (C) transitions. The counter increments (shifts Q0 to Q1, Q1 to Q2, and so forth) when the clock enable input (CE) is High during the Low-to-High clock transition. Clock transitions are ignored when CE is Low. The Q4 output is inverted and fed back to input Q0 to provide continuous counting operation.

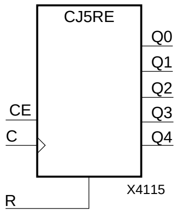
The default initial state of all flip-flops is zero.

Inputs			Outputs				
CLR	CE	C	Q0	Q1	Q2	Q3	Q4
1	X	X	0	0	0	0	0
0	0	X	------------No Change------------				
0	1	↑	$\overline{q4}$	q0	q1	q2	q3

qn = state of referenced output one set-up time prior to active clock transition

CJ5RE

5-Bit Johnson Counter with Clock Enable and Synchronous Reset



XC7000	XC9000
Macro	Macro

CJ5RE is a resettable Johnson / shift counter. The synchronous reset (R) input, when High, overrides all other inputs and causes the data outputs (Q4 – Q0) to go to logic zero during the Low-to-High clock transition. The counter increments (shifts Q0 to Q1, Q1 to Q2, and so forth) when the clock enable input (CE) is High during the Low-to-High clock transition. Clock transitions are ignored when CE is Low. The Q4 output is inverted and fed back to input Q0 to provide continuous counting operation.

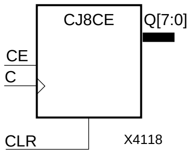
The default initial state of all flip-flops is zero.

Inputs			Outputs				
R	CE	C	Q0	Q1	Q2	Q3	Q4
1	X	↑	0	0	0	0	0
0	0	X	------------No Change------------				
0	1	↑	$\overline{q4}$	q0	q1	q2	q3

qn = state of referenced output one set-up time prior to active clock transition

CJ8CE

8-Bit Johnson Counter with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

CJ8CE is a clearable Johnson / shift counter. The asynchronous clear (CLR) input, when High, overrides all other inputs and causes the data outputs (Q7 – Q0) to go to logic level zero, independent of clock (C) transitions. The counter increments (shifts Q0 to Q1, Q1 to Q2, and so forth) when the clock enable input (CE) is High during the Low-to-High clock transition. Clock transitions are ignored when CE is Low. The Q7 output is inverted and fed back to input Q0 to provide continuous counting operation.

The default initial state of all flip-flops is zero.

Inputs			Outputs	
CLR	**CE**	**C**	**Q0**	**Q1 – Q7**
1	X	X	0	0
0	0	X	--No Change--	
0	1	↑	$\overline{q7}$	q0 – q6

qn = state of referenced output one set-up time prior to active clock transition

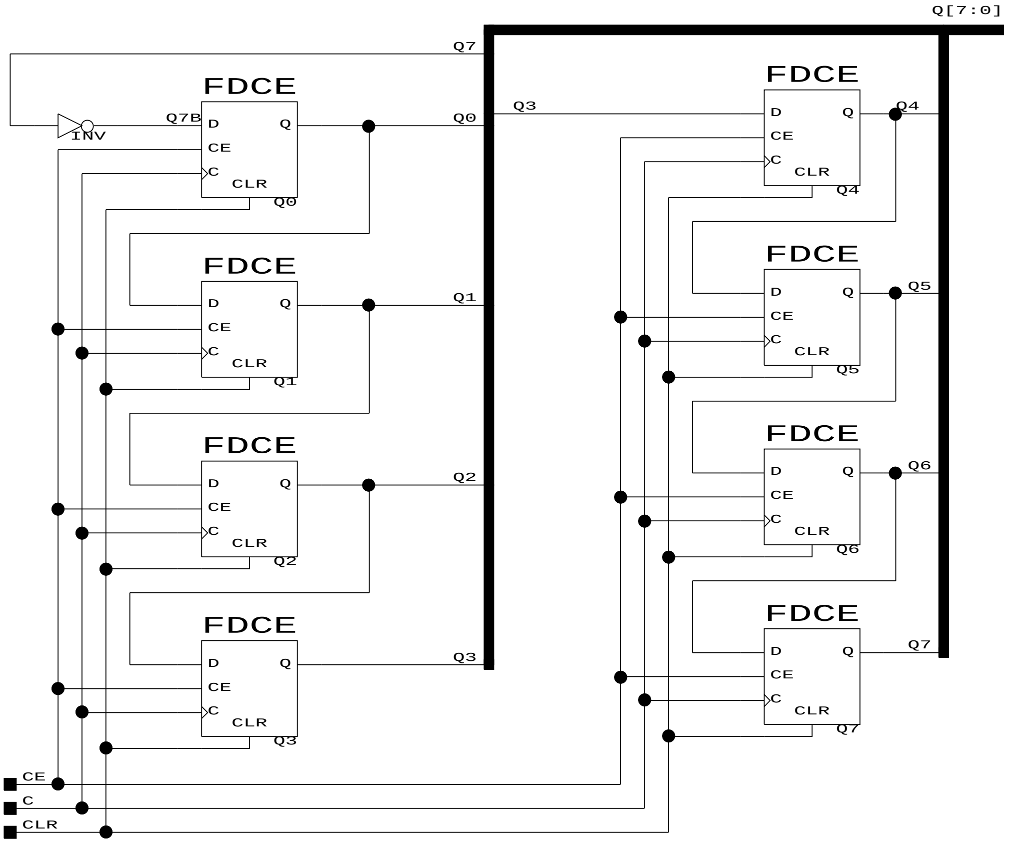
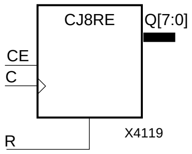


Figure 3-57 CJ8CE Implementation (XC7000 and XC9000)

CJ8RE

8-Bit Johnson Counter with Clock Enable and Synchronous Reset



XC7000	XC9000
Macro	Macro

CJ8RE is a resettable Johnson / shift counter. The synchronous reset (R) input, when High, overrides all other inputs and causes the data outputs (Q7 – Q0) to go to logic level zero during the Low-to-High clock transition. The counter increments (shifts Q0 to Q1, Q1 to Q2, and so forth) when the clock enable input (CE) is High during the Low-to-High clock transition. Clock transitions are ignored when CE is Low. The Q7 output is inverted and fed back to input Q0 to provide continuous counting operation.

The default initial state of all flip-flops is zero.

Inputs			Outputs	
R	CE	C	Q0	Q1 – Q7
1	X	↑	0	0
0	0	X	--No Change--	
0	1	↑	$\overline{q7}$	q0 – q6

qn = state of referenced output one set-up time prior to active clock transition

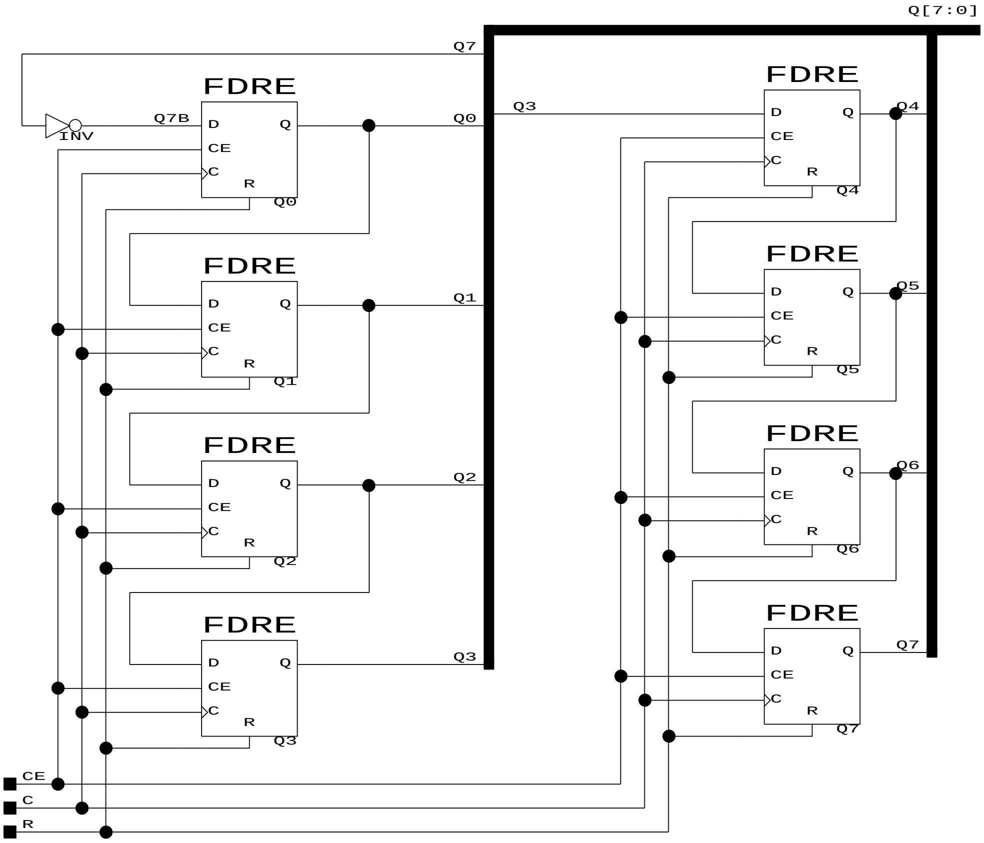
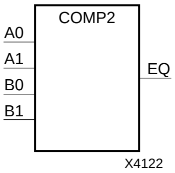


Figure 3-58   CJ8RE Implementation (XC7000 and XC9000)

COMP2

2-Bit Identity Comparator

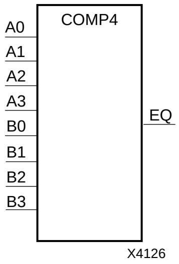


XC7000	XC9000
Macro	Macro

The equal output (EQ) of the COMP2 2-bit, identity comparator is High when the two words A1 – A0 and B1 – B0 are equal. Equality is determined by a bit comparison of the two words. When any two of the corresponding bits from each word are not the same, the EQ output is Low.

COMP4

4-Bit Identity Comparator

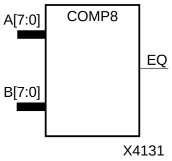


XC7000	XC9000
Macro	Macro

The equal output (EQ) of the COMP4 4-bit, identity comparator is High when the two words A3 – A0 and B3 – B0 are equal. Equality is determined by a bit comparison of the two words. When any two of the corresponding bits from each word are not the same, the EQ output is Low.

COMP8

8-Bit Identity Comparator



XC7000	XC9000
Macro	Macro

The equal output (EQ) of the COMP8 8-bit, identity comparator is High when the two words A7 – A0 and B7 – B0 are equal. Equality is determined by a bit comparison of the two words. When any two of the corresponding bits from each word are not the same, the EQ output is Low.

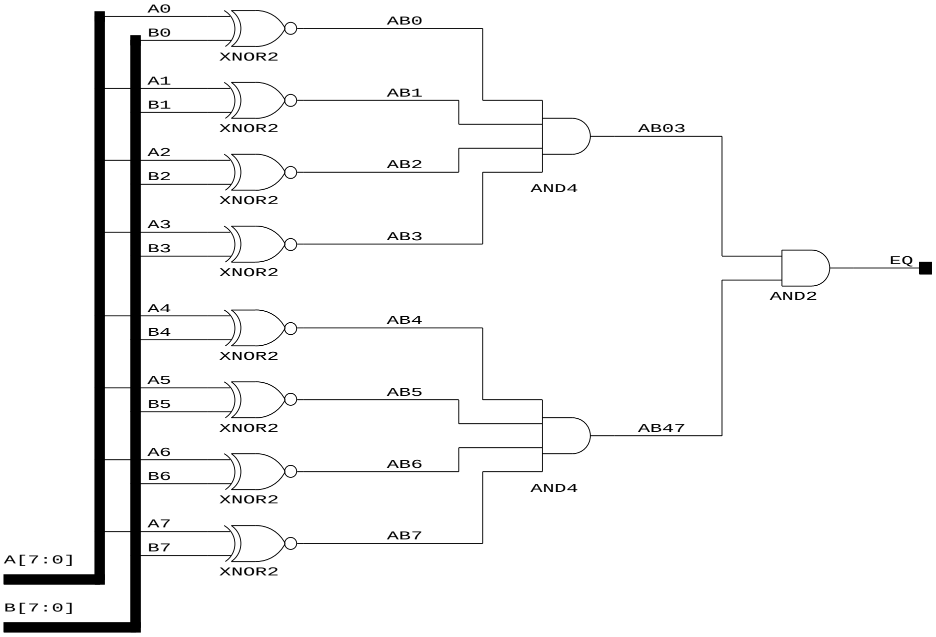
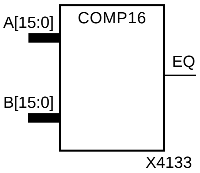


Figure 3-59 COMP8 Implementation (XC7000 and XC9000)

COMP16

16-Bit Identity Comparator

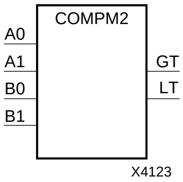


XC7000	XC9000
Macro	Macro

The equal output (EQ) of the COMP16 16-bit, identity comparator is High when the two words A15 – A0 and B15 – B0 are equal. Equality is determined by a bit comparison of the two words. When any two of the corresponding bits from each word are not the same, the EQ output is Low.

COMPM2

2-Bit Magnitude Comparator



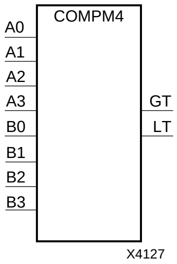
XC7000	XC9000
Macro	Macro

COMPM2 is a 2-bit, magnitude comparator that compares two positive binary-weighted words A1 – A0 and B1 – B0, where A1 and B1 are the most significant bits. The greater-than output (GT) is High when A>B, and the less-than output (LT) is High when A<B. When the two words are equal, both GT and LT are Low. Equality can be measured with this macro by comparing both outputs with a NOR gate.

Inputs				Outputs	
A1	B1	A0	B0	GT	LT
0	0	0	0	0	0
0	0	1	0	1	0
0	0	0	1	0	1
0	0	1	1	0	0
1	1	0	0	0	0
1	1	1	0	1	0
1	1	0	1	0	1
1	1	1	1	0	0
1	0	X	X	1	0
0	1	X	X	0	1

COMPM4

4-Bit Magnitude Comparator



XC7000	XC9000
Macro*	Macro

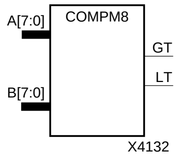
* not supported for XC7336 designs

COMPM4 is a 4-bit, magnitude comparator that compares two positive binary-weighted words A3 – A0 and B3 – B0, where A3 and B3 are the most significant bits. The greater-than output (GT) is High when A>B, and the less-than output (LT) is High when A<B. When the two words are equal, both GT and LT are Low. Equality can be measured with this macro by comparing both outputs with a NOR gate.

Inputs				Outputs	
A3, B3	**A2, B2**	**A1, B1**	**A0, B0**	**GT**	**LT**
A3>B3	X	X	X	1	0
A3<B3	X	X	X	0	1
A3=B3	A2>B2	X	X	1	0
A3=B3	A2<B2	X	X	0	1
A3=B3	A2=B2	A1>B1	X	1	0
A3=B3	A2=B2	A1<B1	X	0	1
A3=B3	A2=A2	A1=B1	A0>B0	1	0
A3=B3	A2=B2	A1=B1	A0<B0	0	1
A3=B3	A2=B2	A1=B1	A0=B0	0	0

COMPM8

8-Bit Magnitude Comparator



XC7000	XC9000
Macro*	Macro

* not supported for XC7336 designs

COMPM8 is an 8-bit, magnitude comparator that compares two positive binary-weighted words A7 – A0 and B7 – B0, where A7 and B7 are the most significant bits. The greater-than output (GT) is High when A>B, and the less-than output (LT) is High when A<B. When the two words are equal, both GT and LT are Low. Equality can be measured with this macro by comparing both outputs with a NOR gate. Refer to the “COMPM4” section earlier in this chapter for a representative truth table.

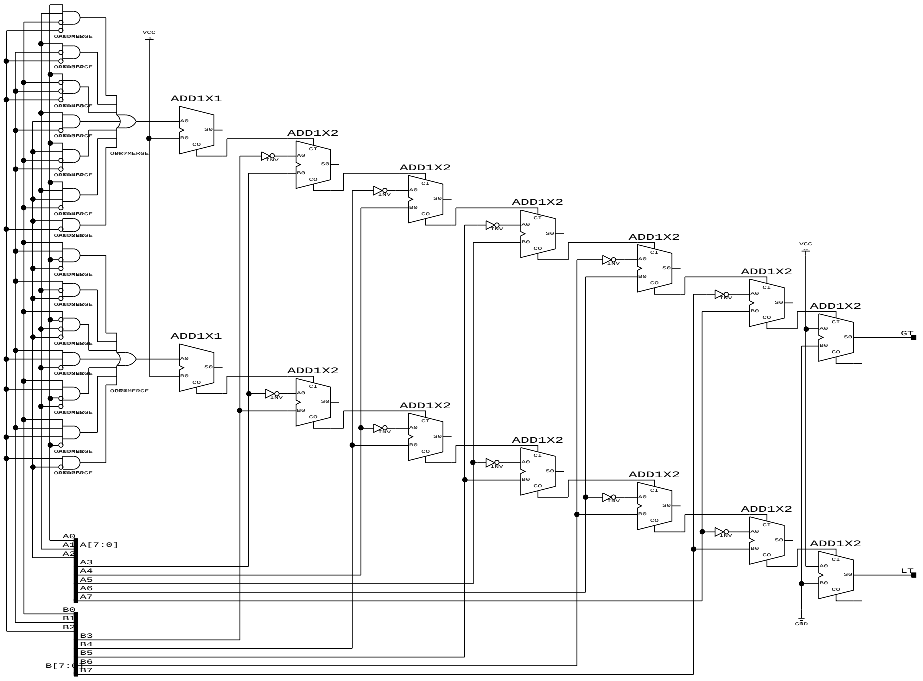


Figure 3-60 COMPM8 Implementation (XC7000)

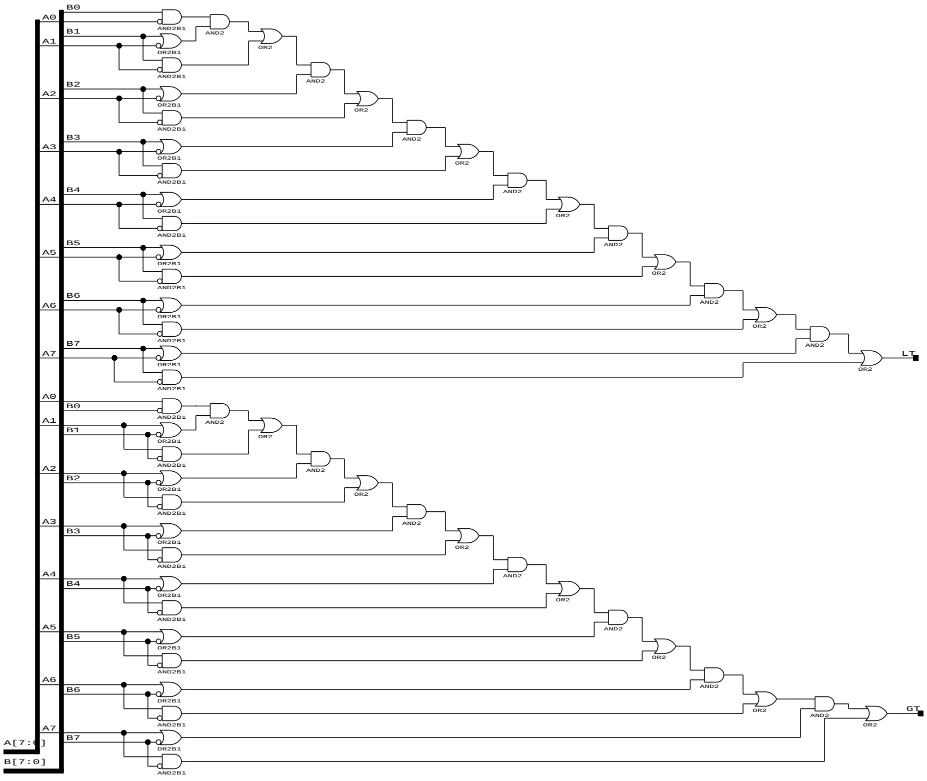
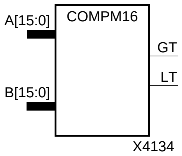


Figure 3-61 COMPM8 Implementation (XC9000)

COMPM16

16-Bit Magnitude Comparator

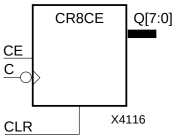


XC7000	XC9000
NA	Macro

COMPM16 is a 16-bit, magnitude comparator that compares two positive binary-weighted words A15 – A0 and B15 – B0, where A15 and B15 are the most significant bits. The greater-than output (GT) is High when A>B, and the less-than output (LT) is High when A<B. When the two words are equal, both GT and LT are Low. Equality can be measured with this macro by comparing both outputs with a NOR gate. Refer to the “COMPM4” section earlier in this chapter for a representative truth table.

CR8CE

8-Bit Negative-Edge Binary Ripple Counter with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

CR8CE is an 8-bit, cascadable, clearable, binary, ripple counter. The asynchronous clear (CLR), when High, overrides all other inputs and causes the outputs (Q7 – Q0) to go to logic level zero. The counter increments when the clock enable input (CE) is High during the High-to-Low clock (C) transition. The counter ignores clock transitions when CE is Low.

Larger counters can be created by connecting the Q7 output of the first stage to the clock input of the next stage. CLR and CE inputs are connected in parallel. The clock period is not affected by the overall length of a ripple counter. The overall clock-to-output propagation is *n*(TC - Q), where *n* is the number of stages and TC - Q is the C-to-Q7 propagation delay of each stage.

The default initial state of all flip-flops is zero. For XC7000, the clock (C) cannot be driven by a FastCLK (BUFG).

Inputs			Outputs
CLR	**CE**	**C**	**Q7 – Q0**
1	X	X	0
0	0	X	No Chg
0	1	↓	Inc

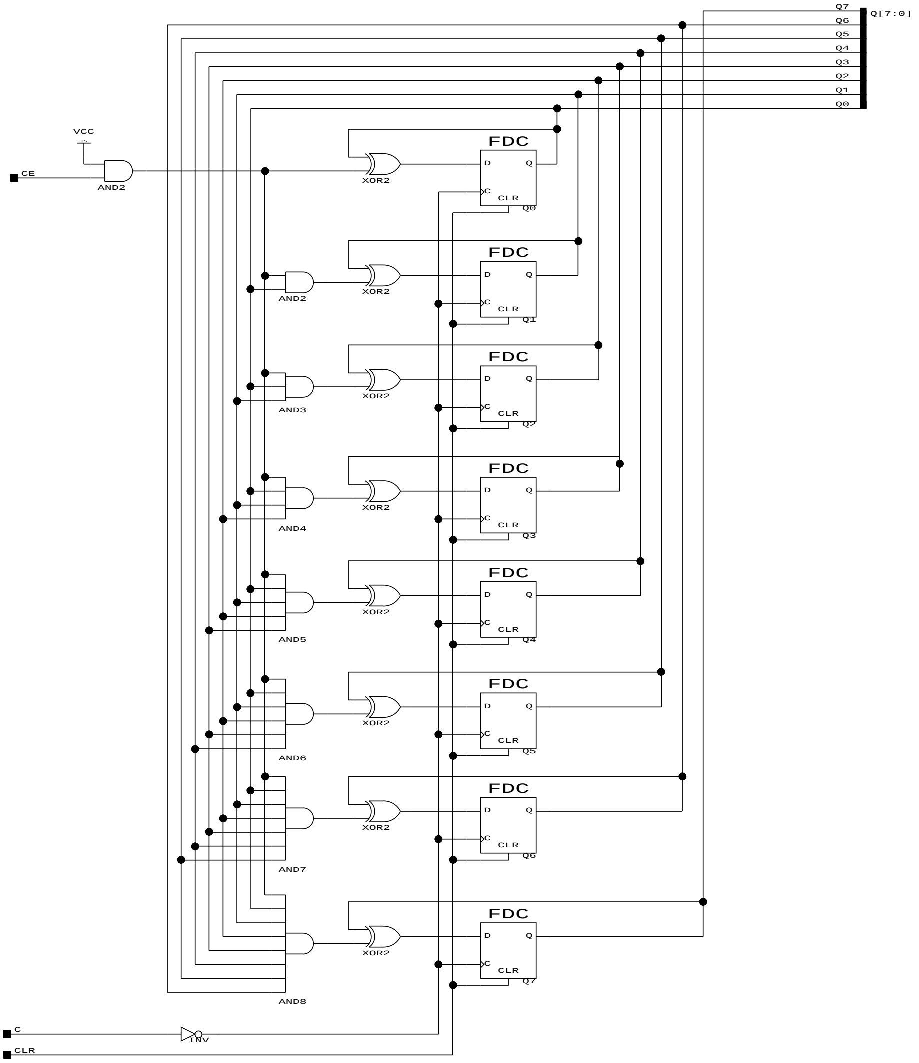
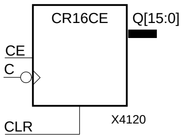


Figure 3-62 CR8CE Implementation (XC7000 and XC9000)

CR16CE

16-Bit Negative-Edge Binary Ripple Counter with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

CR16CE is a 16-bit, cascadable, clearable, binary, ripple counter. The asynchronous clear (CLR), when High, overrides all other inputs and causes the outputs (Q15 – Q0) to go to logic level zero. The counter increments when the clock enable input (CE) is High during the High-to-Low clock (C) transition. The counter ignores clock transitions when CE is Low.

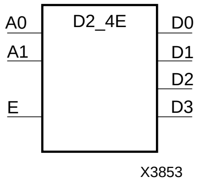
Larger counters can be created by connecting the Q15 output of the first stage to the clock input of the next stage. CLR and CE inputs are connected in parallel. The clock period is not affected by the overall length of a ripple counter. The overall clock-to-output propagation is *n*(TC - Q), where *n* is the number of stages and TC - Q is the C-to-Q15 propagation delay of each stage.

The default initial state of all flip-flops is zero. For XC7000, the clock (C) cannot be driven by a FastCLK (BUFG).

Inputs			Outputs
CLR	CE	C	Q15 – Q0
1	X	X	0
0	0	X	No Chg
0	1	↓	Inc

D2_4E

2- to 4-Line Decoder/Demultiplexer with Enable



XC7000	XC9000
Macro	Macro

When the enable (EN) input of the D2_4E decoder / demultiplexer is High, one of four active-High outputs (D3 – D0) is selected with a 2-bit binary address (A1 – A0) input. The non-selected outputs are Low. Also, when the EN input is Low, all outputs are Low. In demultiplexer applications, the EN input is the data input.

Inputs			Outputs			
A1	**A0**	**E**	**D3**	**D2**	**D1**	**D0**
X	X	0	0	0	0	0
0	0	1	0	0	0	1
0	1	1	0	0	1	0
1	0	1	0	1	0	0
1	1	1	1	0	0	0

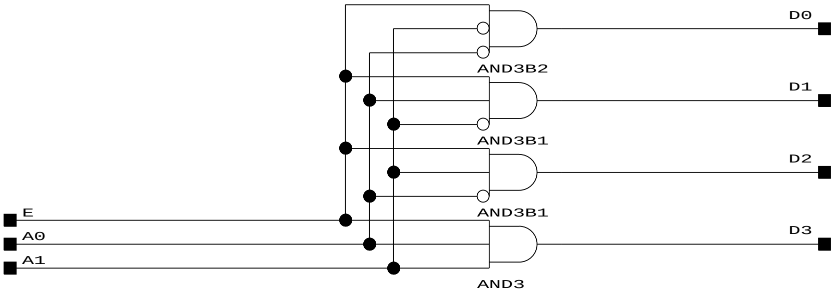
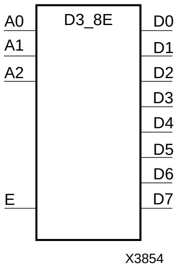


Figure 3-63 D2_4E Implementation (XC7000 and XC9000)

D3_8E

3- to 8-Line Decoder/Demultiplexer with Enable



XC7000	XC9000
Macro	Macro

When the enable (EN) input of the D3_8E decoder / demultiplexer is High, one of eight active-High outputs (D7 – D0) is selected with a 3-bit binary address (A2 – A0) input. The non-selected outputs are Low. Also, when the EN input is Low, all outputs are Low. In demultiplexer applications, the EN input is the data input.

Inputs				Outputs							
A2	A1	A0	E	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	1
0	0	1	1	0	0	0	0	0	0	1	0
0	1	0	1	0	0	0	0	0	1	0	0
0	1	1	1	0	0	0	0	1	0	0	0
1	0	0	1	0	0	0	1	0	0	0	0
1	0	1	1	0	0	1	0	0	0	0	0
1	1	0	1	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0

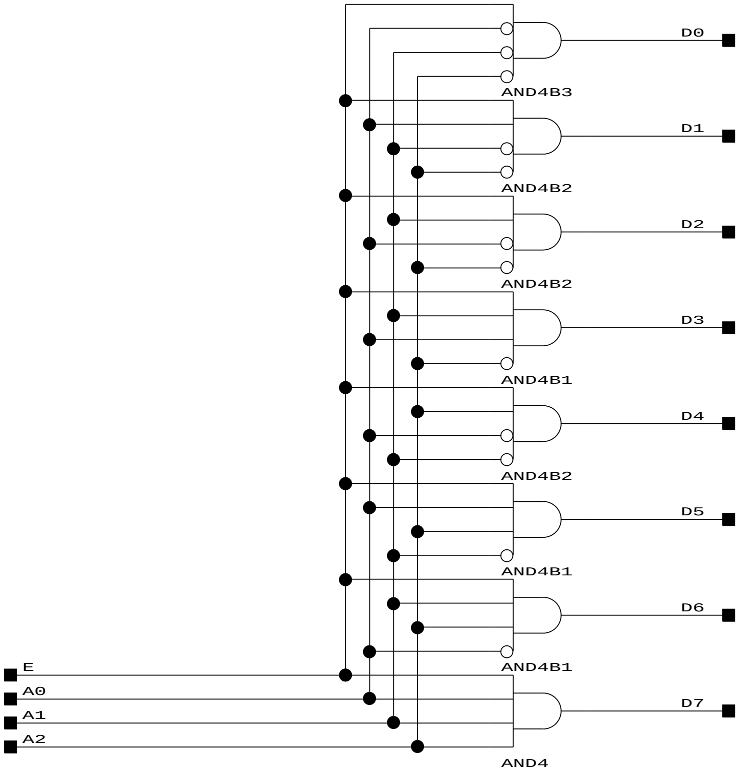
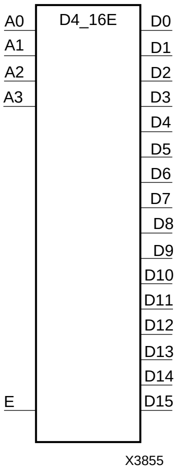


Figure 3-64   D3_8E Implementation (XC7000 and XC9000)

D4_16E

4- to 16-Line Decoder/Demultiplexer with Enable



XC7000	XC9000
Macro	Macro

When the enable (EN) input of the D4_16E decoder / demultiplexer is High, one of 16 active-High outputs (D15 – D0) is selected with a 4-bit binary address (A3 – A0) input. The non-selected outputs are Low. Also, when the EN input is Low, all outputs are Low. In demultiplexer applications, the EN input is the data input. Refer to “D3_8E” for truth table derivation.

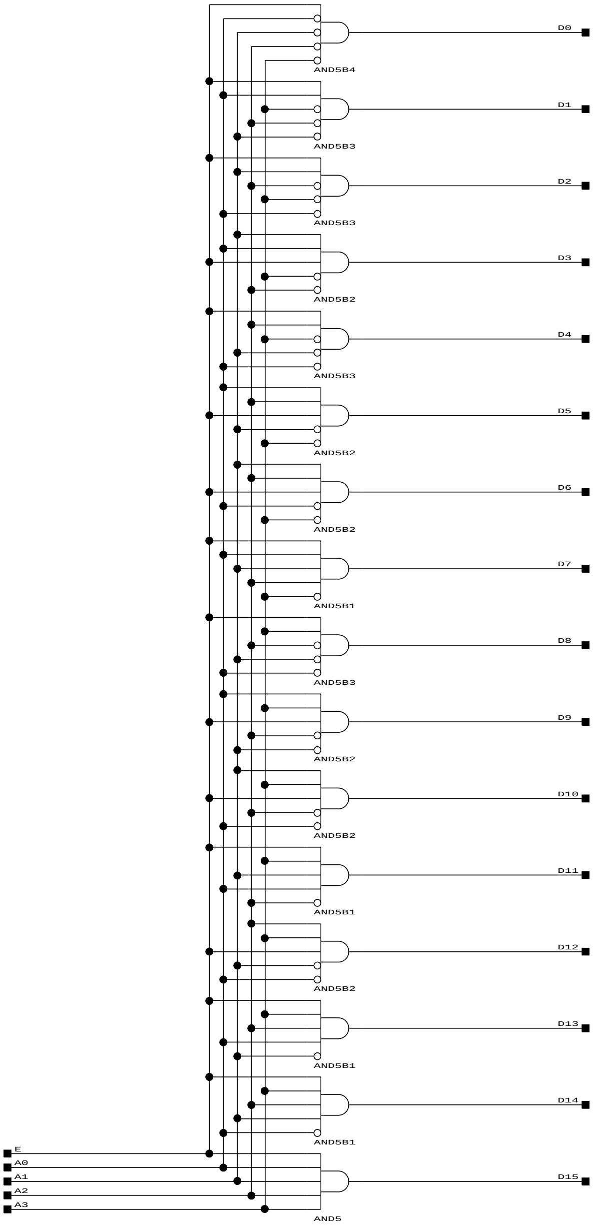
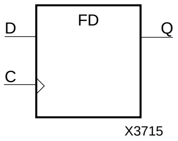


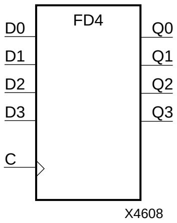
Figure 3-65 D4_16E Implementation (XC7000 and XC9000)

FD, FD4, FD8, and FD16

Single and Multiple D Flip-Flops

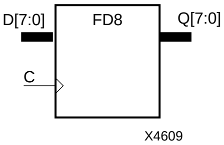


Element	XC7000	XC9000
FD FD4, FD8, FD16	Macro	Macro

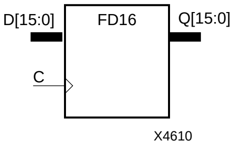


FD is a single D-type flip-flop with data input (D) and data output (Q). FD4, FD8, and FD16 are 4-bit, 8-bit, and 16-bit registers, each with a common clock (C). The data on the D inputs is loaded into the flip-flop during the Low-to-High clock (C) transition.

The default initial state of all flip-flops is zero.



Inputs		Outputs
D	C	Q
0	↑	0
1	↑	1



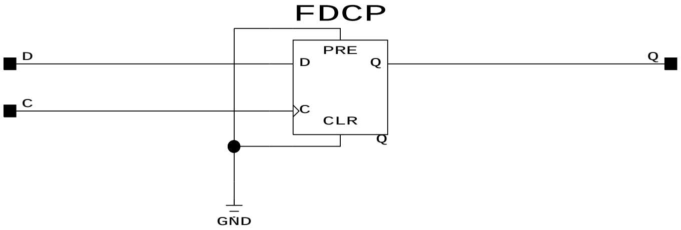


Figure 3-66 FD Implementation (XC7000 and XC9000)

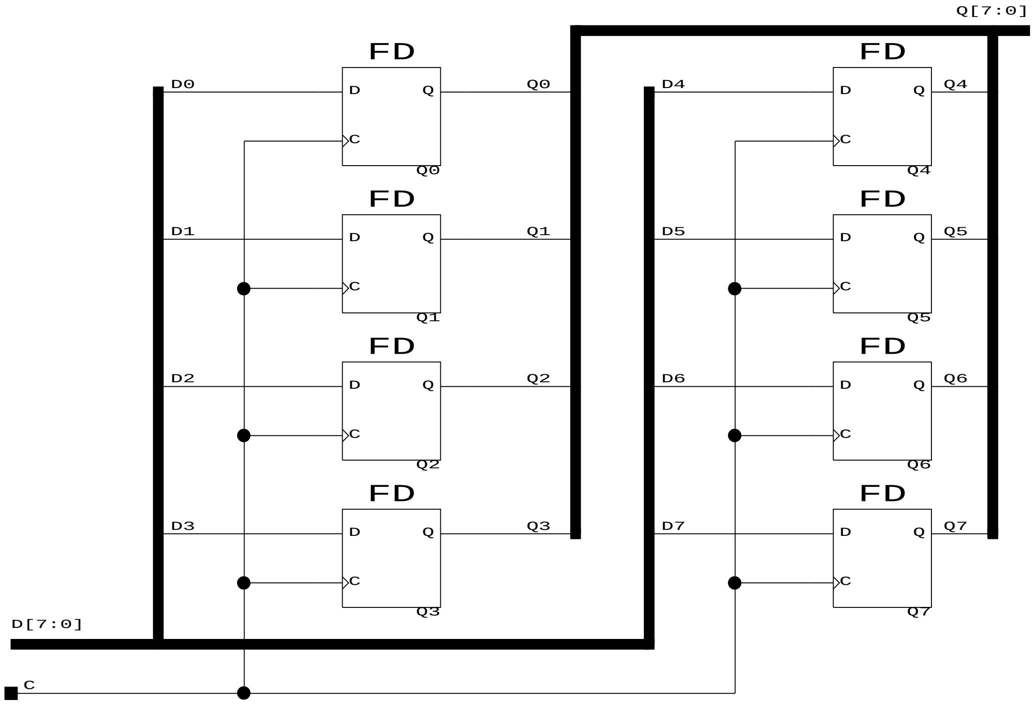
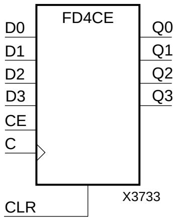


Figure 3-67 FD8 Implementation (XC7000 and XC9000)

FD4CE

4-Bit Data Register with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

When clock enable (CE) is High, and asynchronous clear (CLR) is Low, the data on the four data inputs (D3 – D0) of FD4CE is transferred to the corresponding data outputs (Q3 – Q0) during the Low-to-High clock (C) transition. When CLR is High, it overrides all other inputs and resets the data outputs (Q3 – Q0) Low. When CE is Low, clock transitions are ignored.

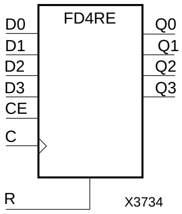
The default initial state of all flip-flops is zero.

Inputs				Outputs
CLR	CE	D3 – D0	C	Q3 – Q0
1	X	X	X	0
0	0	X	X	No Chg
0	1	Dn	↑	dn

dn = state of corresponding input one set-up time prior to active clock transition

FD4RE

4-Bit Data Register with Clock Enable and Synchronous Reset



XC7000	XC9000
Macro	Macro

When the clock enable (CE) input is High, and the synchronous reset (R) input is Low, the data on the four data inputs (D3 – D0) of FD4RE is transferred to the corresponding data outputs (Q3 – Q0) during the Low-to-High clock (C) transition. When R is High, it overrides all other inputs and resets the data outputs (Q3 – Q0) Low on the Low-to-High clock transition. When CE is Low, clock transitions are ignored.

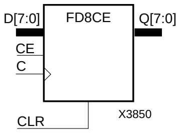
The default initial state of all flip-flops is zero.

Inputs				Outputs
R	CE	D3 – D0	C	Q3 – Q0
1	X	X	↑	0
0	0	X	X	No Chg
0	1	Dn	↑	dn

dn = state of corresponding input one set-up time prior to active clock transition

FD8CE

8-Bit Data Register with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

When clock enable (CE) is High, and asynchronous clear (CLR) is Low, the data on the eight data inputs (D7 – D0) of FD8CE is transferred to the corresponding data outputs (Q7 – Q0) during the Low-to-High clock (C) transition. When CLR is High, it overrides all other inputs and resets the data outputs (Q7 – Q0) Low. When CE is Low, clock transitions are ignored.

The default initial state of all flip-flops is zero.

Inputs				Outputs
CLR	CE	D7 – D0	C	Q7 – Q0
1	X	X	X	0
0	0	X	X	No Chg
0	1	Dn	↑	dn

dn = state of corresponding input one set-up time prior to active clock transition

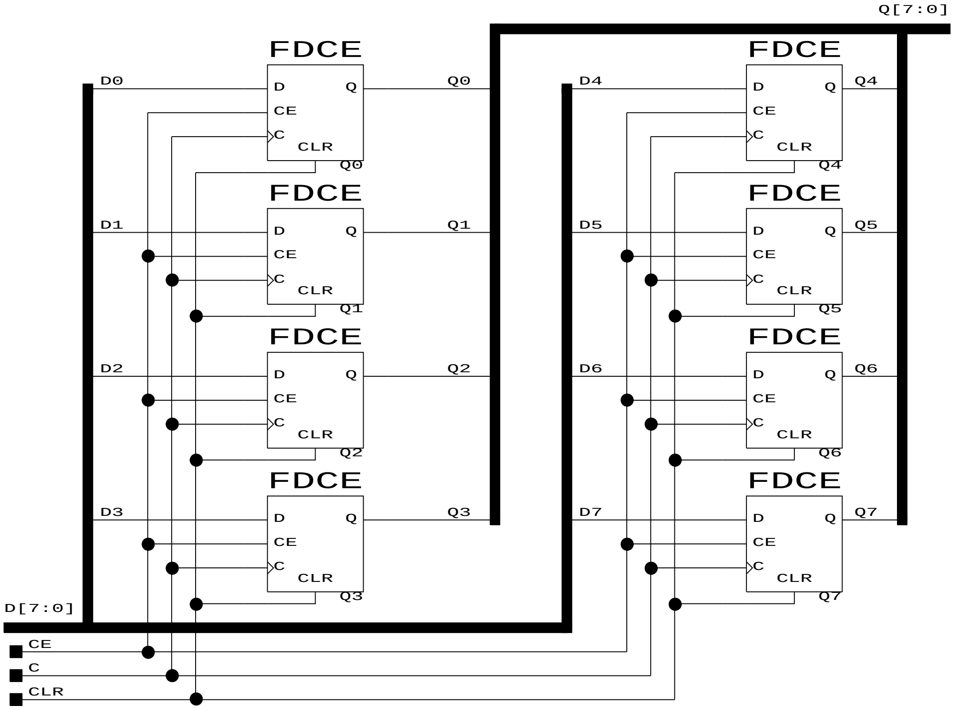
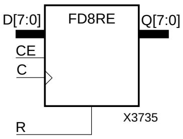


Figure 3-68 FD8CE Implementation (XC7000 and XC9000)

FD8RE

8-Bit Data Register with Clock Enable and Synchronous Reset



XC7000	XC9000
Macro	Macro

When the clock enable (CE) input is High, and the synchronous reset (R) input is Low, the data on the eight data inputs (D7 – D0) of FD8RE is transferred to the corresponding data outputs (Q7 – Q0) during the Low-to-High clock (C) transition. When R is High, it overrides all other inputs and resets the data outputs (Q7 – Q0) Low on the Low-to-High clock transition. When CE is Low, clock transitions are ignored.

The default initial state of all flip-flops is zero.

Inputs				Outputs
R	CE	D7 – D0	C	Q7 – Q0
1	X	X	↑	0
0	0	X	X	No Chg
0	1	Dn	↑	dn

dn = state of corresponding input one set-up time prior to active clock transition

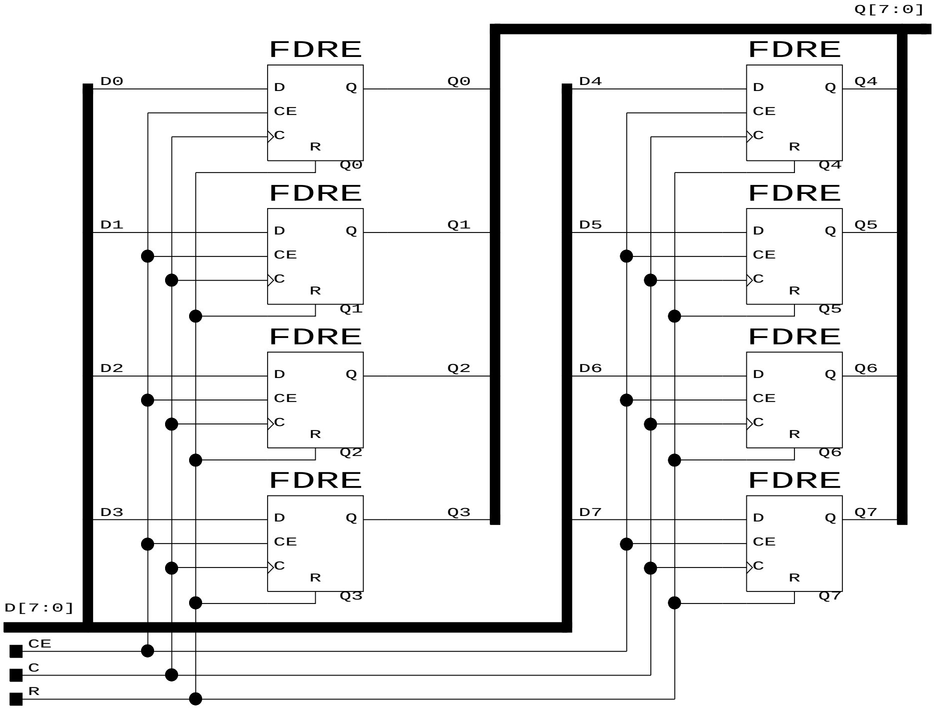
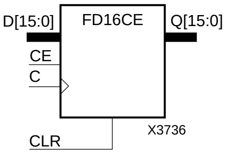


Figure 3-69 FD8RE Implementation (XC7000 and XC9000)

FD16CE

16-Bit Data Register with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

When clock enable (CE) is High, and asynchronous clear (CLR) is Low, the data on the 16 data inputs (D15 – D0) of FD16CE is transferred to the corresponding data outputs (Q15 – Q0) during the Low-to-High clock (C) transition. When CLR is High, it overrides all other inputs and resets the data outputs (Q15 – Q0) Low. When CE is Low, clock transitions are ignored.

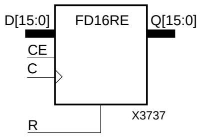
The default initial state of all flip-flops is zero.

Inputs				Outputs
CLR	CE	D15 – D0	C	Q15 – Q0
1	X	X	X	0
0	0	X	X	No Chg
0	1	Dn	↑	dn

dn = state of corresponding input one set-up time prior to active clock transition

FD16RE

16-Bit Data Register with Clock Enable and Synchronous Reset



XC7000	XC9000
Macro	Macro

When the clock enable (CE) input is High, and the synchronous reset (R) input is Low, the data on the 16 data inputs (D15 – D0) of FD16RE is transferred to the corresponding data outputs (Q15 – Q0) during the Low-to-High clock (C) transition. When R is High, it overrides all other inputs and resets the data outputs (Q15 – Q0) Low on the Low-to-High clock transition. When CE is Low, clock transitions are ignored.

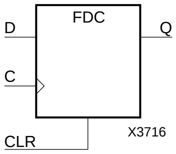
The default initial state of all flip-flops is zero.

Inputs				Outputs
R	**CE**	**D15 – D0**	**C**	**Q15 – Q0**
1	X	X	↑	0
0	0	X	X	No Chg
0	1	Dn	↑	dn

dn = state of corresponding input one set-up time prior to active clock transition

FDC

D Flip-Flop with Asynchronous Clear



XC7000	XC9000
Macro	Macro

FDC is a single D-type flip-flop with data (D) and asynchronous clear (CLR) inputs and data output (Q). The asynchronous CLR, when High, overrides all other inputs and sets the Q output Low. The data on the D input is loaded into the flip-flop when CLR is Low on the Low-to-High clock transition.

The default initial state of the flip-flop is zero.

Inputs			Outputs
CLR	D	C	Q
1	X	X	0
0	1	↑	1
0	0	↑	0

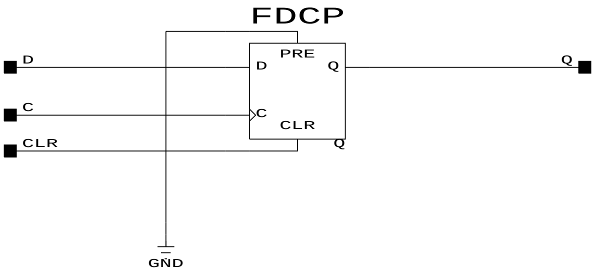
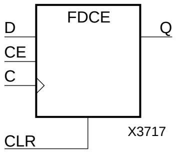


Figure 3-70 FDC Implementation (XC7000 and XC9000)

FDCE

D Flip-Flop with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

When clock enable (CE) is High, and asynchronous clear (CLR) is Low, the data on the data input (D) of FDCE is transferred to the corresponding data output (Q) during the Low-to-High clock (C) transition. When CLR is High, it overrides all other inputs and resets the data output (Q) Low. When CE is Low, clock transitions are ignored.

The default initial state of the flip-flop is zero.

Inputs				Outputs
CLR	**CE**	**D**	**C**	**Q**
1	X	X	X	0
0	0	X	X	No Chg
0	1	1	↑	1
0	1	0	↑	0

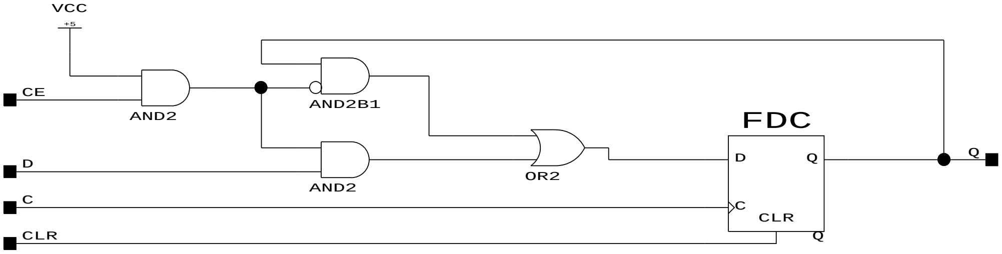
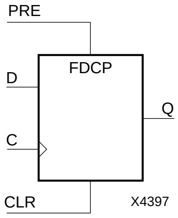


Figure 3-71 FDCE Implementation (XC7000 and XC9000)

FDCP

D Flip-Flop with Asynchronous Preset and Clear



XC7000	XC9000
Primitive	Primitive

FDCP is a single D-type flip-flop with data (D), asynchronous preset (PRE) and clear (CLR) inputs, and data output (Q). The asynchronous PRE, when High, sets the Q output High; CLR, when High, resets the output Low. When both PRE and CLR are active, the flip-flop output is unpredictable. Data on the D input is loaded into the flip-flop when PRE and CLR are Low on the Low-to-High clock (C) transition.

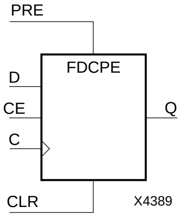
The default initial state of the flip-flop is zero.

Inputs				Outputs
CLR	**PRE**	**D**	**C**	**Q**
1	0	X	X	0
0	1	X	X	1
0	0	0	↑	0
0	0	1	↑	1

Note: For the XC7336, the PRE and CLR inputs cannot both be used.

FDCPE

D Flip-Flop with Clock Enable and Asynchronous Preset and Clear



XC7000	XC9000
Macro	Macro

FDCPE is a single D-type flip-flop with data (D), clock enable (CE), asynchronous preset (PRE), and asynchronous clear (CLR) inputs and data output (Q). The asynchronous PRE, when High, sets the Q output High; CLR, when High, resets the output Low. When both PRE and CLR are active, the flip-flop output is unpredictable. Data on the D input is loaded into the flip-flop when PRE and CLR are Low and CE is High. When CE is Low, the clock transitions are ignored.

The default initial state of the flip-flop is zero.

Inputs					Outputs
CLR	**PRE**	**CE**	**D**	**C**	**Q**
1	0	X	X	X	0
0	1	X	X	X	1
0	0	0	X	X	No Chg
0	0	1	0	↑	0
0	0	1	1	↑	1

Note: For the XC7336, the PRE and CLR inputs cannot both be used.

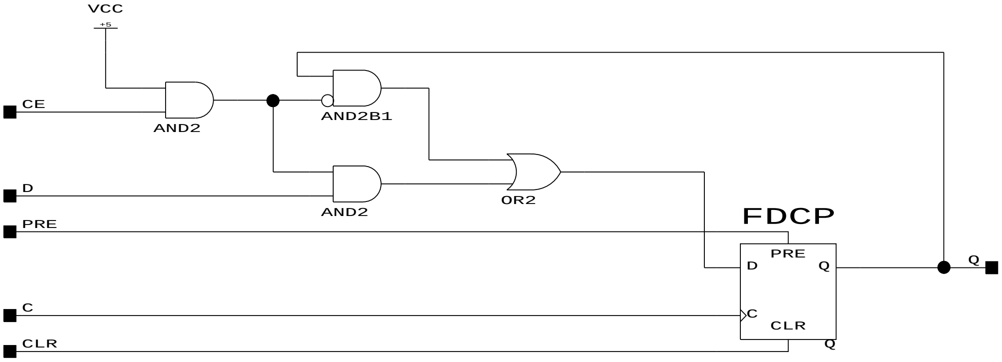
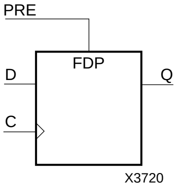


Figure 3-72 FDCPE Implementation (XC7000 and XC9000)

FDP

D Flip-Flop with Asynchronous Preset



XC7000	XC9000
Macro	Macro

FDP is a single D-type flip-flop with data (D) and asynchronous preset (PRE) inputs, and data output (Q). The asynchronous PRE, when High, overrides all other inputs and presets the Q output High. The data on the D input is loaded into the flip-flop when PRE is Low on the Low-to-High clock (C) transition.

The default initial state of the flip-flop is zero.

Inputs			Outputs
PRE	C	D	Q
1	X	X	1
0	↑	1	1
0	↑	0	0

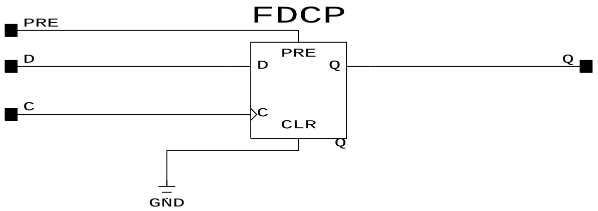
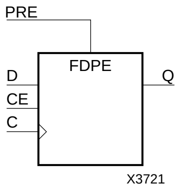


Figure 3-73 FDP Implementation (XC7000 and XC9000)

FDPE

D Flip-Flop with Clock Enable and Asynchronous Preset



XC7000	XC9000
Macro	Macro

FDPE is a single D-type flip-flop with data (D), clock enable (CE), and asynchronous preset (PRE) inputs and data output (Q). The asynchronous PRE, when High, overrides all other inputs and sets the Q output High. Data on the D input is loaded into the flip-flop when PRE is Low and CE is High on the Low-to-High clock (C) transition. When CE is Low, the clock transitions are ignored. The default initial state of the flip-flop is zero.

Inputs				Outputs
PRE	CE	D	C	Q
1	X	X	X	1
0	0	X	X	No Chg
0	1	0	↑	0
0	1	1	↑	1

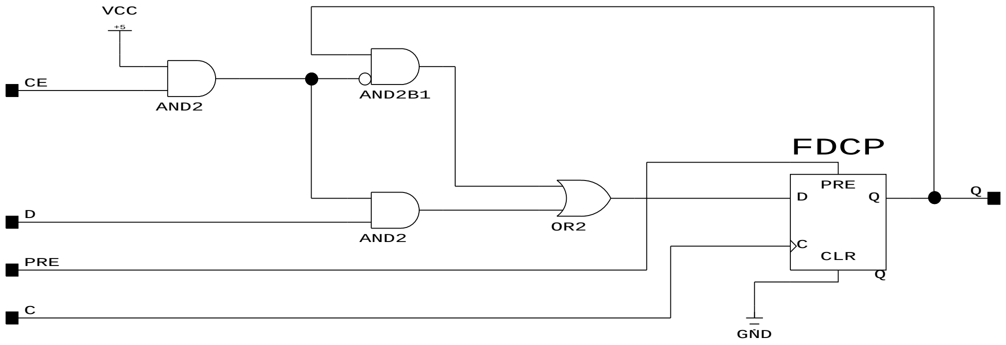
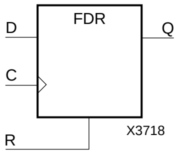


Figure 3-74 FDPE Implementation (XC7000 and XC9000)

FDR

D Flip-Flop with Synchronous Reset



XC7000	XC9000
Macro	Macro

FDR is a single D-type flip-flop with data (D) and synchronous reset (R) inputs and data output (Q). The synchronous reset (R) input, when High, overrides all other inputs and resets the Q output Low on the Low-to-High clock (C) transition. The data on the D input is loaded into the flip-flop when R is Low during the Low-to-High clock transition.

The default initial state of the flip-flop is zero.

Inputs			Outputs
R	D	C	Q
1	X	↑	0
0	1	↑	1
0	0	↑	0

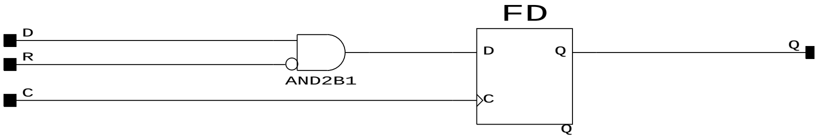
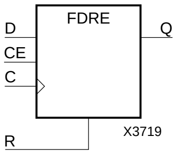


Figure 3-75 FDR Implementation (XC7000 and XC9000)

FDRE

D Flip-Flop with Clock Enable and Synchronous Reset



XC7000	XC9000
Macro	Macro

FDRE is a single D-type flip-flop with data (D), clock enable (CE), and synchronous reset (R) inputs and data output (Q). The synchronous reset (R) input, when High, overrides all other inputs and resets the Q output Low on the Low-to-High clock (C) transition. The data on the D input is loaded into the flip-flop when R is Low and CE is High during the Low-to-High clock transition.

The default initial state of the flip-flop is zero.

Inputs				Outputs
R	CE	D	C	Q
1	X	X	↑	0
0	0	X	X	No Chg
0	1	1	↑	1
0	1	0	↑	0

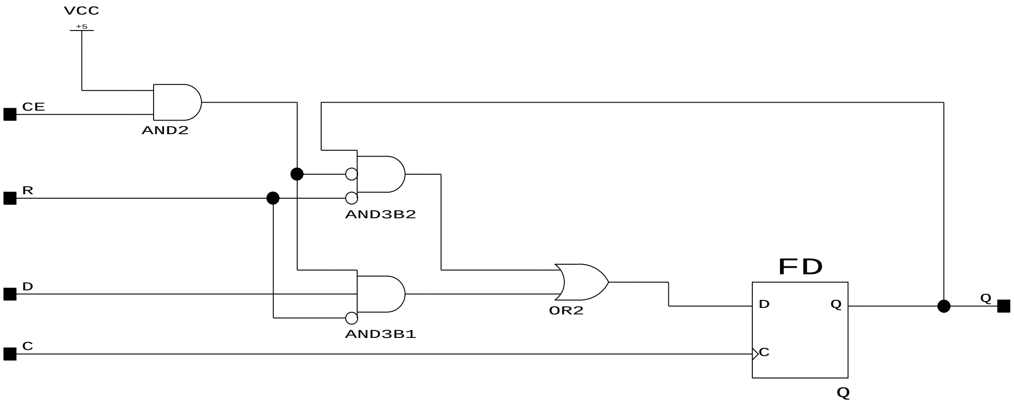
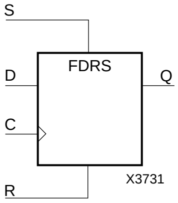


Figure 3-76 FDRE Implementation (XC7000 and XC9000)

FDRS

D Flip-Flop with Synchronous Reset and Synchronous Set



XC7000	XC9000
Macro	Macro

FDRS is a single D-type flip-flop with data (D), synchronous set (S), and synchronous reset (R) inputs and data output (Q). The synchronous reset (R) input, when High, overrides all other inputs and resets the Q output Low during the Low-to-High clock (C) transition. (Reset has precedence over Set.) When S is High and R is Low, the flip-flop is set, output High, during the Low-to-High clock transition. When R and S are Low, data on the (D) input is loaded into the flip-flop during the Low-to-High clock transition.

The default initial state of the flip-flop is zero.

Inputs				Outputs
R	S	D	C	Q
1	X	X	↑	0
0	1	X	↑	1
0	0	1	↑	1
0	0	0	↑	0

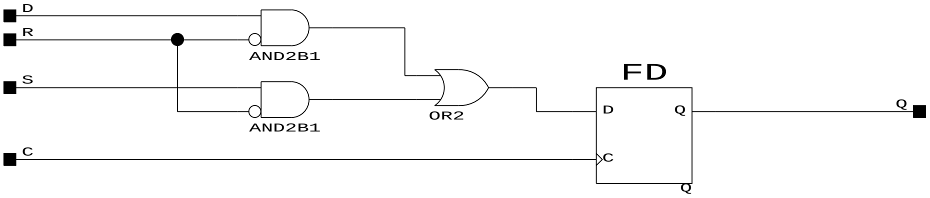
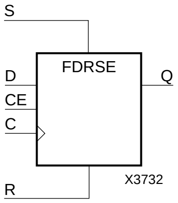


Figure 3-77 FDRS Implementation (XC7000 and XC9000)

FDRSE

D Flip-Flop with Synchronous Reset and Set and Clock Enable



XC7000	XC9000
Macro	Macro

FDRSE is a single D-type flip-flop with synchronous reset (R), synchronous set (S), and clock enable (CE) inputs and data output (Q). The reset (R) input, when High, overrides all other inputs and resets the Q output Low during the Low-to-High clock transition. (Reset has precedence over Set.) When the set (S) input is High and R is Low, the flip-flop is set, output High, during the Low-to-High clock (C) transition. Data on the D input is loaded into the flip-flop when R and S are Low and CE is High during the Low-to-High clock transition.

The default initial state of the flip-flop is zero.

Inputs					Outputs
R	S	CE	D	C	Q
1	X	X	X	↑	0
0	1	X	X	↑	1
0	0	0	X	X	No Chg
0	0	1	1	↑	1
0	0	1	0	↑	0

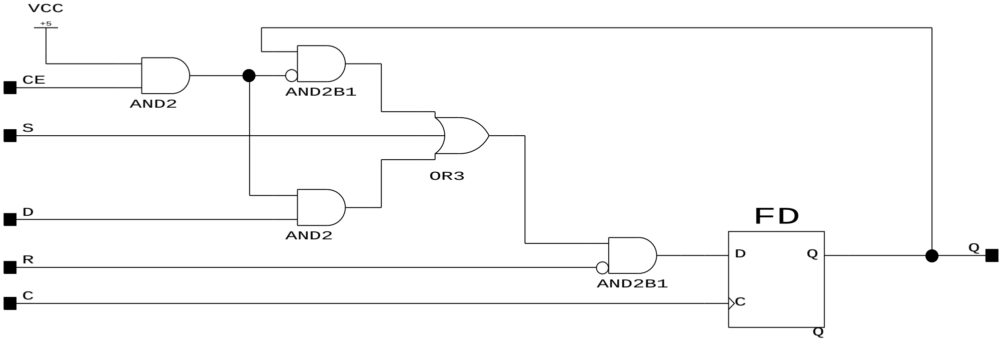
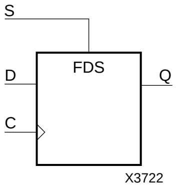


Figure 3-78 FDRSE Implementation (XC7000 and XC9000)

FDS

D Flip-Flop with Synchronous Set



XC7000	XC9000
Macro	Macro

FDS is a single D-type flip-flop with data (D) and synchronous set (S) inputs and data output (Q). The synchronous set input, when High, sets the Q output High on the Low-to-high clock (C) transition. The data on the D input is loaded into the flip-flop when S is Low during the Low-to-High clock (C) transition.

The default initial state of the flip-flop is zero.

Inputs			Outputs
S	D	C	Q
1	X	↑	1
0	1	↑	1
0	0	↑	0

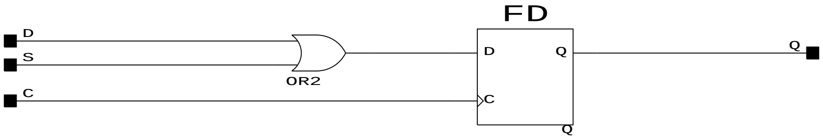
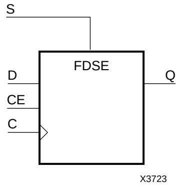


Figure 3-79 FDS Implementation (XC7000 and XC9000)

FDSE

D Flip-Flop with Clock Enable and Synchronous Set



XC7000	XC9000
Macro	Macro

FDSE is a single D-type flip-flop with data (D), clock enable (CE), and synchronous set (S) inputs and data output (Q). The synchronous set (S) input, when High, overrides the clock enable (CE) input and sets the Q output High during the Low-to-High clock (C) transition. The data on the D input is loaded into the flip-flop when S is Low and CE is High during the Low-to-High clock (C) transition.

The default initial state of the flip-flop is zero.

Inputs				Outputs
S	CE	D	C	Q
1	X	X	↑	1
0	0	X	X	No Chg
0	1	1	↑	1
0	1	0	↑	0

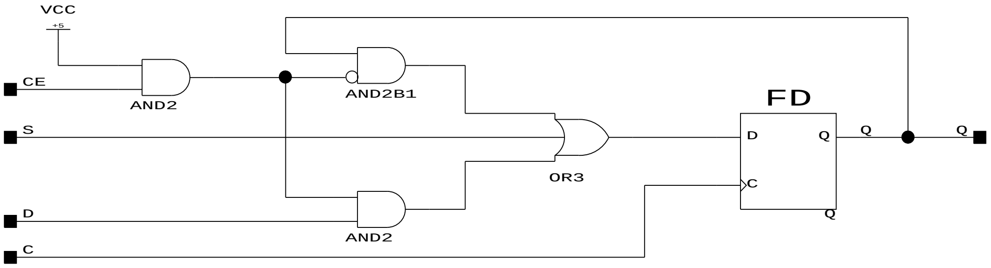
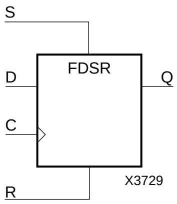


Figure 3-80 FDSE Implementation (XC7000 and XC9000)

FDSR

D Flip-Flop with Synchronous Set and Reset



XC7000	XC9000
Macro	Macro

FDSR is a single D-type flip-flop with data (D), synchronous reset (R) and synchronous set (S) inputs and data output (Q). When the set (S) input is High, it overrides all other inputs and sets the Q output High during the Low-to-High clock transition. (Set has precedence over Reset.) When reset (R) is High and S is Low, the flip-flop is reset, output Low, on the Low-to-High clock transition. Data on the D input is loaded into the flip-flop when S and R are Low on the Low-to-High clock transition.

The default initial state of the flip-flop is zero.

Inputs				Outputs
S	R	D	C	Q
1	X	X	↑	1
0	1	X	↑	0
0	0	1	↑	1
0	0	0	↑	0

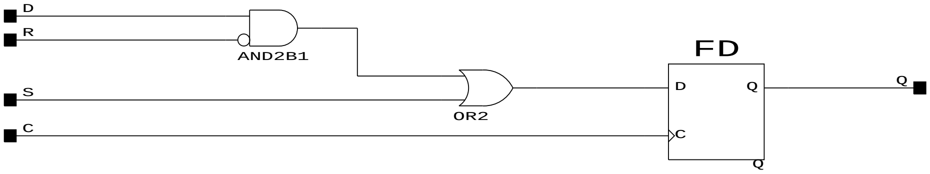
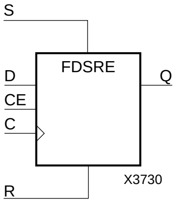


Figure 3-81 FDSR Implementation (XC7000 and XC9000)

FDSRE

D Flip-Flop with Synchronous Set and Reset and Clock Enable



XC7000	XC9000
Macro	Macro

FDSRE is a single D-type flip-flop with synchronous set (S), synchronous reset (R), and clock enable (CE) inputs and data output (Q). When synchronous set (S) is High, it overrides all other inputs and sets the Q output High during the Low-to-High clock transition. (Set has precedence over Reset.) When synchronous reset (R) is High and S is Low, output Q is reset Low during the Low-to-High clock transition. Data is loaded into the flip-flop when S and R are Low and CE is High during the Low-to-high clock transition. When CE is Low, clock transitions are ignored.

The default initial state of the flip-flop is zero.

Inputs					Outputs
S	R	CE	D	C	Q
1	X	X	X	↑	1
0	1	X	X	↑	0
0	0	0	X	X	No Chg
0	0	1	1	↑	1
0	0	1	0	↑	0

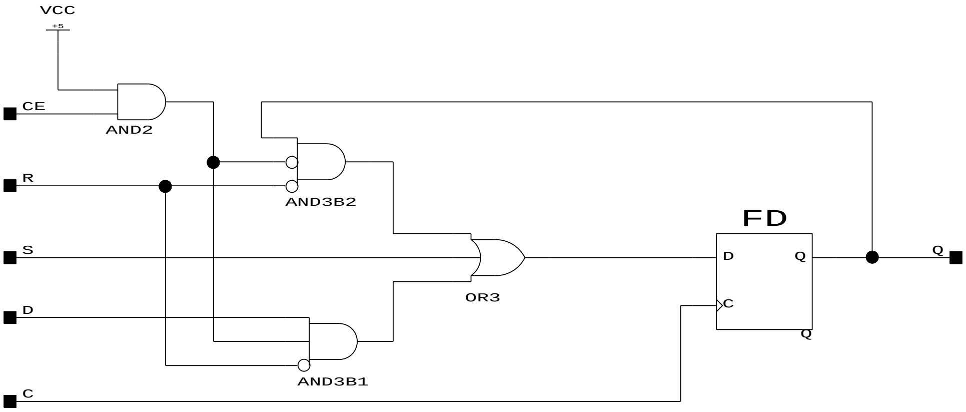
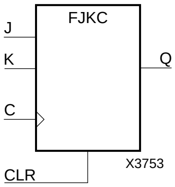


Figure 3-82 FDSRE Implementation (XC7000 and XC9000)

FJKC

J-K Flip-Flop with Asynchronous Clear



XC7000	XC9000
Macro	Macro

FJKC is a single J-K-type flip-flop with J, K, and asynchronous clear (CLR) inputs and data output (Q). The asynchronous clear (CLR) input, when High, overrides all other inputs and resets the Q output Low. When CLR is Low, the output responds to the state of the J and K inputs, as shown in the following truth table, during the Low-to-High clock (C) transition.

The default initial state of the flip-flop is zero.

Inputs				Outputs
CLR	J	K	C	Q
1	X	X	X	0
0	0	0	↑	No Chg
0	0	1	↑	0
0	1	0	↑	1
0	1	1	↑	Toggle

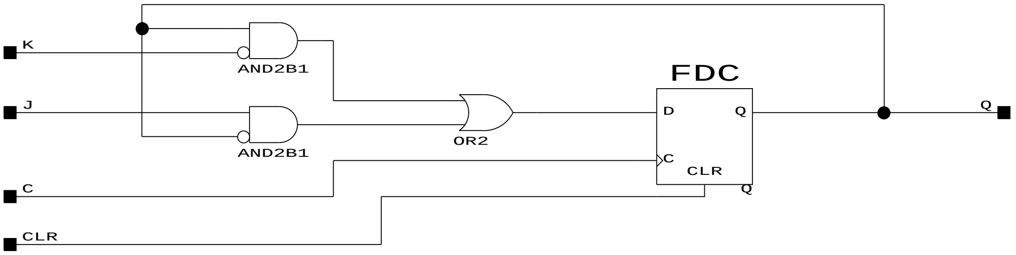
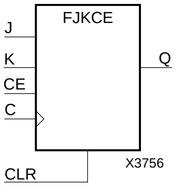


Figure 3-83 FJKC Implementation (XC7000 and XC9000)

FJKCE

J-K Flip-Flop with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

FJKCE is a single J-K-type flip-flop with J, K, clock enable (CE), and asynchronous clear (CLR) inputs and data output (Q). The asynchronous clear (CLR), when High, overrides all other inputs and resets the Q output Low. When CLR is Low and CE is High, Q responds to the state of the J and K inputs, as shown in the following truth table, during the Low-to-High clock transition. When CE is Low, the clock transitions are ignored.

The default initial state of the flip-flop is zero.

Inputs					Outputs
CLR	CE	J	K	C	Q
1	X	X	X	X	0
0	0	X	X	X	No Chg
0	1	0	0	X	No Chg
0	1	0	1	↑	0
0	1	1	0	↑	1
0	1	1	1	↑	Toggle

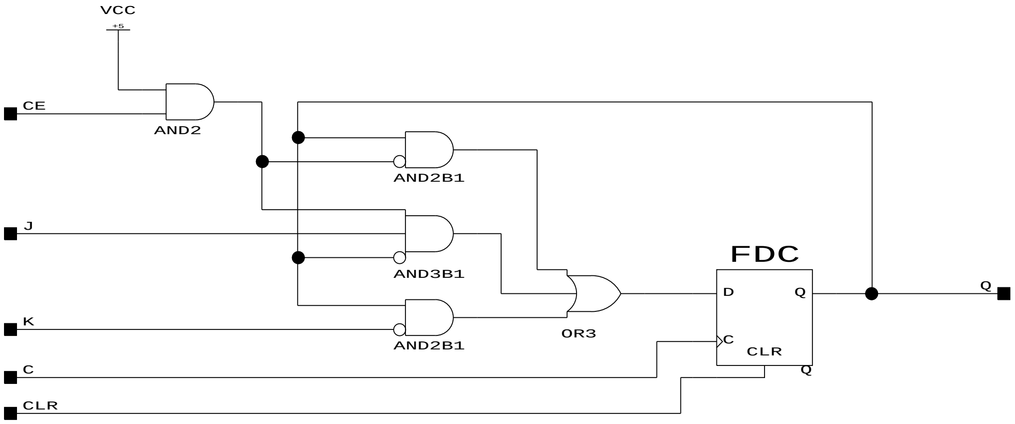
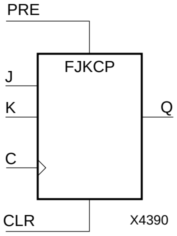


Figure 3-84 FJKCE Implementation (XC7000 and XC9000)

FJKCP

J-K Flip-Flop with Asynchronous Clear and Preset



XC7000	XC9000
Macro*	Macro

* not supported for XC7336 designs

FJKCP is a single J-K-type flip-flop with J, K, asynchronous clear (CLR), and asynchronous preset (PRE) inputs and data output (Q). The asynchronous clear input (CLR), when High, overrides all other inputs and resets the Q output Low. The asynchronous preset (PRE) input, when High, overrides all other inputs and sets the Q output High. When both CLR and PRE are active, the flip-flop output is unpredictable. When CLR and PRE are Low, Q responds to the state of the J and K inputs during the Low-to-High clock transition, as shown in the following truth table.

The default initial state of the flip-flop is zero.

Inputs					Outputs
CLR	**PRE**	**J**	**K**	**C**	**Q**
1	0	X	X	X	0
0	1	X	X	X	1
0	0	0	0	X	No Chg
0	0	0	1	↑	0
0	0	1	0	↑	1
0	0	1	1	↑	Toggle

Note: For XC7336, the PRE and CLR inputs cannot both be used.

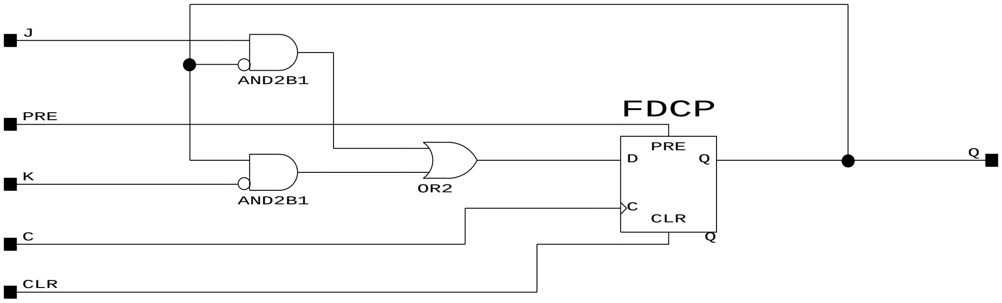
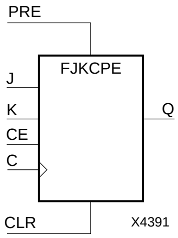


Figure 3-85   FJKCP Implementation (XC7000 and XC9000)

FJKCPE

J-K Flip-Flop with Asynchronous Clear and Preset and Clock Enable



XC7000	XC9000
Macro*	Macro

* not supported for XC7336 designs

FJKCPE is a single J-K-type flip-flop with J, K, asynchronous clear (CLR), asynchronous preset (PRE), and clock enable (CE) inputs and data output (Q). The asynchronous clear input (CLR), when High, overrides all other inputs and resets the Q output Low. The asynchronous preset (PRE) input, when High, overrides all other inputs and sets the Q output High. When both CLR and PRE are active, the flip-flop output is unpredictable. When CLR and PRE are Low and CE is High, Q responds to the state of the J and K inputs, as shown in the following truth table, during the Low-to-High clock transition. Clock transitions are ignored when CE is Low.

The default initial state of the flip-flop is zero.

Inputs						Outputs
CLR	PRE	CE	J	K	C	Q
1	0	X	X	X	X	0
0	1	X	X	X	X	1
0	0	0	0	X	X	No Chg
0	0	1	0	0	X	No Chg
0	0	1	0	1	↑	0
0	0	1	1	0	↑	1
0	0	1	1	1	↑	Toggle

Note: For XC7336, the PRE and CLR inputs cannot both be used.

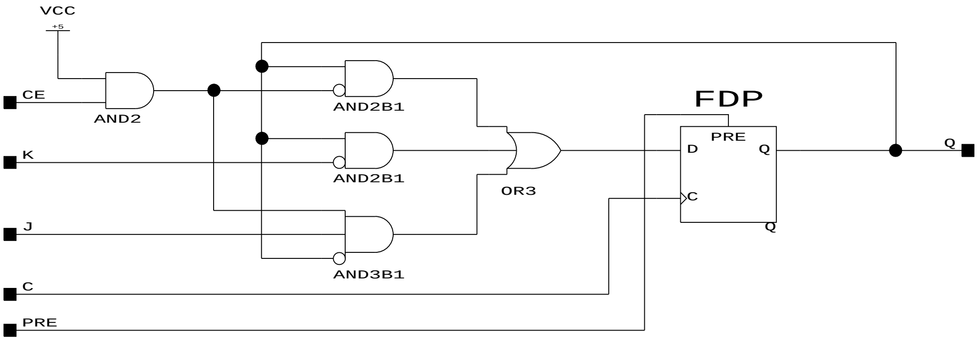
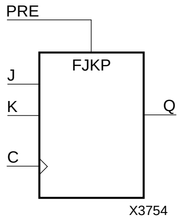


Figure 3-86 FJKCPE Implementation (XC7000 and XC9000)

FJKP

J-K Flip-Flop with Asynchronous Preset



XC7000	XC9000
Macro	Macro

FJKP is a single J-K-type flip-flop with J, K, and asynchronous preset (PRE) inputs and data output (Q). The asynchronous preset (PRE) input, when High, overrides all other inputs and sets the Q output High on the Low-to-High clock (C) transition. When PRE is Low, the Q output responds to the state of the J and K inputs, as shown in the following truth table, during the Low-to-High clock transition. The default initial state of the flip-flop is zero.

Inputs				Outputs
PRE	J	K	C	Q
1	X	X	X	1
0	0	0	X	No Chg
0	0	1	↑	0
0	1	0	↑	1
0	1	1	↑	Toggle

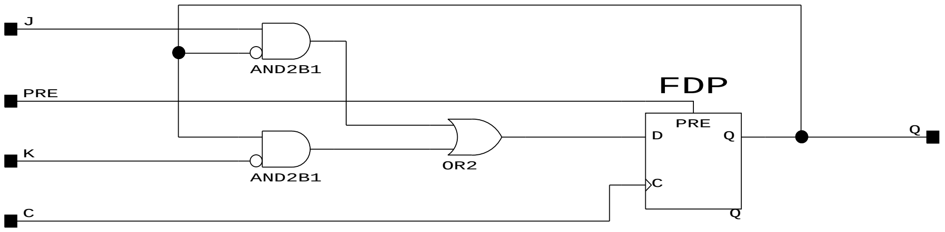
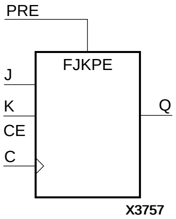


Figure 3-87 FJKP Implementation (XC7000 and XC9000)

FJKPE

J-K Flip-Flop with Clock Enable and Asynchronous Preset



XC7000	XC9000
Macro	Macro

FJKPE is a single J-K-type flip-flop with J, K, clock enable (CE), and asynchronous preset (PRE) inputs and data output (Q). The asynchronous preset (PRE), when high, overrides all other inputs and sets the Q output High. When PRE is Low and CE is High, the Q output responds to the state of the J and K inputs, according to the following truth table, during the Low-to-High clock (C) transition. When CE is Low, clock transitions are ignored. The default initial state of the flip-flop is zero.

Inputs					Outputs
PRE	CE	J	K	C	Q
1	X	X	X	X	1
0	0	X	X	X	No Chg
0	1	0	0	X	No Chg
0	1	0	1	↑	0
0	1	1	0	↑	1
0	1	1	1	↑	Toggle

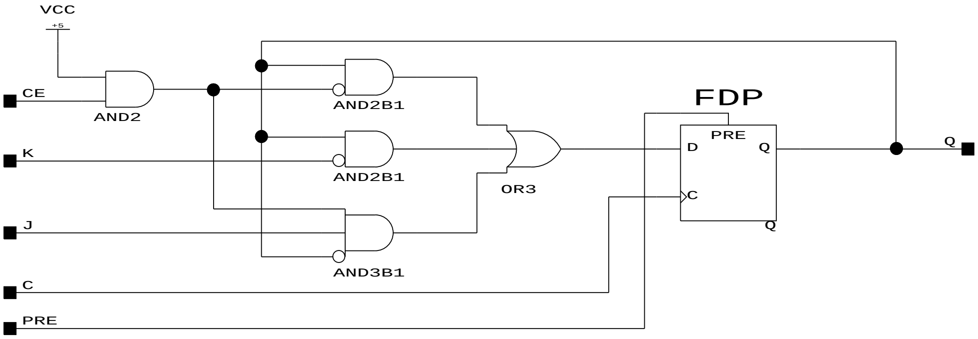
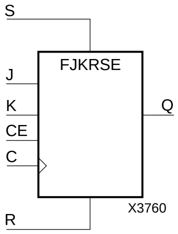


Figure 3-88 FJKPE Implementation (XC7000 and XC9000)

FJKRSE

J-K Flip-Flop with Clock Enable and Synchronous Reset and Set



XC7000	XC9000
Macro	Macro

FJKRSE is a single J-K-type flip-flop with J, K, synchronous reset (R), synchronous set (S), and clock enable (CE) inputs and data output (Q). When synchronous reset (R) is High, all other inputs are ignored and output Q is reset Low. (Reset has precedence over Set.) When synchronous set (S) is High and R is Low, output Q is set High. When R and S are Low and CE is High, output Q responds to the state of the J and K inputs, according to the following truth table, during the Low-to-High clock (C) transition. When CE is Low, clock transitions are ignored.

The default initial state of the flip-flop is zero.

Inputs						Outputs
R	S	CE	J	K	C	Q
1	X	X	X	X	↑	0
0	1	X	X	X	↑	1
0	0	0	X	X	X	No Chg
0	0	1	0	0	X	No Chg
0	0	1	0	1	↑	0
0	0	1	1	0	↑	1
0	0	1	1	1	↑	Toggle

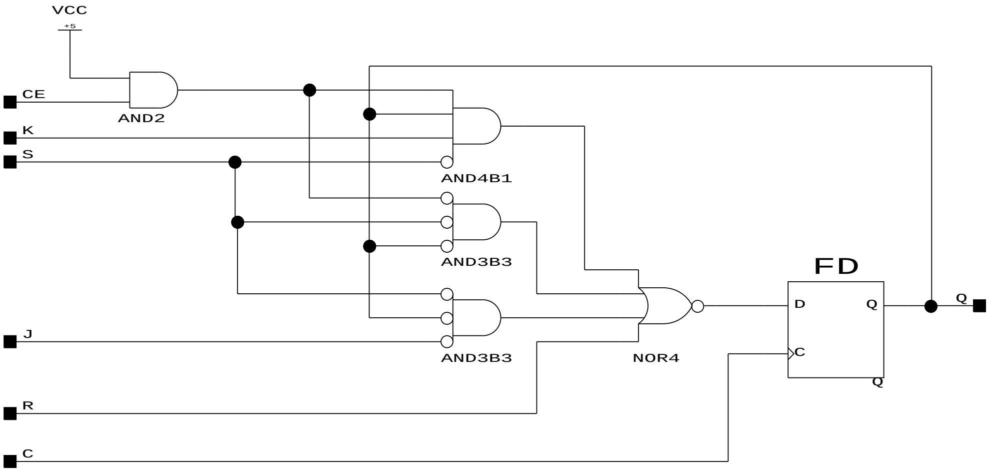
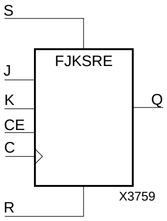


Figure 3-89 FJKRSE Implementation (XC7000 and XC9000)

FJKSRE

J-K Flip-Flop with Clock Enable and Synchronous Set and Reset



XC7000	XC9000
Macro	Macro

FJKSRE is a single J-K-type flip-flop with J, K, synchronous set (S), synchronous reset (R), and clock enable (CE) inputs and data output (Q). When synchronous set (S) is High, all other inputs are ignored and output Q is set High. (Set has precedence over Reset.) When synchronous reset (R) is High and S is Low, output Q is reset Low. When S and R are Low and CE is High, output Q responds to the state of the J and K inputs, as shown in the following truth table, during the Low-to-High clock (C) transition. When CE is Low, clock transitions are ignored.

The default initial state of the flip-flop is zero.

Inputs						Outputs
S	**R**	**CE**	**J**	**K**	**C**	**Q**
1	X	X	X	X	↑	1
0	1	X	X	X	↑	0
0	0	0	X	X	X	No Chg
0	0	1	0	0	X	No Chg
0	0	1	0	1	↑	0
0	0	1	1	0	↑	1
0	0	1	1	1	↑	Toggle

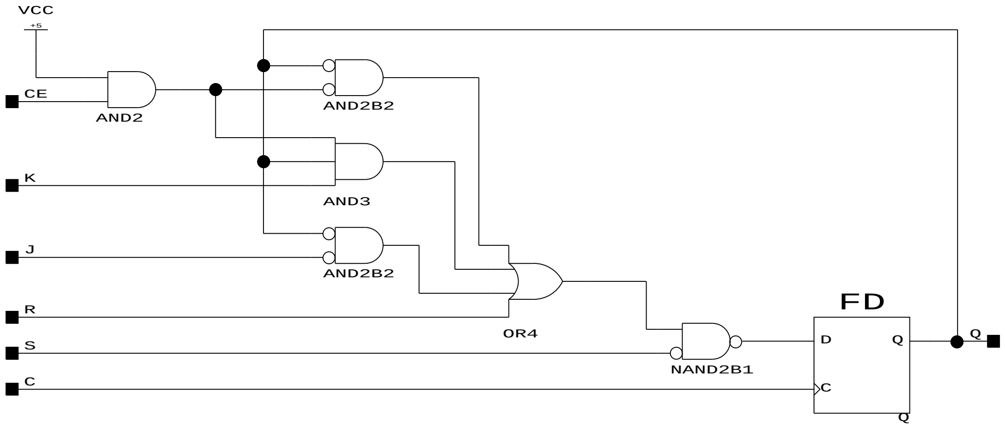
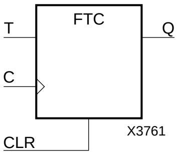


Figure 3-90 FJKSRE Implementation (XC7000 and XC9000)

FTC

Toggle Flip-Flop with Toggle Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

FTC is a synchronous, resettable toggle flip-flop. The asynchronous clear (CLR) input, when High, overrides all other inputs and resets the data output (Q) Low. The Q output toggles, or changes state, when the toggle enable (T) input is High and CLR is Low during the Low-to-High clock transition.

The default initial state of the flip-flop is zero.

Inputs			Outputs
CLR	T	C	Q
1	X	X	0
0	0	X	No Chg
0	1	↑	Toggle

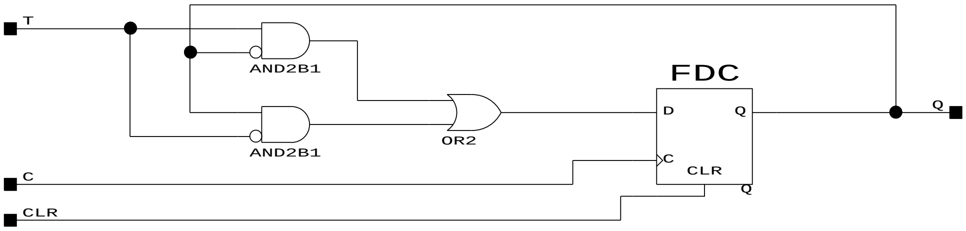
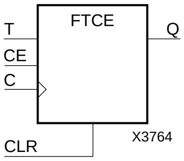


Figure 3-91 FTC Implementation (XC7000 and XC9000)

FTCE

Toggle Flip-Flop with Toggle and Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

When the asynchronous clear (CLR) input is High, all other inputs are ignored and the data output (Q) is reset Low. When CLR is Low and toggle enable (T) and clock enable (CE) are High, Q output toggles, or changes state, during the Low-to-High clock (C) transition. When CE is Low, clock transitions are ignored.

The default initial state of the flip-flop is zero.

Inputs				Outputs
CLR	CE	T	C	Q
1	X	X	X	0
0	0	X	X	No Chg
0	1	0	X	No Chg
0	1	1	↑	Toggle

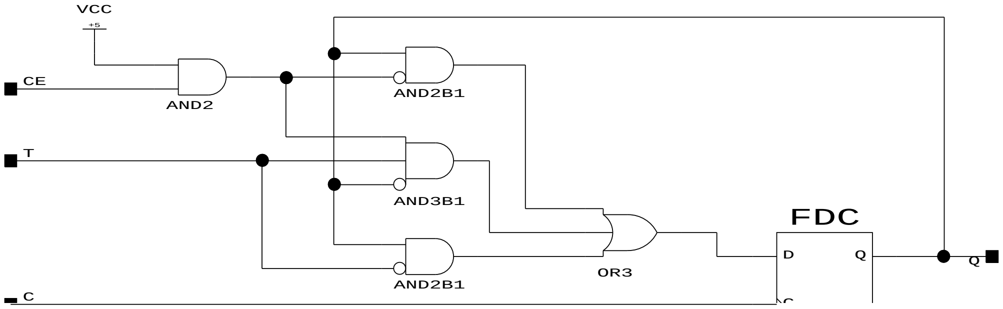
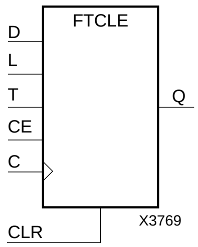


Figure 3-92 FTCE Implementation (XC7000 and XC9000)

FTCLE

Toggle/Loadable Flip-Flop with Toggle and Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

When the asynchronous clear input (CLR) is High, all other inputs are ignored and output Q is reset Low. When load enable input (L) is High and CLR is Low, clock enable (CE) is overridden and the data on data input (D) is loaded into the flip-flop during the Low-to-High clock (C) transition. When toggle enable (T) and CE are High and L and CLR are Low, output Q toggles, or changes state, during the Low- to-High clock transition. When CE is Low, clock transitions are ignored.

The default initial state of the flip-flop is zero.

Inputs						Outputs
CLR	L	CE	T	D	C	Q
1	X	X	X	X	X	0
0	1	X	X	1	↑	1
0	1	X	X	0	↑	0
0	0	0	X	X	X	No Chg
0	0	1	0	X	X	No Chg
0	0	1	1	X	↑	Toggle

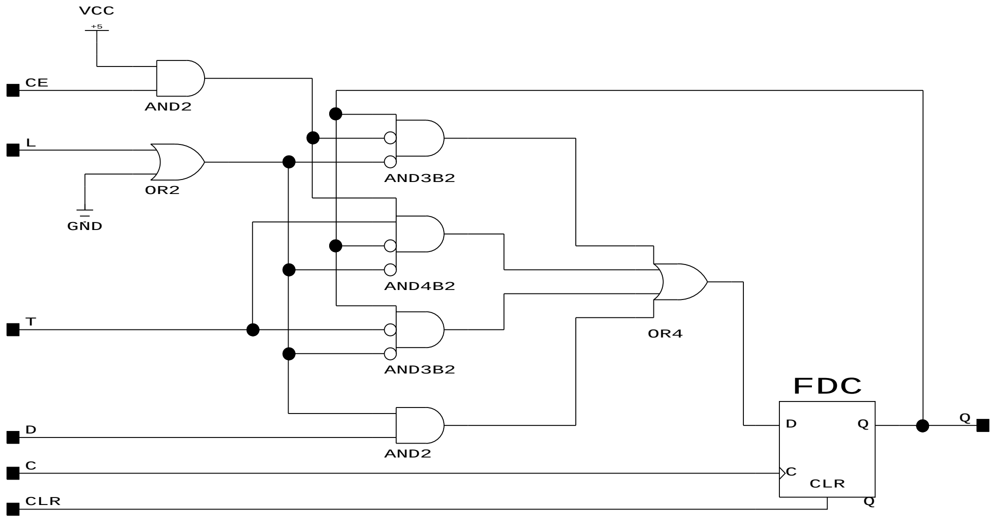
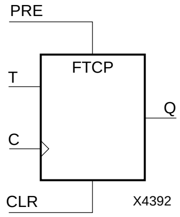


Figure 3-93 FTCLE Implementation (XC7000 and XC9000)

FTCP

Toggle Flip-Flop with Toggle Enable and Asynchronous Clear and Preset



XC7000	XC9000
Macro*	Macro

When the asynchronous clear (CLR) input is High, all other inputs are ignored and the output (Q) is reset Low. When the asynchronous preset (PRE) input is High, all other inputs are ignored and Q is set High. When the toggle enable input (T) is High and CLR and PRE are Low, output Q toggles, or changes state, during the Low-to-High clock (C) transition. When both PRE and CLR are active, the flip-flop output is unpredictable.

The default initial state of the flip-flop is zero.

Inputs				Outputs
CLR	PRE	T	C	Q
1	0	X	X	0
0	1	X	X	1
0	0	0	X	No Chg
0	0	1	↑	Toggle

Note: For XC7336, the PRE and CLR inputs cannot both be used.

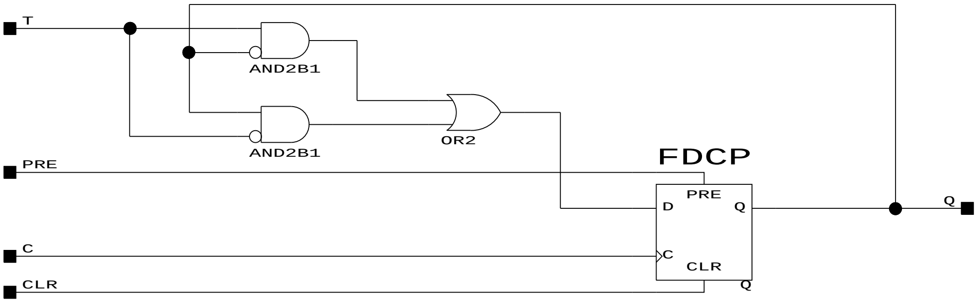
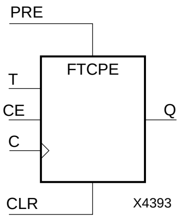


Figure 3-94 FTCP Implementation (XC7000 and XC9000)

FTCPE

Toggle Flip-Flop with Toggle and Clock Enable and Asynchronous Clear and Preset



XC7000	XC9000
Macro*	Macro

* not supported for XC7336 designs

When the asynchronous clear (CLR) input is High, all other inputs are ignored and the output (Q) is reset Low. When the asynchronous preset (PRE) input is High, all other inputs are ignored and Q is set High. When the toggle enable input (T) and the clock enable input (CE) are High and CLR and PRE are Low, output Q toggles, or changes state, during the Low-to-High clock (C) transition. Clock transitions are ignored when CE is Low. When both CLR and PRE are active, the flip-flop output is unpredictable.

The default initial state of the flip-flop is zero.

Inputs					Outputs
CLR	**PRE**	**CE**	**T**	**C**	**Q**
1	0	X	X	X	0
0	1	X	X	X	1
0	0	0	X	X	No Chg
0	0	1	0	X	No Chg
0	0	1	1	↑	Toggle

Note: For XC7336, the PRE and CLR inputs cannot both be used.

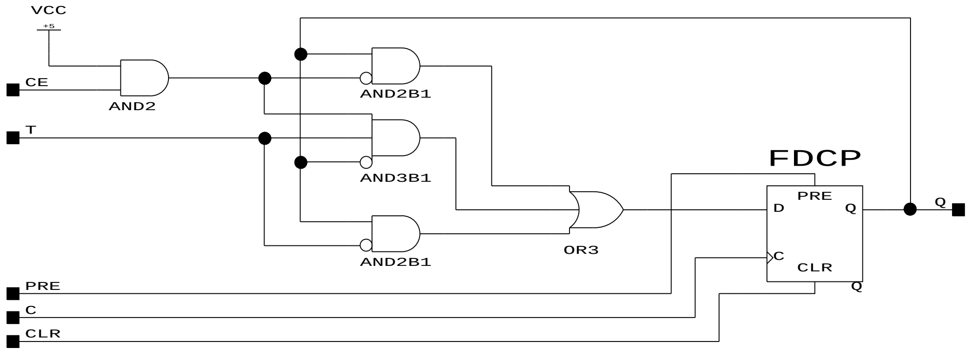
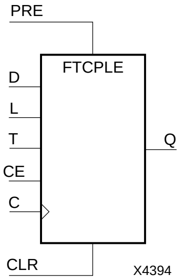


Figure 3-95 FTCPE Implementation (XC7000 and XC9000)

FTCPLE

Loadable Toggle Flip-Flop with Toggle and Clock Enable and Asynchronous Clear and Preset



XC7000	XC9000
Macro*	Macro

* not supported for XC7336 designs

When the asynchronous clear (CLR) input is High, all other inputs are ignored and the output (Q) is reset Low. When the asynchronous preset (PRE) input is High, all other inputs are ignored and Q is set High. The load input (L) loads the data on input D into the flip-flop on the Low-to-High clock transition, regardless of the state of the clock enable (CE). When the toggle enable input (T) and the clock enable input (CE) are High and CLR, PRE, and L are Low, output Q toggles, or changes state, during the Low-to-High clock (C) transition. Clock transitions are ignored when CE is Low. When both CLR and PRE are active, the flip-flop output is unpredictable.

The default initial state of the flip-flop is zero.

Inputs							Outputs
CLR	PRE	L	CE	T	C	D	Q
1	0	X	X	X	X	X	0
0	1	X	X	X	X	X	1
0	0	1	X	X	↑	0	0
0	0	1	X	X	↑	1	1
0	0	0	0	X	X	X	No Chg
0	0	0	1	0	X	X	No Chg
0	0	0	1	1	↑	X	Toggle

Note: For XC7336, the PRE and CLR inputs cannot both be used.

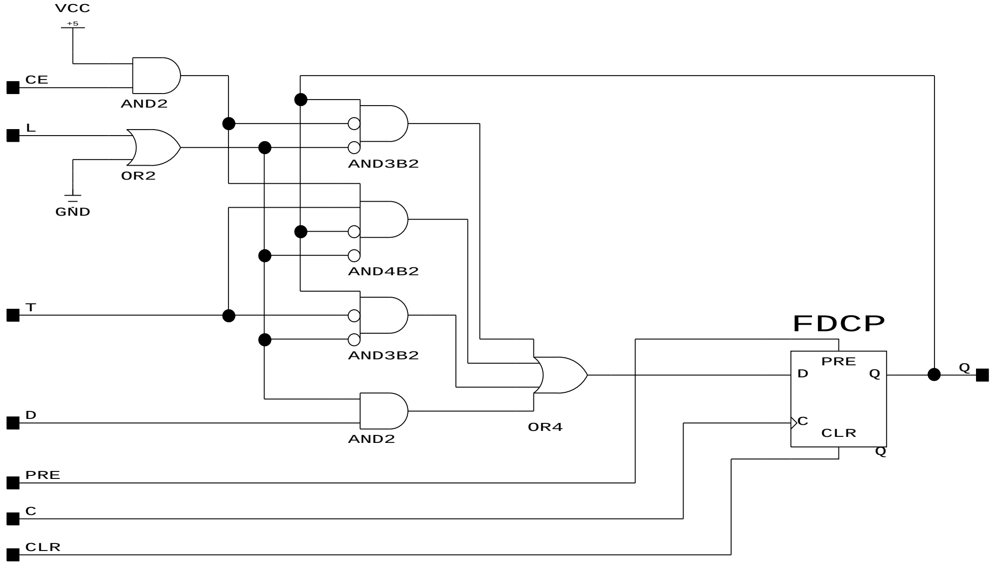
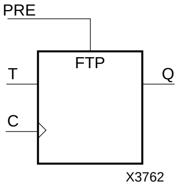


Figure 3-96 FTCPLE Implementation (XC7000 and XC9000)

FTP

Toggle Flip-Flop with Toggle Enable and Asynchronous Preset



XC7000	XC9000
Macro	Macro

When the asynchronous preset (PRE) input is High, all other inputs are ignored and output Q is set High. When toggle enable input (T) is High and PRE is Low, output Q toggles, or changes state, during the Low-to-High clock (C) transition. The default initial state of the flip-flop is zero.

Inputs			Outputs
PRE	T	C	Q
1	X	X	1
0	0	X	No Chg
0	1	↑	Toggle

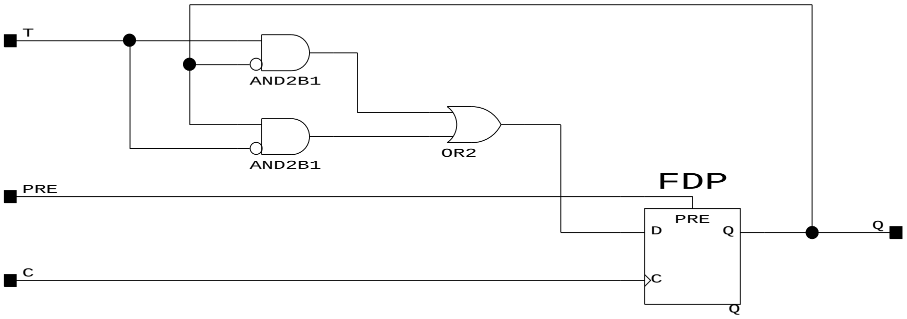
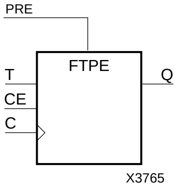


Figure 3-97 FTP Implementation (XC7000 and XC9000)

FTPE

Toggle Flip-Flop with Toggle and Clock Enable and Asynchronous Preset



XC7000	XC9000
Macro	Macro

When the asynchronous preset (PRE) input is High, all other inputs are ignored and output Q is set High during the Low-to-High clock (C) transition. When the toggle enable input (T) is High, clock enable (CE) is High, and PRE is Low, output Q toggles, or changes state, during the Low-to-High clock transition. When CE is Low, clock transitions are ignored. The default initial state of the flip-flop is zero.

Inputs				Outputs
PRE	CE	T	C	Q
1	X	X	X	1
0	0	X	X	No Chg
0	1	0	X	No Chg
0	1	1	↑	Toggle

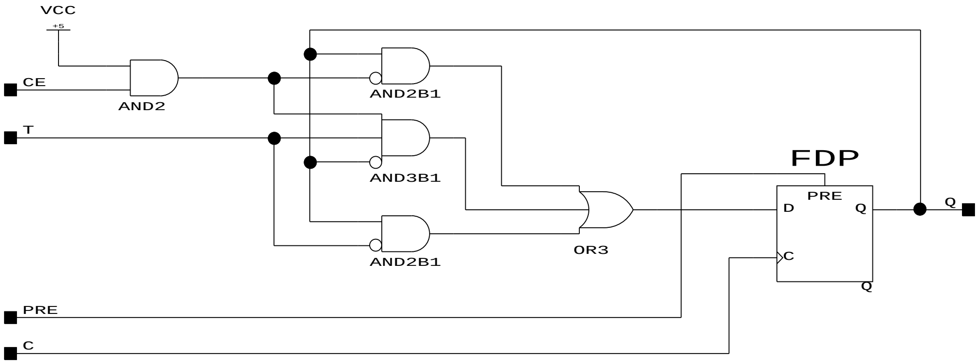
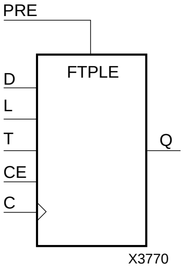


Figure 3-98 FTPE Implementation (XC7000 and XC9000)

FTPLE

Toggle/Loadable Flip-Flop with Toggle and Clock Enable and Asynchronous Preset



XC7000	XC9000
Macro	Macro

When the asynchronous preset input (PRE) is High, all other inputs are ignored and output Q is set High during the Low-to-High clock (C) transition. When the load enable input (L) is High and PRE is Low, the clock enable (CE) is overridden and the data on input (D) is loaded into the flip-flop during the Low-to-High clock transition. When L and PRE are Low and toggle enable input (T) and CE are High, output Q toggles, or changes state, during the Low-to-High clock transition. When CE is Low, clock transitions are ignored. The default initial state of the flip-flop is zero.

Inputs						Outputs
PRE	**L**	**CE**	**T**	**D**	**C**	**Q**
1	X	X	X	X	X	1
0	1	X	X	1	↑	1
0	1	X	X	0	↑	0
0	0	0	X	X	X	No Chg
0	0	1	0	X	X	No Chg
0	0	1	1	X	↑	Toggle

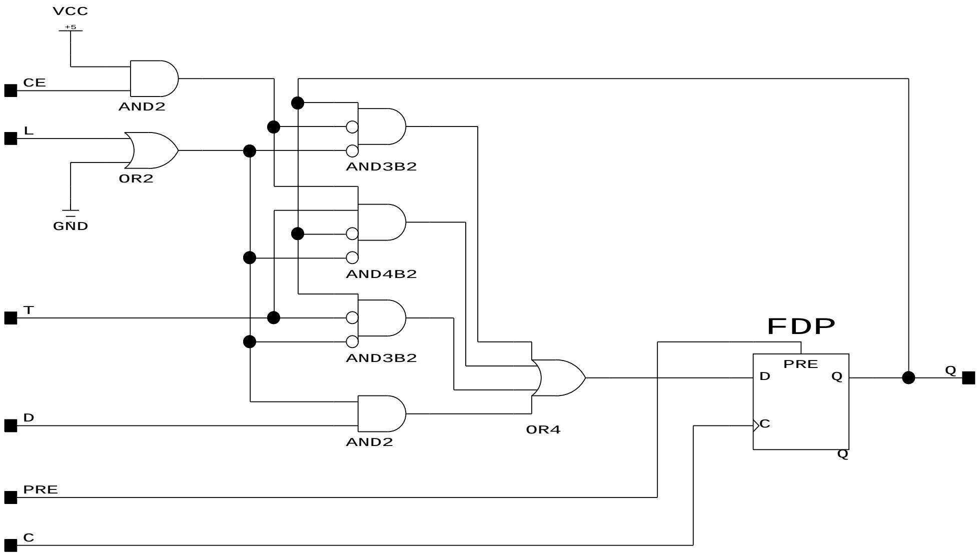
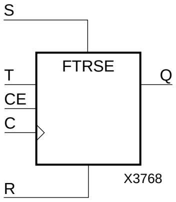


Figure 3-99   FTPLE Implementation (XC7000 and XC9000)

FTRSE

Toggle Flip-Flop with Toggle and Clock Enable and Synchronous Reset and Set



XC7000	XC9000
Macro	Macro

When the synchronous reset input (R) is High, it overrides all other inputs and the data output (Q) is reset Low. When the synchronous set input (S) is High and R is Low, clock enable input (CE) is overridden and output Q is set High. (Reset has precedence over Set.) When toggle enable input (T) and CE are High and R and S are Low, output Q toggles, or changes state, during the Low-to-High clock transition.

The default initial state of the flip-flop is zero.

Inputs					Outputs
R	**S**	**CE**	**T**	**C**	**Q**
1	X	X	X	↑	0
0	1	X	X	↑	1
0	0	0	X	X	No Chg
0	0	1	0	X	No Chg
0	0	1	1	↑	Toggle

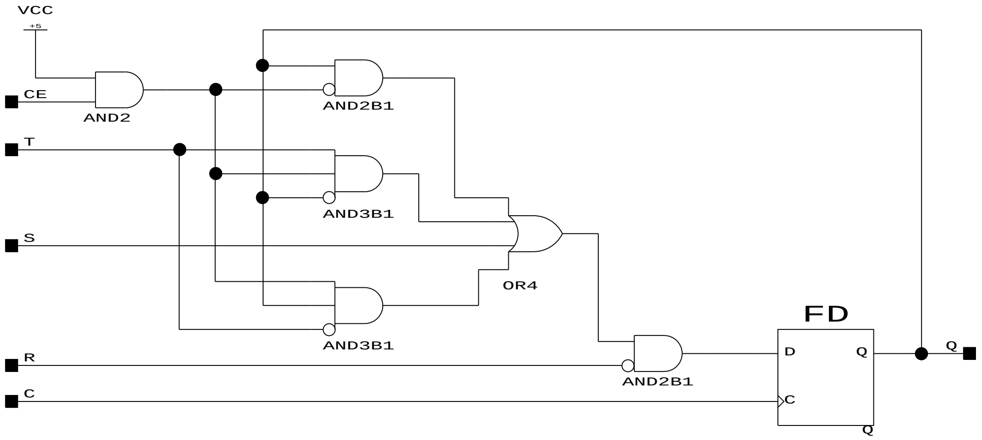
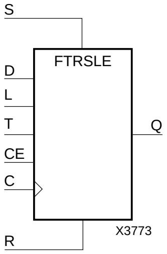


Figure 3-100 FTRSE Implementation (XC7000 and XC9000)

FTRSLE

Toggle/Loadable Flip-Flop with Toggle and Clock Enable and Synchronous Reset and Set



XC7000	XC9000
Macro	Macro

The synchronous reset input (R), when High, overrides all other inputs and resets the data output (Q) Low. (Reset has precedence over Set.) When R is Low and synchronous set input (S) is High, the clock enable input (CE) is overridden and output Q is set High. When R and S are Low and load enable input (L) is High, CE is overridden and data on data input (D) is loaded into the flip-flop during the Low-to-High clock transition. When R, S, and L are Low and CE is High, output Q toggles, or changes state, during the Low-to-High clock transition. When CE is Low, clock transitions are ignored.

The default initial state of the flip-flop is zero.

Inputs							Outputs
R	S	L	CE	T	D	C	Q
1	0	X	X	X	X	↑	0
0	1	X	X	X	X	↑	1
0	0	1	X	X	1	↑	1
0	0	1	X	X	0	↑	0
0	0	0	0	X	X	X	No Chg
0	0	0	1	0	X	X	No Chg
0	0	0	1	1	X	↑	Toggle

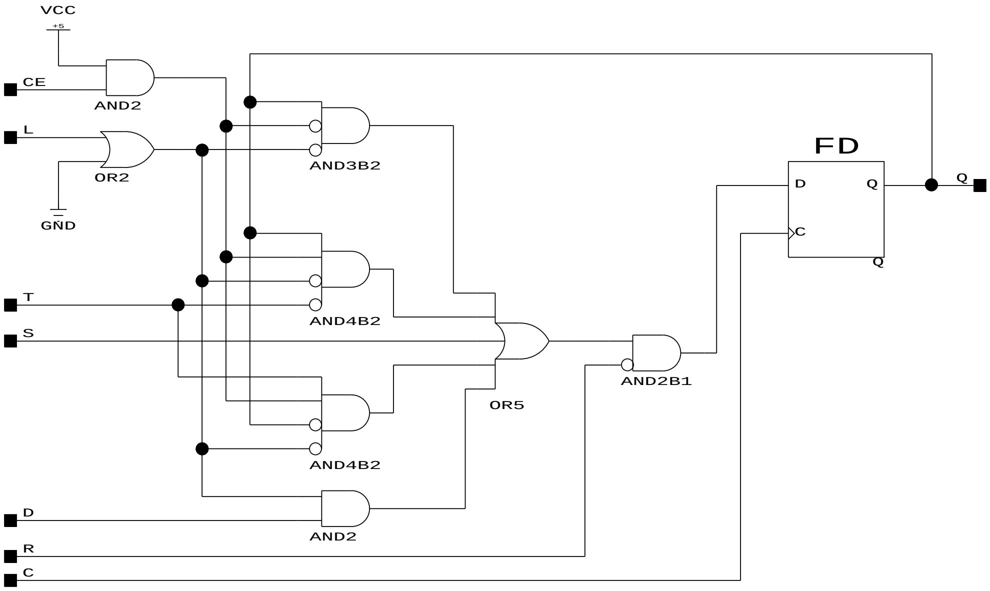
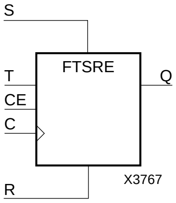


Figure 3-101 FTRSLE Implementation (XC7000 and XC9000)

FTSRE

Toggle Flip-Flop with Toggle and Clock Enable and Synchronous Set and Reset



XC7000	XC9000
Macro	Macro

The synchronous set input, when High, overrides all other inputs and sets data output (Q) High. (Set has precedence over Reset.) When synchronous reset input (R) is High and S is Low, clock enable input (CE) is overridden and output Q is reset Low. When toggle enable input (T) and CE are High and S and R are Low, output Q toggles, or changes state, during the Low-to-High clock transition.

The default initial state of the flip-flop is zero.

Inputs					Outputs
S	R	CE	T	C	Q
1	X	X	X	↑	1
0	1	X	X	↑	0
0	0	0	X	X	No Chg
0	0	1	0	X	No Chg
0	0	1	1	↑	Toggle

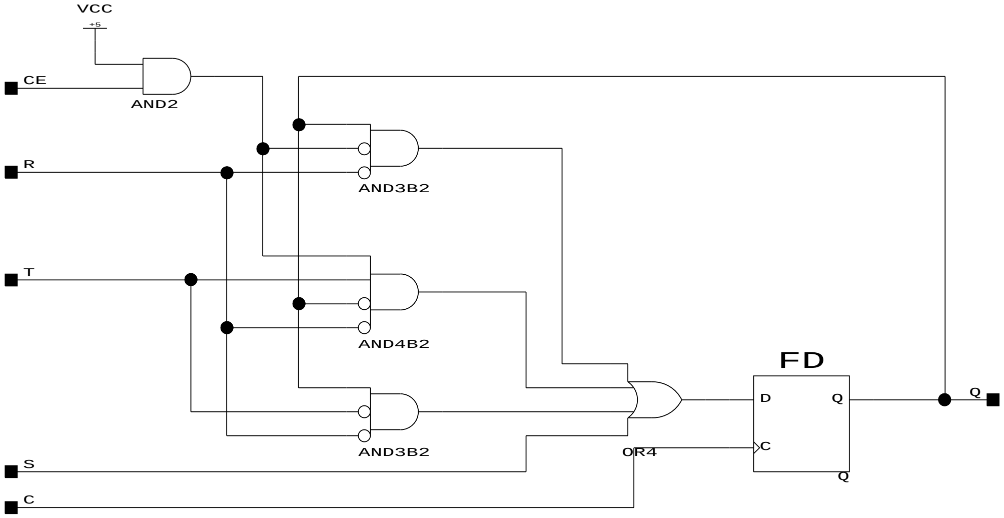
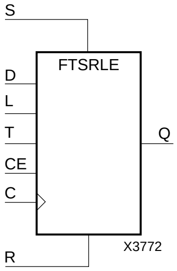


Figure 3-102   FTSRE Implementation (XC7000 and XC9000)

FTSRLE

Toggle/Loadable Flip-Flop with Toggle and Clock Enable and Synchronous Set and Reset



XC7000	XC9000
Macro	Macro

The synchronous set input (S), when High, overrides all other inputs and sets data output (Q) High. (Set has precedence over Reset.) When synchronous reset (R) is High and S is Low, clock enable input (CE) is overridden and output Q is reset Low. When load enable input (L) is High and S and R are Low, CE is overridden and data on data input (D) is loaded into the flip-flop during the Low-to-High clock transition. When the toggle enable input (T) and CE are High and S, R, and L are Low, output Q toggles, or changes state, during the Low-to-High clock transition. When CE is Low, clock transitions are ignored.

The default initial state of the flip-flop is zero.

Inputs							Outputs
S	R	L	CE	T	D	C	Q
1	0	X	X	X	X	↑	1
0	1	X	X	X	X	↑	0
0	0	1	X	X	1	↑	1
0	0	1	X	X	0	↑	0
0	0	0	0	X	X	X	No Chg
0	0	0	1	0	X	X	No Chg
0	0	0	1	1	X	↑	Toggle

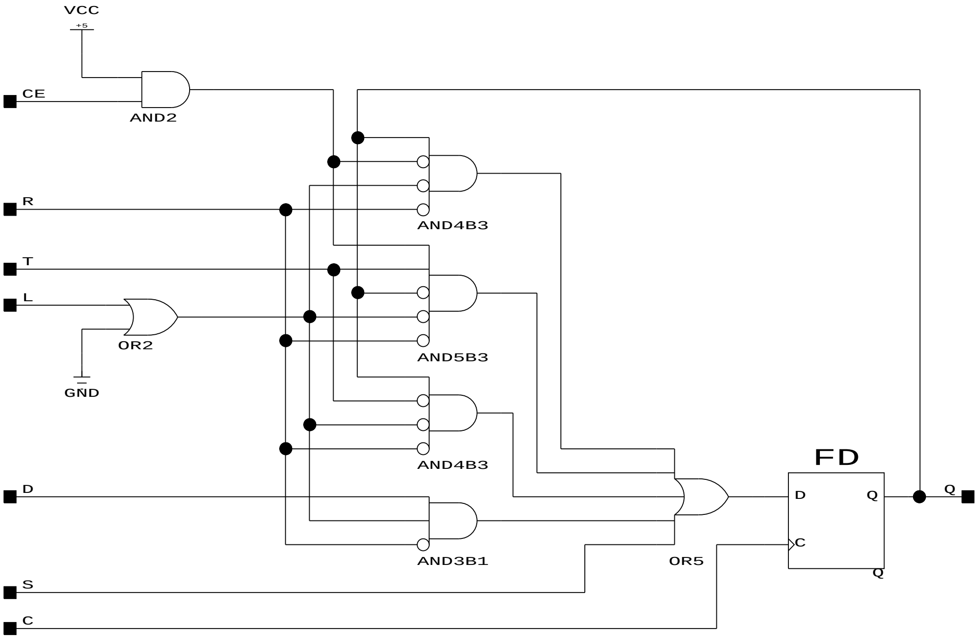
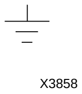


Figure 3-103 FTSRLE Implementation (XC7000 and XC9000)

GND

Ground-Connection Signal Tag



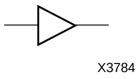
XC7000	XC9000
Primitive	Primitive

The GND signal tag, or parameter, forces a net or input function to a Low logic level. A net tied to GND cannot have any other source.

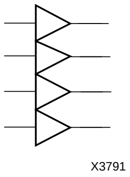
When the software encounters a net or input function tied to GND, it removes any logic that is disabled by the GND signal. The GND signal is only implemented when the disabled logic cannot be removed.

IBUF, IBUF4, IBUF8, and IBUF16

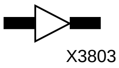
Single- and Multiple-Input Buffers



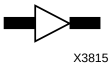
IBUF4



IBUF8



IBUF16



Name	XC7000	XC9000
IBUF	Primitive	Primitive
IBUF4, IBUF8, IBUF16	Macro	Macro

IBUF, IBUF4, IBUF8, and IBUF16 are single and multiple input buffers. An IBUF isolates the internal circuit from the signals coming into a chip. IBUFs are contained in input/output blocks (IOB). IBUF inputs (I) are connected to an IPAD or an IOPAD. IBUF outputs (O) are connected to the internal circuit.

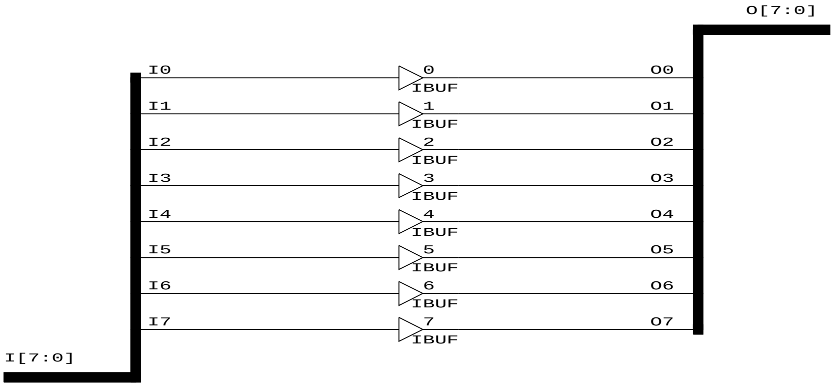
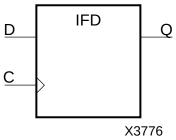


Figure 3-104 IBUF8 Implementation (XC7000 and XC9000)

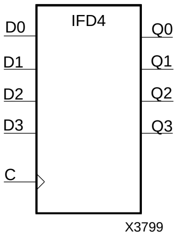
IFD, IFD4, IFD8, and IFD16

Single- and Multiple-Input D Flip-Flops



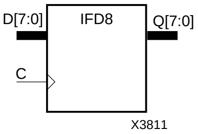
Name	XC7000	XC9000
IFD IFD4, IFD8, IFD16	Macro*	Macro

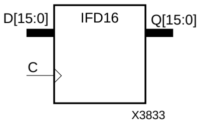
* not supported for XC7336 designs



The IFD D-type flip-flop is contained in an input/output block (IOB). The input (D) of the flip-flop is connected to an IPAD or an IOPAD (without using an IBUF). The D input provides data input for the flip-flop, which synchronizes data entering the chip. The data on input D is loaded into the flip-flop during the Low-to-High clock (C) transition and appears at the output (Q). The clock input is controlled by the internal circuit. For XC7000 CPLDs, the clock (C) can only be driven by a FastCLK represented by the BUFG symbol.

The initial state of all flip-flops is zero, except for XC7000 CPLDs, where the initial state is always one.





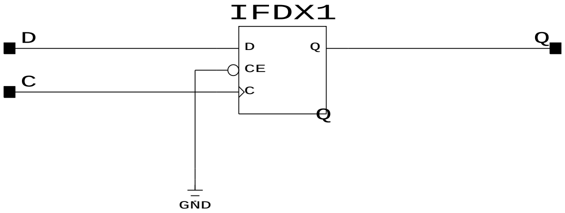


Figure 3-105 IFD Implementation (XC7000)

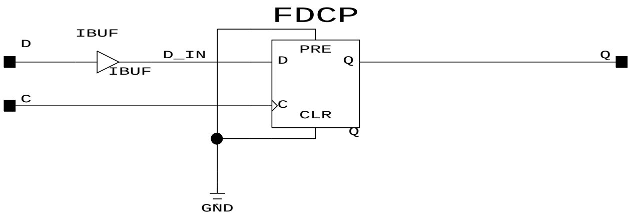


Figure 3-106 IFD Implementation (XC9000)

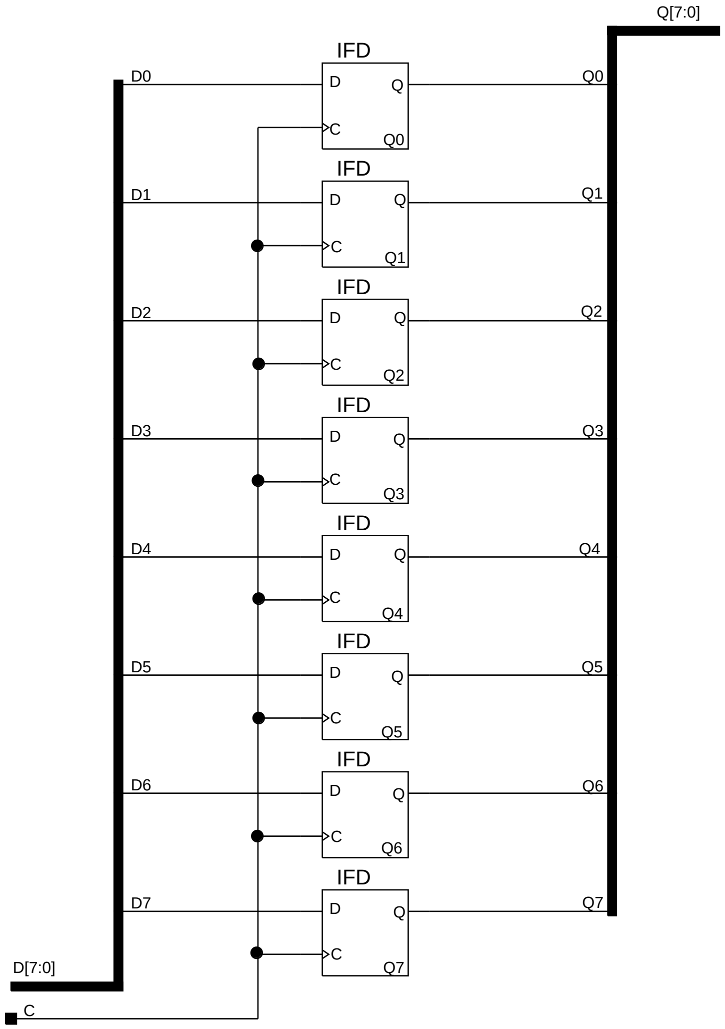
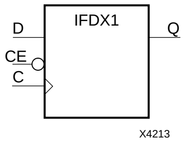
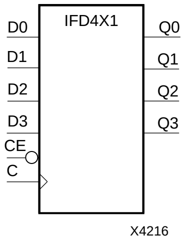


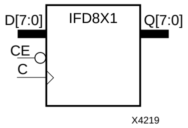
Figure 3-107 IFD8 Implementation (XC7000 and XC9000)

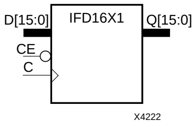
IFDX1, IFD4X1, IFD8X1, and IFD16X1

Input D Flip-Flops for XC7000









Name	XC7000	XC9000
IFDX1	Primitive*	NA
IFD4X1, IFD8X1, IFD16X1	Macro*	NA

* not supported for XC7336 designs

The IFDX1 symbols are D-type flip-flops with synchronous clock enable implemented in the input blocks of an XC7000 device. They are commonly used to synchronize and store data entering a chip. The data input (D) of the flip-flop is connected directly to an IPAD or an IOPAD (without using an IBUF). When the clock enable (CE) input is Low, the data on input D is loaded into the flip-flop during the Low-to-High clock (C) transition and appears at the output (Q). The flip-flop ignores clock transitions when CE is High.

The clock input (C) must be driven by a global FastCLK net of the XC7000 device, represented by the BUFG symbol. The clock enable input (CE) must be driven by a global clock enable net of the XC7000 device, represented by the BUFCE symbol.

The initial state of all flip-flops is zero, except for XC7000 CPLDs, where the initial state is always one.

Inputs			Outputs
D	**CE**	**C**	**Q**
X	1	X	No Chg
0	0	↑	0
1	0	↑	1

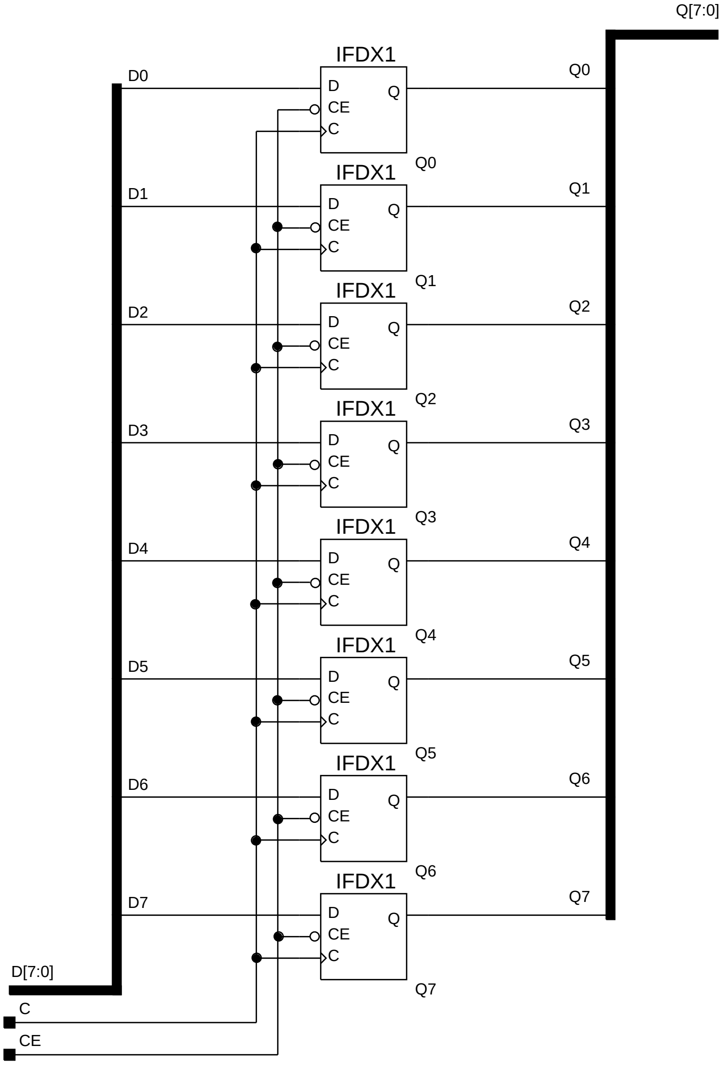
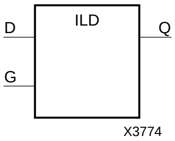


Figure 3-108 IFD8X1 Implementation (XC7000)

ILD, ILD4, ILD8, and ILD16

Input Transparent Data Latches

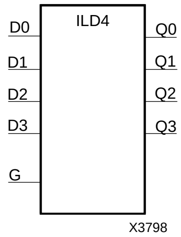


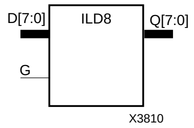
Name	XC7000	XC9000
ILD	Primitive*	NA
ILD4, ILD8, ILD16	Macro*	NA

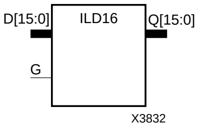
* not supported for XC7336 designs

ILD, ILD4, ILD8, and ILD16 are single or multiple transparent data latches, which can be used to hold transient data entering a chip. The latch input (D) is connected to an IPAD or an IOPAD (without using an IBUF). When the gate input (G) is High, data on the inputs (D) appears on the outputs (Q). Data on the D inputs during the High-to-Low G transition is stored in the latch. For XC7000 CPLDs, the gate input (G) must be driven by a FastCLK, represented by the BUFG symbol.

The default state of all latches is zero, except for XC7000 CPLDs, where the initial state is always one.







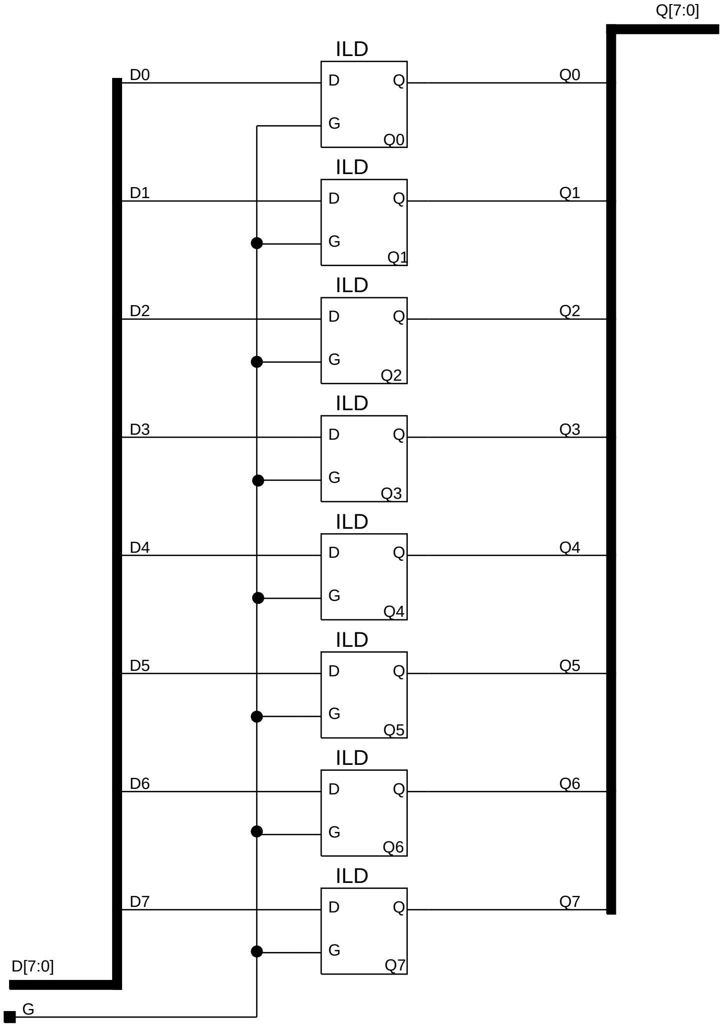
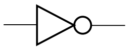


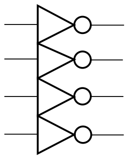
Figure 3-109 ILD8 Implementation (XC7000)

INV, INV4, INV8, and INV16

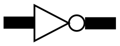
Single and Multiple Inverters



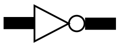
X3788



X3795



X3807



X3819

Name	XC7000	XC9000
INV	Primitive	Primitive
INV4, INV8, INV16	Macro	Macro

These single and multiple inverters identify signal inversions in a schematic.

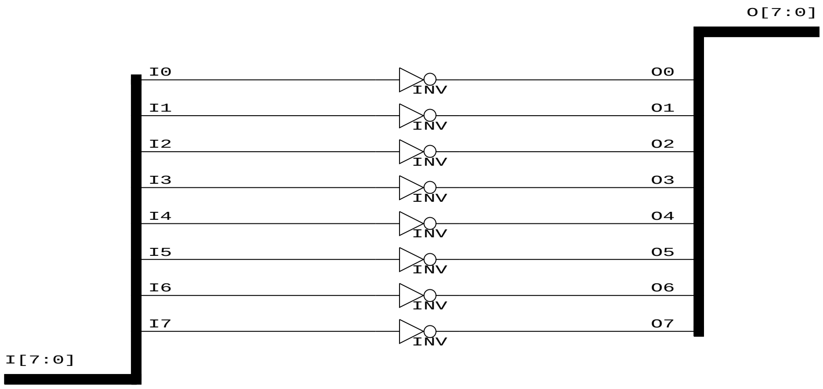
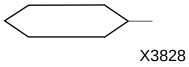
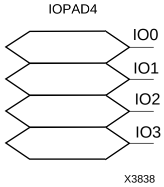


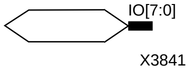
Figure 3-110 INV8 Implementation (XC7000 and XC9000)

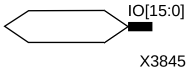
IOPAD, IOPAD4, IOPAD8, and IOPAD16

Input/Output Pads









Name	XC7000	XC9000
IOPAD	Primitive	Primitive
IOPD4, IOPAD8, IOPAD16	Macro	Macro

IOPAD, IOPAD4, IOPAD8, and IOPAD16 are single and multiple input/output pads. The IOPAD is a connection point from a device pin, used as a bidirectional signal, to a PLD device. The IOPAD is connected internally to an input/output block (IOB), which is configured by the XACT software as a bidirectional block. Bidirectional blocks can consist of any combinations of a 3-state output buffer (such as OBUFT or OFDE) and any available input buffer (such as IBUF or IFD).

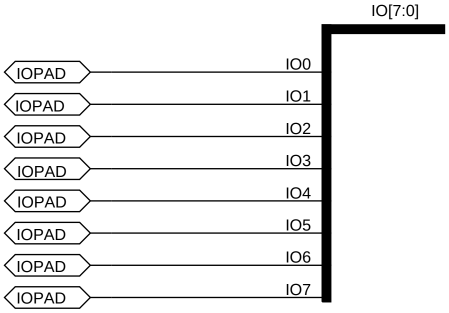
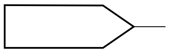


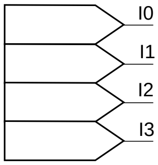
Figure 3-111   IOPAD8 Implementation (XC7000 and XC9000)

IPAD, IPAD4, IPAD8, and IPAD16

Single- and Multiple-Input Pads



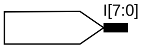
X3827



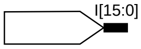
X3837

Name	XC7000	XC9000
IPAD	Primitive	Primitive
IPAD4, IPAD8, IPAD16	Macro	Macro

IPAD, IPAD4, IPAD8, and IPAD16 are single and multiple input pads (IPADs). The IPAD is a connection point from a device pin used for an input signal to the PLD device. It is connected internally to an input/output block (IOB), which is configured by the XACT software as an IBUF, IFD or ILD.



X3840



X3844

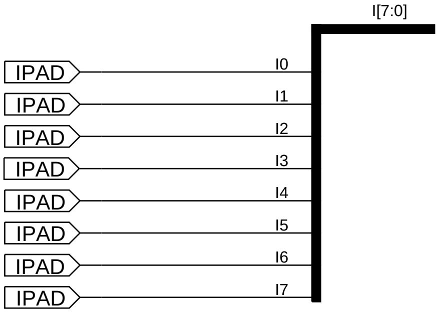
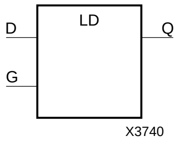
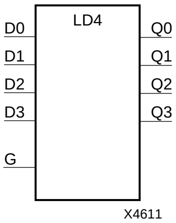


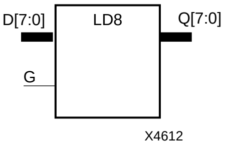
Figure 3-112 IPAD8 Implementation (XC7000 and XC9000)

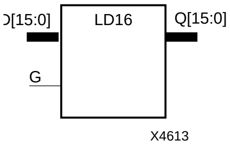
LD, LD4, LD8, and LD16

Single and Multiple Transparent Data Latches









Element	XC7000	XC9000
LD LD4, LD8, LD16	Macro*	Macro

* not supported for XC7336 designs

The data output (Q) of the latch reflects the data (D) input while the gate enable (G) input is High. The data on the D input during the High-to-Low gate transition is stored in the latch. The data on the Q output remains unchanged as long as G remains Low. LD4, LD8, and LD16 have 4, 8, and 16 transparent latches, respectively, with a common Gate enable (G).

The default initial state of all latches is zero. For XC7000 and XC9000 designs, the G input may not be driven by a global clock signal (BUFG).

Inputs		Outputs
G	D	Q
1	0	0
1	1	1
0	X	No Chg
↓	D	d

d = state of input one set-up time prior to High-to-Low gate transition

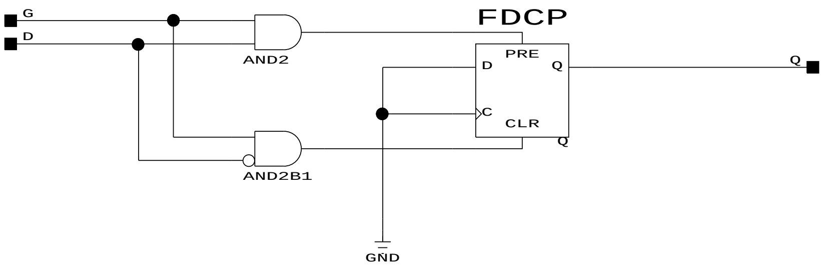


Figure 3-113 LD Implementation (XC7000 and XC9000)

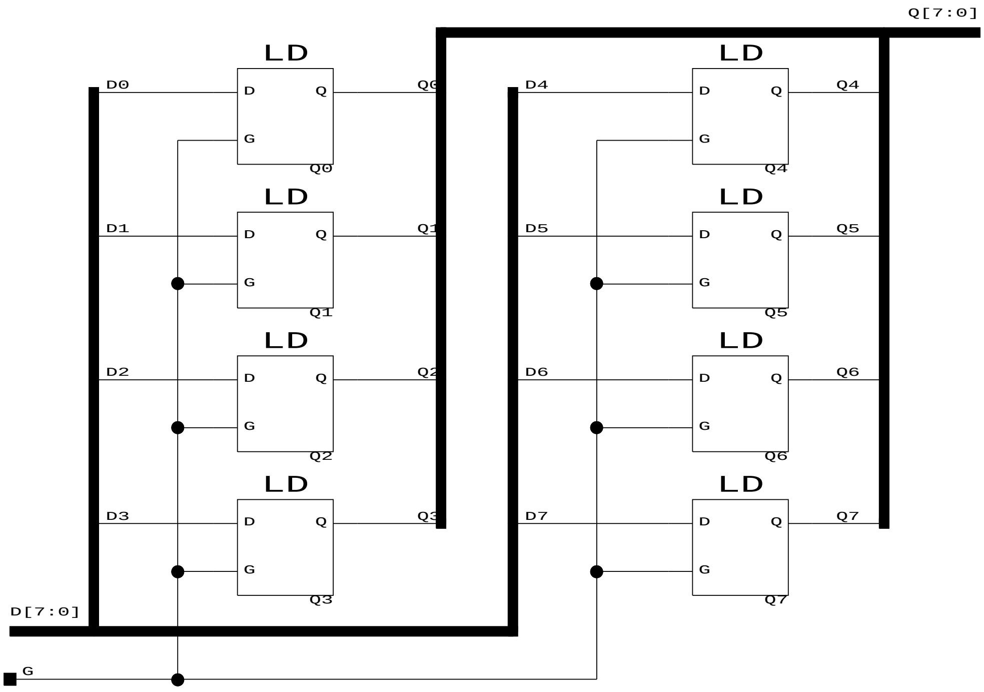
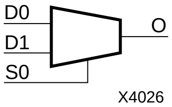


Figure 3-114 LD8 Implementation (XC7000 and XC9000)

M2_1

2-to-1 Multiplexer



XC7000	XC9000
Macro	Macro

The M2_1 multiplexer chooses one data bit from two sources (D1 or D0) under the control of the select input (S0). The output (O) reflects the state of the selected data input. When Low, S0 selects D0 and when High, S0 selects D1.

Inputs			Outputs
S0	D1	D0	O
1	1	X	1
1	0	X	0
0	X	1	1
0	X	0	0

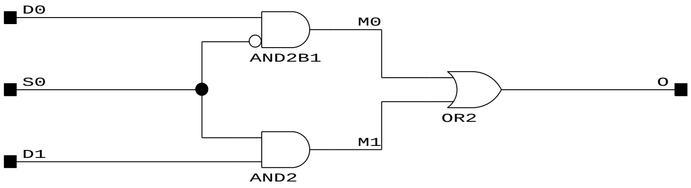
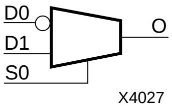


Figure 3-115 M2_1 Implementation (XC7000 and XC9000)

M2_1B1

2-to-1 Multiplexer with D0 Inverted



XC7000	XC9000
Macro	Macro

The M2_1B1 multiplexer chooses one data bit from two sources (D1 or D0) under the control of select input (S0). When S0 is Low, the output (O) reflects the state of $\overline{D0}$. When S0 is High, O reflects the state of D1.

Inputs			Outputs
S0	D1	D0	O
1	1	X	1
1	0	X	0
0	X	1	0
0	X	0	1

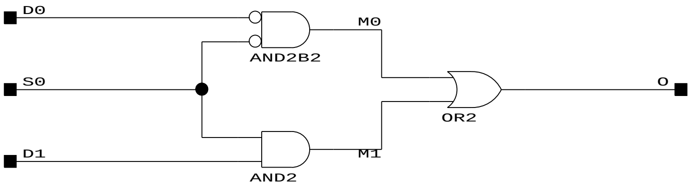
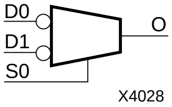


Figure 3-116 M2_1B1 Implementation (XC7000 and XC9000)

M2_1B2

2-to-1 Multiplexer with D0 and D1 Inverted



XC7000	XC9000
Macro	Macro

The M2_1B2 multiplexer chooses one data bit from two sources (D1 or D0) under the control of select input (S0). When S0 is Low, the output (O) reflects the state of $\overline{D0}$. When S0 is High, O reflects the state of $\overline{D1}$.

Inputs			Outputs
S0	D1	D0	O
1	1	X	0
1	0	X	1
0	X	1	0
0	X	0	1

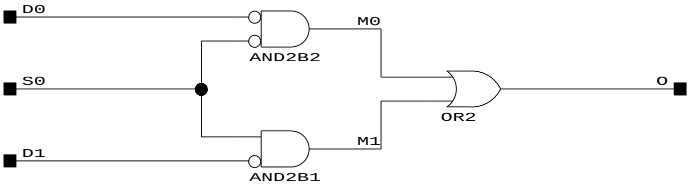
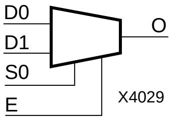


Figure 3-117 M2_1B2 Implementation (XC7000 and XC9000)

M2_1E

2-to-1 Multiplexer with Enable



XC7000	XC9000
Macro	Macro

When the enable input (E) is High, the M2_1E chooses one data bit from two sources (D1 or D0) under the control of select input (S0). When E is High, the output (O) reflects the state of the selected input. When Low, S0 selects D0 and when High, S0 selects D1. When E is Low, the output is Low.

Inputs				Outputs
E	S0	D1	D0	O
0	X	X	X	0
1	0	X	1	1
1	0	X	0	0
1	1	1	X	1
1	1	0	X	0

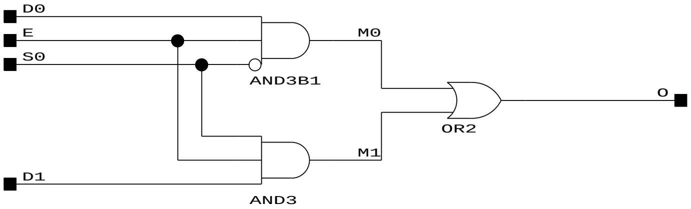
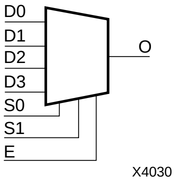


Figure 3-118 M2_1E Implementation (XC7000 and XC9000)

M4_1E

4-to-1 Multiplexer with Enable



XC7000	XC9000
Macro	Macro

When the enable input (E) is High, the M4_1E multiplexer chooses one data bit from four sources (D3, D2, D1, or D0) under the control of the select inputs (S1 – S0). The output (O) reflects the state of the selected input as shown in the truth table. When E is Low, the output is Low.

Inputs							Outputs
E	S1	S0	D0	D1	D2	D3	O
0	X	X	X	X	X	X	0
1	0	0	D0	X	X	X	D0
1	0	1	X	D1	X	X	D1
1	1	0	X	X	D2	X	D2
1	1	1	X	X	X	D3	D3

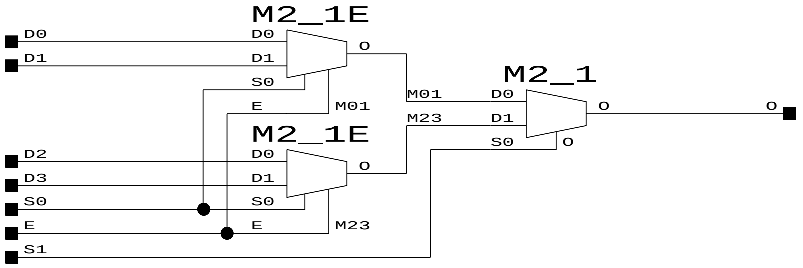
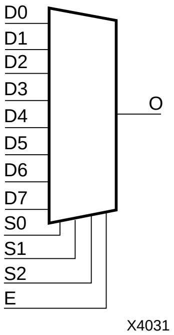


Figure 3-119 M4_1E Implementation (XC7000 and XC9000)

M8_1E

8-to-1 Multiplexer with Enable



XC7000	XC9000
Macro	Macro

When the enable input (E) is High, the M8_1E multiplexer chooses one data bit from eight sources (D7 – D0) under the control of the select inputs (S2 – S0). The output (O) reflects the state of the selected input as shown in the truth table. When E is Low, the output is Low.

Inputs					Outputs
E	S2	S1	S0	D7 – D0	O
0	X	X	X	X	0
1	0	0	0	D0	D0
1	0	0	1	D1	D1
1	0	1	0	D2	D2
1	0	1	1	D3	D3
1	1	0	0	D4	D4
1	1	0	1	D5	D5
1	1	1	0	D6	D6
1	1	1	1	D7	D7

Dn represents signal on the Dn input; all other data inputs are don’t-cares (X).

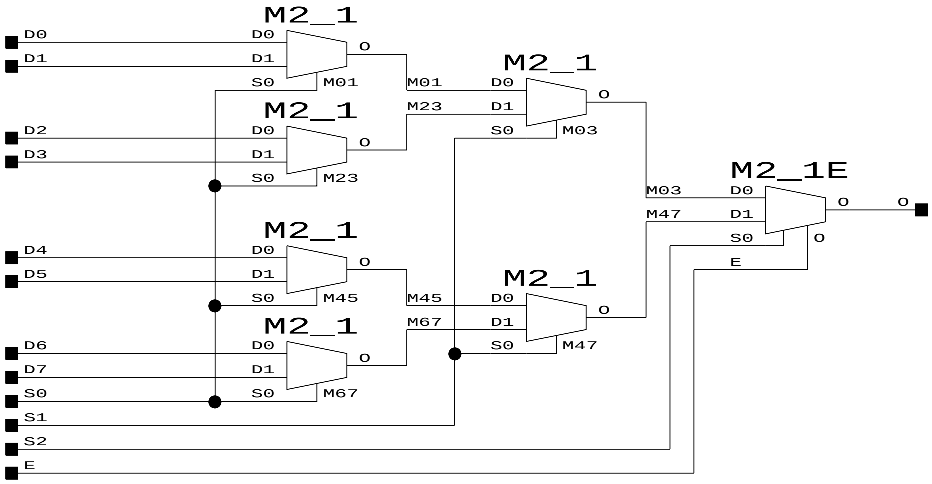
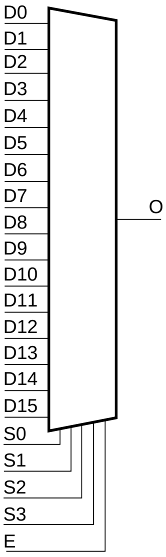


Figure 3-120 M8_1E Implementation (XC7000 and XC9000)

M16_1E

16-to-1 Multiplexer with Enable



XC7000	XC9000
Macro	Macro

When the enable input (E) is High, the M16_1E multiplexer chooses one data bit from 16 sources (D15 – D0) under the control of the select inputs (S3 – S0). The output (O) reflects the state of the selected input as shown in the truth table. When E is Low, the output is Low.

Inputs						Outputs
E	S3	S2	S1	S0	D15 – D0	O
0	X	X	X	X	X	0
1	0	0	0	0	D0	D0
1	0	0	0	1	D1	D1
1	0	0	1	0	D2	D2
1	0	0	1	1	D3	D3
. . .	. . .	. . .	. . .	. . .	. . .	. . .
1	1	1	0	0	D12	D12
1	1	1	0	1	D13	D13
1	1	1	1	0	D14	D14
1	1	1	1	1	D15	D15

Dn represents signal on the Dn input; all other data inputs are don’t-cares (X).

NAND

2- to 9-Input NAND Gates with Inverted and Non-Inverted Inputs

Name	XC7000	XC9000
NAND2 – NAND4B4	Primitive	Primitive
NAND5 – NAND5B5	Primitive	Primitive
NAND6 – NAND9	Primitive	Primitive

NAND functions of up to five inputs are available in any combination of inverting and non-inverting inputs. NAND functions of six to nine inputs are available with only non-inverting inputs. To invert some or all inputs, use external inverters.

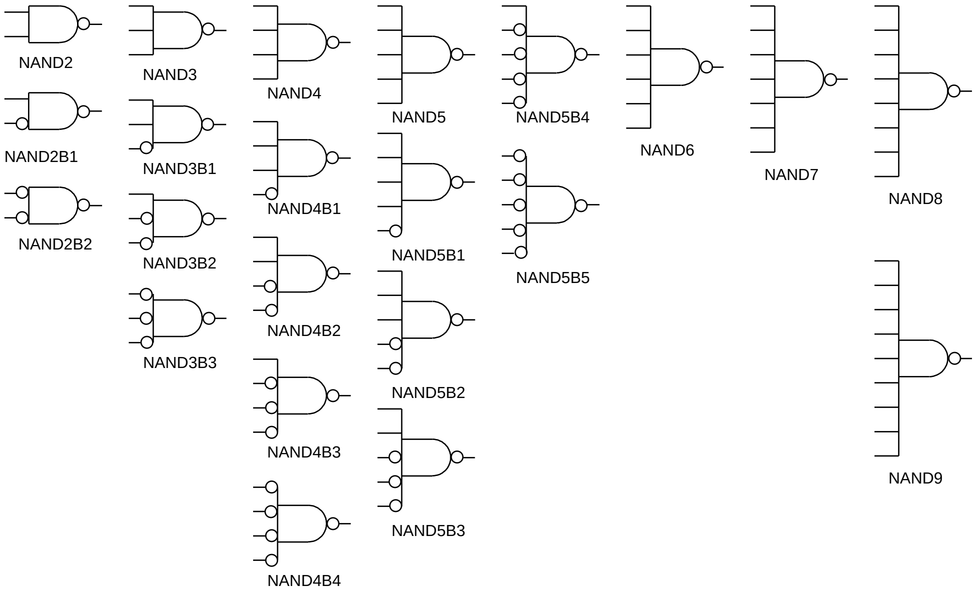


Figure 3-121 NAND Gate Representations

NOR

2- to 9-Input NOR Gates with Inverted and Non-Inverted Inputs

Name	XC7000	XC9000
NOR2 – NOR4B4	Primitive	Primitive
NOR5 – NOR5B5	Primitive	Primitive
NOR6 – NOR9	Primitive	Primitive

NOR functions of up to five inputs are available in any combination of inverting and non-inverting inputs. NOR functions of six to nine inputs are available with only non-inverting inputs. To invert some or all inputs, use external inverters.

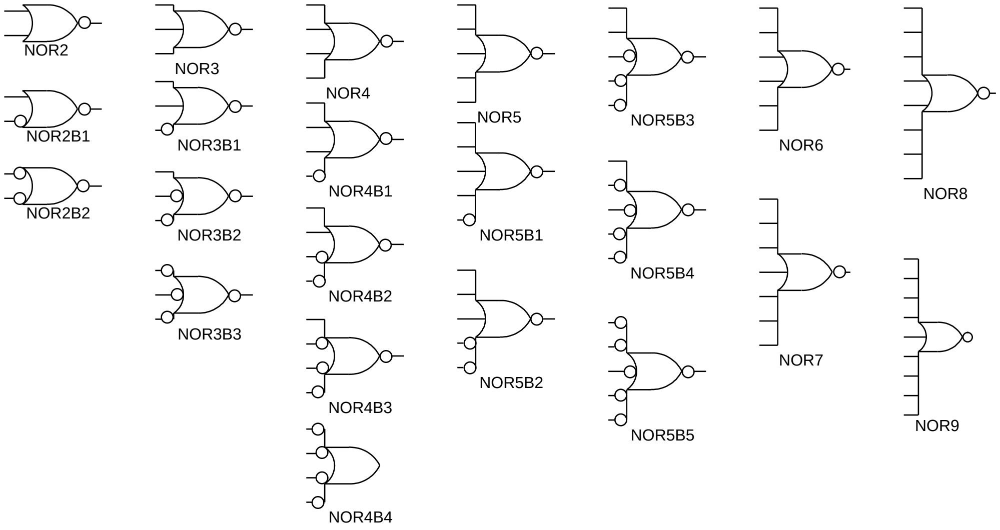
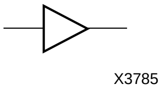
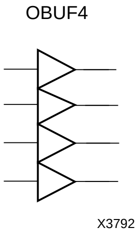


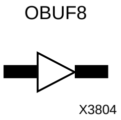
Figure 3-122 NOR Gate Representations

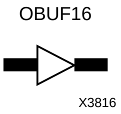
OBUF, OBUF4, OBUF8, and OBUF16

Single- and Multiple-Output Buffers









Name	XC7000	XC9000
OBUF	Primitive	Primitive
OBUF4, OBUF8, OBUF16	Macro	Macro

OBUF, OBUF4, OBUF8, and OBUF16 are single and multiple output buffers. An OBUF isolates the internal circuit and provides drive current for signals leaving a chip. OBUFs exist in input/output blocks (IOB). The output (O) of an OBUF is connected to an OPAD or an IOPAD. For XC7000 and XC9000 CPLDs, if a high impedance (Z) signal from an on-chip 3-state buffer (like BUFE) is applied to the input of an OBUF, it is propagated to the CPLD device output pin.

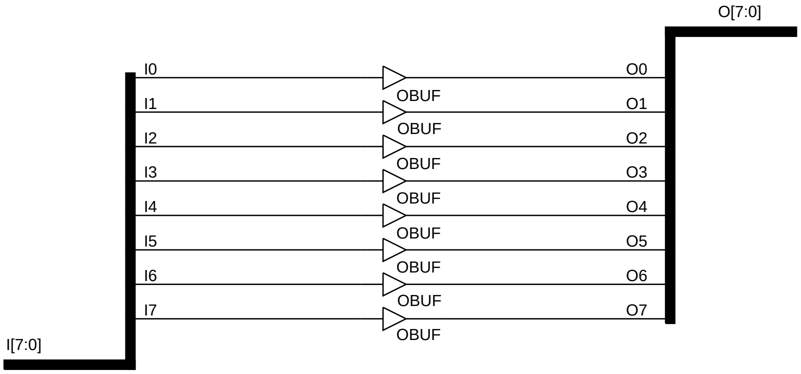
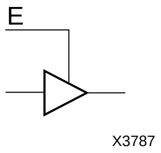


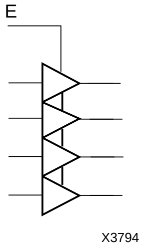
Figure 3-123 OBUF8 XC7000 Implementation

OBUFE, OBUFE4, OBUFE8, and OBUFE16

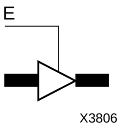
3-State Output Buffers with Active-High Output Enable



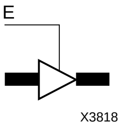
OBUFE4



OBUFE8



OBUFE16



XC7000	XC9000
Macro	Macro

OBUFE, OBUFE4, OBUFE8, and OBUFE16 are single or multiple 3-state buffers with inputs I, I3 – I0, I7 – I0, and so forth, outputs O, O3 – O0, O7 – O0, and so forth, and active-High output enable (E). When E is High, data on the inputs of the buffers is transferred to the corresponding outputs. When E is Low, the output is High impedance (off or Z state). An OBUFE isolates the internal circuit and provides drive current for signals leaving a chip. An OBUFE output is connected to an OPAD or an IOPAD. An OBUFE input is connected to the internal circuit.

Inputs		Outputs
E	I	O
0	X	Z
1	1	1
1	0	0

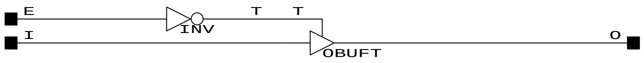
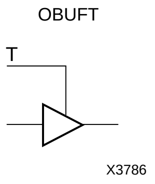


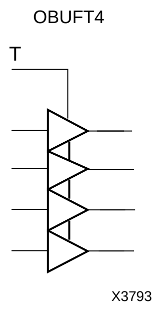
Figure 3-124 OBUFE Implementation (XC7000 and XC9000)

OBUFT, OBUFT4, OBUFT8, and OBUFT16

Single and Multiple 3-State Output Buffers with Active-Low Output Enable

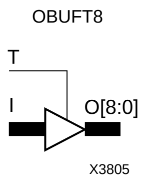


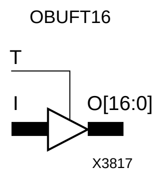
Name	XC7000	
OBUFT	Primitive	Primitive
OBUFT4, OBUFT8, OBUFT16	Macro	Macro



OBUFT, OBUFT4, OBUFT8, and OBUFT16 are single and multiple 3-state output buffers with inputs I, I3 – I0, I7 – I0, I15 – I0, outputs O, O3 – O0, O7 – O0, O15 – O0, and active-Low output enables (T). When T is Low, data on the inputs of the buffers is transferred to the corresponding outputs. When T is High, the output is high impedance (off or Z state). OBUFTs isolate the internal circuit and provide extra drive current for signals leaving a chip. An OBUFT output is connected to an OPAD or an IOPAD.

Inputs		Outputs
T	I	O
1	X	Z
0	1	1
0	0	0





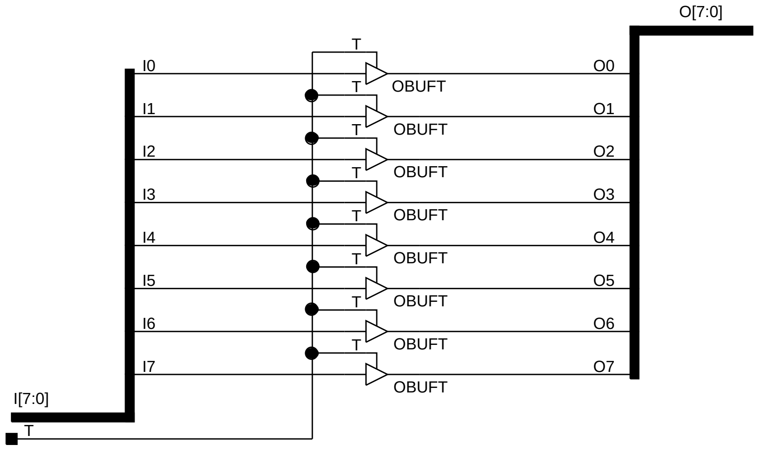
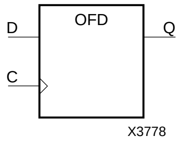


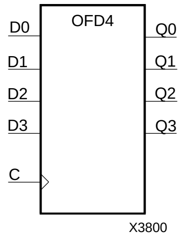
Figure 3-125 OBUFT8 Implementation (XC7000 and XC9000)

OFD, OFD4, OFD8, and OFD16

Single- and Multiple-Output D Flip-Flops

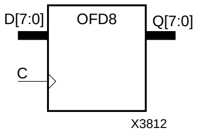


Name	XC7000	XC9000
OFD OFD4, OFD8, OFD16	Macro	Macro



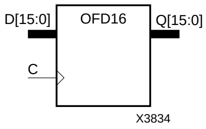
OFD, OFD4, OFD8, and OFD16 are single and multiple output D flip-flops. The outputs (for example, Q3 – Q0) are connected to OPADs or IOPADs. The data on the D inputs is loaded into the flip-flops during the Low-to-High clock (C) transition and appears on the Q outputs.

The default initial state of all flip-flops is zero.



Inputs		Outputs
D	C	Q
D	↑	dn

dn = state of referenced input one set-up time prior to active clock transition



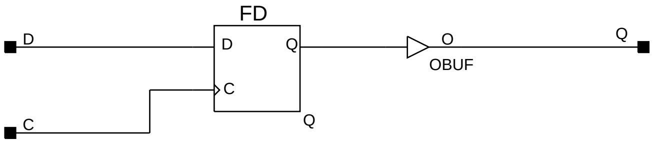


Figure 3-126 OFD Implementation (XC7000 and XC9000)

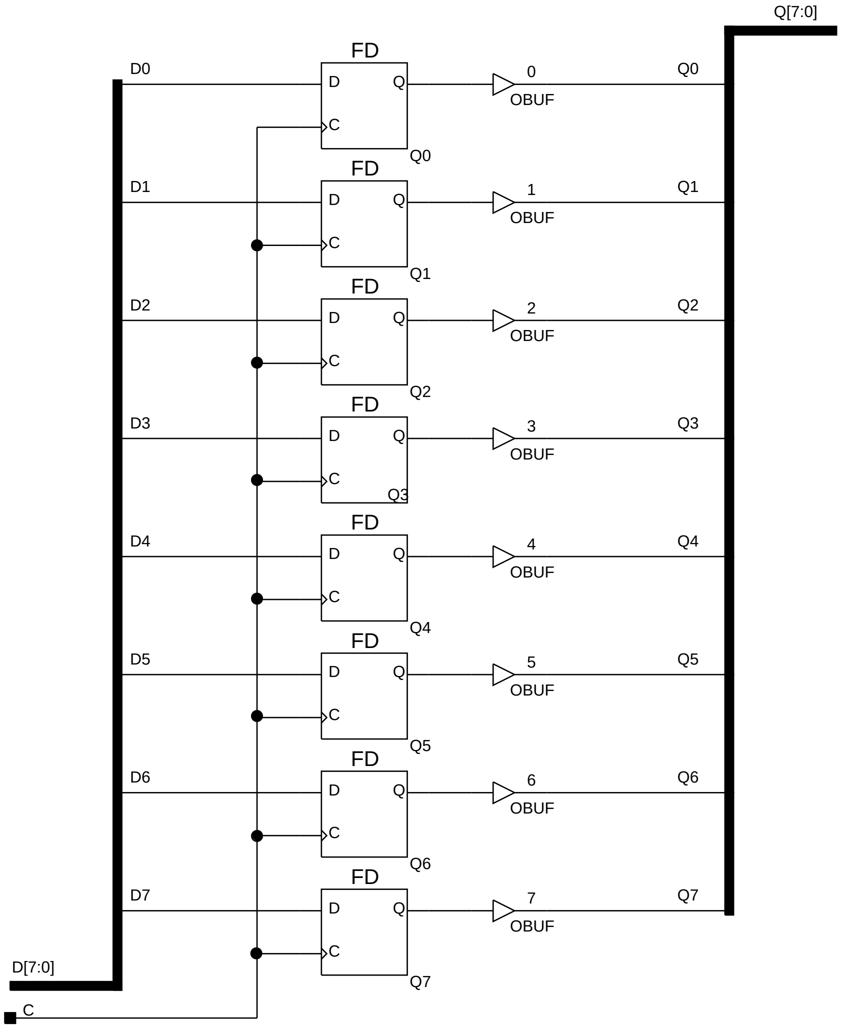
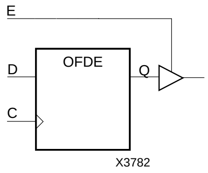


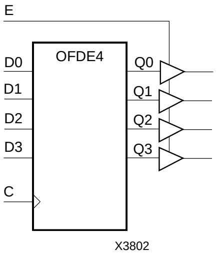
Figure 3-127 OFD8 Implementation (XC7000 and XC9000)

OFDE, OFDE4, OFDE8, and OFDE16

D Flip-Flops with Active-High Enable Output Buffers

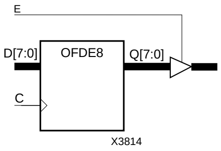


Name	XC7000	XC9000
OFDE OFDE4, OFDE8, OFDE16	Macro	Macro

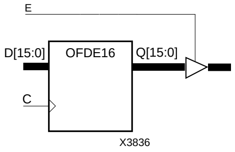


OFDE, OFDE4, OFDE8, and OFDE16 are single or multiple D flip-flops whose outputs are enabled by 3-state buffers. The flip-flop data outputs (Q) are connected to the inputs of output buffers (OBUFE). The OBUFE outputs (O) are connected to OPADs or IOPADs. The data on the data inputs (D) is loaded into the flip-flops during the Low-to-High clock (C) transition. When the active-High enable inputs (E) are High, the data on the flip-flop outputs (Q) appears on the O outputs. When E is Low, outputs are high impedance (Z state or off).

The default initial state of all flip-flops is zero.



Inputs			Outputs
E	D	C	O
0	X	X	Z, not off
1	1	↑	1
1	0	↑	0



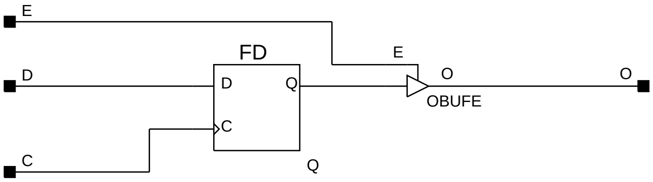


Figure 3-128 OFDE Implementation (XC7000 and XC9000)

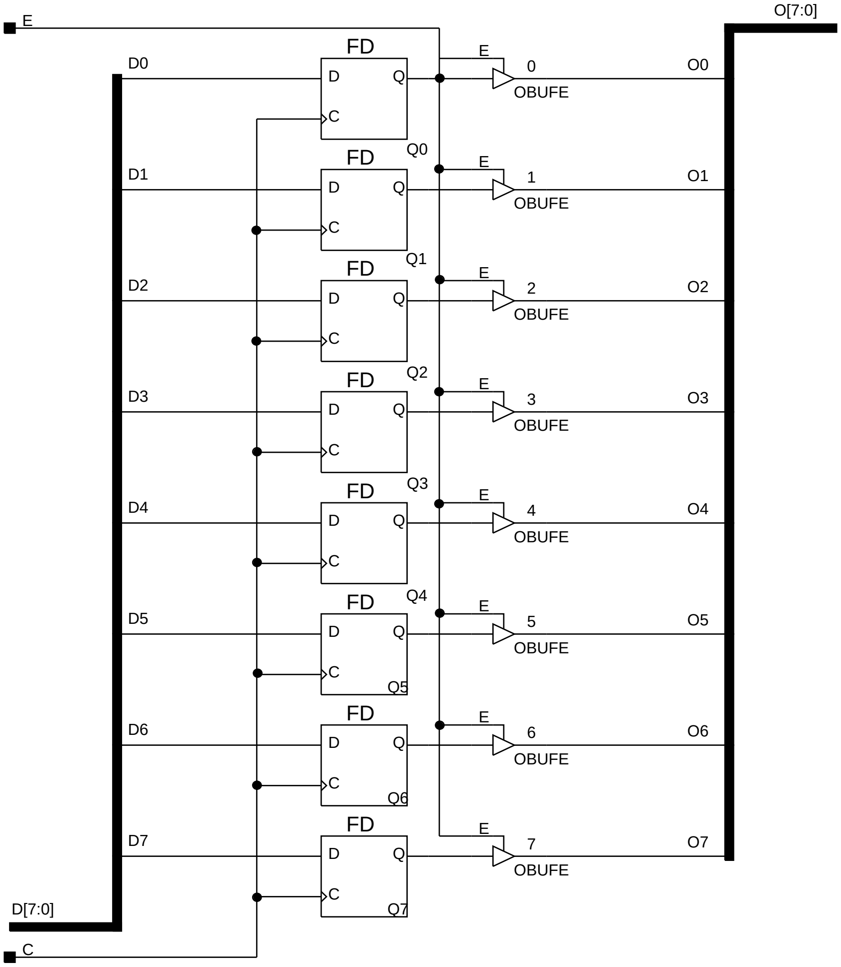
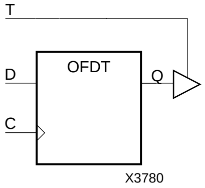


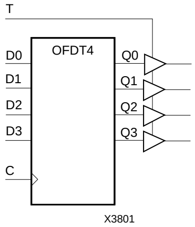
Figure 3-129 OFDE8 Implementation (XC7000 and XC9000)

OFDT, OFDT4, OFDT8, and OFDT16

Single and Multiple D Flip-Flops with Active-High 3-State Active-Low Output Enable Buffers

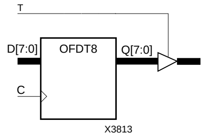


Name	XC7000	XC9000
OFDT OFDT4, OFDT8, OFDT16	Macro*	Macro

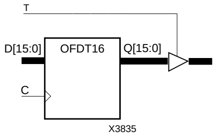


OFDT, OFDT4, OFDT8, and OFDT16 are single or multiple D flip-flops whose outputs are enabled by a 3-state buffers. The data outputs (Q) of the flip-flops are connected to the inputs of output buffers (OBUFT). The outputs of the OBUFTs (O) are connected to OPADs or IOPADs. The data on the data inputs (D) is loaded into the flip-flops during the Low-to-High clock (C) transition. When the active-Low enable inputs (T) are Low, the data on the flip-flop outputs (Q) appears on the O outputs. When T is High, outputs are high impedance (off).

The default initial state of all flip-flops is zero.



Inputs			Outputs
T	D	C	O
1	X	X	Z
0	D	↑	d



d = state of referenced input one set-up time prior to active clock transition

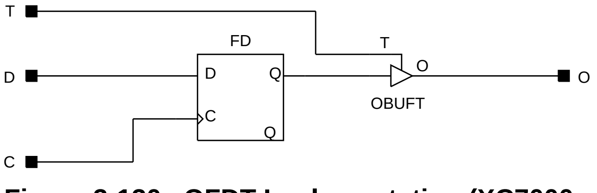


Figure 3-130 OFDT Implementation (XC7000 and XC9000)

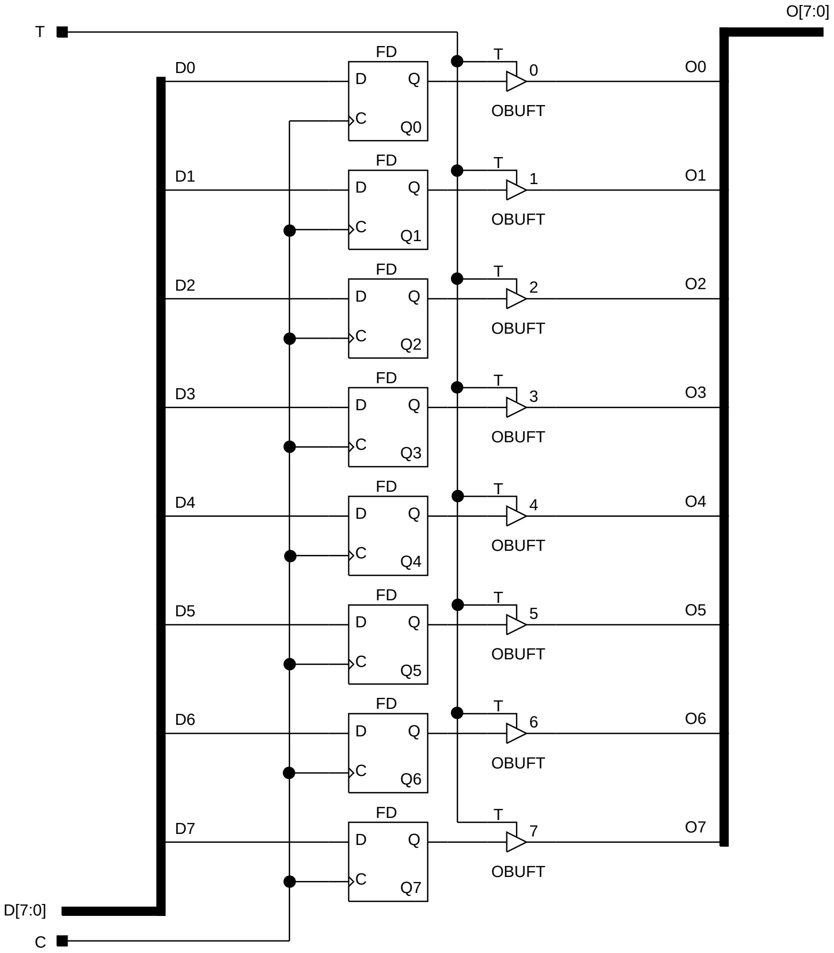
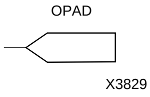
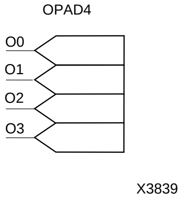


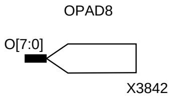
Figure 3-131 OFDT8 Implementation (XC7000 and XC9000)

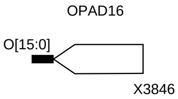
OPAD, OPAD4, OPAD8, and OPAD16

Single- and Multiple-Output Pads









Name	XC7000	XC9000
OPAD	Primitive	Primitive
OPAD4, OPAD8, OPAD16	Macro	Macro

OPAD, OPAD4, OPAD8, and OPAD16 are single and multiple output pads. An OPAD connects a device pin to an output signal of a PLD. It is internally connected to an input/output block (IOB), which is configured by the XACT software as an OBUF, an OBUFT, an OBUFE, an OFD, or an OFDT.

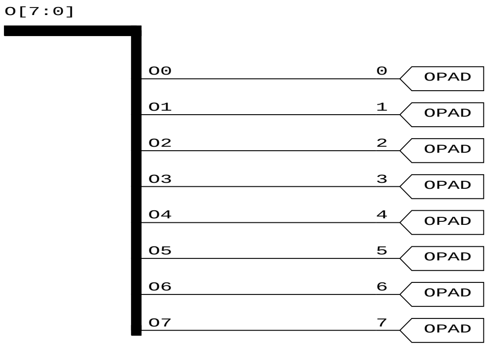


Figure 3-132 OPAD8 Implementation (XC7000 and XC9000)

OR

2- to 9-Input OR Gates with Inverted and Non-Inverted Inputs

Name	XC7000	XC9000
OR2 – OR4B4	Primitive	Primitive
OR5 – OR5B5	Primitive	Primitive
OR6 – OR9	Primitive	Primitive

OR functions of up to five inputs are available in any combination of inverting and non-inverting inputs. OR functions of six to nine inputs are available with only non-inverting inputs. To invert some or all inputs, use external inverters.

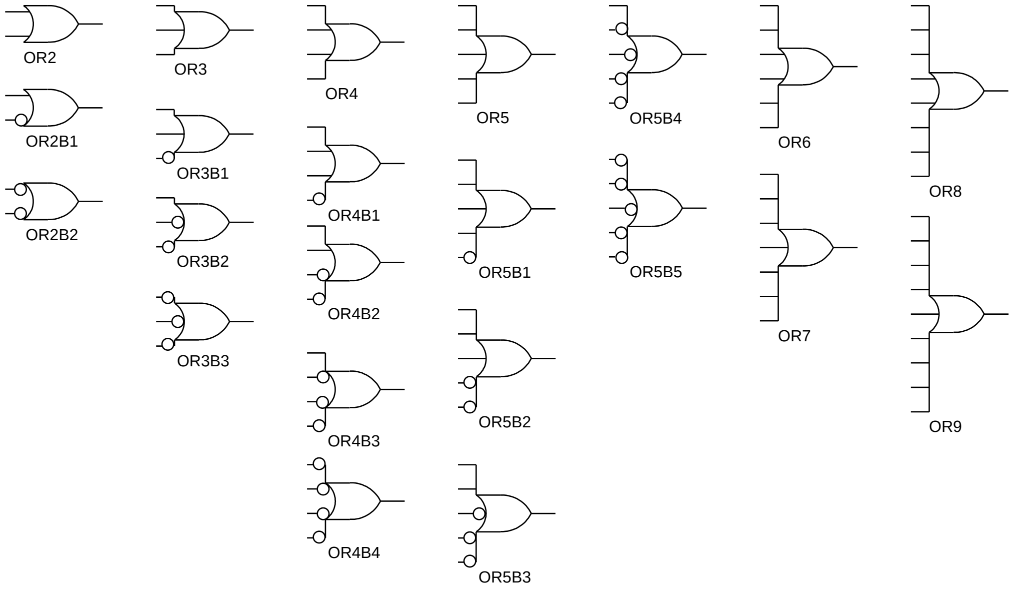


Figure 3-133 OR Gate Representations

SOP

Sum Of Products

Name	XC7000	XC9000
SOP3 – SOP3B3	Macro	Macro
SOP4 – SOP4B4	Macro	Macro

Sum Of Products macros and primitives provide common logic functions by OR gating the outputs of two AND functions or the output of one AND function with one direct input. Variations of inverting and non-inverting inputs are available.

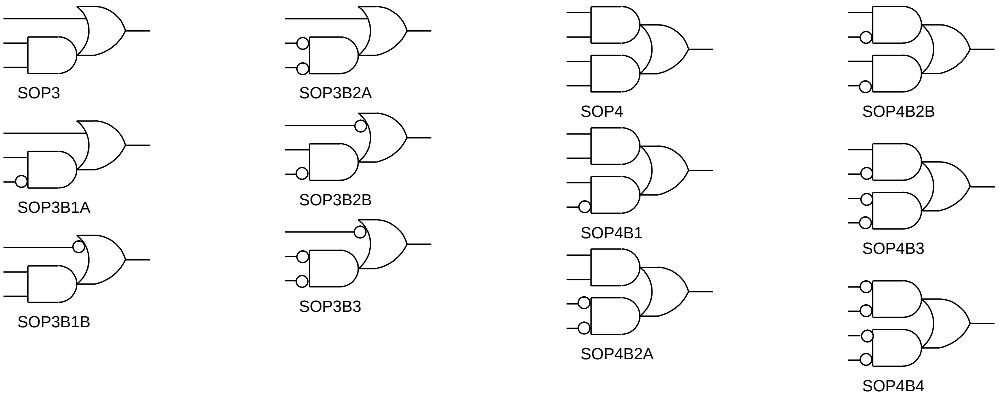
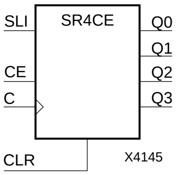


Figure 3-134 SOP Gate Representations

SR4CE

4-Bit Serial-In Parallel-Out Shift Register with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

SR4CE is a 4-bit shift register with a shift-left serial input (SLI), parallel outputs (Q3 – Q0), and clock enable (CE) and asynchronous clear (CLR) inputs. The CLR input, when High, overrides all other inputs and resets the data outputs (Q3 – Q0) Low. When CE is High and CLR is Low, the data on the SLI input is loaded into the first bit of the shift register during the Low-to-High clock (C) transition and appears on the Q0 output. During subsequent Low-to-High clock transitions, when CE is High and CLR is Low, data is shifted to the next highest bit position as new data is loaded into Q0 (SLI➝Q0, Q0➝Q1, Q1➝Q2, and so forth). The register ignores clock transitions when CE is Low.

Registers can be cascaded by connecting the Q3 output of one stage to the SLI input of the next stage and connecting clock, CE, and CLR in parallel.

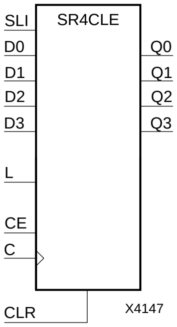
The default initial state of all flip-flops is zero.

Inputs				Outputs	
CLR	CE	SLI	C	Q0	Q3 – Q1
1	X	X	X	0	0
0	0	X	X	---No Change---	
0	1	1	↑	1	qn-1
0	1	0	↑	0	qn-1

qn-1 = state of referenced output one set-up time prior to active clock transition

SR4CLE

4-Bit Loadable Serial/Parallel-In Parallel-Out Shift Register with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

SR4CLE is a 4-bit shift register with a shift-left serial input (SLI), parallel inputs (D3 – D0), parallel outputs (Q3 – Q0), and three control inputs – clock enable (CE), load enable (L), and asynchronous clear (CLR).The register ignores clock transitions when L and CE are Low. The asynchronous CLR, when High, overrides all other inputs and resets the data outputs (Q3 – Q0) Low. When L is High and CLR is Low, data on the D3 – D0 inputs is loaded into the corresponding Q3 – Q0 bits of the register. When CE is High and L and CLR are Low, data on the SLI input is loaded into the first bit of the shift register during the Low-to-High clock (C) transition and appears on the Q0 output. During subsequent clock transitions, when CE is High and L and CLR are Low, the data is shifted to the next highest bit position as new data is loaded into Q0 (SLI→Q0, Q0→Q1, Q1→Q2, and so forth).

Registers can be cascaded by connecting the Q3 output of one stage to the SLI input of the next stage and connecting clock, CE, L, and CLR inputs in parallel.

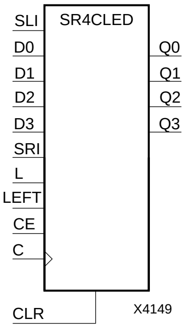
The default initial state of all flip-flops is zero.

Inputs						Outputs	
CLR	**L**	**CE**	**SLI**	**D3 – D0**	**C**	**Q0**	**Q3 – Q1**
1	X	X	X	X	X	0	0
0	1	X	X	D3 – D0	↑	d0	dn
0	0	1	SLI	X	↑	SLI	qn-1
0	0	0	X	X	X	--No Change--	

dn or qn-1 = state of referenced input or output one set-up time prior to active clock transition

SR4CLED

4-Bit Shift Register with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

SR4CLED is a 4-bit shift register with shift-left (SLI) and shift-right (SRI) serial inputs, four parallel inputs (D3 – D0), and four control inputs – clock enable (CE), load enable (L), shift left / right (LEFT), and asynchronous clear (CLR). The register ignores clock transitions when CE and L are Low. The asynchronous clear, when High, overrides all other inputs and resets the data outputs (Q3 – Q0) Low. When L is High and CLR is Low, the data on the D3 – D0 inputs is loaded into the corresponding Q3 – Q0 bits of the register. When CE is High and L and CLR are Low, data is shifted right or left, depending on the state of the LEFT input. If LEFT is High, data on the SLI is loaded into Q0 during the Low-to-High clock transition and shifted left (to Q1, Q2, and so forth) during subsequent clock transitions. If LEFT is Low, data on the SRI is loaded into Q3 during the Low-to-High clock transition and shifted right (to Q2, Q1, and so forth) during subsequent clock transitions. The truth table indicates the state of the Q3 – Q0 outputs under all input conditions.

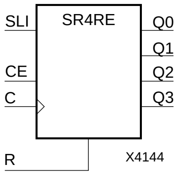
The default initial state of all flip-flops is zero.

Inputs								Outputs		
CLR	L	CE	LEFT	SLI	SRI	D3 – D0	C	Q0	Q3	Q2 – Q1
1	X	X	X	X	X	X	X	0	0	0
0	1	X	X	X	X	D3 – D0	↑	d0	d3	dn
0	0	0	X	X	X	X	X	----No Change-----		
0	0	1	1	SLI	X	X	↑	SLI	q2	qn-1
0	0	1	0	X	SRI	X	↑	q1	SRI	qn+1

dn, qn-1 or qn+1 = state of referenced input one set-up time prior to active clock transition

SR4RE

4-Bit Serial-In Parallel-Out Shift Register with Clock Enable and Synchronous Reset



XC7000	XC9000
Macro	Macro

SR4RE is a 4-bit shift register with shift-left serial input (SLI), parallel outputs (Q3 – Q0), clock enable (CE), and synchronous reset (R) inputs. The R input, when High, overrides all other inputs and resets the data outputs (Q3 – Q0) Low. When CE is High and R is Low, the data on the SLI is loaded into the first bit of the shift register during the Low-to-High clock (C) transition and appears on the Q0 output. During subsequent Low-to-High clock transitions, when CE is High and R is Low, data is shifted to the next highest bit position as new data is loaded into Q0 (SLI→Q0, Q0→Q1, Q1→Q2, and so forth). The register ignores clock transitions when CE is Low.

Registers can be cascaded by connecting the Q3 output of one stage to the SLI input of the next stage and connecting clock, CE, and R in parallel.

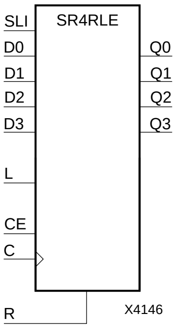
The default initial state of all flip-flops is zero.

Inputs				Outputs	
R	**CE**	**SLI**	**C**	**Q0**	**Q3 – Q1**
1	X	X	↑	0	0
0	0	X	X	---No Change---	
0	1	1	↑	1	qn-1
0	1	0	↑	0	qn-1

qn-1 = state of referenced output one set-up time prior to active clock transition

SR4RLE

4-Bit Loadable Serial/Parallel-In Parallel-Out Shift Register with Clock Enable and Synchronous Reset



XC7000	XC9000
Macro	Macro

SR4RLE is a 4-bit shift register with shift-left serial input (SLI), four parallel inputs (D3 – D0), four parallel outputs (Q3 – Q0), and three control inputs – clock enable (CE), load enable (L), and synchronous reset (R). The register ignores clock transitions when L and CE are Low. The synchronous R, when High, overrides all other inputs and resets the data outputs (Q3 – Q0) Low. When L is High and R is Low, data on the D3 – D0 inputs is loaded into the corresponding Q3 – Q0 bits of the register. When CE is High and L and R are Low, data on the SLI input is loaded into the first bit of the shift register during the Low-to-High clock (C) transition and appears on the Q0 output. During subsequent clock transitions, when CE is High and L and R are Low, the data is shifted to the next highest bit position as new data is loaded into Q0 (SLI→Q0, Q0→Q1, Q1→Q2, and so forth).

Registers can be cascaded by connecting the Q3 output of one stage to the SLI input of the next stage and connecting clock, CE, L, and R inputs in parallel.

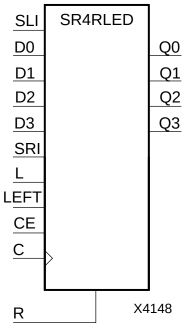
The default initial state of all flip-flops is zero.

Inputs						Outputs	
R	**L**	**CE**	**SLI**	**D3 – D0**	**C**	**Q0**	**Q3 – Q1**
1	X	X	X	X	↑	0	0
0	1	X	X	D3 – D0	↑	d0	dn
0	0	1	SLI	X	↑	SLI	qn-1
0	0	0	X	X	X	--No Change--	

dn or qn-1 = state of referenced input one set-up time prior to active clock transition

SR4RLED

4-Bit Shift Register with Clock Enable and Synchronous Reset



XC7000	XC9000
Macro	Macro

SR4RLED is a 4-bit shift register with shift-left (SLI) and shift-right (SRI) serial inputs, four parallel inputs (D3 – D0), and four control inputs – clock enable (CE), load enable (L), shift left / right (LEFT), and synchronous reset (R). The register ignores clock transitions when CE and L are Low. The synchronous R, when High, overrides all other inputs and resets the data outputs (Q3 – Q0) Low. When L is High and R is Low, the data on the D3 – D0 inputs is loaded into the corresponding Q3 – Q0 bits of the register. When CE is High and L and R are Low, data is shifted right or left, depending on the state of the LEFT input. If LEFT is High, data on SLI is loaded into Q0 during the Low-to-High clock transition and shifted left (to Q1, Q2, and so forth) during subsequent clock transitions. If LEFT is Low, data on the SRI is loaded into Q3 during the Low-to-High clock transition and shifted right (to Q2, Q1, and so forth) during subsequent clock transitions. The truth table indicates the state of the Q3 – Q0 outputs under all input conditions.

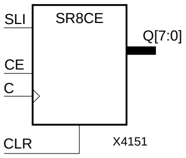
The default initial state of all flip-flops is zero.

Inputs								Outputs		
R	L	CE	LEFT	SLI	SRI	D3 – D0	C	Q0	Q3	Q2 – Q1
1	X	X	X	X	X	X	↑	0	0	0
0	1	X	X	X	X	D3 – D0	↑	d0	d3	dn
0	0	0	X	X	X	X	X	----No Change-----		
0	0	1	1	SLI	X	X	↑	SLI	q2	qn-1
0	0	1	0	X	SRI	X	↑	q1	SRI	qn+1

dn, qn-1 or qn+1 = state of referenced input one set-up time prior to active clock transition

SR8CE

8-Bit Serial-In Parallel-Out Shift Register with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

SR8CE is an 8-bit shift-left serial input (SLI), parallel output (Q7 – Q0) shift register with clock enable (CE) and asynchronous clear (CLR) inputs. The CLR input, when High, overrides all other inputs and resets the data outputs (Q7 – Q0) Low. When CE is High and CLR is Low, the data on the SLI is loaded into the first bit of the shift register during the Low-to-High clock (C) transition and appears on the Q0 output. During subsequent Low-to-High clock transitions, when CE is High and CLR is Low, data is shifted to the next highest bit position as new data is loaded into Q0 (SLI→Q0, Q0→Q1, Q1→Q2, and so forth). The register ignores clock transitions when CE is Low.

Registers can be cascaded by connecting the Q7 output of one stage to the SLI input of the next stage and connecting clock, CE, and CLR in parallel.

The default initial state of all flip-flops is zero.

Inputs				Outputs	
CLR	**CE**	**SLI**	**C**	**Q0**	**Q7 – Q1**
1	X	X	X	0	0
0	0	X	X	--No Change---	
0	1	1	↑	1	qn-1
0	1	0	↑	0	qn-1

qn-1 = state of referenced output one set-up time prior to active clock transition

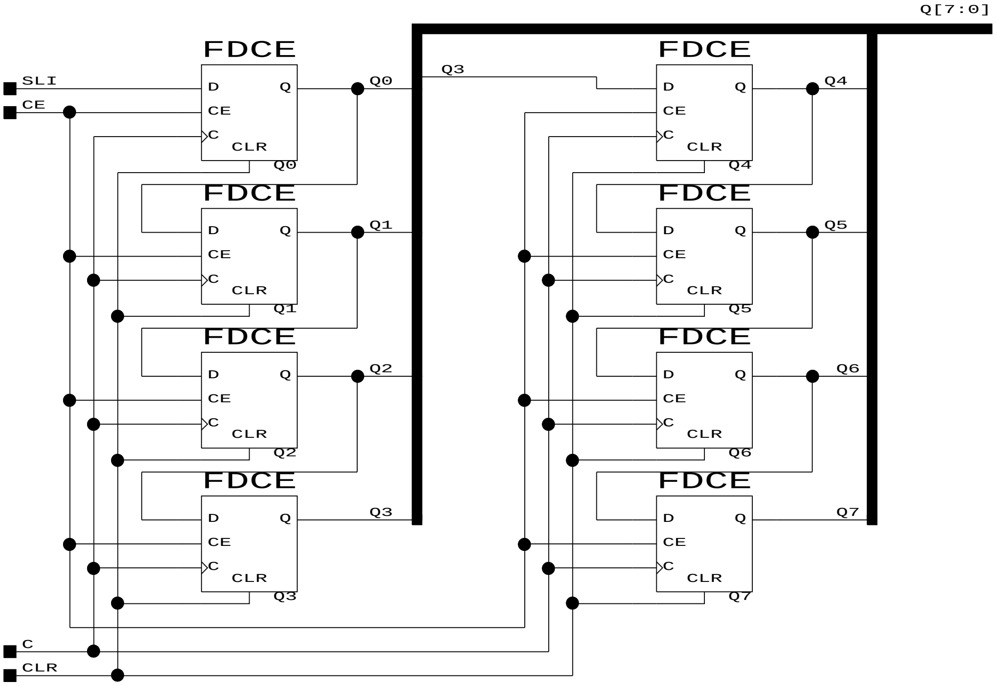
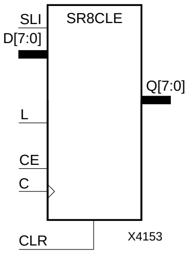


Figure 3-135 SR8CE Implementation (XC7000 and XC9000)

SR8CLE

8-Bit Loadable Serial/Parallel-In Parallel-Out Shift Register with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

SR8CLE is an 8-bit shift register with a shift-left serial input (SLI), parallel inputs (D7 – D0), parallel outputs (Q7 – Q0), and three control inputs – clock enable (CE), load enable (L) and asynchronous clear (CLR). The register ignores clock transitions when L and CE are Low. The asynchronous CLR, when High, overrides all other inputs and resets the data outputs (Q7 – Q0) Low. When L is High and CLR is Low, data on the D7 – D0 inputs is loaded into the corresponding Q7 – Q0 bits of the register. When CE is High and L and CLR are Low, data on the SLI input is loaded into the first bit of the shift register during the Low-to-High clock (C) transition and appears on the Q0 output. During subsequent clock transitions, when CE is High and L and CLR are Low, the data is shifted to the next highest bit position as new data is loaded into Q0 (SLI→Q0, Q0→Q1, Q1→Q2, and so forth).

Registers can be cascaded by connecting the Q7 output of one stage to the SLI input of the next stage and connecting clock, CE, L, and CLR inputs in parallel.

The default initial state of all flip-flops is zero.

Inputs						Outputs	
CLR	L	CE	SLI	D7 – D0	C	Q0	Q7 – Q1
1	X	X	X	X	X	0	0
0	1	X	X	D7 – D0	↑	d0	dn
0	0	1	SLI	X	↑	SLI	qn-1
0	0	0	X	X	X	--No Change--	

dn or qn-1 = state of referenced input one set-up time prior to active clock transition

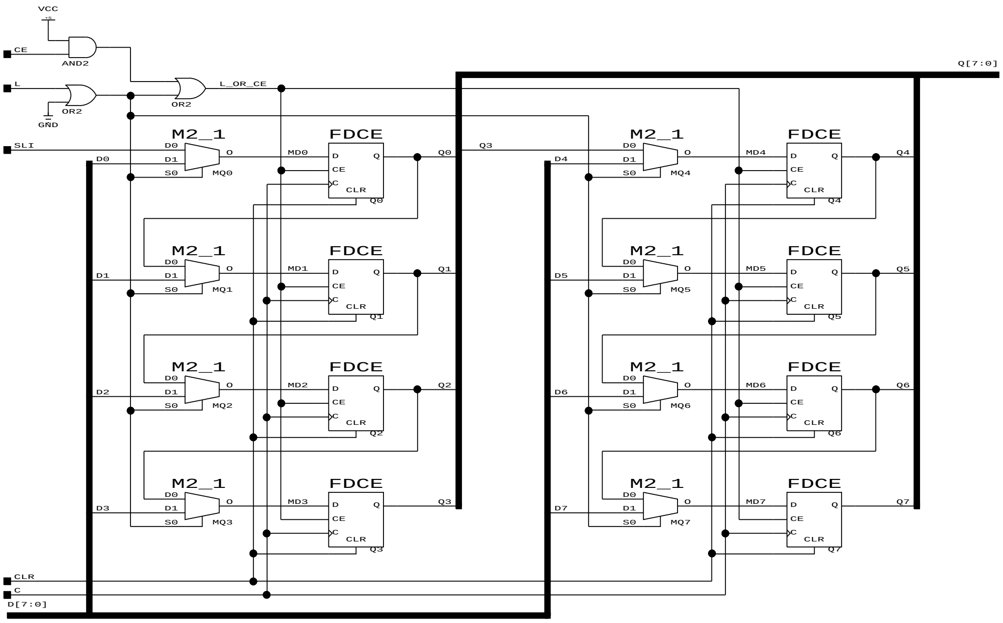
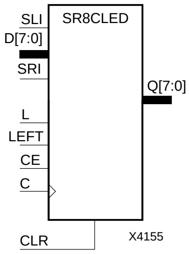


Figure 3-136 SR8CLE Implementation (XC7000 and XC9000)

SR8CLED

8-Bit Shift Register with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

SR8CLED is an 8-bit shift register with shift-left (SLI) and shift-right (SRI) serial inputs, parallel inputs (D7 – D0), and four control inputs – clock enable (CE), load enable (L), shift left / right (LEFT), and asynchronous clear (CLR). The register ignores clock transitions when CE and L are Low. The asynchronous CLR, when High, overrides all other inputs and resets the data outputs (Q7 – Q0) Low. When L is High and CLR is Low, data on the D7 – D0 inputs is loaded into the corresponding Q7 – Q0 bits of the register. When CE is High and L and CLR are Low, data is shifted right or left depending on the state of the LEFT input. If LEFT is High, data on the SLI is loaded into Q0 during the Low-to-High clock transition and shifted left (to Q1, Q2, and so forth) during subsequent clock transitions. If LEFT is Low, data on the SRI is loaded into Q7 during the Low-to-High clock transition and shifted right (to Q6, Q5, and so forth) during subsequent clock transitions. The truth table indicates the state of the Q7 – Q0 outputs under all input conditions.

The default initial state of all flip-flops is zero.

Inputs								Outputs		
CLR	**L**	**CE**	**LEFT**	**SLI**	**SRI**	**D7 – D0**	**C**	**Q0**	**Q7**	**Q6 – Q1**
1	X	X	X	X	X	X	X	0	0	0
0	1	X	X	X	X	D7 – D0	↑	d0	d7	dn
0	0	0	X	X	X	X	X	-----No Change-----		
0	0	1	1	SLI	X	X	↑	SLI	q6	qn-1
0	0	1	0	X	SRI	X	↑	q1	SRI	qn+1

dn, qn-1 or qn+1 = state of referenced input one set-up time prior to active clock transition

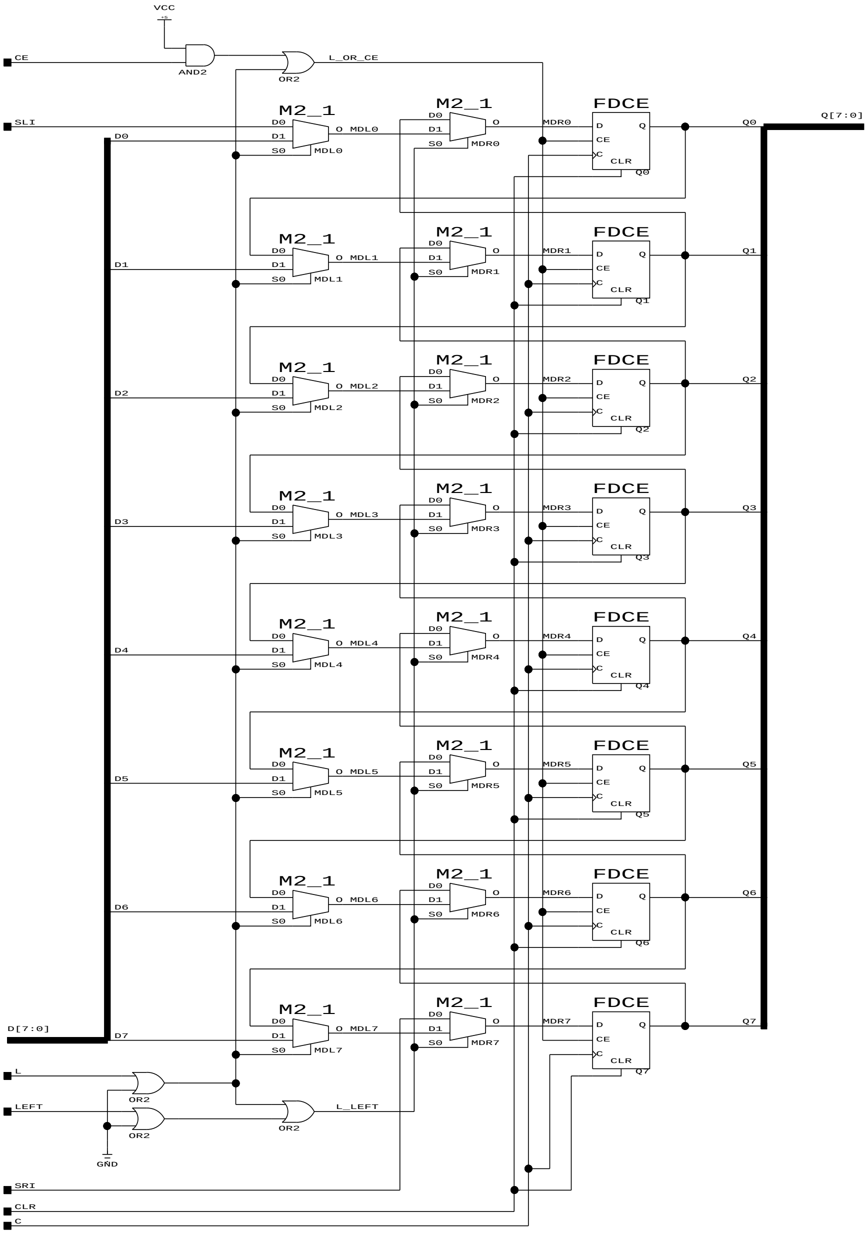
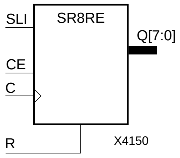


Figure 3-137 SR8CLED Implementation (XC7000 and XC9000)

SR8RE

8-Bit Serial-In Parallel-Out Shift Register with Clock Enable and Synchronous Reset



XC7000	XC9000
Macro	Macro

SR8RE is an 8-bit shift-left serial input (SLI), parallel output (Q7 – Q0) shift register with clock enable (CE) and synchronous reset (R) inputs. The R input, when High, overrides all other inputs and resets the data outputs (Q7 – Q0) Low. When CE is High and R is Low, the data on the SLI is loaded into the first bit of the shift register during the Low-to-High clock (C) transition and appears on the Q0 output. During subsequent Low-to-High clock transitions, when CE is High and R is Low, data is shifted to the next highest bit position as new data is loaded into Q0 (SLI→Q0, Q0→Q1, Q1→Q2, and so forth). The register ignores clock transitions when CE is Low.

Registers can be cascaded by connecting the Q7 output of one stage to the SLI input of the next stage and by connecting clock, CE, and R in parallel.

The default initial state of all flip-flops is zero.

Inputs				Outputs	
R	**CE**	**SLI**	**C**	**Q0**	**Q7 – Q1**
1	X	X	↑	0	0
0	0	X	X	--No Change---	
0	1	1	↑	1	qn-1
0	1	0	↑	0	qn-1

qn-1 = state of referenced output one set-up time prior to active clock transition

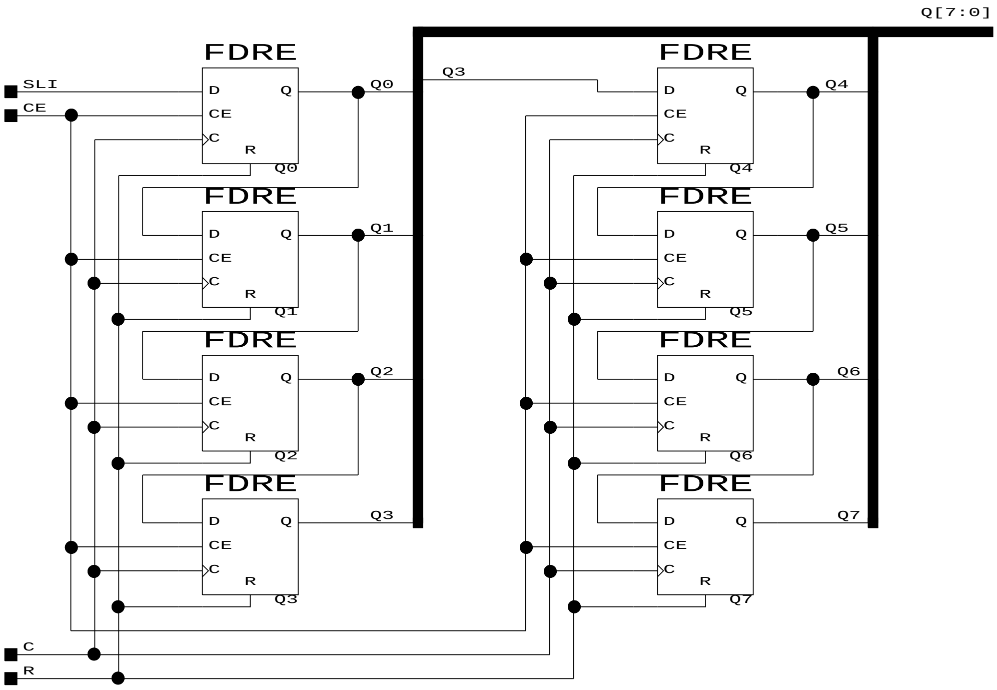
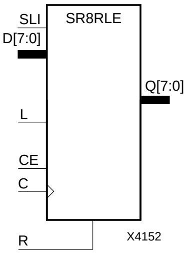


Figure 3-138 SR8RE Implementation (XC7000 and XC9000)

SR8RLE

8-Bit Loadable Serial/Parallel-In Parallel-Out Shift Register with Clock Enable and Synchronous Reset



XC7000	XC9000
Macro	Macro

SR8RLE is an 8-bit shift register with shift-left serial input (SLI), parallel inputs (D7 – D0), parallel outputs (Q7 – Q0), and three control inputs – clock enable (CE), load enable (L), and synchronous reset (R). The register ignores clock transitions when L and CE are Low. The synchronous R, when High, overrides all other inputs and resets the data outputs (Q7 – Q0) Low. When L is High and R is Low, data on the D7 – D0 inputs is loaded into the corresponding Q7 – Q0 bits of the register. When CE is High and L and R are Low, data on the SLI is loaded into the first bit of the shift register during the Low-to-High clock (C) transition and appears on the Q0 output. During subsequent clock transitions, when CE is High and L and R are Low, the data is shifted to the next highest bit position as new data is loaded into Q0 (SLI→Q0, Q0→Q1, Q1→Q2, and so forth). Registers can be cascaded by connecting the Q7 output of one stage to the SLI input of the next stage and connecting clock, CE, L, and R inputs in parallel.

The default initial state of all flip-flops is zero.

Inputs						Outputs	
R	**L**	**CE**	**SLI**	**D7 – D0**	**C**	**Q0**	**Q7 – Q1**
1	X	X	X	X	↑	0	0
0	1	X	X	D7 – D0	↑	d0	dn
0	0	1	SLI	X	↑	SLI	qn-1
0	0	0	X	X	X	--No Change--	

dn or qn-1 = state of referenced input one set-up time prior to active clock transition

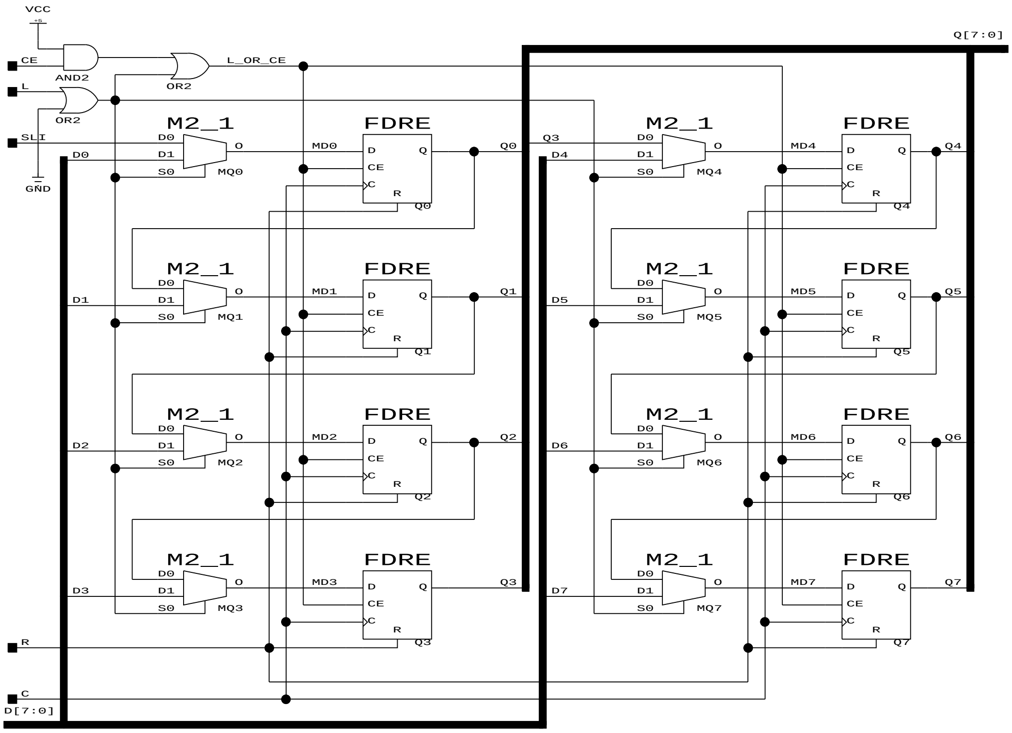
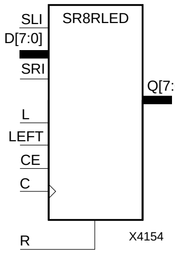


Figure 3-139 SR8RLE Implementation (XC7000 and XC9000)

SR8RLED

8-Bit Shift Register with Clock Enable and Synchronous Reset



XC7000	XC9000
Macro	Macro

SR8RLED is an 8-bit shift register with shift-left (SLI) and shift-right (SRI) serial inputs, parallel inputs (D7 – D0), and four control inputs – clock enable (CE), load enable (L), shift left / right (LEFT), and synchronous reset (R). The register ignores clock transitions when CE and L are Low. The synchronous R, when High, overrides all other inputs and resets the data outputs (Q7 – Q0) Low. When L is High and R is Low, the data on the D7 – D0 inputs is loaded into the corresponding Q7 – Q0 bits of the register. When CE is High and L and R are Low, data is shifted right or left depending on the state of the LEFT input. If LEFT is High, data on SLI is loaded into Q0 during the Low-to-High clock transition and shifted left (to Q1, Q2, and so forth) during subsequent clock transitions. If LEFT is Low, data on SRI is loaded into Q7 bit during the Low-to-High clock transition and shifted right (to Q6, Q5, and so forth) during subsequent clock transitions. The truth table indicates the state of the Q7 – Q0 outputs under all input conditions.

The default initial state of all flip-flops is zero.

Inputs								Outputs		
R	**L**	**CE**	**LEFT**	**SLI**	**SRI**	**D7– D0**	**C**	**Q0**	**Q7**	**Q6 – Q1**
1	X	X	X	X	X	X	↑	0	0	0
0	1	X	X	X	X	D7 – D0	↑	d0	d7	dn
0	0	0	X	X	X	X	X	----No Change-----		
0	0	1	1	SLI	X	X	↑	SLI	q6	qn-1
0	0	1	0	X	SRI	X	↑	q1	SRI	qn+1

dn, qn-1 or qn+1 = state of referenced input one set-up time prior to active clock transition

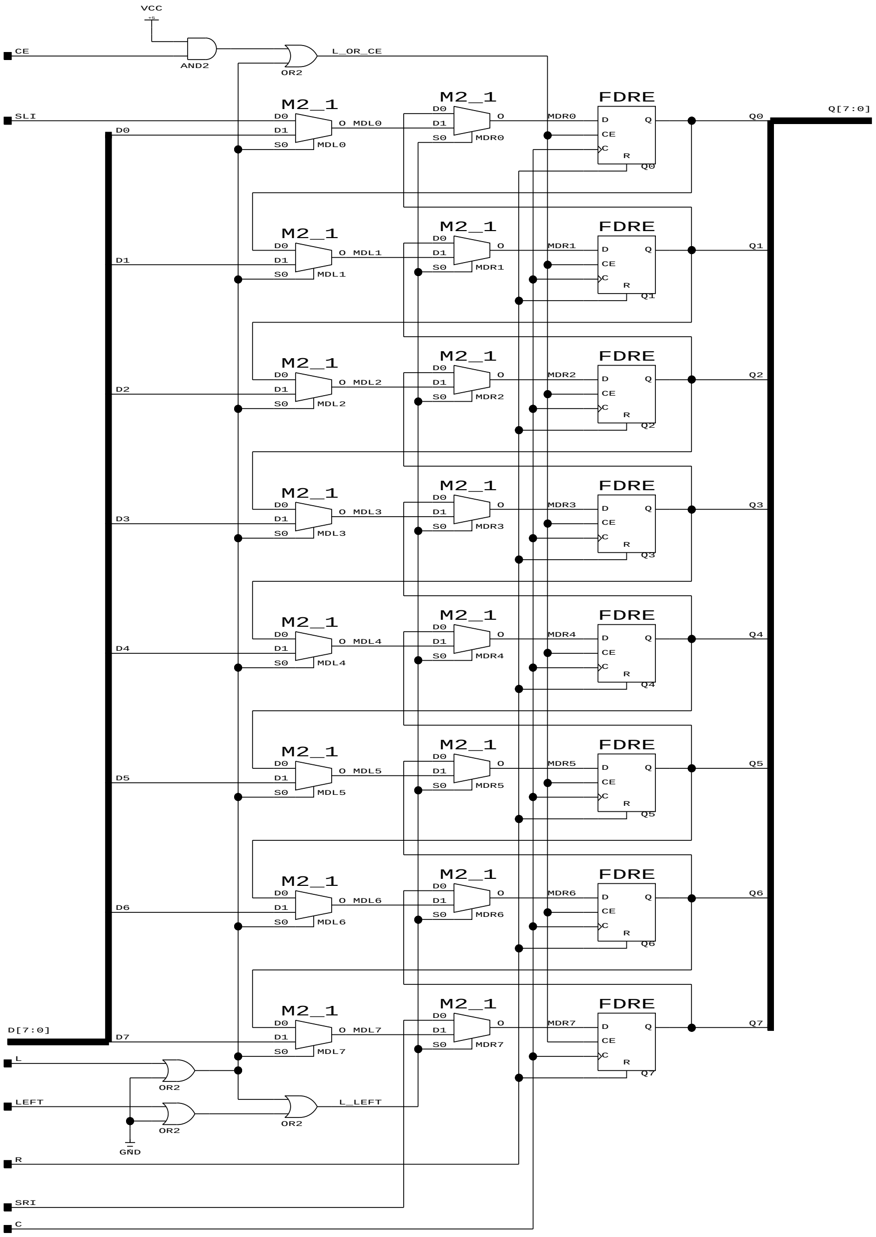
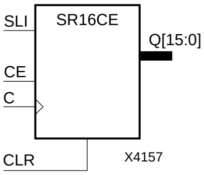


Figure 3-140 SR8RLED Implementation (XC7000 and XC9000)

SR16CE

16-Bit Serial-In Parallel-Out Shift Register with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

SR16CE is a 16-bit a shift-left serial input (SLI), parallel outputs (Q15 – Q0) shift register with clock enable (CE) and asynchronous clear (CLR) inputs. The CLR input, when High, overrides all other inputs and resets the data outputs (Q15 – Q0) Low. When CE is High and CLR is Low, the data on the SLI input is loaded into the first bit of the shift register during the Low-to-High clock (C) transition and appears on the Q0 output. During subsequent Low-to-High clock transitions, when CE is High and CLR is Low, data is shifted to the next highest bit position as new data is loaded into Q0 (SLI→Q0, Q0→Q1, Q1→Q2, and so forth). The register ignores clock transitions when CE is Low. Registers can be cascaded by connecting the Q15 output of one stage to the Shift Left Input (SLI) of the next stage and connecting clock, CE, and CLR in parallel.

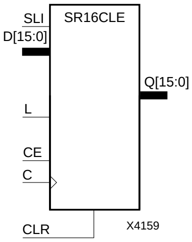
The default initial state of all flip-flops is zero.

Inputs				Outputs	
CLR	CE	SLI	C	Q0	Q15 – Q1
1	X	X	X	0	0
0	0	X	X	---No Change---	
0	1	1	↑	1	qn-1
0	1	0	↑	0	qn-1

qn-1 = state of referenced output one set-up time prior to active clock transition

SR16CLE

16-Bit Loadable Serial/Parallel-In Parallel-Out Shift Register with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

SR16CLE is a 16-bit shift register with shift-left serial input (SLI), parallel inputs (D15 – D0), parallel outputs (Q15 – Q0), and three control inputs – clock enable (CE), load enable (L), and asynchronous clear (CLR). The register ignores clock transitions when L and CE are Low. The asynchronous CLR, when High, overrides all other inputs and resets the data outputs (Q15 – Q0) Low. When L is High and CLR is Low, data on the D15 – D0 inputs is loaded into the corresponding Q15 – Q0 bits of the register. When CE is High and L and CLR are Low, data on the SLI is loaded into the first bit of the shift register during the Low-to-High clock (C) transition and appears on the Q0 output. During subsequent clock transitions, when CE is High and L and CLR are Low, the data is shifted to the next highest bit position as new data is loaded into Q0 (SLI→Q0, Q0→Q1, Q1→Q2, and so forth). Registers can be cascaded by connecting the Q15 output of one stage to the Shift Left Input (SLI) of the next stage and connecting clock, CE, L, and CLR inputs in parallel.

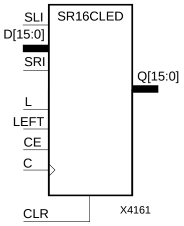
The default initial state of all flip-flops is zero.

Inputs						Outputs	
CLR	**L**	**CE**	**SLI**	**D15 – D0**	**C**	**Q0**	**Q15 – Q1**
1	X	X	X	X	X	0	0
0	1	X	X	D15 – D0	↑	d0	dn
0	0	1	SLI	X	↑	SLI	qn-1
0	0	0	X	X	X	--No Change--	

dn or qn-1 = state of referenced input one set-up time prior to active clock transition

SR16CLED

16-Bit Shift Register with Clock Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

SR16CLED is a 16-bit shift register with shift-left (SLI) and shift-right (SRI) serial inputs, parallel inputs (D15 – D0), and four control inputs – clock enable (CE), load enable (L), shift left / right (LEFT), and asynchronous clear (CLR). The register ignores clock transitions when CE and L are Low. The asynchronous CLR, when High, overrides all other inputs and resets the data outputs (Q15 – Q0) Low. When L is High and CLR is Low the data on the D15 – D0 inputs is loaded into the corresponding Q15 – Q0 bits of the register. When CE is High and L and CLR are Low, data is shifted right or left depending on the state of the LEFT input. If LEFT is High, data on SLI is loaded into Q0 during the Low-to-High clock transition and shifted left (to Q1, Q2, and so forth) during subsequent clock transitions. If LEFT is Low, data on SRI is loaded into Q15 during the Low-to-High clock transition and shifted right (to Q14, Q13, and so forth) during subsequent clock transitions. The truth table indicates the state of the Q15 – Q0 outputs under all input conditions.

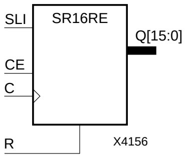
The default initial state of all flip-flops is zero.

Inputs								Outputs		
CLR	**L**	**CE**	**LEFT**	**SLI**	**SRI**	**D15 – D0**	**C**	**Q0**	**Q15**	**Q14 – Q1**
1	X	X	X	X	X	X	X	0	0	0
0	1	X	X	X	X	D15 – D0	↑	d0	d15	dn
0	0	0	X	X	X	X	X	----No Change-----		
0	0	1	1	SLI	X	X	↑	SLI	q14	qn-1
0	0	1	0	X	SRI	X	↑	q1	SRI	qn+1

dn, qn-1 or qn+1 = state of referenced input one set-up time prior to active clock transition

SR16RE

16-Bit Serial-In Parallel-Out Shift Register with Clock Enable and Synchronous Reset



XC7000	XC9000
Macro	Macro

SR16RE is a 16-bit shift-left serial input (SLI), parallel output (Q15 – Q0) shift register with clock enable (CE) and synchronous reset (R) inputs. The R input, when High, overrides all other inputs and resets the data outputs (Q15 – Q0) Low. When CE is High and R is Low, the data on the SLI is loaded into the first bit of the shift register during the Low-to-High clock (C) transition and appears on the Q0 output. During subsequent Low-to-High clock transitions, when CE is High and R is Low, data is shifted to the next highest bit position as new data is loaded into Q0 (SLI→Q0, Q0→Q1, Q1→Q2, and so forth). The register ignores clock transitions when CE is Low. Registers can be cascaded by connecting the Q15 output of one stage to the SLI input of the next stage and connecting clock, C, and R in parallel.

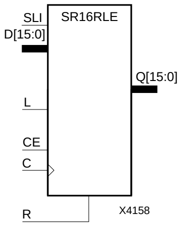
The default initial state of all flip-flops is zero.

Inputs				Outputs	
R	CE	SLI	C	Q0	Q15 – Q1
1	X	X	↑	0	0
0	0	X	X	---No Change---	
0	1	1	↑	1	qn-1
0	1	0	↑	0	qn-1

qn-1 = state of referenced output one set-up time prior to active clock transition

SR16RLE

16-Bit Loadable Serial/Parallel-In Parallel-Out Shift Register with Clock Enable and Synchronous Reset



XC7000	XC9000
Macro	Macro

SR16RLE is a 16-bit shift register with shift-left serial input (SLI), parallel inputs (D15 – D0), parallel outputs (Q15 – Q0), and control inputs – clock enable (CE), load enable (L), and synchronous reset (R). The register ignores clock transitions when L and CE are Low. The synchronous R, when High, overrides all other inputs and resets the data outputs (Q15 – Q0) Low. When L is High and R is Low, data on the data D15 – D0 inputs is loaded into the corresponding Q15 – Q0 bits of the register. When CE is High and L and R are Low, data on the SLI is loaded into the first bit of the shift register during the Low-to-High clock (C) transition and appears on the Q0 output. During subsequent clock transitions, when CE is High and L and R are Low, the data is shifted to the next highest bit position as new data is loaded into Q0 (SLI→Q0, Q0→Q1, Q1→Q2, and so forth). Registers can be cascaded by connecting the Q15 output of one stage to the Shift Left Input (SLI) of the next stage and connecting clock, CE, L, and R inputs in parallel.

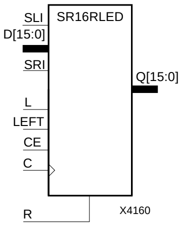
The default initial state of all flip-flops is zero.

Inputs						Outputs	
R	L	CE	SLI	D15 – D0	C	Q0	Q15 – Q1
1	X	X	X	X	↑	0	0
0	1	X	X	D15 – D0	↑	d0	dn
0	0	1	SLI	X	↑	SLI	qn-1
0	0	0	X	X	X	--No Change--	

dn or qn-1 = state of referenced input one set-up time prior to active clock transition

SR16RLED

16-Bit Shift Register with Clock Enable and Synchronous Reset



XC7000	XC9000
Macro	Macro

SR16RLED is a 16-bit shift register with shift-left (SLI) and shift-right (SRI) serial inputs, parallel inputs (D15 – D0), and four control inputs – clock enable (CE), load enable (L), shift left / right (LEFT), and synchronous reset (R). The register ignores clock transitions when CE and L are Low. The synchronous R, when High, overrides all other inputs and resets the data Q15 – Q0 outputs Low. When L is High and R is Low, the data on the D15 – D0 inputs is loaded into the corresponding Q15 – Q0 bits of the register. When CE is High and L and R are Low, data is shifted right or left depending on the state of the LEFT input. If LEFT is High, data on SLI is loaded into Q0 during the Low-to-High clock transition and shifted left (to Q1, Q2, and so forth) during subsequent clock transitions. If LEFT is Low, data on SRI is loaded into Q15 during the Low-to-High clock transition and shifted right (to Q14, Q13, and so forth) during subsequent clock transitions. The truth table indicates the state of the Q15 – Q0 outputs under all input conditions.

The default initial state of all flip-flops is zero.

Inputs								Outputs		
R	L	CE	LEFT	SLI	SRI	D15 – D0	C	Q0	Q15	Q14 – Q1
1	X	X	X	X	X	X	↑	0	0	0
0	1	X	X	X	X	D15 – D0	↑	d0	d15	dn
0	0	0	X	X	X	X	X	----No Change-----		
0	0	1	1	SLI	X	X	↑	SLI	q14	qn-1
0	0	1	0	X	SRI	X	↑	q1	SRI	qn+1

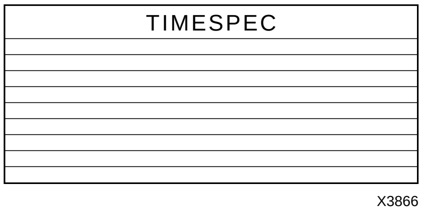
dn, qn-1 or qn+1 = state of referenced input one set-up time prior to active clock transition

TIMESPEC

Schematic-Level Timing Requirement Table

XC7000	XC9000
Primitive	Primitive

The TIMESPEC primitive is a table that specifies up to eight timing attributes (TS). TS attributes can be any length, but only 30 characters are displayed in the TIMESPEC window. The TIMESPEC table is displayed in the follow figure.

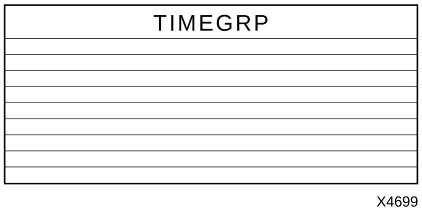


TIMEGRP

Schematic-Level Table of Basic Timing Specification Groups

XC7000	XC9000
Primitive	Primitive

The TIMEGRP primitive table defines timing groups used in “from-to” TIMESPEC statements in terms of other groups. The TIMEGRP table is shown in the following figure.



These groups can include predefined groups, such as “ffs,” groups created by using TNM attributes, such as TNM-reg on schematics, and other groups defined by a statement in the TIMEGRP symbol.

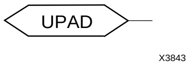
The following example statement defines groups in a TIMEGRP symbol.

```
=all_but_regs=ffs:except:regs
```

The table can contain up to 8 statements of any character length, but only 30 characters are displayed in the symbol.

UPAD

Unbonded I/O Pad

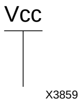


XC7000	XC9000
Primitive	Primitive

A UPAD allows the use of any unbonded IOBs in a device. It is used the same way as a IOPAD, except that the signal output is not visible on any external device pins.

VCC

VCC-Connection Signal Tag



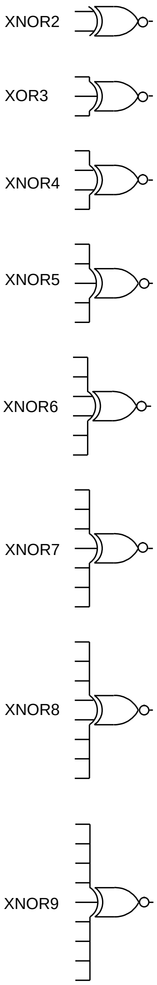
XC7000	XC9000
Primitive	Primitive

The V_{CC} signal tag or parameter forces a net or input function to a logic High level. A net tied to V_{CC} cannot have any other source.

When the encounters a net or input function tied to V_{CC}, it removes any logic that is disabled by the V_{CC} signal. The V_{CC} signal is only implemented when the disabled logic cannot be removed.

XNOR

2- to 9-Input XNOR Gates with Non-Inverted Inputs



Name	XC7000	XC9000
XNOR2 – XNOR4	Primitive	Primitive
XNOR5 – XNOR9	Macro	Macro

XNOR functions of up to nine inputs are available. All inputs are non-inverting.

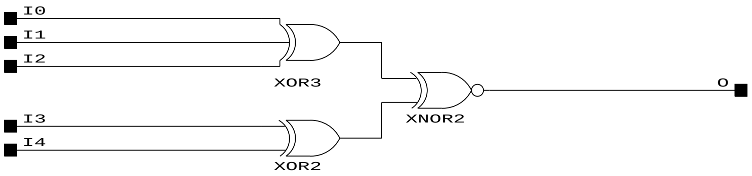


Figure 3-141 XNOR5 Implementation (XC7000 and XC9000)

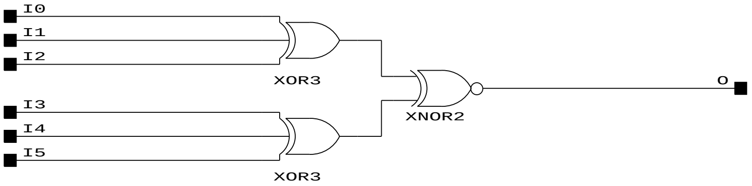


Figure 3-142 XNOR6 Implementation (XC7000 and XC9000)

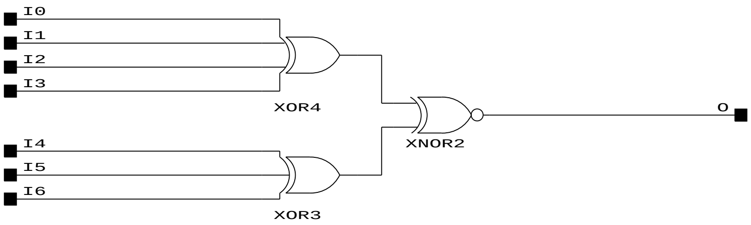


Figure 3-143 XNOR7 Implementation (XC7000 and XC9000)

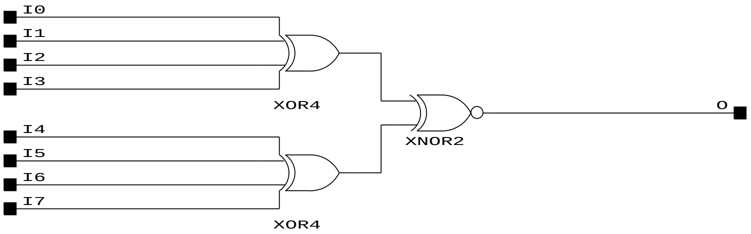


Figure 3-144 XNOR8 Implementation (XC7000 and XC9000)

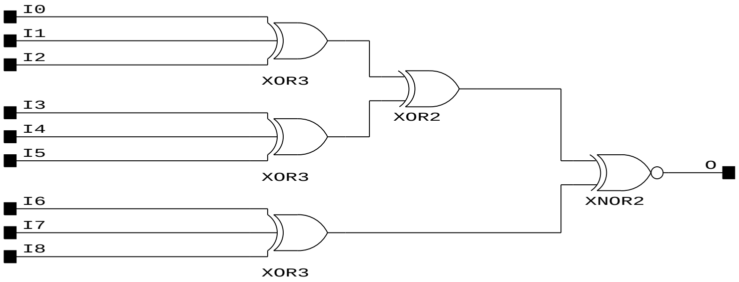
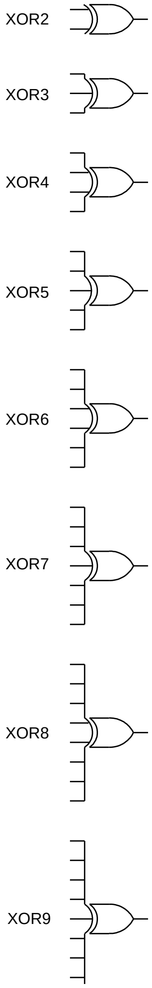


Figure 3-145 XNOR9 Implementation (XC7000 and XC9000)

XOR

2- to 9-Input XOR Gates with Non-Inverted Inputs



Name	XC7000	XC9000
XOR2 – XOR4	Primitive	Primitive
XOR5 – XOR9	Macro	Macro

XOR functions of up to nine inputs are available. All inputs are non-inverting.

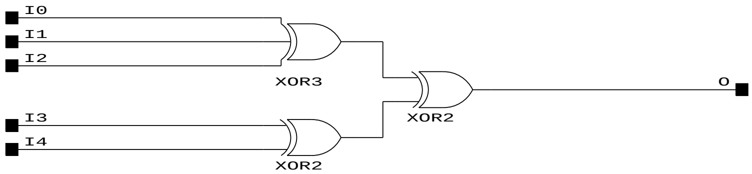


Figure 3-146   XOR5 Implementation (XC7000 and XC9000)

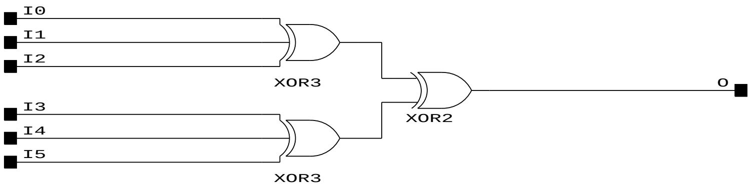


Figure 3-147   XOR6 Implementation (XC7000 and XC9000)

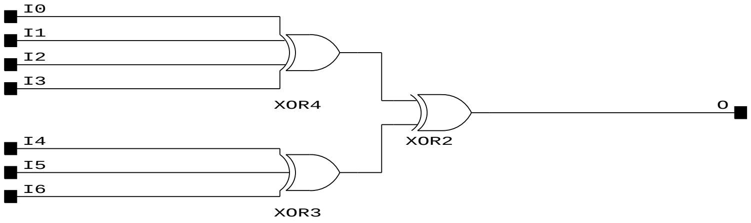


Figure 3-148 XOR7 Implementation (XC7000 and XC9000)

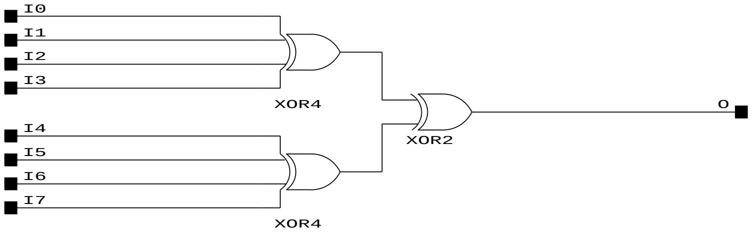


Figure 3-149 XOR8 Implementation (XC7000 and XC9000)

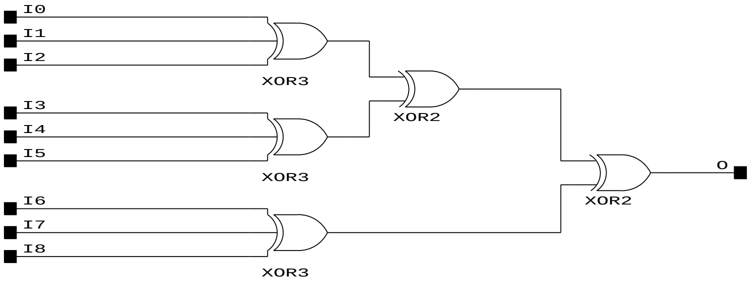
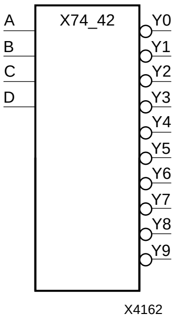


Figure 3-150 XOR9 Implementation (XC7000 and XC9000)

X74_42

4- to 10-Line BCD-to-Decimal Decoder with Active-Low Outputs



XC7000	XC9000
Macro	Macro

X74_42 decodes the 4-bit BCD number on the data inputs (A – D). Only one of the ten outputs (Y9 – Y0) is active (Low) at a time, which reflects the decimal equivalent of the BCD number on inputs A – D. All outputs are inactive (High) during any one of six illegal states, as shown in the truth table.

Inputs				Outputs
D	C	B	A	Selected (Low) Output*
0	0	0	0	Y0
0	0	0	1	Y1
0	0	1	0	Y2
0	0	1	1	Y3
0	1	0	0	Y4
0	1	0	1	Y5
0	1	1	0	Y6
0	1	1	1	Y7
1	0	0	0	Y8
1	0	0	1	Y9
1	0	1	0	All Outputs High
1	0	1	1	All Outputs High
1	1	0	0	All Outputs High
1	1	0	1	All Outputs High
1	1	1	0	All Outputs High
1	1	1	1	All Outputs High

* Selected output is Low (0) and all others are High.

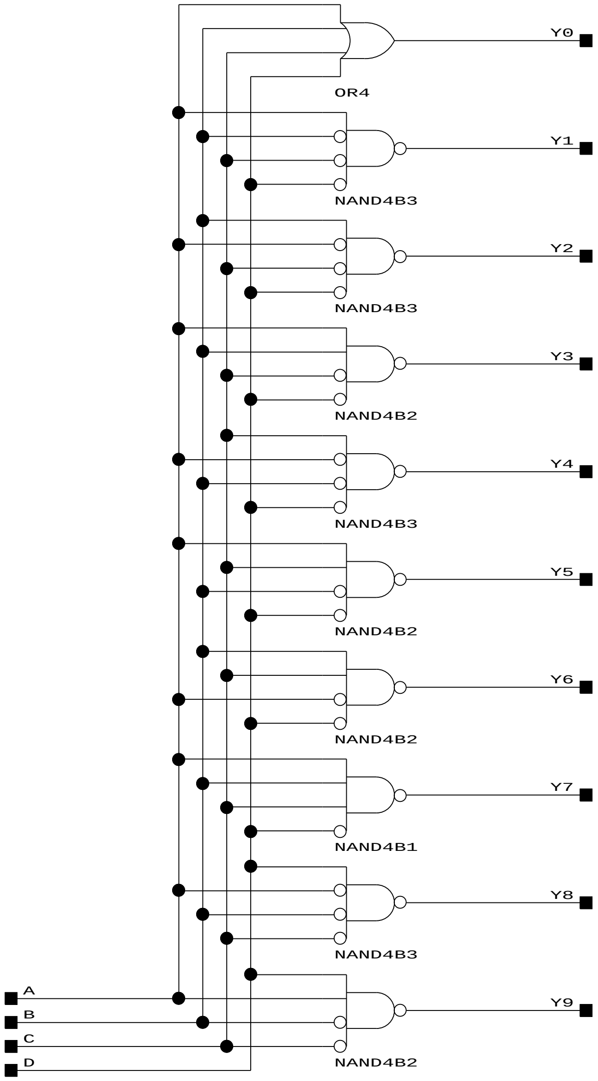
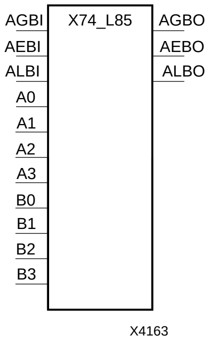


Figure 3-151 X74_42 Implementation (XC7000 and XC9000)

X74_L85

4-Bit Expandable Magnitude Comparator



XC7000	XC9000
Macro	Macro

X74_L85 is a 4-bit magnitude comparator that compares two 4-bit binary-weighted words A3 – A0 and B3 – B0, where A3 and B3 are the most significant bits. The greater-than output, AGBO, is High when A>B. The less-than output, ALBO, is High when A<B, and the equal output, AEBO, is High when A=B. The expansion inputs, AGBI, ALBI, and AEBI, are the least significant bits. Words of greater length can be compared by cascading the comparators. The AGBO, ALBO, and AEBO outputs of the stage handling less-significant bits are connected to the corresponding AGBI, ALBI, and AEBI inputs of the next stage handling more- significant bits. For proper operation, the stage handling the least significant bits must have AGBI and ALBI tied Low and AEBI tied High.

Inputs							Outputs		
A3, B3	**A2, B2**	**A1, B1**	**A0, B0**	**AGBI**	**ALBI**	**AEBI**	**AGBO**	**ALBO**	**AEBO**
A3>B3	X	X	X	X	X	X	1	0	0
A3<B3	X	X	X	X	X	X	0	1	0
A3=B3	A2>B2	X	X	X	X	X	1	0	0
A3=B3	A2<B2	X	X	X	X	X	0	1	0
A3=B3	A2=B2	A1>B1	X	X	X	X	1	0	0
A3=B3	A2=B2	A1<B1	X	X	X	X	0	1	0
A3=B3	A2=B2	A1=B1	A0>B0	X	X	X	1	0	0
A3=B3	A2=B2	A1=B1	A0<B0	X	X	X	0	1	0
A3=B3	A2=B2	A1=B1	A0=B0	1	0	0	1	0	0
A3=B3	A2=B2	A1=B1	A0=B0	0	1	0	0	1	0
A3=B3	A2=B2	A1=B1	A0=B0	0	0	1	0	0	1
A3=B3	A2=B2	A1=B1	A0=B0	0	1	1	0	1	1
A3=B3	A2=B2	A1=B1	A0=B0	1	0	1	1	0	1
A3=B3	A2=B2	A1=B1	A0=B0	1	1	1	1	1	1
A3=B3	A2=B2	A1=B1	A0=B0	1	1	0	1	1	0
A3=B3	A2=B2	A1=B1	A0=B0	0	0	0	0	0	0

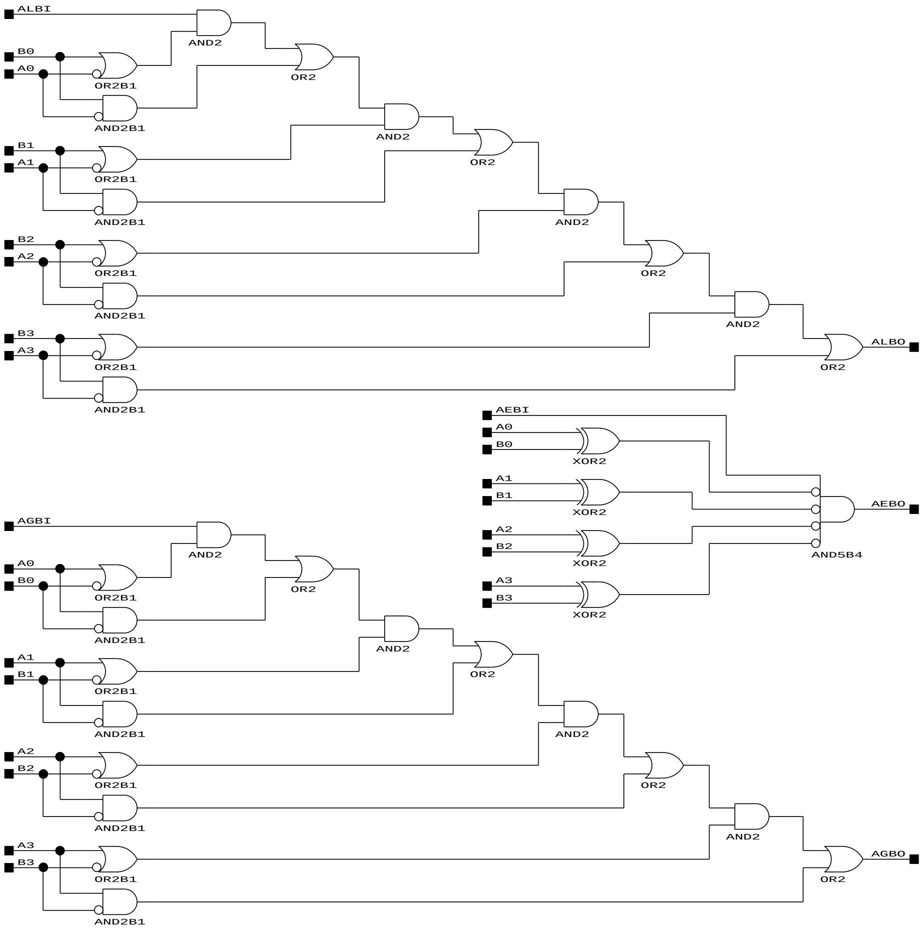
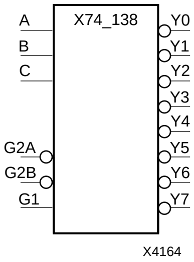


Figure 3-152 X74_L85 Implementation (XC7000 and XC9000)

X74_138

3- to 8-Line Decoder/Demultiplexer with Active-Low Outputs and Three Enables



XC7000	XC9000
Macro	Macro

X74_138 is an expandable decoder / demultiplexer with one active-High enable input (G1), two active-Low enable inputs (G2A and G2B), and eight active-Low outputs (Y7 – Y0). When G1 is High and G2A and G2B are Low, one of the eight active-Low outputs is selected with a 3-bit binary address on address inputs A, B, and C. The non-selected outputs are High. When G1 is Low or when G2A or G2B is High, all outputs are High.

X74_138 can be used as an 8-output active-Low demultiplexer by tying the data input to one of the enable inputs.

Inputs						Outputs							
C	B	A	G1	G2A	G2B	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
0	0	0	1	0	0	1	1	1	1	1	1	1	0
0	0	1	1	0	0	1	1	1	1	1	1	0	1
0	1	0	1	0	0	1	1	1	1	1	0	1	1
0	1	1	1	0	0	1	1	1	1	0	1	1	1
1	0	0	1	0	0	1	1	1	0	1	1	1	1
1	0	1	1	0	0	1	1	0	1	1	1	1	1
1	1	0	1	0	0	1	0	1	1	1	1	1	1
1	1	1	1	0	0	0	1	1	1	1	1	1	1
X	X	X	0	X	X	1	1	1	1	1	1	1	1
X	X	X	X	1	X	1	1	1	1	1	1	1	1
X	X	X	X	X	1	1	1	1	1	1	1	1	1

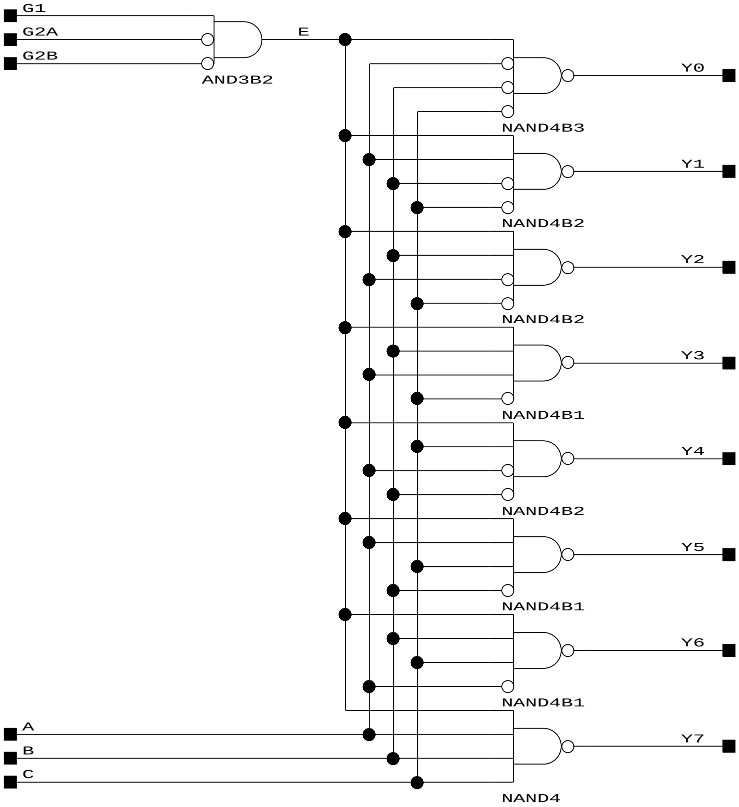
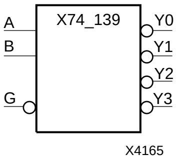


Figure 3-153 X74_138 Implementation (XC7000 and XC9000)

X74_139

2- to 4-Line Decoder/Demultiplexer with Active-Low Outputs and Active-Low Enable



XC7000	XC9000
Macro	Macro

X74_139 implements one half of a standard 74139 dual 2- to 4-line decoder / demultiplexer. When the active-Low enable input (G) is Low, one of the four active-Low outputs (Y3 – Y0) is selected with the 2-bit binary address on the A and B address input lines. B is the High-order address bit. The non-selected outputs are High. Also, when G is High all outputs are High.

X74_139 can be used as a 4-output active-Low demultiplexer by tying the data input to G.

Inputs			Outputs			
G	B	A	Y3	Y2	Y1	Y0
0	0	0	1	1	1	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	1
1	X	X	1	1	1	1

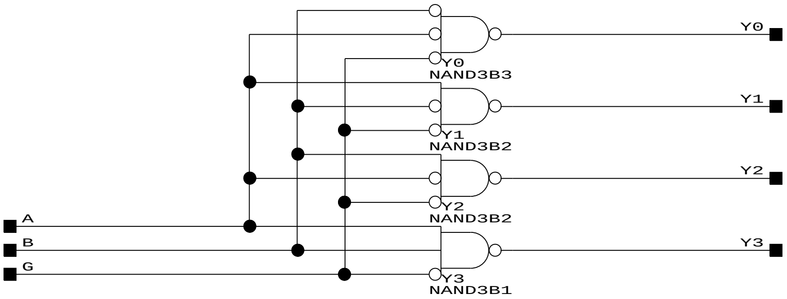
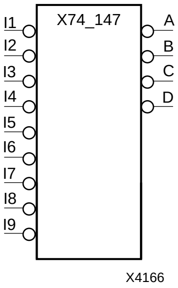


Figure 3-154 X74_139 Implementation (XC7000 and XC9000)

X74_147

10- to 4-Line Priority Encoder with Active-Low Inputs and Outputs



XC7000	XC9000
Macro	Macro

X74_147 is a 10-line-to-BCD-priority encoder that accepts data from nine active-Low inputs (I9 – I1) and produces a binary-coded decimal (BCD) representation on the four active-Low outputs A, B, C, and D. The data inputs are weighted, so when more than one input is active, only the one with the highest priority is encoded, with I9 having the highest priority. Only nine inputs are provided, because the implied “zero” condition requires no data input. “Zero” is encoded when all data inputs are High.

Inputs									Outputs			
I9	I8	I7	I6	I5	I4	I3	I2	I1	D	C	B	A
1	1	1	1	1	1	1	1	0	1	1	1	0
1	1	1	1	1	1	1	0	X	1	1	0	1
1	1	1	1	1	1	0	X	X	1	1	0	0
1	1	1	1	1	0	X	X	X	1	0	1	1
1	1	1	1	0	X	X	X	X	1	0	1	0
1	1	1	0	X	X	X	X	X	1	0	0	1
1	1	0	X	X	X	X	X	X	1	0	0	0
1	0	X	X	X	X	X	X	X	0	1	1	1
0	X	X	X	X	X	X	X	X	0	1	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1

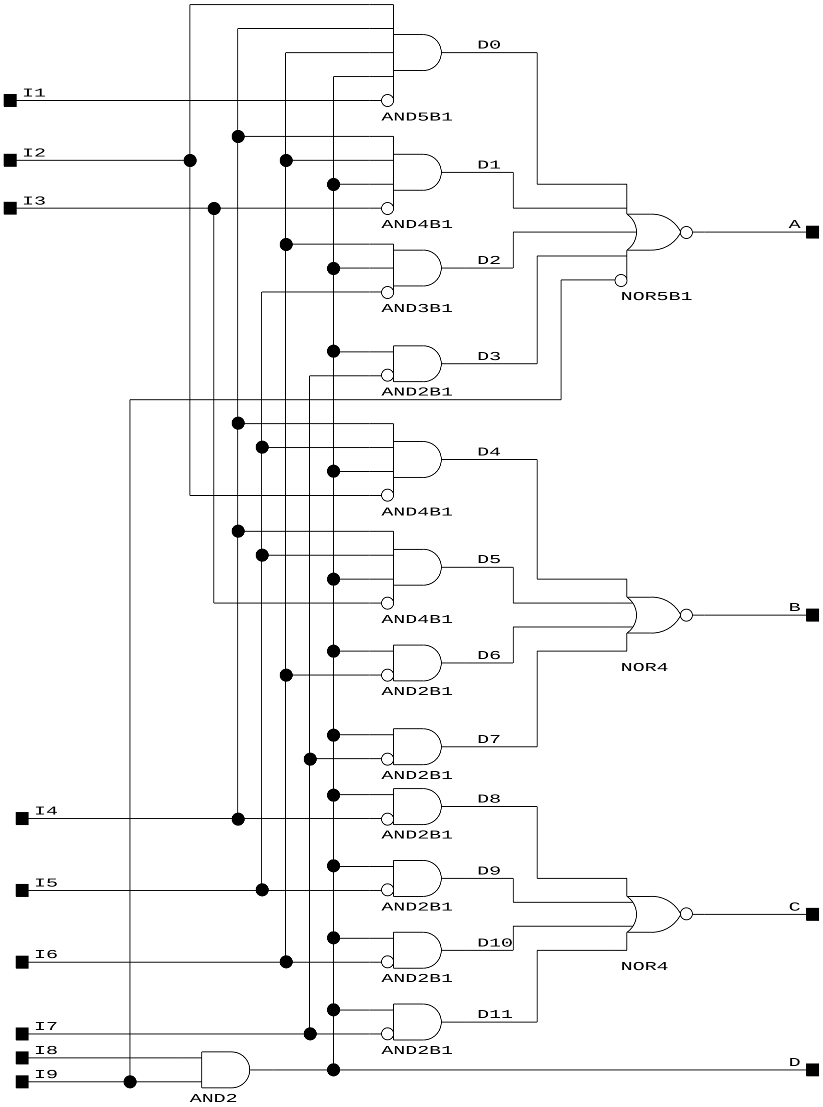
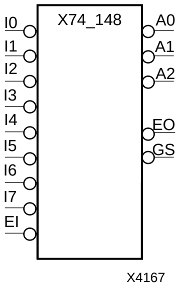


Figure 3-155 X74_147 Implementation (XC7000 and XC9000)

X74_148

8- to 3-Line Cascadable Priority Encoder with Active-Low Inputs and Outputs



XC7000	XC9000
Macro	Macro

X74_148 8-input priority encoder accepts data from eight active-Low inputs (I7 – I0) and produces a binary representation on the three active-Low outputs (A2 – A0). The data inputs are weighted, so when more than one of the inputs is active, only the input with the highest priority is encoded, I7 having the highest priority. The active-Low group signal (GS) is Low whenever one of the data inputs is Low and the active-Low enable input (EI) is Low.

The active-Low enable input (EI) and active-Low enable output (EO) are used to cascade devices and retain priority control. The EO of the highest priority stage is connected to the EI of the next-highest priority stage. When EI is High, the data outputs and EO are High. When EI is Low, the encoder output represents the highest-priority Low data input, and the EO is High. When EI is Low and all the data inputs are High, the EO output is Low to enable the next-lower priority stage.

Inputs									Outputs				
EI	I7	I6	I5	I4	I3	I2	I1	I0	A2	A1	A0	GS	EO
1	X	X	X	X	X	X	X	X	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0	1	1	1	0	1
0	1	1	1	1	1	1	0	X	1	1	0	0	1
0	1	1	1	1	1	0	X	X	1	0	1	0	1
0	1	1	1	1	0	X	X	X	1	0	0	0	1
0	1	1	1	0	X	X	X	X	0	1	1	0	1
0	1	1	0	X	X	X	X	X	0	1	0	0	1
0	1	0	X	X	X	X	X	X	0	0	1	0	1
0	0	X	X	X	X	X	X	X	0	0	0	0	1

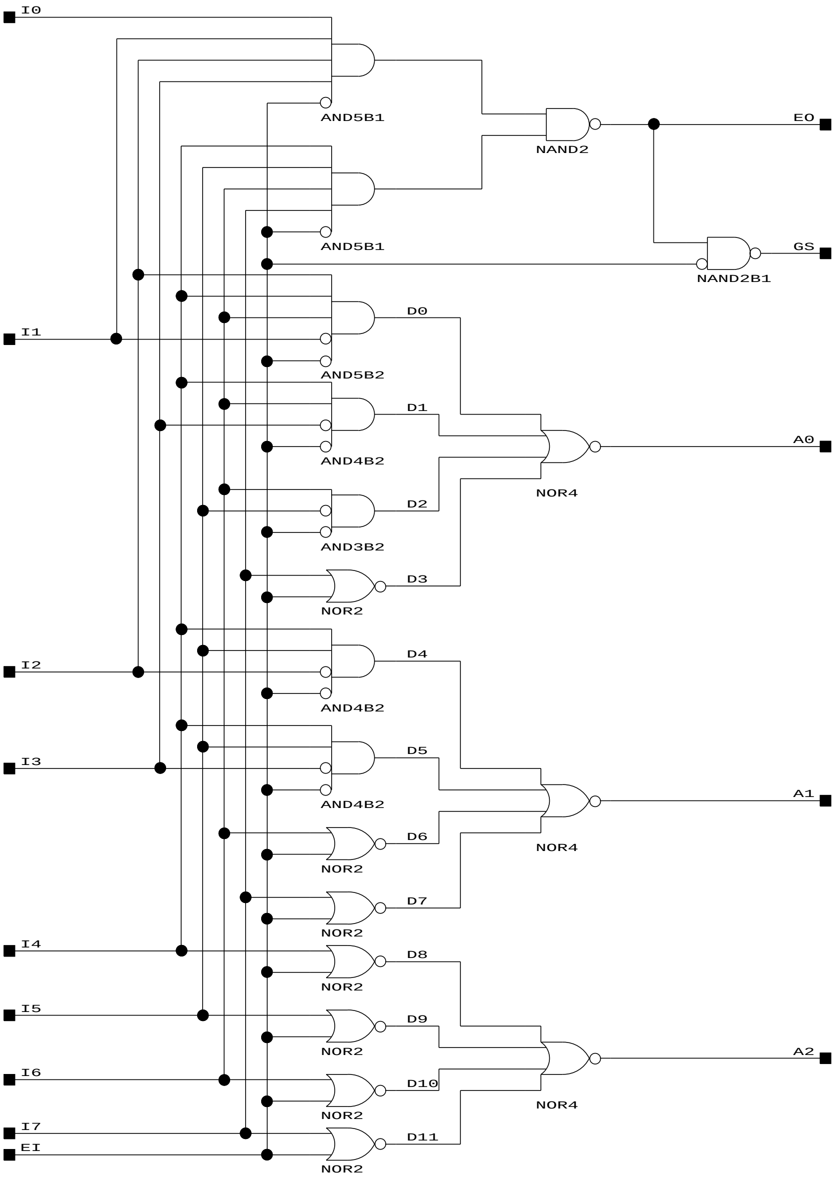
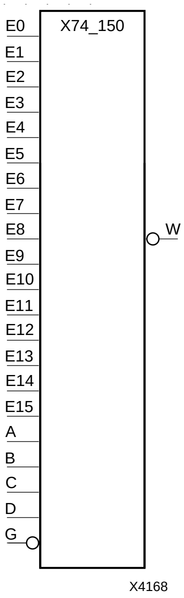


Figure 3-156   X74_148 Implementation (XC7000 and XC9000)

X74_150

16-to-1 Multiplexer with Active-Low Enable and Output



XC7000	XC9000
Macro	Macro

When the active-Low enable input (G) is Low, the X74_150 multiplexer chooses one data bit from 16 sources (E15 – E0) under the control of select inputs A, B, C, and D. The active-Low output (W) reflects the inverse of the selected input, as shown in the truth table. When the enable input (G) is High, the output (W) is High.

Inputs					Outputs
G	D	C	B	A	Selected Input Appears (Inverted) on W
1	X	X	X	X	1
0	0	0	0	0	$\overline{E0}$
0	0	0	0	1	$\overline{E1}$
0	0	0	1	0	$\overline{E2}$
0	0	0	1	1	$\overline{E3}$
0	0	1	0	0	$\overline{E4}$
0	0	1	0	1	$\overline{E5}$
0	0	1	1	0	$\overline{E6}$
0	0	1	1	1	$\overline{E7}$
0	1	0	0	0	$\overline{E8}$
0	1	0	0	1	$\overline{E9}$
0	1	0	1	0	$\overline{E10}$
0	1	0	1	1	$\overline{E11}$
0	1	1	0	0	$\overline{E12}$
0	1	1	0	1	$\overline{E13}$
0	1	1	1	0	$\overline{E14}$
0	1	1	1	1	$\overline{E15}$

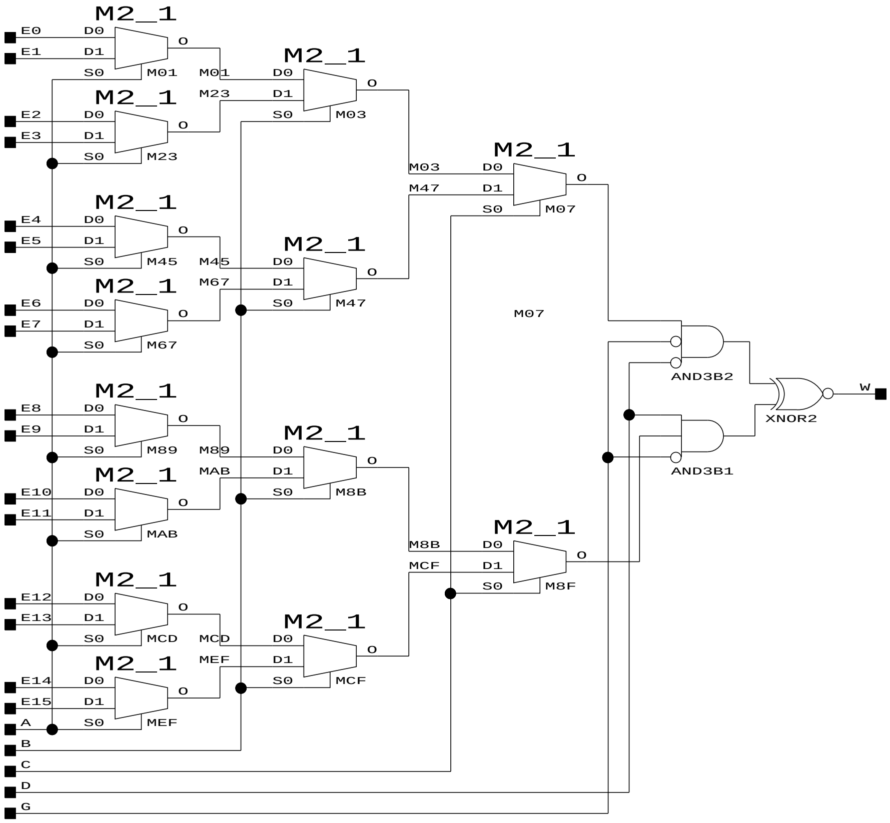
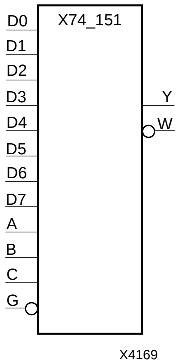


Figure 3-157 X74_150 Implementation (XC7000 and XC9000)

X74_151

8-to-1 Multiplexer with Active-Low Enable and Complementary Outputs



XC7000	XC9000
Macro	Macro

When the active-Low enable (G) is Low, the X74_151 multiplexer chooses one data bit from eight sources (D7 – D0) under control of the select inputs A, B, and C. The output (Y) reflects the state of the selected input, and the active-Low output (W) reflects the inverse of the selected input as shown in the truth table. When G is High, the Y output is Low, and the W output is High.

Inputs				Outputs	
G	C	B	A	Y	W
1	X	X	X	1	0
0	0	0	0	D0	$\overline{D0}$
0	0	0	1	D1	$\overline{D1}$
0	0	1	0	D2	$\overline{D2}$
0	0	1	1	D3	$\overline{D3}$
0	1	0	0	D4	$\overline{D4}$
0	1	0	1	D5	$\overline{D5}$
0	1	1	0	D6	$\overline{D6}$
0	1	1	1	D7	$\overline{D7}$

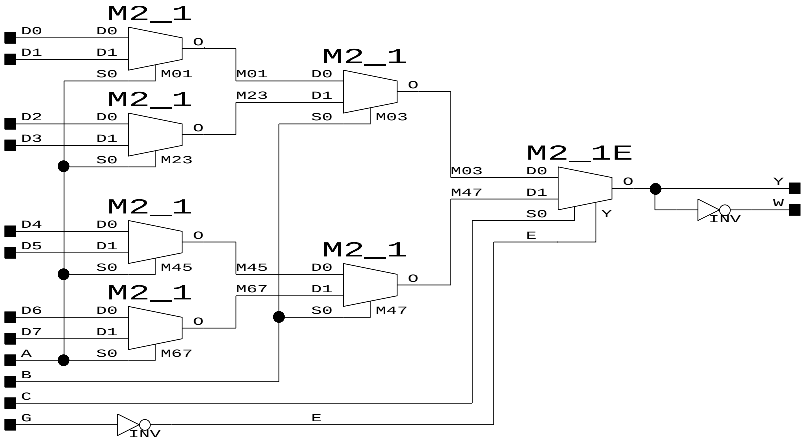
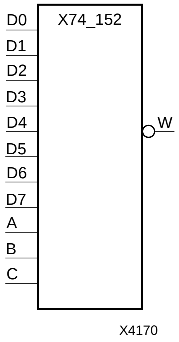


Figure 3-158 X74_151 Implementation (XC7000 and XC9000)

X74_152

8-to-1 Multiplexer with Active-Low Output



XC7000	XC9000
Macro	Macro

X74_152 multiplexer chooses one data bit from eight sources (D7 – D0) under control of the select inputs A, B, and C. The active-Low output (W) reflects the inverse of the selected data input, as shown in the truth table.

Inputs			Outputs
C	B	A	W
0	0	0	$\overline{D0}$
0	0	1	$\overline{D1}$
0	1	0	$\overline{D2}$
0	1	1	$\overline{D3}$
1	0	0	$\overline{D4}$
1	0	1	$\overline{D5}$
1	1	0	$\overline{D6}$
1	1	1	$\overline{D7}$

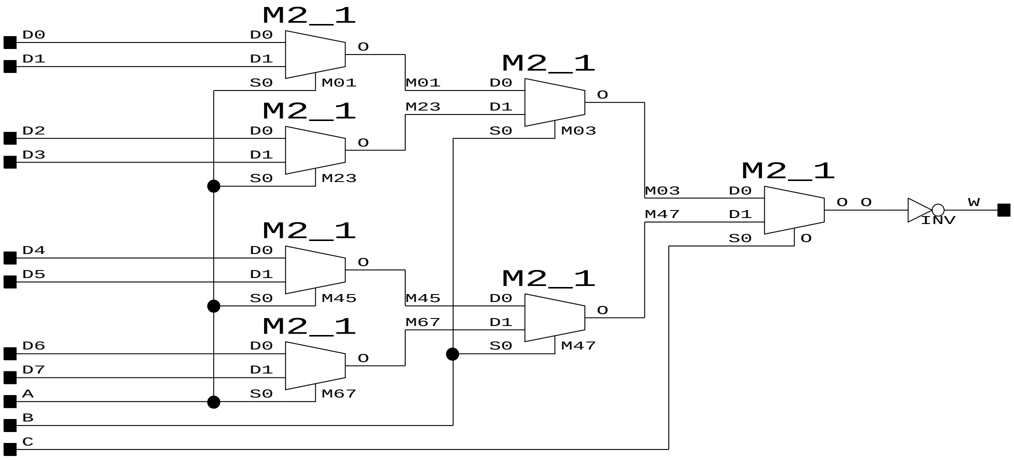
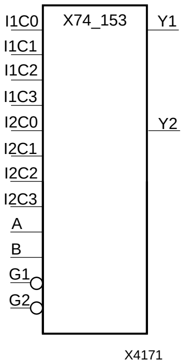


Figure 3-159 X74_152 Implementation (XC7000 and XC9000)

X74_153

Dual 4-to-1 Multiplexer with Active-Low Enables and Common Select Input



XC7000	XC9000
Macro	Macro

When the active-Low enable inputs G1 and G2 are Low, the data output Y1, reflects the data input chosen by select inputs A and B from data inputs I1C3 – I1C0. The data output Y2 reflects the data input chosen by select inputs A and B from data inputs I2C3 – I2C0. When G1 or G2 is High, the corresponding output, Y1 or Y2 respectively, is Low.

Inputs			Outputs
G	B	A	Y
1	X	X	0
0	0	0	IC0
0	0	1	IC1
0	1	0	IC2
0	1	1	IC3

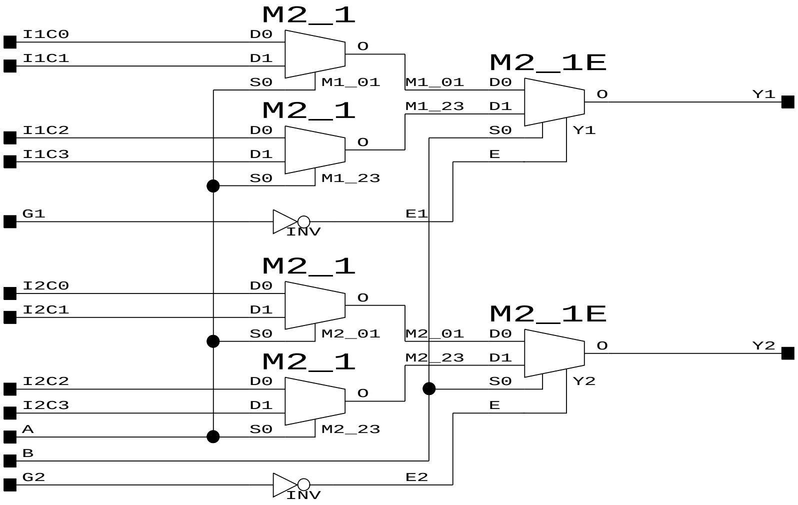
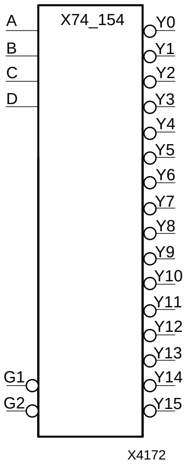


Figure 3-160 X74_153 Implementation (XC7000 and XC9000)

X74_154

4- to 16-Line Decoder/Demultiplexer with Two Enables and Active-Low Outputs



XC7000	XC9000
Macro	Macro

When the active-Low enable inputs G1 and G2 of the X74_154 decoder/demultiplexer are Low, one of 16 active-Low outputs, Y15 – Y0, is selected under the control of four binary address inputs A, B, C, and D. The non-selected inputs are High. Also, when either input G1 or G2 is High, all outputs are High.

The X74_154 can be used as a 16-to-1 demultiplexer by tying the data input to one of the G inputs and tying the other G input Low.

Inputs						Outputs							
G1	**G2**	**D**	**C**	**B**	**A**	**Y15**	**Y14**	**Y13**	**Y12**	**Y11**	**Y10**	**Y9**	**... Y0**
1	X	X	X	X	X	1	1	1	1	1	1	1	... 1
X	1	X	X	X	X	1	1	1	1	1	1	1	... 1
0	0	1	1	1	1	0	1	1	1	1	1	1	... 1
0	0	1	1	1	0	1	0	1	1	1	1	1	... 1
0	0	1	1	0	1	1	1	0	1	1	1	1	... 1
-	-	-	-	-	-	-	-	-	-	-	-	-	... -
-	-	-	-	-	-	-	-	-	-	-	-	-	... -
-	-	-	-	-	-	-	-	-	-	-	-	-	... -
0	0	0	0	0	0	1	1	1	1	1	1	1	... 0

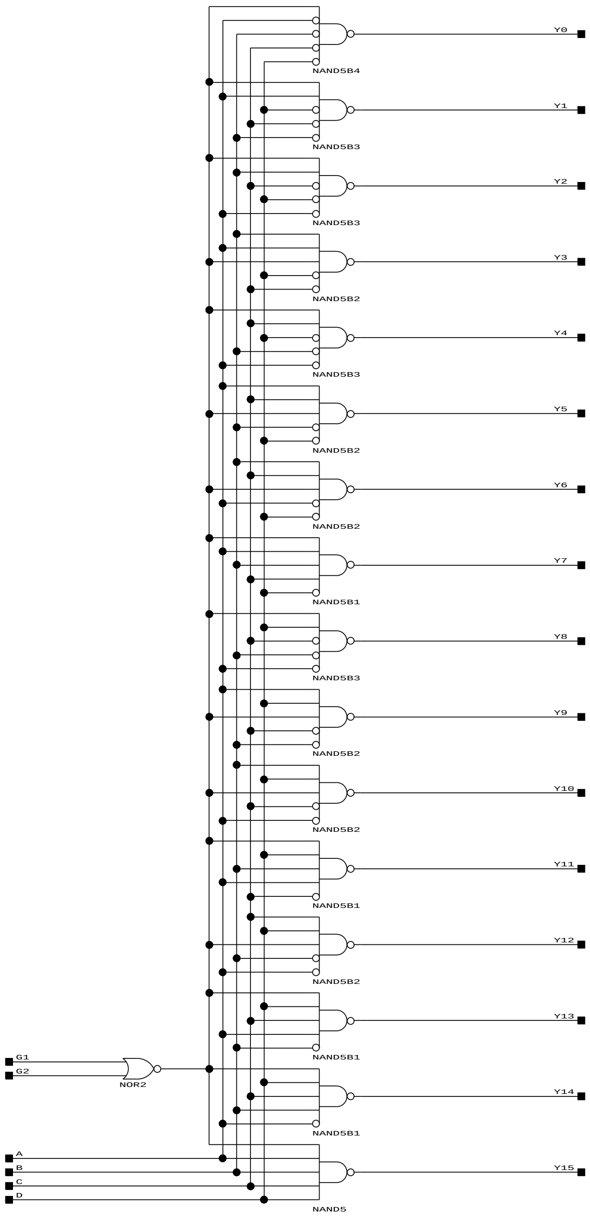
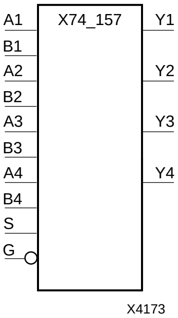


Figure 3-161 X74_154 Implementation (XC7000 and XC9000)

X74_157

Quadruple 2-to-1 Multiplexer with Common Select and Active-Low Enable



XC7000	XC9000
Macro	Macro

When the active-Low enable input (G) is Low, a 4-bit word is selected from one of two sources (A3 – A0 or B3 – B0) under the control of the select input (S) and is reflected on the four outputs (Y4 – Y1). When S is Low, the outputs reflect A3 – A0; when S is High, the outputs reflect B3 – B0. When G is High, the outputs are Low.

Inputs				Outputs
G	S	B	A	Y
1	X	X	X	0
0	1	1	X	1
0	1	0	X	0
0	0	X	1	1
0	0	X	0	0

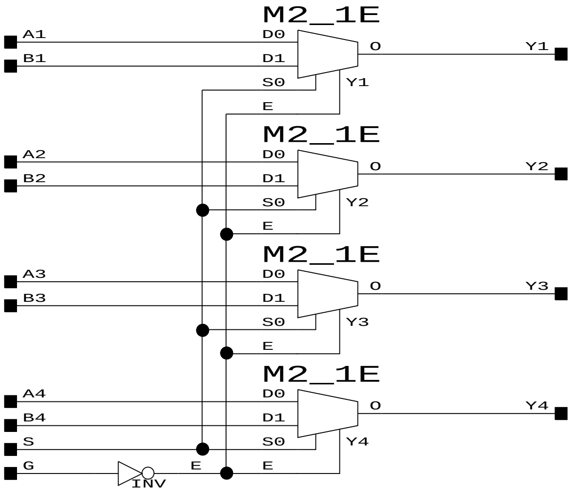
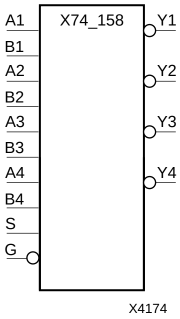


Figure 3-162 X74_157 Implementation (XC7000 and XC9000)

X74_158

Quadruple 2-to-1 Multiplexer with Common Select, Active-Low Enable, and Active-Low Outputs



XC7000	XC9000
Macro	Macro

When the active-Low enable (G) is Low, a 4-bit word is selected from one of two sources (A3 – A0 or B3 – B0) under the control of the common select input (S). The inverse of the selected word is reflected on the active-Low outputs (Y4 – Y1). When S is Low, $\overline{A3}$ – $\overline{A0}$ appear on the outputs; when S is High, $\overline{B3}$ – $\overline{B0}$ appear on the outputs. When G is High, the outputs are High.

Inputs				Outputs
G	S	B	A	Y
1	X	X	X	1
0	1	1	X	0
0	1	0	X	1
0	0	X	1	0
0	0	X	0	1

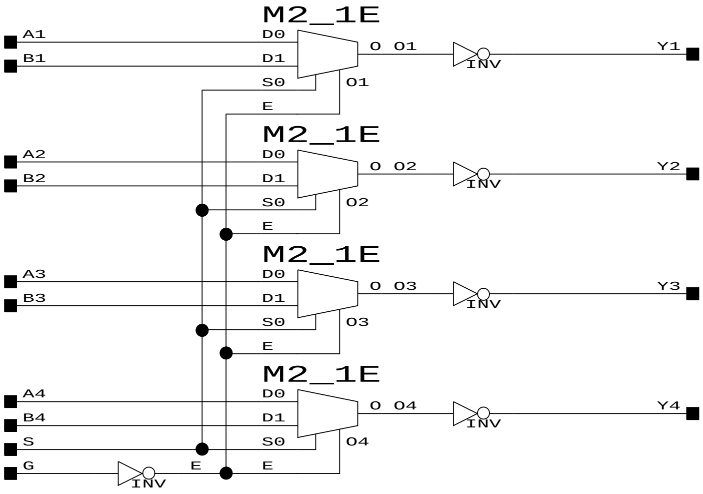
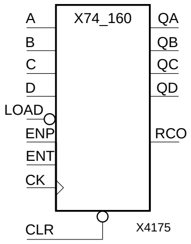


Figure 3-163 X74_158 Implementation (XC7000 and XC9000)

X74_160

4-Bit BCD Counter with Parallel and Trickle Enables, Active-Low Load Enable, and Asynchronous Clear



XC7000	XC9000
Macro	Macro

X74_160 is a 4-stage, 4-bit, synchronous, loadable, resettable, cascadable, binary-coded decimal (BCD) counter. The active-Low asynchronous clear (CLR), when Low, overrides all other inputs and resets the data (QD, QC, QB, QA) and ripple carry-out (RCO) outputs Low during the Low-to-High clock (C) transition. When the active-Low load enable input (LOAD) is Low, parallel clock enable (ENP), and trickle clock enable (ENT) are overridden and data on inputs A, B, C, and D are loaded into the counter during the Low-to-High clock transition. The data outputs (QD, QC, QB, QA) increment when ENP, ENT LOAD, and CLR are High during the Low-to-High clock transition. The counter ignores clock transitions when ENP or ENT are Low and LOAD is High. RCO is High when QD, QA, and ENT are High and QC and QB are Low. The default initial state of all flip-flops is zero.

Inputs						Outputs	
CLR	LOAD	ENP	ENT	D – A	CK	QD – QA	RCO
0	X	X	X	X	X	0	0
1	0	X	X	D – A	↑	d – a	RCO
1	1	0	X	X	X	No Chg	RCO
1	1	X	0	X	X	No Chg	0
1	1	1	1	X	↑	Inc	RCO

$RCO = (QD \bullet \overline{QC} \bullet \overline{QB} \bullet QA \bullet ENT)$

d – a = state of referenced input one set-up time prior to active clock transition

The carry-lookahead design allows cascading of large counters without extra gating. Both ENT and ENP must be High to count. ENT is fed forward to enable RCO, which produces a High output pulse with the approximate duration of the QA output. The following figure illustrates a carry-lookahead design.

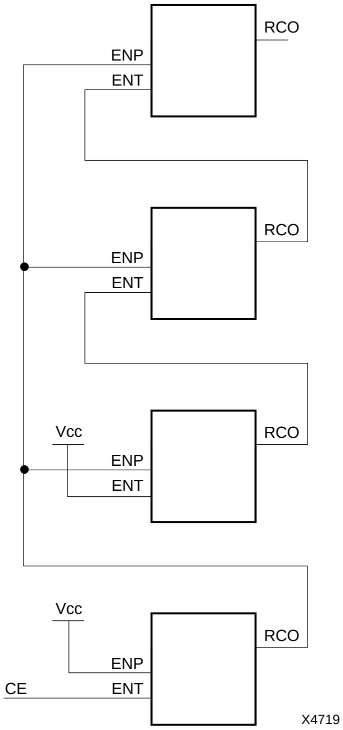


Figure 3-164 Carry-Lookahead Design

The RCO output of the first stage of the ripple carry is connected to the ENP input of the second stage and all subsequent stages. The RCO output of the second stage and all subsequent stages is connected to the ENT input of the next stage. The ENT of the second stage is always enabled / tied to VCC. CE is always connected to the ENT input of the first stage. This cascading method allows the first stage of the ripple carry to be built as a prescaler. In other words, the first stage is built to count very fast.

The counter recovers from any of six possible illegal states and returns to a normal count sequence within two clock cycles.

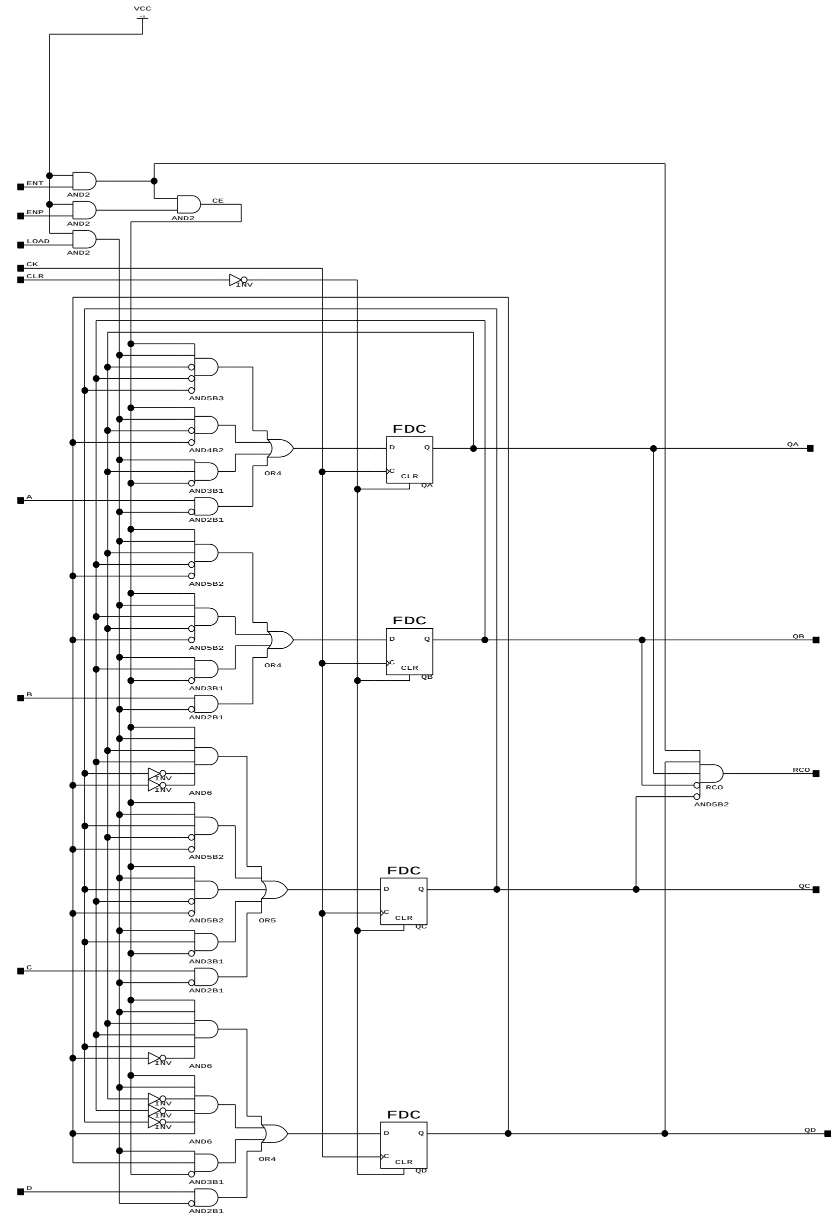
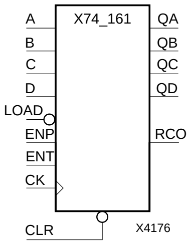


Figure 3-165 X74_160 Implementation (XC7000 and XC9000)

X74_161

4-Bit Counter with Parallel and Trickle Enables Active-Low Load Enable and Asynchronous Clear



XC7000	XC9000
Macro	Macro

X74_161 is a 4-stage, 4-bit, synchronous, loadable, resettable, cascadable binary counter. The active-Low asynchronous clear (CLR), when Low, overrides all other inputs and resets the data outputs (QD, QC, QB, QA) and the ripple carry-out output (RCO) Low. When the active-Low load enable (LOAD) is Low and CLR is High, parallel clock enable (ENP) and trickle clock enable (ENT) are overridden and the data on inputs A, B, C, and D is loaded into the counter during the Low-to-High clock (C) transition. The data outputs (QD, QC, QB, QA) increment when LOAD, ENP, ENT, and CLR are High during the Low-to-High clock transition. The counter ignores clock transitions when LOAD is High and ENP or ENT are Low. RCO is High when QD – QA and ENT are High. The default initial state of all flip-flops is zero.

Inputs						Outputs	
CLR	LOAD	ENP	ENT	D – A	CK	QD – QA	RCO
0	X	X	X	X	X	0	0
1	0	X	X	D – A	↑	d – a	RCO
1	1	0	X	X	X	No Chg	RCO
1	1	X	0	X	X	No Chg	0
1	1	1	1	X	↑	Inc	RCO

RCO = (QD•QC•QB•QA•ENT)

d – a = state of referenced input one set-up time prior to active clock transition

The carry-lookahead design accommodates large counters without extra gating. Both the ENT and ENP inputs must be High to count. ENT is fed forward to enable RCO, which produces a High output with the approximate duration of the QA output. The following figure illustrates a carry-lookahead design.

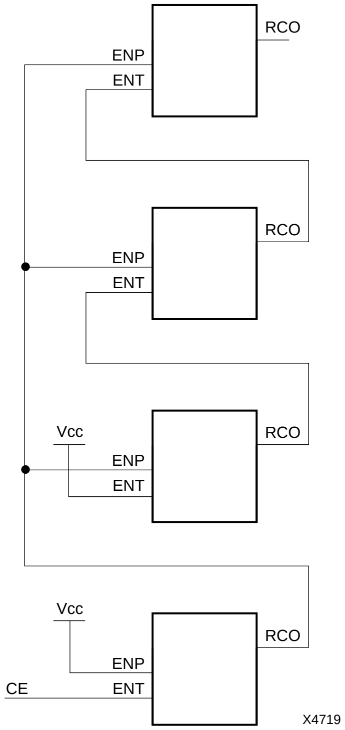


Figure 3-166 Carry-Lookahead Design

The RCO output of the first stage of the ripple carry is connected to the ENP input of the second stage and all subsequent stages. The RCO output of the second stage and all subsequent stages is connected to the ENT input of the next stage. The ENT of the second stage is always enabled / tied to VCC. CE is always connected to the ENT input of the first stage. This cascading method allows the first stage of the ripple carry to be built as a prescaler. In other words, the first stage is built to count very fast.

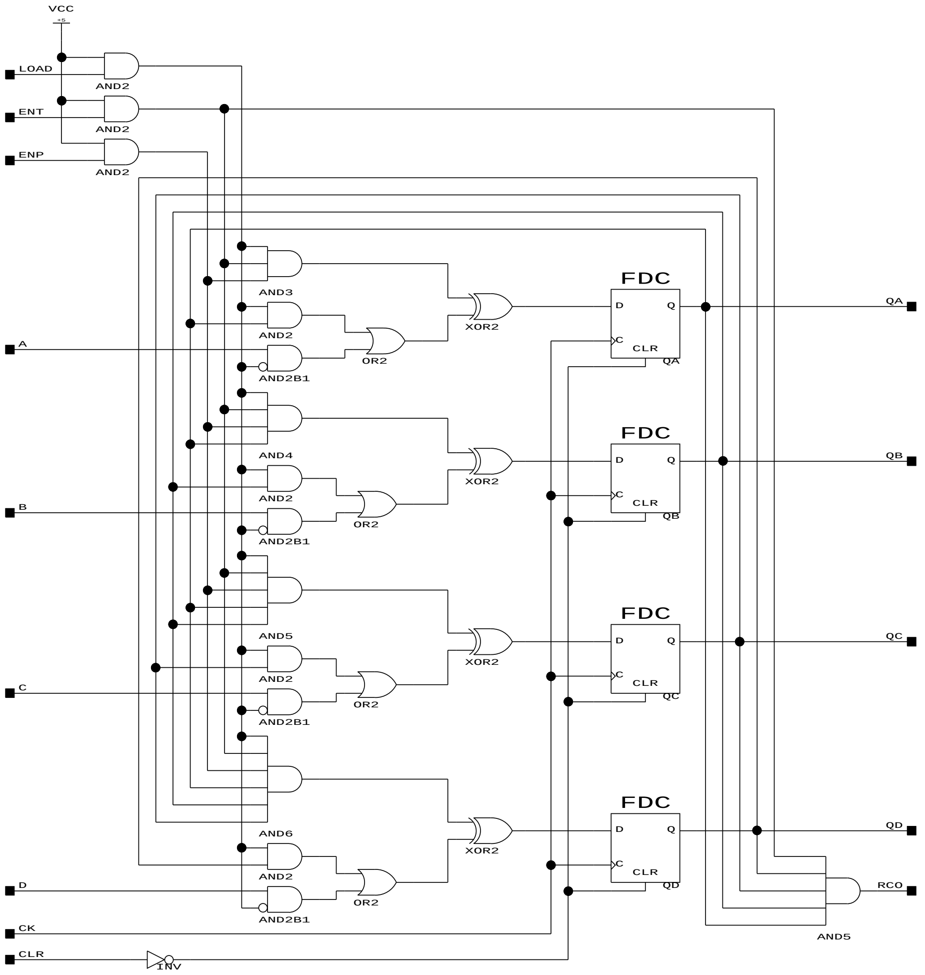
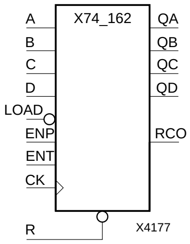


Figure 3-167 X74_161 Implementation (XC7000 and XC9000)

X74_162

4-Bit Counter with Parallel and Trickle Enables and Active-Low Load Enable and Synchronous Reset



XC7000	XC9000
Macro	Macro

X74_162 is a 4-stage, 4-bit, synchronous, loadable, resettable, cascadable binary-coded decimal (BCD) counter. The active-Low synchronous reset (R), when Low, overrides all other inputs and resets the data (QD, QC, QB, QA) and ripple carry-out (RCO) outputs Low during the Low-to-High clock (C) transition. When the active-Low load enable input (LOAD) is Low, parallel clock enable (ENP) and trickle clock enable (ENT) are overridden and data on inputs A, B, C, and D is loaded into the counter during the Low-to-High clock transition. The data outputs (QD, QC, QB, QA) increment when ENP, ENT, LOAD, and R are High during the Low-to-High clock transition. The counter ignores clock transitions when ENP or ENT are Low and LOAD is High. RCO is High when QD, QA, and ENT are High and QC and QB are Low. The default initial state of all flip-flops is zero.

Inputs						Outputs	
R	LOAD	ENP	ENT	D – A	CK	QD – QA	RCO
0	X	X	X	X	↑	0	0
1	0	X	X	D – A	↑	d – a	RCO
1	1	0	X	X	X	No Chg	RCO
1	1	X	0	X	X	No Chg	0
1	1	1	1	X	↑	Inc	RCO

$RCO = (QD \bullet \overline{QC} \bullet \overline{QB} \bullet QA \bullet ENT)$

d – a = state of referenced input one set-up time prior to active clock transition

The carry-lookahead design accommodates cascading large counters without extra gating. Both ENT and ENP must be High to count. The ENT is fed forward to enable RCO, which produces a High output pulse with the approximate duration of the QA output. The following figure illustrates a carry-lookahead design.

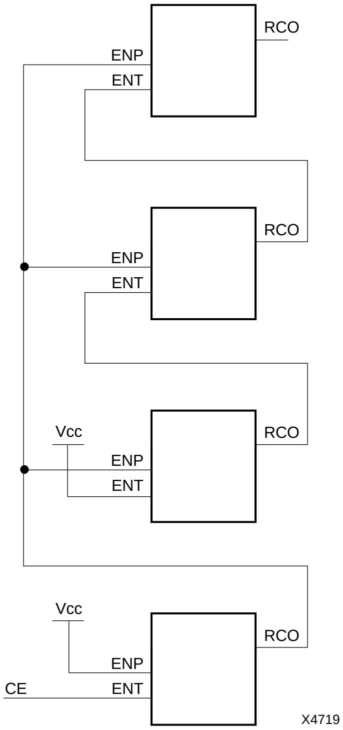


Figure 3-168 Carry-Lookahead Design

The RCO output of the first stage of the ripple carry is connected to the ENP input of the second stage and all subsequent stages. The RCO output of the second stage and all subsequent stages is connected to the ENT input of the next stage. The ENT of the second stage is always enabled / tied to VCC. CE is always connected to the ENT input of the first stage. This cascading method allows the first stage of the ripple carry to be built as a prescaler. In other words, the first stage is built to count very fast.

The counter recovers from any of six possible illegal states and returns to a normal count sequence within two clock cycles.

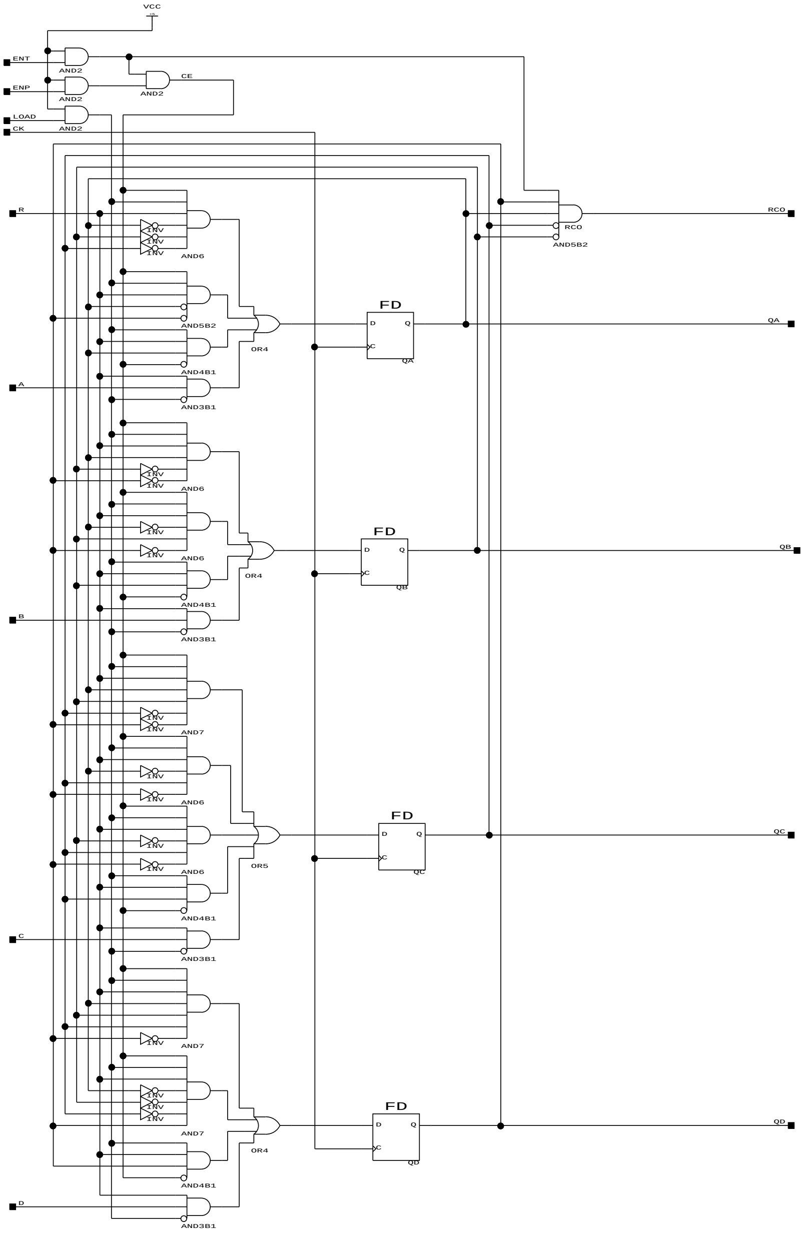
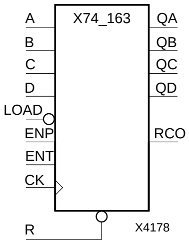


Figure 3-169 X74_162 Implementation (XC7000 and XC9000)

X74_163

4-Bit Counter with Parallel and Trickle Enables, Active-Low Load Enable, and Synchronous Reset



XC7000	XC9000
Macro	Macro

X74_163 is a 4-stage, 4-bit, synchronous, loadable, resettable, cascadable binary counter. The active-Low synchronous reset (R), when Low, overrides all other inputs and resets the data outputs (QD, QC, QB, QA) and the ripple carry-out output (RCO) Low. When the active-Low load enable (LOAD) is Low and R is High, parallel clock enable (ENP) and trickle clock enable (ENT) are overridden and the data on inputs (A, B, C, D) is loaded into the counter during the Low-to-High clock (C) transition. The outputs (QD, QC, QB, QA) increment when LOAD, ENP, ENT, and R are High during the Low-to-High clock transition. The counter ignores clock transitions when LOAD is High and ENP or ENT are Low; RCO is High when QD – QA and ENT are High. The default initial state of all flip-flops is zero.

Inputs						Outputs	
R	**LOAD**	**ENP**	**ENT**	**D – A**	**CK**	**QD – QA**	**RCO**
0	X	X	X	X	↑	0	0
1	0	X	X	D – A	↑	d – a	RCO
1	1	0	X	X	X	No Chg	RCO
1	1	X	0	X	X	No Chg	0
1	1	1	1	X	↑	Inc	RCO

RCO = (QD•QC•QB•QA•ENT)

d – a = state of referenced input one set-up time prior to active clock transition

The carry-lookahead design accommodates large counters without extra gating. Both the ENT and ENP inputs must be High to count. ENT is propagated forward to enable RCO, which produces a High output with the approximate duration of the QA output. The following figure illustrates a carry-lookahead design.

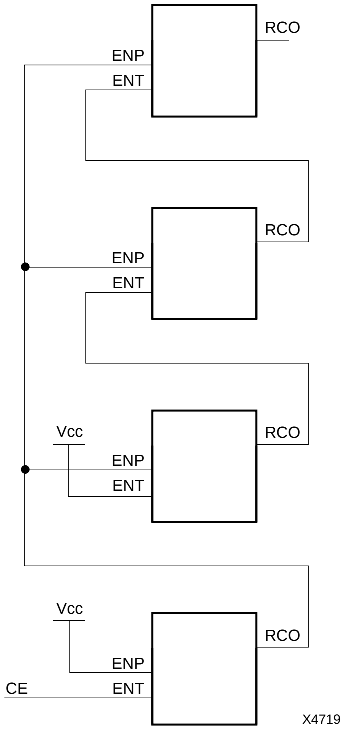


Figure 3-170 Carry-Lookahead Design

The RCO output of the first stage of the ripple carry is connected to the ENP input of the second stage and all subsequent stages. The RCO output of the second stage and all subsequent stages is connected to the ENT input of the next stage. The ENT of the second stage is always enabled / tied to VCC. CE is always connected to the ENT input of the first stage. This cascading method allows the first stage of the ripple carry to be built as a prescaler. In other words, the first stage is built to count very fast.

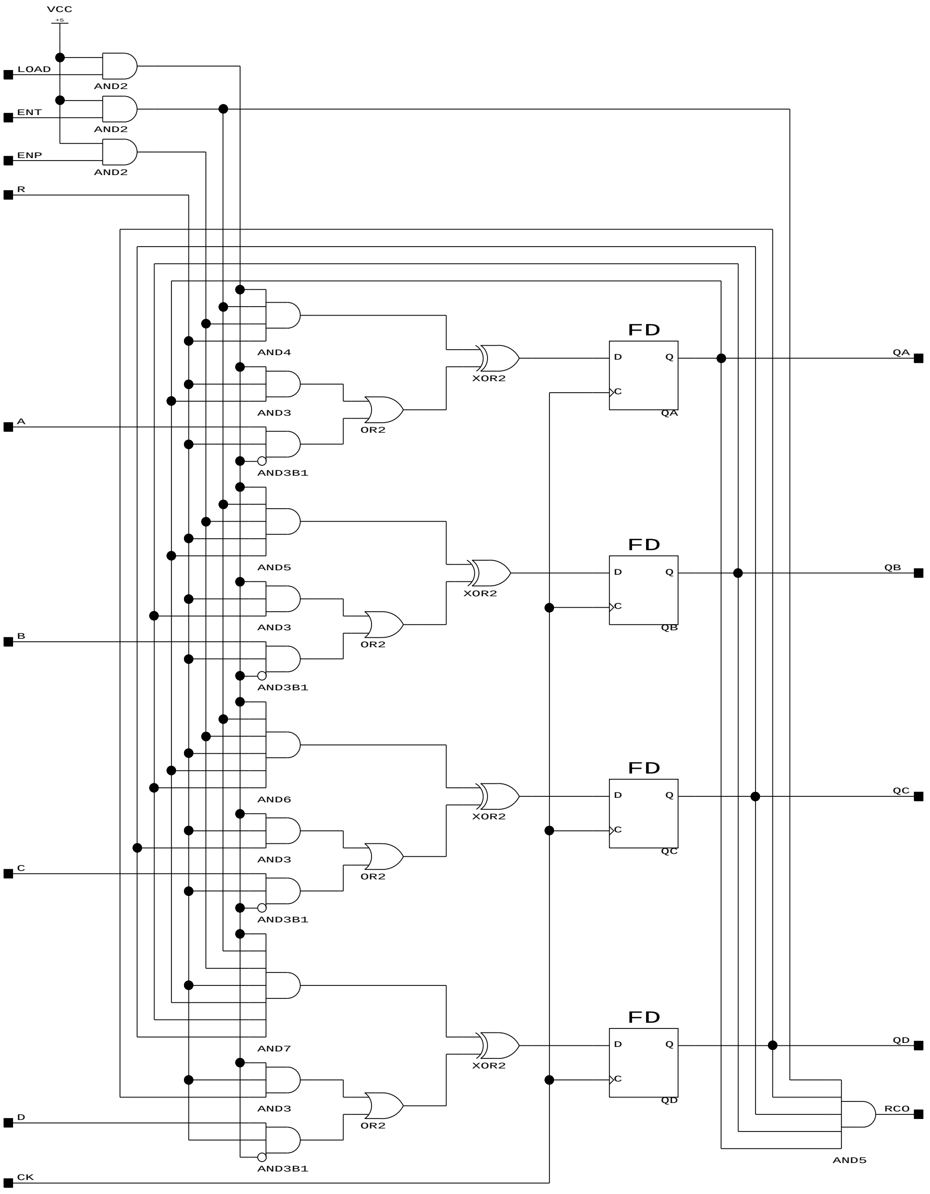
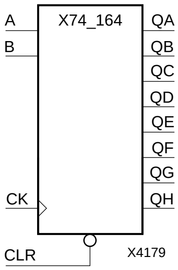


Figure 3-171 X74_163 Implementation (XC7000 and XC9000)

X74_164

8-Bit Serial-In Parallel-Out Shift Register with Active-Low Asynchronous Clear



XC7000	XC9000
Macro	Macro

X74_164 is an 8-bit, serial input (A and B), parallel output (QH – QA) shift register with an active-Low asynchronous clear (CLR) input. The asynchronous CLR, when Low, overrides the clock input and sets the data outputs (QH – QA) Low. When CLR is High, the AND function of the two data inputs (A and B) is loaded into the first bit of the shift register during the Low-to-High clock (C) transition and appears on the QA output. During subsequent Low-to-High clock transitions, with CLR High, the data is shifted to the next-highest bit position as new data is loaded into QA (A and B→QA, QA→QB, QB→QC, and so forth). The default initial state of all flip-flops is zero.

Registers can be cascaded by connecting the QH output of one stage to the A input of the next stage, by tying B High, and by connecting the clock and CLR inputs in parallel.

Inputs				Outputs	
CLR	A	B	CK	QA	QB – QH
0	X	X	X	0	0
1	1	1	↑	1	qA – qG
1	0	X	↑	0	qA – qG
1	X	0	↑	0	qA – qG

qA – qG = state of referenced output one set-up time prior to active clock transition

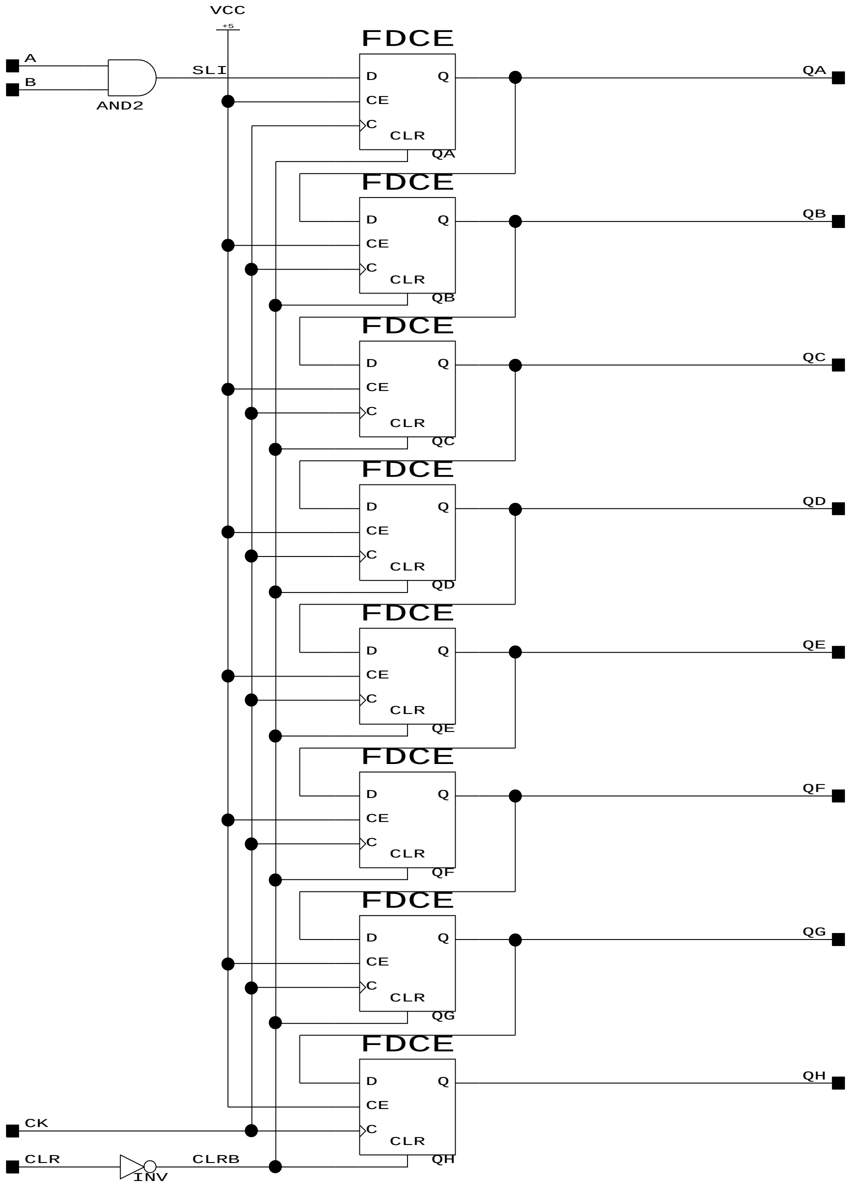
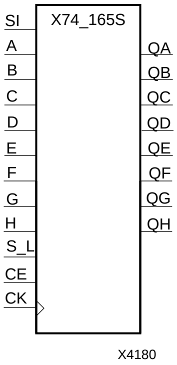


Figure 3-172 X74_164 Implementation (XC7000 and XC9000)

X74_165S

8-Bit Loadable Serial/Parallel-In Parallel-Out Shift Register with Clock Enable



XC7000	XC9000
Macro	Macro

X74_165S is an 8-bit shift register with serial-input (SI), parallel-inputs (H – A), parallel-outputs (QH – QA), and two control inputs – clock enable (CE) and active-Low shift/load enable (S_L). When S_L is Low, data on the H – A inputs is loaded into the corresponding QH – QA bits of the register on the Low-to-High clock (C) transition. When CE and S_L are High, data on the SI input is loaded into the first bit of the register during the Low-to-High clock transition. During subsequent Low-to-High clock transitions, with CE and S_L High, the data is shifted to the next-highest bit position (shift right) as new data is loaded into QA (SI→QA, QA→QB, QB→QC, and so forth). The register ignores clock transitions when CE is Low and S_L is High. The default initial state of all flip-flops is zero.

Registers can be cascaded by connecting the QH output of one stage to the SI input of the next stage and connecting clock, CE, and S_L inputs in parallel.

Inputs					Outputs	
S_L	CE	SI	A – H	CK	QA	QB – QH
0	X	X	A – H	↑	qa	qb – qh
1	0	X	X	X	--No Change---	
1	1	SI	X	↑	si	qA – qG

si, qn represent state of referenced input or output one set-up time prior to active clock transition.

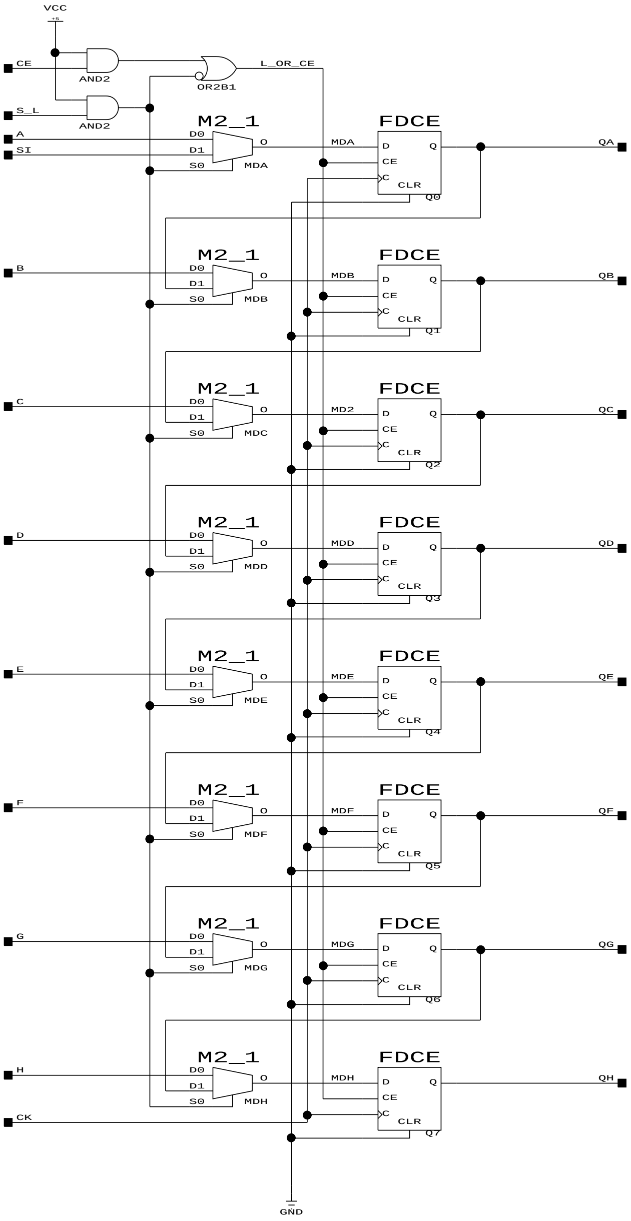
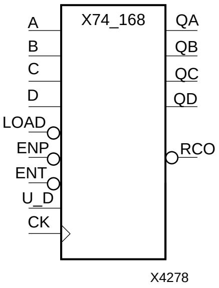


Figure 3-173 X74_165S Implementation (XC7000 and XC9000)

X74_168

4-Bit BCD Bidirectional Counter with Parallel and Trickle Clock Enables and Active-Low Load Enable



XC7000	XC9000
Macro	Macro

X74_168 is a 4-stage, 4-bit, synchronous, loadable, cascadable, bidirectional binary-coded-decimal (BCD) counter. The data on the D – A inputs is loaded into the counter when the active-Low load enable (LOAD) is Low during the Low-to-High clock (C) transition. The LOAD input, when Low, has priority over parallel clock enable (ENP), trickle clock enable (ENT), and the bidirectional (U_D) control. The outputs (QD – QA) increment when U_D and LOAD are High and ENP and ENT are Low during the Low-to-High clock transition. The outputs decrement when LOAD is High and ENP, ENT, and U_D are Low during the Low-to-High clock transition. The counter ignores clock transitions when LOAD and either ENP or ENT are High. The default initial state of all flip-flops is zero.

Inputs						Outputs	
LOAD	ENP	ENT	U_D	A – D	CK	QA – QD	RCO
0	X	X	X	A – D	↑	qa – qd	RCO
1	0	0	1	X	↑	Inc	RCO
1	0	0	0	X	↑	Dec	RCO
1	1	0	X	X	X	No Chg	RCO
1	X	1	X	X	X	No Chg	1

RCO = $(Q3 \bullet \overline{Q2} \bullet \overline{Q1} \bullet Q0 \bullet U\_D \bullet \overline{ENT}) + (\overline{Q3} \bullet \overline{Q2} \bullet \overline{Q1} \bullet \overline{Q0} \bullet \overline{U\_D} \bullet \overline{ENT})$

qa – qd = state of referenced input one set-up time prior to active clock transition

The active-Low ripple carry-out output (RCO) is Low when QD, QA, and U_D are High and QC, QB, and ENT are Low. RCO is also Low when all outputs, ENT and U_D are Low. The following figure illustrates a carry-lookahead design.

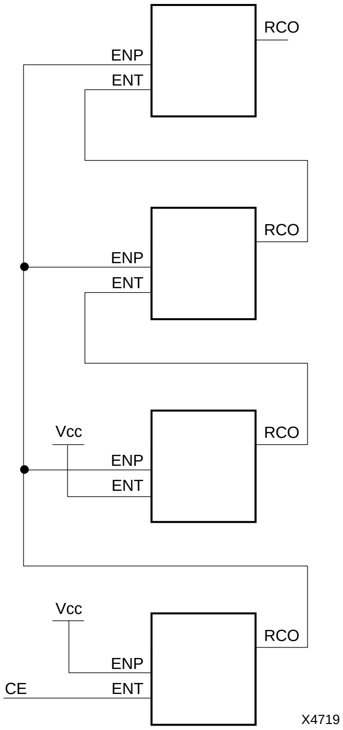


Figure 3-174 Carry-Lookahead Design

The RCO output of the first stage of the ripple carry is connected to the ENP input of the second stage and all subsequent stages. The RCO output of second stage and all subsequent stages is connected to the ENT input of the next stage. The ENT of the second stage is always enabled / tied to VCC. CE is always connected to the ENT input of the first stage. This cascading method allows the first stage of the ripple carry to be built as a prescaler. In other words, the first stage is built to count very fast.

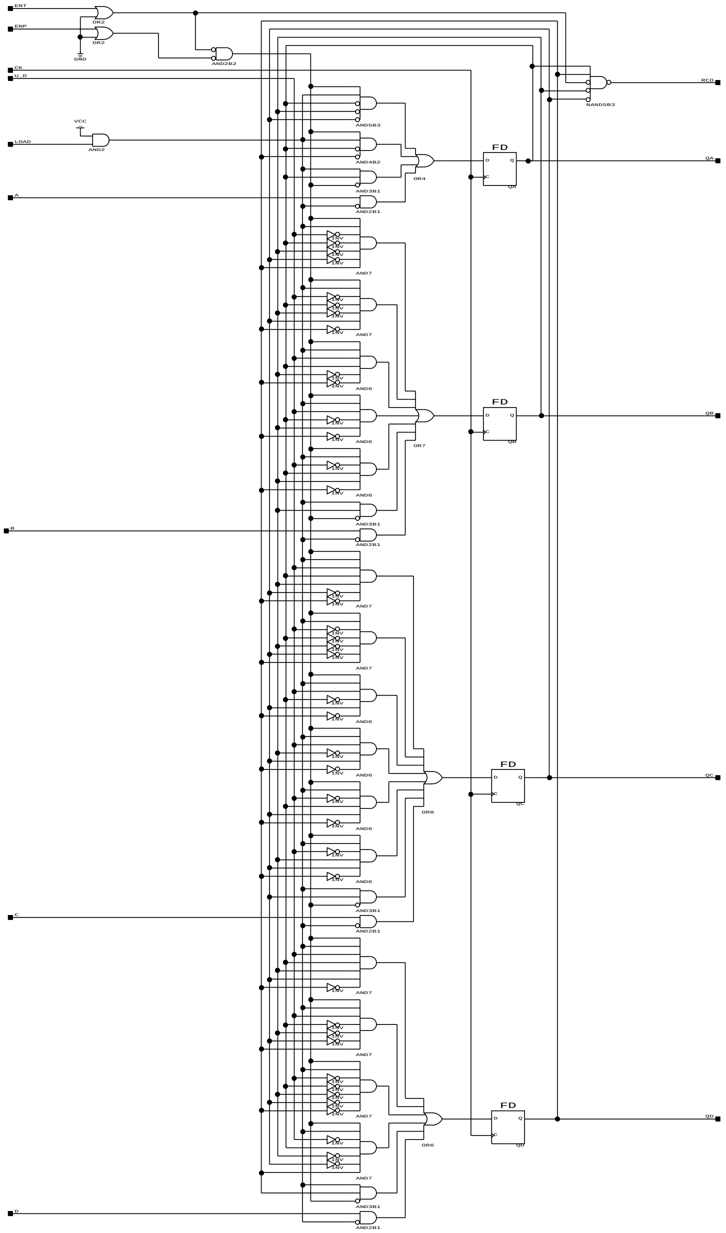
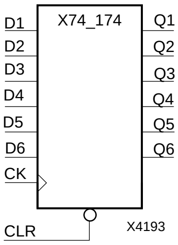


Figure 3-175 X74_168 Implementation (XC7000 and XC9000)

X74_174

6-Bit Data Register with Active-Low Asynchronous Clear



XC7000	XC9000
Macro	Macro

The active-Low asynchronous clear input (CLR), when Low, overrides the clock and resets the six data outputs (Q6 – Q1) Low. When CLR is High, the data on the six data inputs (D6 – D1) is transferred to the corresponding data outputs on the Low-to-High clock (C) transition. The default initial state of all flip-flops is zero.

Inputs			Outputs
CLR	**D6 – D1**	**CK**	**Q6 – Q1**
0	X	X	0
1	D6 – D1	↑	d6 – d1

dn = state of referenced input one set-up time prior to active clock transition

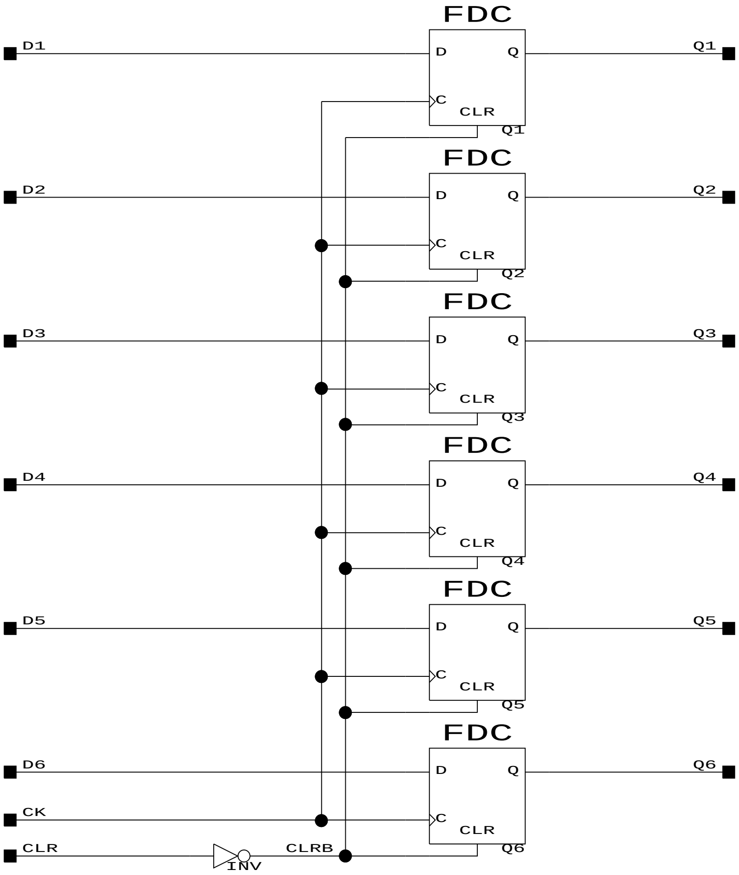
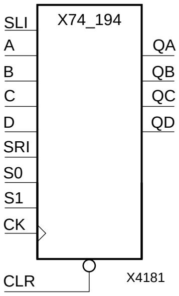


Figure 3-176 X74_174 Implementation (XC7000 and XC9000)

X74_194

4-Bit Loadable Bidirectional Serial/Parallel-In Parallel-Out Shift Register



XC7000	XC9000
Macro	Macro

X74_194 is a 4-bit shift register with shift-right serial input (SRI), shift-left serial input (SLI), parallel inputs (D – A), parallel outputs (QD – QA), two control inputs (S1, S0), and active-Low asynchronous clear (CLR). The shift register performs the following functions.

- Clear — When CLR is Low, all other inputs are ignored and outputs QD – QA go to logic state zero during the Low-to-High clock transition.
- Load — When S1 and S0 are High, the data on inputs D – A is loaded into the corresponding output bits QD – QA during the Low-to-High clock transition.
- Shift Right — When S1 is Low and S0 is High, the data is shifted to the next-highest bit position (right) as new data is loaded into QA (SRI→QA, QA→QB, QB→QC, and so forth).
- Shift Left — When S1 is High and S0 is Low, the data is shifted to the next-lowest bit position (left) as new data is loaded into QD (SLI→QD, QD→QC, QC→QB, and so forth).

Registers can be cascaded by connecting the QD output of one stage to the SRI input of the next stage, the QA output of one stage to the SRI input of the next stage, and connecting clock, S1, S0, and CLR inputs in parallel.

The default initial state of all flip-flops is zero.

Inputs							Outputs			
CLR	**S1**	**S0**	**SRI**	**SLI**	**A – D**	**CK**	**QA**	**QB**	**QC**	**QD**
0	X	X	X	X	X	X	0	0	0	0
1	0	0	X	X	X	X	-----No Change--------			
1	1	1	X	X	A – D	↑	a	b	c	d
1	0	1	SRI	X	X	↑	sri	qa	qb	qc
1	1	0	X	SLI	X	↑	qb	qc	qd	sli

Lowercase letters represent state of referenced input or output one set-up time prior to active clock transition.

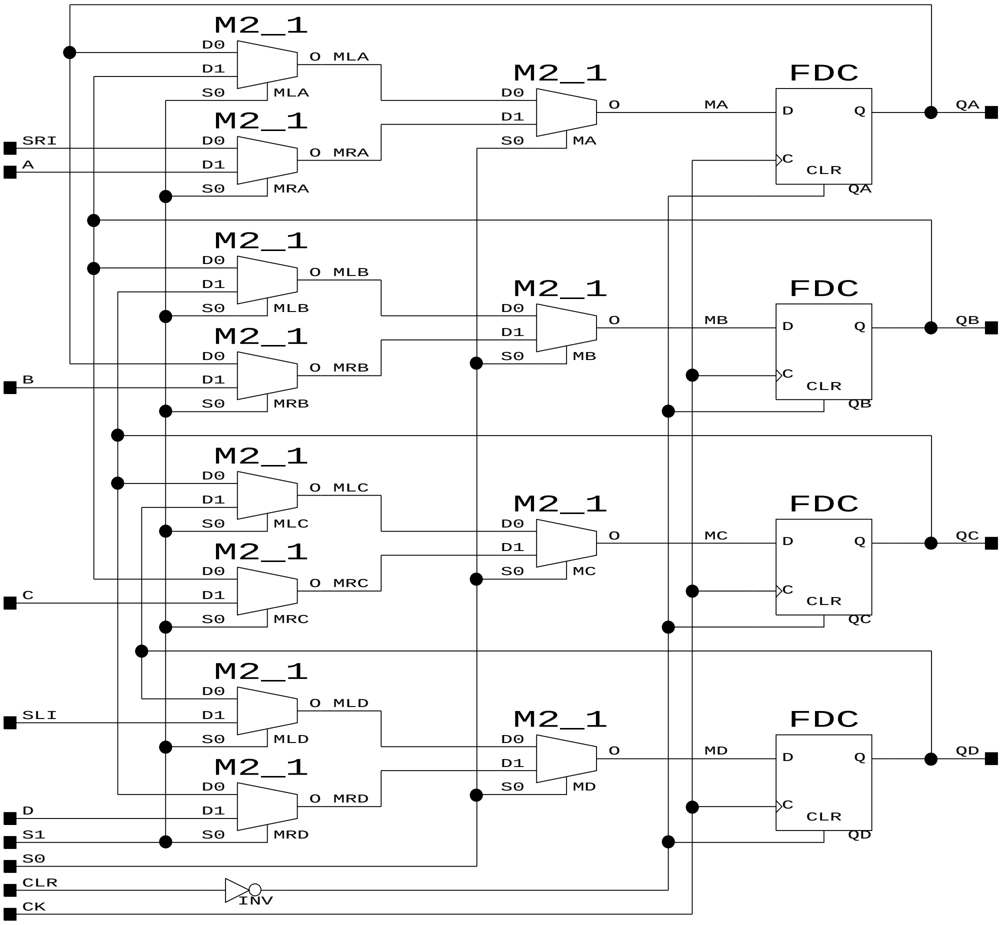
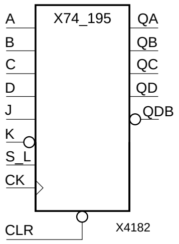


Figure 3-177 X74_194 Implementation (XC7000 and XC9000)

X74_195

4-Bit Loadable Serial/Parallel-In Parallel-Out Shift Register



XC7000	XC9000
Macro	Macro

X74_195 is a 4-bit shift register with shift-right serial inputs (J and K), parallel inputs (D – A), parallel outputs (QD – QA) and QDB, shift / load control input (S_L), and active-Low asynchronous clear (CLR). Asynchronous CLR, when Low, overrides all other inputs and resets data outputs QD – QA Low and QDB High. When S_L is Low and CLR is High, data on the D – A inputs is loaded into the corresponding QD – QA bits of the register during the Low-to-High clock (C) transition. When S_L and CLR are High, the first bit of the register (QA) responds to the J and K inputs during the Low-to-High clock transition, as shown in the truth table. During subsequent Low-to-High clock transitions, with S_L and CLR High, the data is shifted to the next-highest bit position (shift right) as new data is loaded into QA (J, K→QA, QA→QB, QB→QC, and so forth). The default initial state of all flip-flops is zero.

Registers can be cascaded by connecting the QD and QDB outputs of one stage to the J and K inputs, respectively, of the next stage and connecting clock, S_L and CLR inputs in parallel.

Inputs						Outputs				
CLR	**S_L**	**J**	**K**	**A – D**	**CK**	**QA**	**QB**	**QC**	**QD**	**QDB**
0	X	X	X	X	X	0	0	0	0	1
1	0	X	X	A – D	↑	a	b	c	d	$\overline{d}$
1	1	0	0	X	↑	0	qa	qb	qc	$\overline{qc}$
1	1	1	1	X	↑	1	qa	qb	qc	$\overline{qc}$
1	1	0	1	X	↑	qa	qa	qb	qc	$\overline{qc}$
1	1	1	0	X	↑	$\overline{qa}$	qa	qb	qc	$\overline{qc}$

Lowercase letters represent state of referenced input or output one set-up time prior to active clock transition.

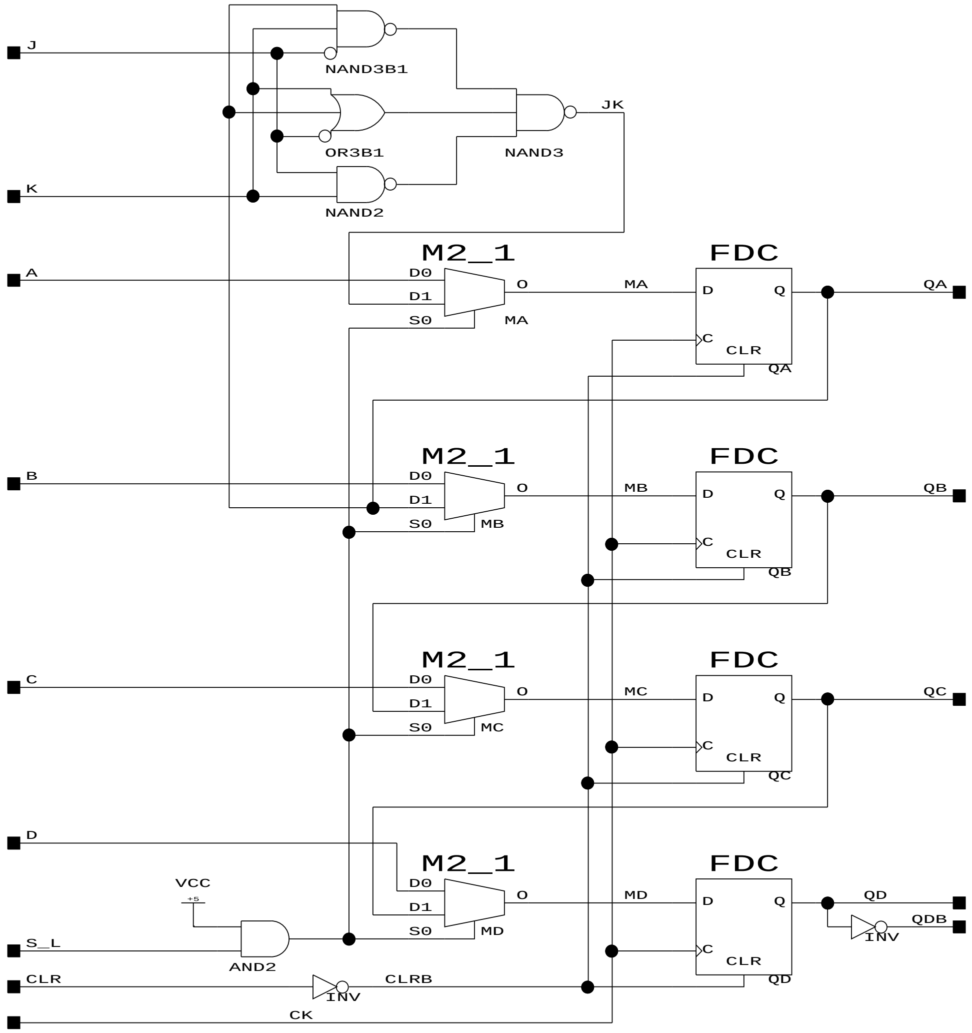
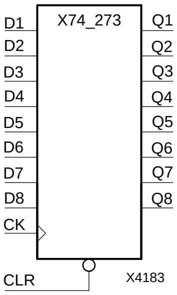


Figure 3-178 X74_195 Implementation (XC7000 and XC9000)

X74_273

8-Bit Data Register with Active-Low Asynchronous Clear



XC7000	XC9000
Macro	Macro

The active-Low asynchronous clear (CLR), when Low, overrides all other inputs and resets the data outputs (Q8 – Q1) Low. When CLR is High, the data on the data inputs (D8 – D1) is transferred to the corresponding data outputs (Q8 – Q1) during the Low-to-High clock transition. The default initial state of all flip-flops is zero.

Inputs			Outputs
CLR	D8 – D1	CK	Q8 – Q1
0	X	X	0
1	D8 – D1	↑	d8 – d1

dn = state of referenced input one set-up time prior to active clock transition

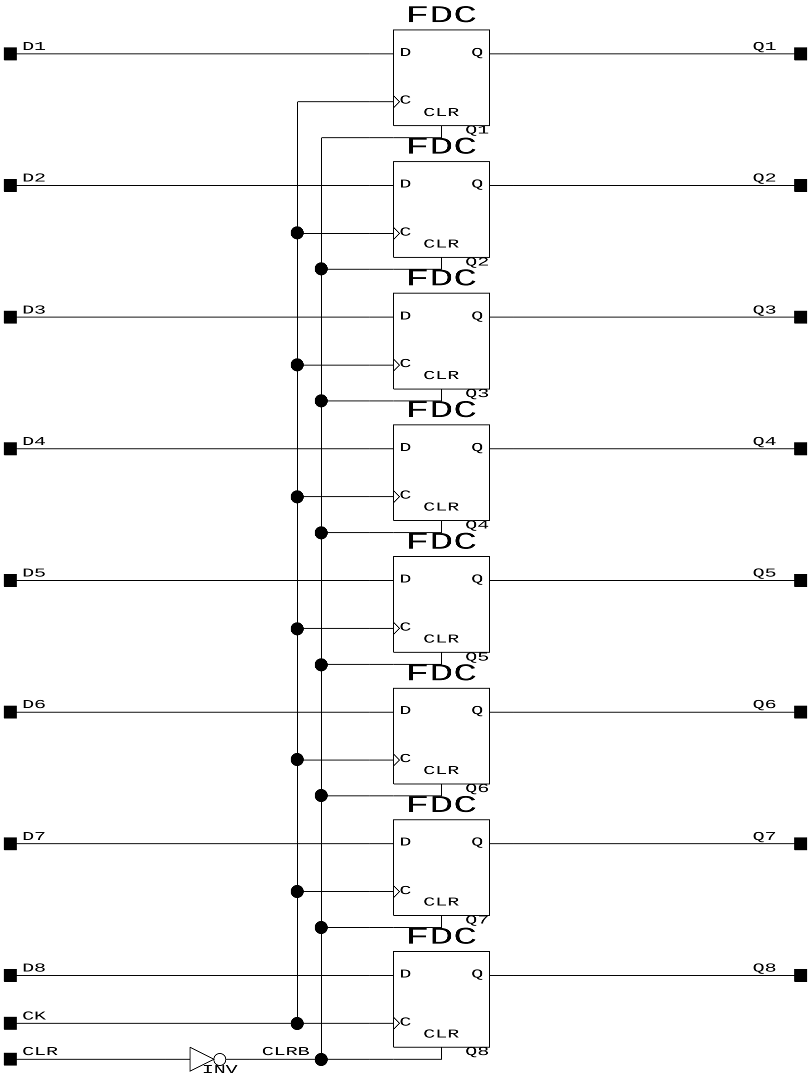
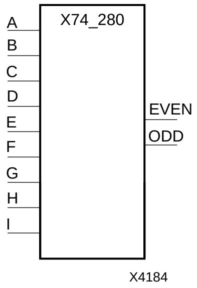


Figure 3-179   X74_273 Implementation (XC7000 and XC9000)

X74_280

9-Bit Odd/Even Parity Generator/Checker



XC7000	XC9000
Macro	Macro

X74_280 parity generator/checker compares up to nine data inputs (I – A) and provides both even (EVEN) and odd parity (ODD) outputs. The EVEN output is High when an even number of inputs is High. The ODD output is High when an odd number of inputs is High.

Expansion to larger word sizes is accomplished by tying the ODD outputs of up to nine parallel components to the data inputs of one more X74_280; all other inputs are tied to ground.

Inputs	Outputs	
Number of Ones on A – I	EVEN	ODD
0, 2, 4, 6, or 8	1	0
1, 3, 5, 7, or 9	0	1

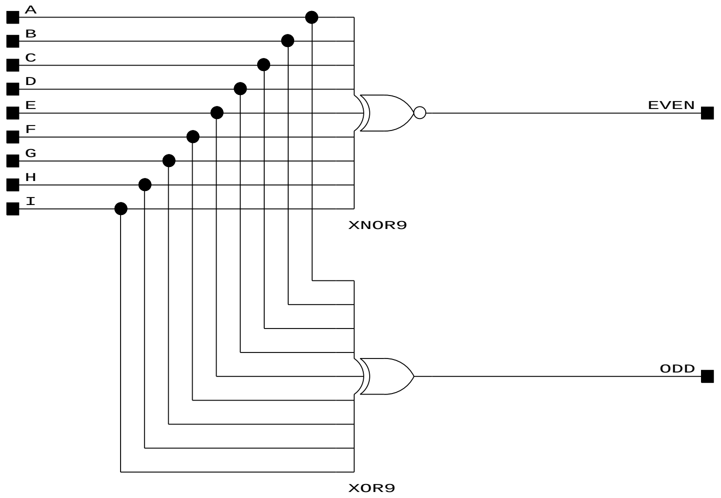
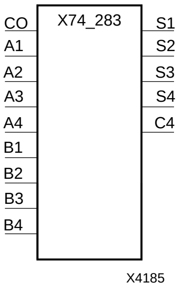


Figure 3-180 X74_280 Implementation (XC7000 and XC9000)

X74_283

4-Bit Full Adder with Carry-In and Carry-Out



XC7000	XC9000
Macro*	Macro

* not supported for XC7336 designs

X74_283, a 4-bit full adder with carry-in and carry-out, adds two 4-bit words (A4 – A1 and B4 – B1) and a carry-in (C0) and produces a binary sum output (S4 – S1) and a carry-out (C4).

16(C4)+8(S4)+4(S3)+2(S2)+S1=8(A4+B4)+4(A3+B3)+2(A2+B2)+(A1+B1)+CO, where “+” = addition.

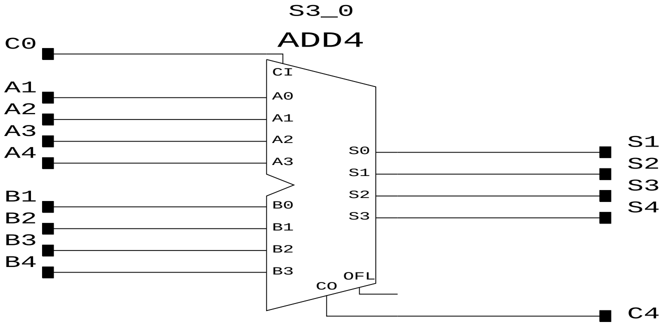
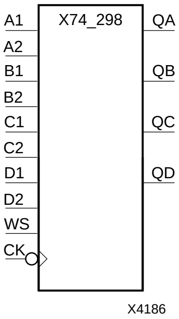


Figure 3-181 X74_283 Implementation (XC7000 and XC9000)

X74_298

Quadruple 2-Input Multiplexer with Storage and Negative-Edge Clock



XC7000	XC9000
Macro*	Macro

* not supported for XC7336 designs

X74_298 selects 4-bits of data from two sources (D1 – A1 or D2 – A2) under the control of a common word select input (WS). When WS is Low, D1 – A1 is chosen, and when WS is High, D2 – A2 is chosen. The selected data is transferred into the output register (QD – QA) during the High-to-Low transition of the negative-edge triggered clock (CK). For XC7000, the CK input cannot be driven by a FastCLK signal (from BUFG). The default initial state of all flip-flops is zero.

Inputs				Outputs
WS	A1 – D1	A2 – D2	CK	QA – QD
0	A1 – D1	X	↓	a1 – d1
1	X	A2 – D2	↓	a2 – d2

an – dn = state of referenced input one set-up time prior to active clock transition

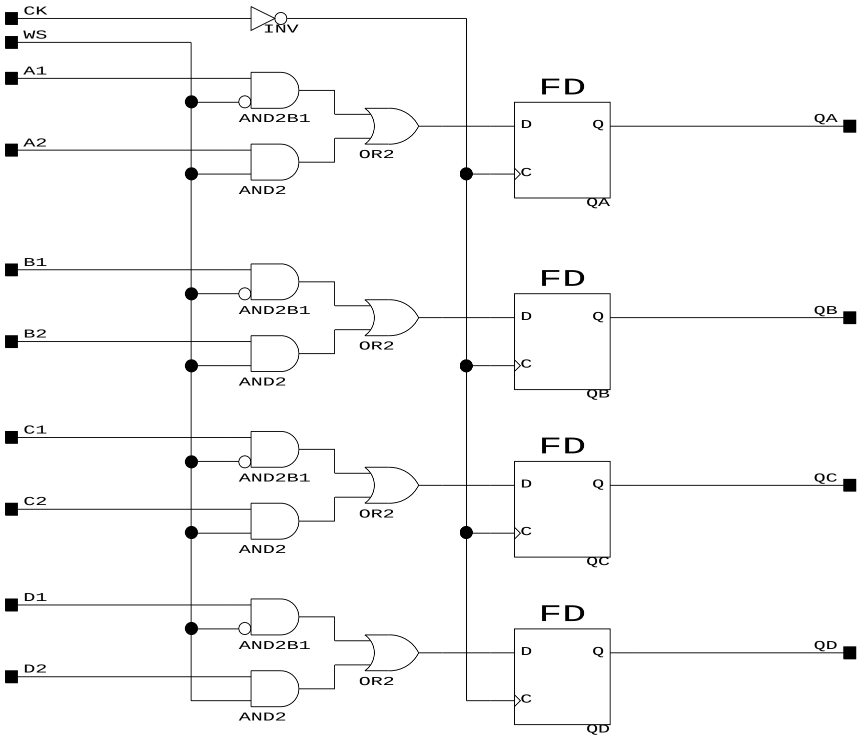
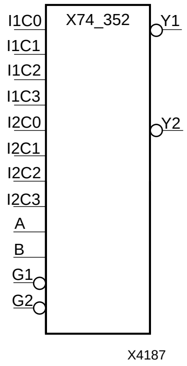


Figure 3-182 X74_298 Implementation (XC7000 and XC9000)

X74_352

Dual 4-to-1 Multiplexer with Active-Low Enables and Outputs



XC7000	XC9000
Macro	Macro

X74_352 comprises two 4-to-1 multiplexers with separate enables (G1 and G2) but with common select inputs (A and B). When an active-Low enable (G1 or G2) is Low, the multiplexer chooses one data bit from the four sources associated with the particular enable (I1C3 – I1C0 for G1 and I2C3 – I2C0 for G2) under the control of the common select inputs (A and B). The active-Low outputs (Y1 and Y2) reflect the inverse of the selected data as shown in truth table. Y1 is associated with G1 and Y2 is associated with G2. When an active-Low enable is High, the associated output is High.

Inputs							Outputs
G	**B**	**A**	**IC0**	**IC1**	**IC2**	**IC3**	**Y**
1	X	X	X	X	X	X	1
0	0	0	IC0	X	X	X	$\overline{\text{IC0}}$
0	0	1	X	IC1	X	X	$\overline{\text{IC1}}$
0	1	0	X	X	IC2	X	$\overline{\text{IC2}}$
0	1	1	X	X	X	IC3	$\overline{\text{IC3}}$

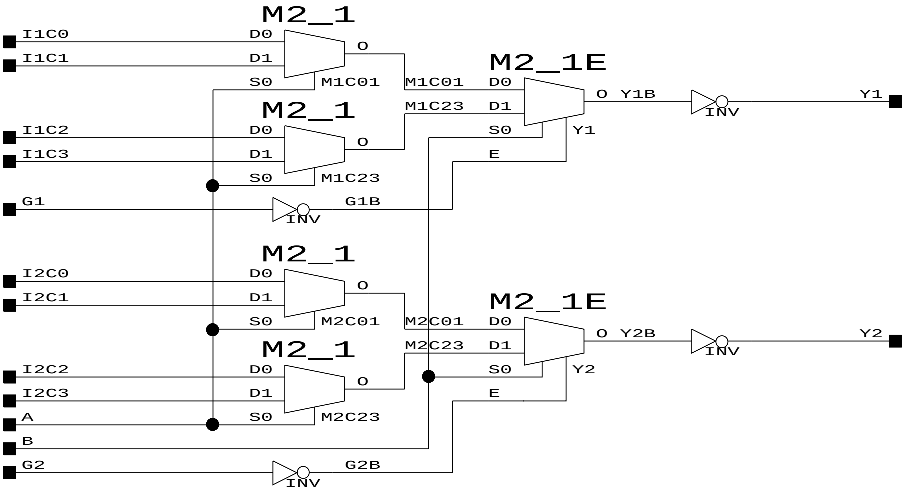
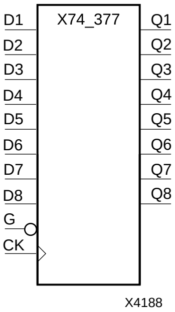


Figure 3-183   X74_352 Implementation (XC7000 and XC9000)

X74_377

8-Bit Data Register with Active-Low Clock Enable



XC7000	XC9000
Macro	Macro

When the active-Low clock enable (G) is Low, the data on the eight data inputs (D8 – D1) is transferred to the corresponding data outputs (Q8 – Q1) during the Low-to-High clock (CK) transition. The register ignores clock transitions when G is High. The default initial state of all flip-flops is zero.

Inputs			Outputs
G	D8 – D1	CK	Q8 – Q1
1	X	X	No Change
0	D8 – D1	↑	d8 – d1

dn = state of referenced input one set-up time prior to active clock transition

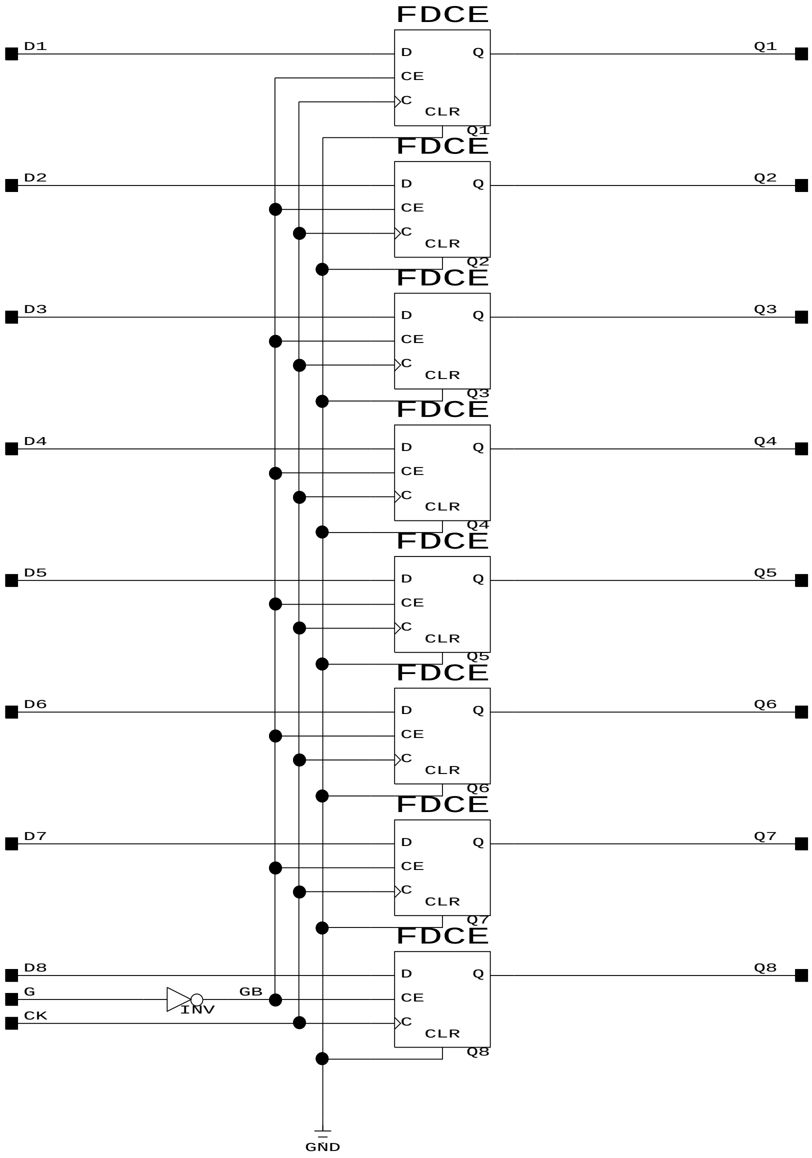
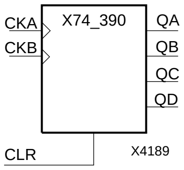


Figure 3-184 X74_377 Implementation (XC7000 and XC9000)

X74_390

4-Bit BCD/Bi-Quinary Ripple Counter with Negative-Edge Clocks and Asynchronous Clear



XC7000	XC9000
Macro*	Macro

* not supported for XC7336 designs

X74_390 is a cascadable, resettable binary-coded decimal (BCD) or bi-quinary counter that can be used to implement cycle lengths equal to whole and / or cumulative multiples of 2 and / or 5. In BCD mode, the output QA is connected to negative-edge clock input (CKB), and data outputs (QD – QA) increment during the High-to-Low clock transition as shown in the truth table, provided asynchronous clear (CLR) is Low. In bi-quinary mode, output QD is connected to the negative-edge clock input (CKA). As shown in the truth table, in bi-quinary mode, QA supplies a divide-by-five output and QB supplies a divide-by-two output, provided asynchronous CLR is Low. When asynchronous CLR is High, the other inputs are overridden, and data outputs (QD – QA) are reset Low. The default initial state of all flip-flops is zero.

Larger ripple counters are created by connecting the QD output (BCD mode) or QA output (biquinary mode) of the first stage to the appropriate clock input of the next stage and connecting the CLR inputs in parallel. For XC7000, CKA and CKB cannot be driven by a FastCLK signal from (BUFG).

Count	BCD				Bi-Quinary			
	QD	**QC**	**QB**	**QA**	**QD**	**QC**	**QB**	**QA**
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	1	0
2	0	0	1	0	0	1	0	0
3	0	0	1	1	0	1	1	0
4	0	1	0	0	1	0	0	0
5	0	1	0	1	0	0	0	1
6	0	1	1	0	0	0	1	1
7	0	1	1	1	0	1	0	1
8	1	0	0	0	0	1	1	1
9	1	0	0	1	1	0	0	1

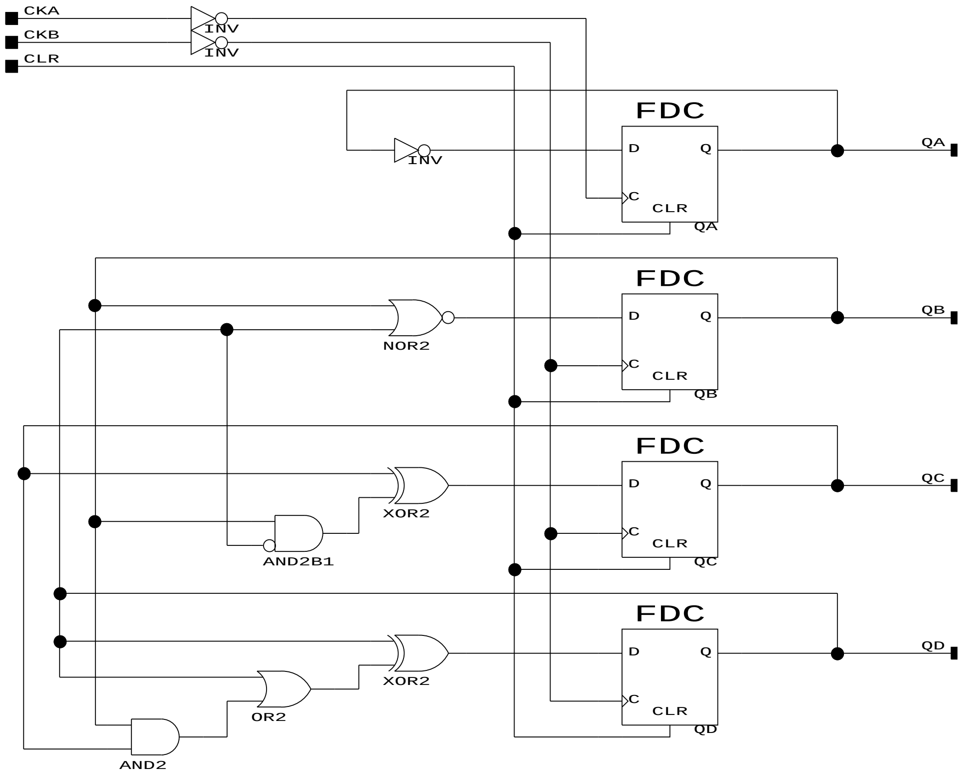
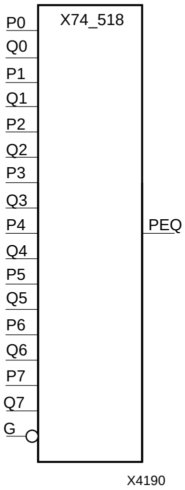


Figure 3-185 X74_390 Implementation (XC7000 and XC9000)

X74_518

8-Bit Identity Comparator with Active-Low Enable



XC7000	XC9000
Macro	Macro

X74_518 is an 8-bit identity comparator with 16 data inputs for two 8-bit words (P7 – P0 and Q7 – Q0), data output (PEQ), and active-Low enable (G). When G is High, the PEQ output is Low. When G is Low and the two input words are equal, PEQ is High. Equality is determined by a bit comparison of the two words. When any of the two equivalent bits from the two words are not equal, PEQ is Low.

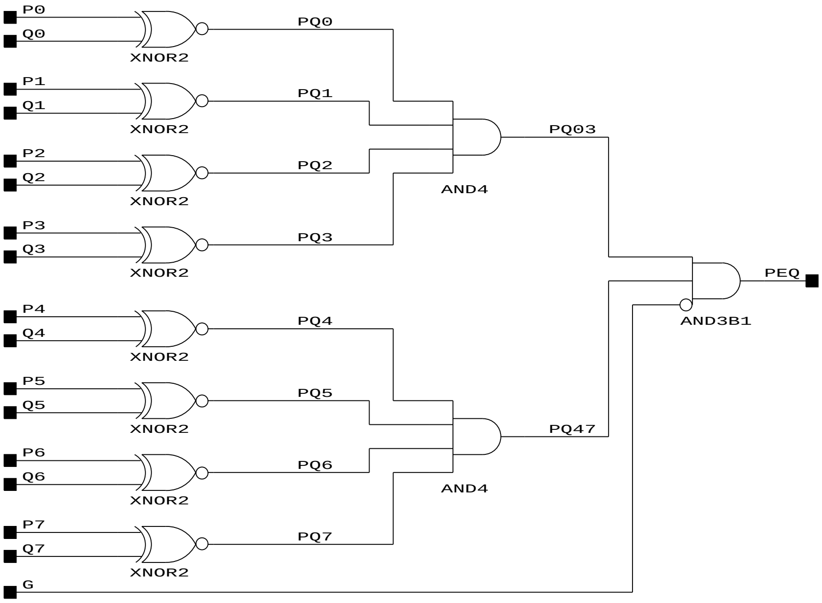
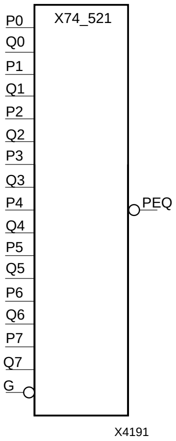


Figure 3-186   X74_518 Implementation (XC7000 and XC9000)

X74_521

8-Bit Identity Comparator with Active-Low Enable and Output



XC7000	XC9000
Macro	Macro

X74_521 is an 8-bit identity comparator with 16 data inputs for two 8-bit words (P7 – P0 and Q7 – Q0), active-Low data output (PEQ), and active-Low enable (G). When G is High, the PEQ output is High. When G is Low and the two input words are equal, PEQ is Low. X74_521 does a bit comparison of the two words to determine equality. When any of the two equivalent bits from the two words are not equal, PEQ is High.

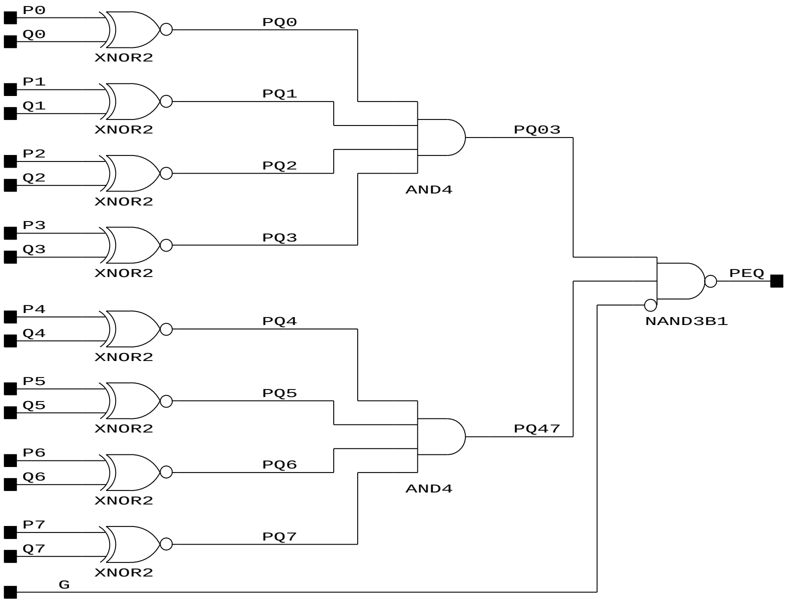


Figure 3-187 X74_521 Implementation (XC7000 and XC9000)