Chapter 10

Design Elements (SOP3 to XORCY_L)

This chapter describes design elements included in the Unified Libraries. The elements are organized in alphanumeric order with all numeric suffixes in ascending order.

Information on the specific architectures supported by each of the following libraries is contained under the Applicable Architectures section of the Unified Libraries Chapter.

- **XC3000 Library**
- **XC4000E Library**
- **XC4000X Library**
- **XC5200 Library**
- **XC9000 Library**
- **Spartan Library**
- **SpartanXL Library**
- **Virtex Library**

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

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

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

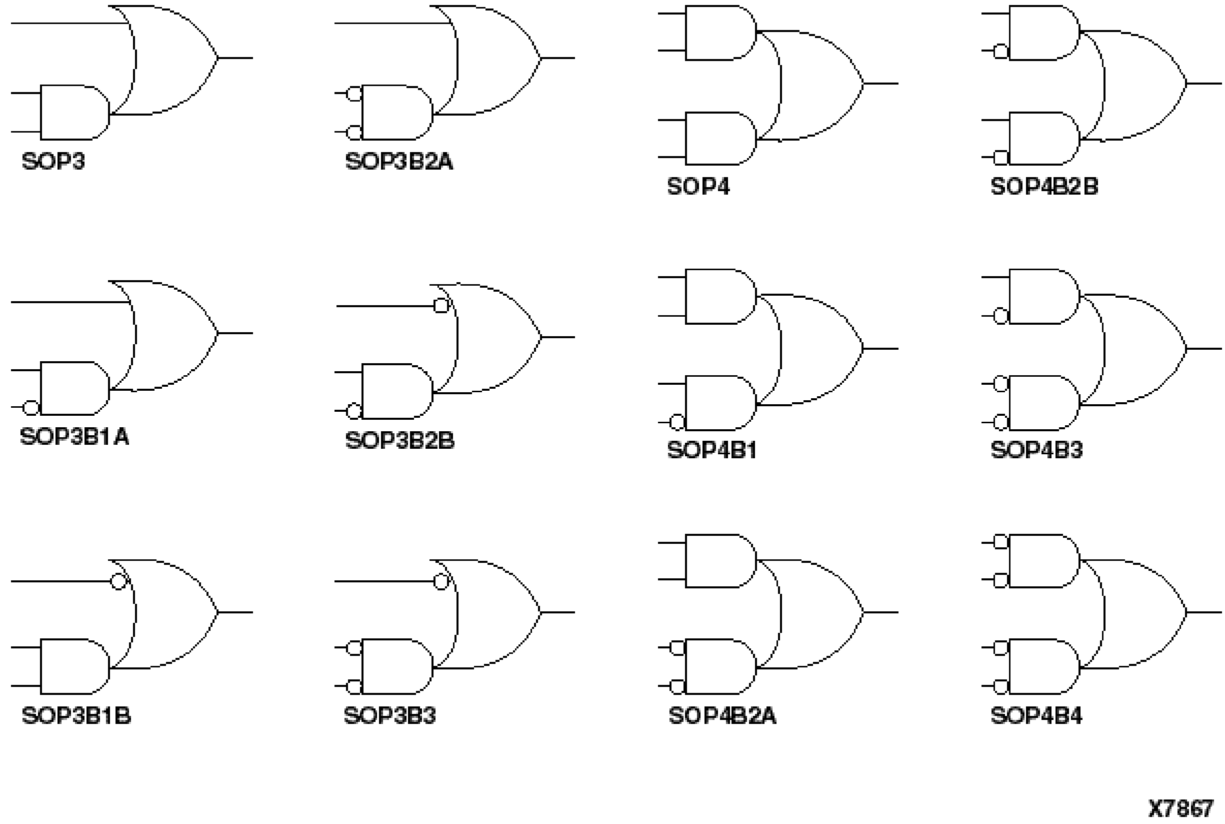
SOP3-4

Sum of Products

Element	XC3000	XC4000E	XC4000X	XC5200	XC9000	Spartan	SpartanXL	Virtex
SOP3, SOP3B1A, SOP3B1B, SOP3B	Macro	Macro	Macro	Macro	Macro	Macro	Macro	Macro

2A,
SOP3B
2B,
SOP3B
3
SOP4,
SOP4B
1,
SOP4B
2A,
SOP4B
2B,
SOP4B
3,
SOP4B
4

Figure 10-1SOP Gate Representations



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 10-2SOP3 Implementation XC3000, XC4000, XC5200, XC9000, Spartans, Virtex

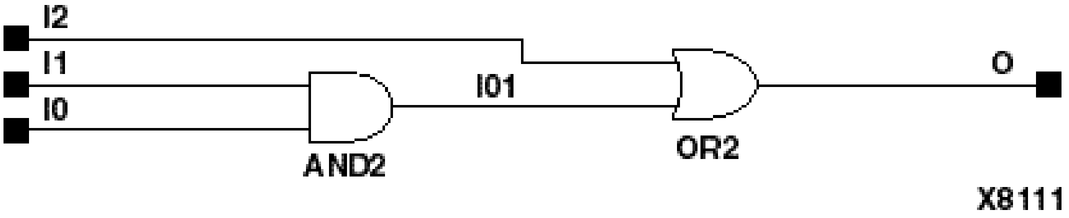
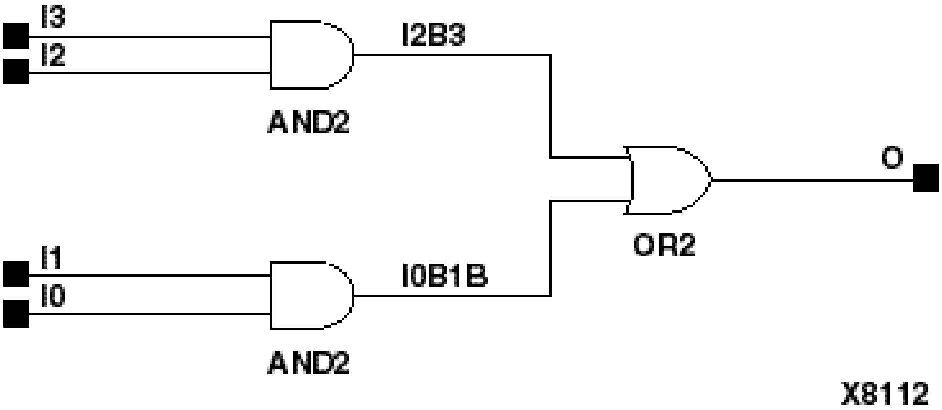


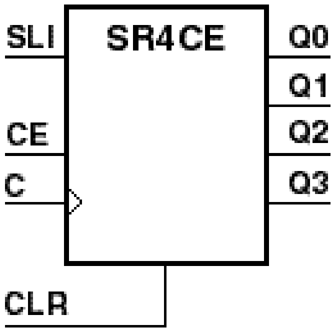
Figure 10-3SOP4 Implementation XC3000, XC4000, XC5200, XC9000, Spartans, Virtex



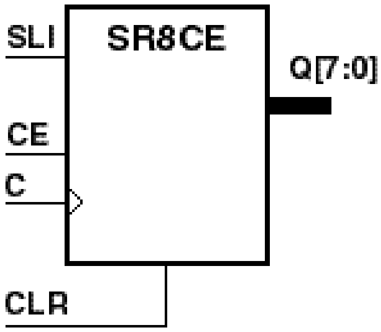
SR4CE, SR8CE, SR16CE

4-, 8-, 16-Bit Serial-In Parallel-Out Shift Registers with Clock Enable and Asynchronous Clear

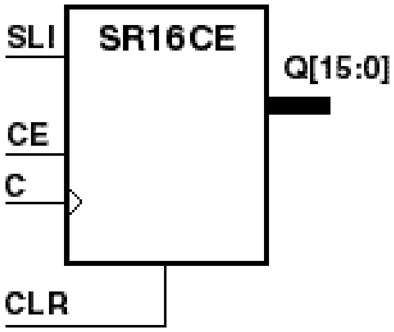
XC3000	XC4000 E	XC4000 X	XC5200	XC9000	Spartan	Spartan XL	Virtex
Macro	Macro	Macro	Macro	Macro	Macro	Macro	Macro



X4145



X4151



X4157

SR4CE, SR8CE, and SR16CE are 4-, 8-, and 16-bit shift registers, respectively, with a shift-left serial input (SLI), parallel outputs (Q), and clock enable (CE) and asynchronous clear (CLR) inputs. The CLR input, when High, overrides all other inputs and resets the data outputs (Q) 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.

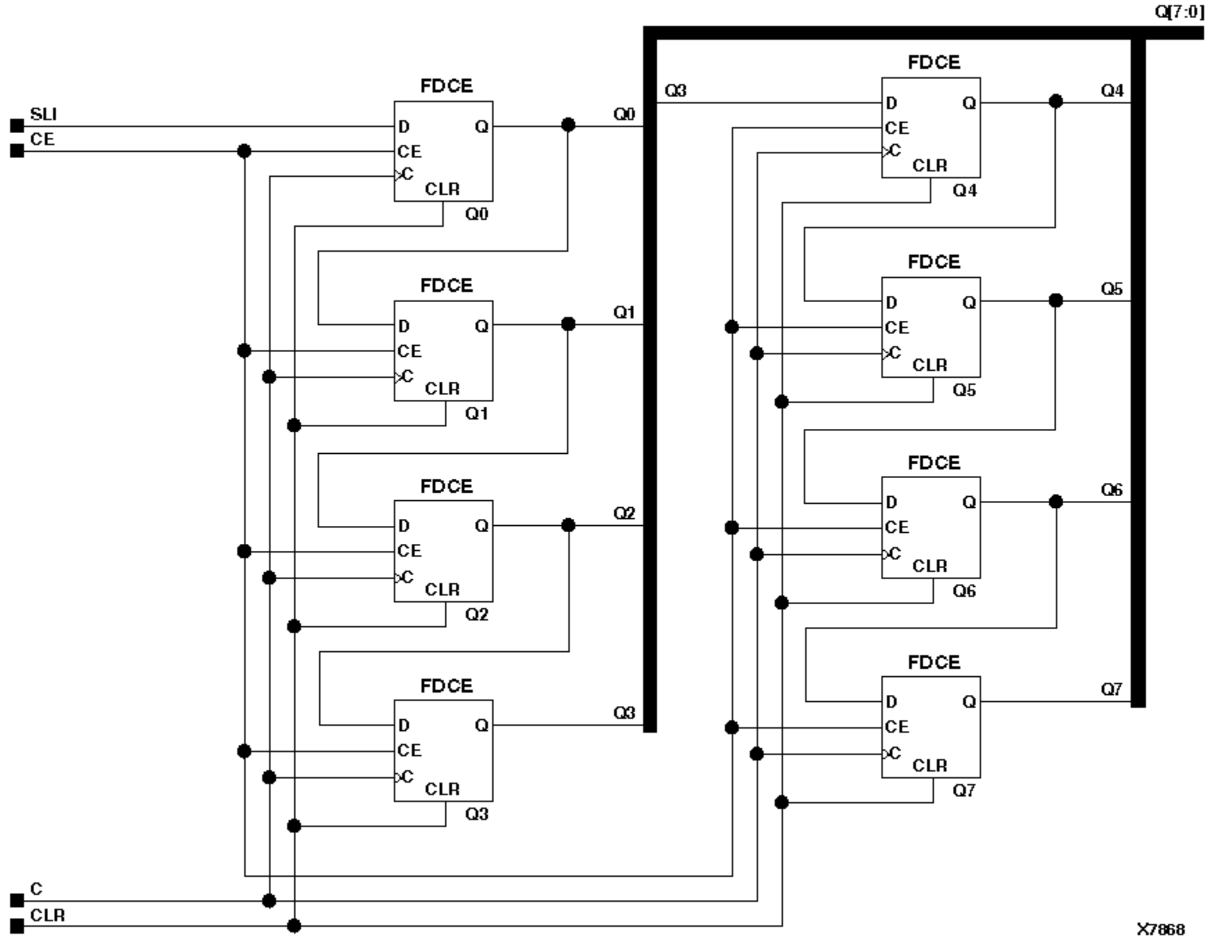
Registers can be cascaded by connecting the last Q output (Q3 for SR4CE, Q7 for SR8CE, or Q15 for SR16CE) of one stage to the SLI input of the next stage and connecting clock, CE, and CLR in parallel.

The register is asynchronously cleared, outputs Low, when power is applied. For CPLDs, the power-on condition can be simulated by applying a High-level pulse on the PRLD global net. FPGAs simulate power-on when global reset (GR) or global set/reset (GSR) is active. GR for XC3000 is active-Low. GR for XC5200 and GSR (XC4000, Spartans, Virtex) default to active-High but can be inverted by adding an inverter in front of the GR/GSR input of the STARTUP or STARTUP_VIRTEX symbol.

Inputs				Outputs	
CLR	**CE**	**SLI**	**C**	**Q0**	**Qz – Q1**
1	X	X	X	0	0
0	0	X	X	No Chg	No Chg
0	1	1	↑	1	qn-1
0	1	0	↑	0	qn-1

z = 3 for SR4CE; z = 7for SR8CE; z = 15 for SR16CE
qn-1 = state of referenced output one setup time prior to active clock transition

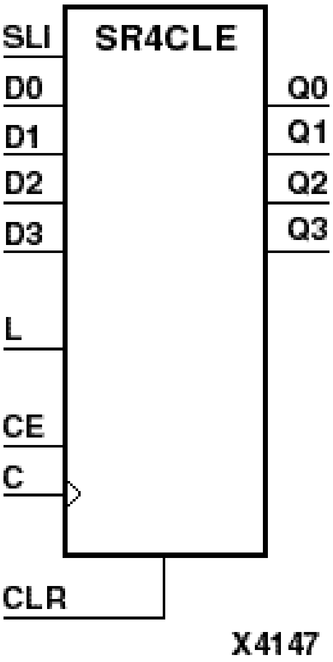
Figure 10-4SR8CE Implementation XC3000, XC4000, XC5200, XC9000, Spartans, Virtex

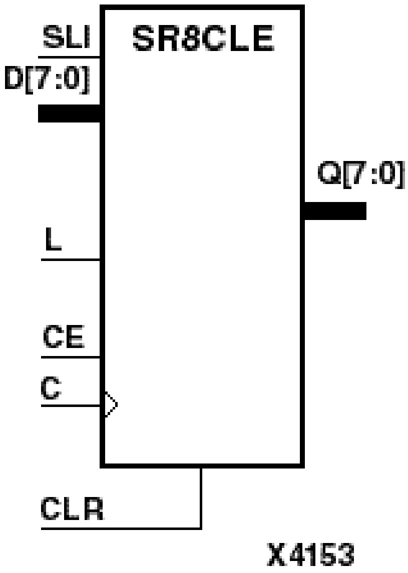


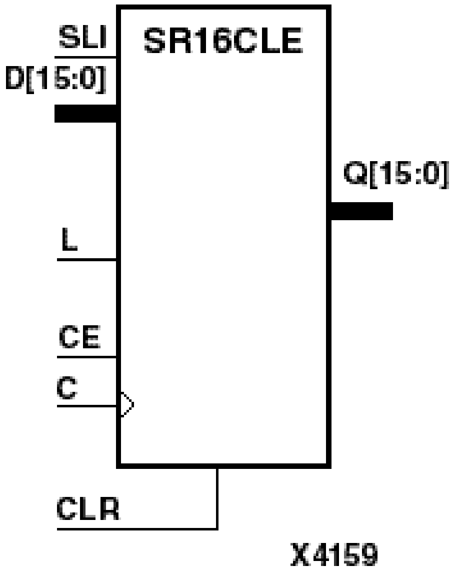
SR4CLE, SR8CLE, SR16CLE

4-, 8-, 16-Bit Loadable Serial/Parallel-In Parallel-Out Shift Registers with Clock Enable and Asynchronous Clear

XC3000	XC4000 E	XC4000 X	XC5200	XC9000	Spartan	Spartan XL	Virtex
Macro	Macro	Macro	Macro	Macro	Macro	Macro	Macro







SR4CLE, SR8CLE, and SR16CLE are 4-, 8-, and 16-bit shift registers, respectively, with a shift-left serial input (SLI), parallel inputs (D), parallel outputs (Q), 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 (Q) Low. When L is High and CLR is Low, data on the Dn – D0 inputs is loaded into the corresponding Qn – 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).

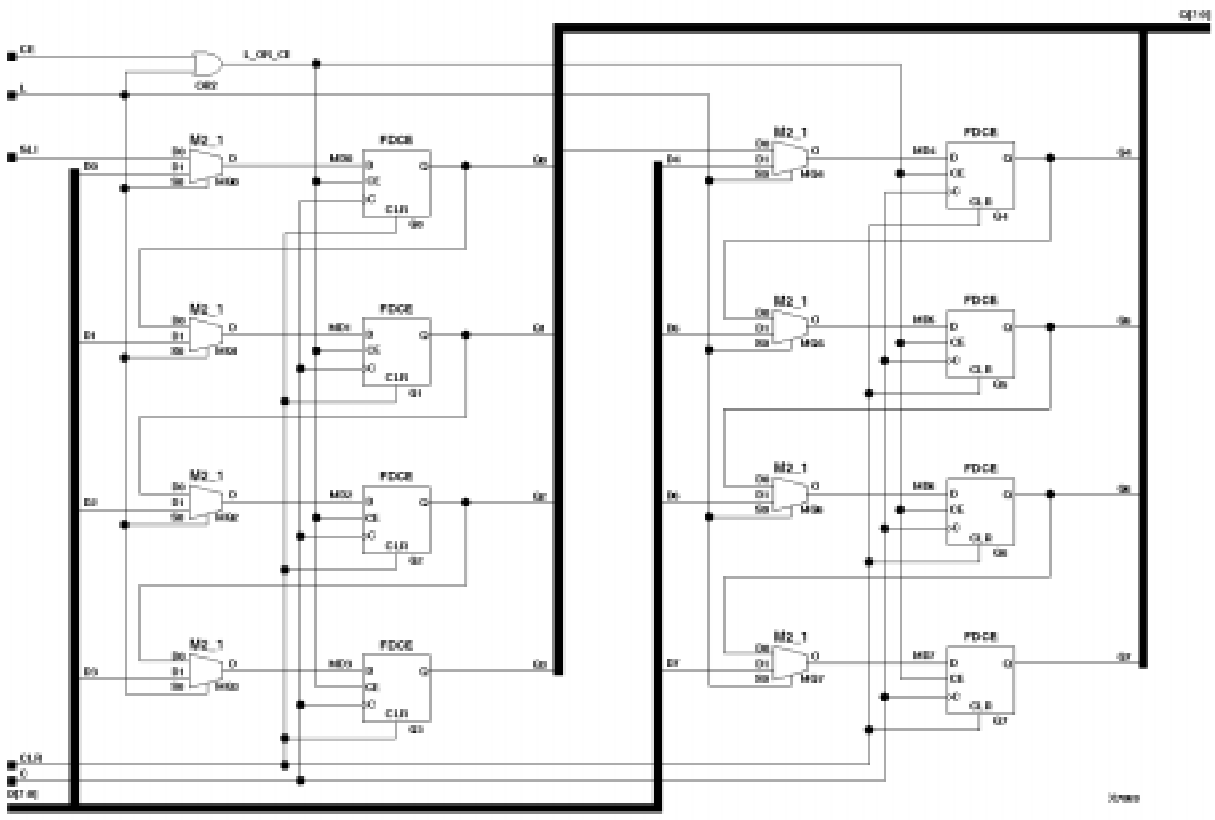
Registers can be cascaded by connecting the last Q output (Q3 for SR4CLE, Q7 for SR8CLE, or Q15 for SR16CLE) of one stage to the SLI input of the next stage and connecting clock, CE, L, and CLR inputs in
parallel.

The register is asynchronously cleared, outputs Low, when power is applied. For CPLDs, the power-on condition can be simulated by applying a High-level pulse on the PRLD global net. FPGAs simulate power-on when global reset (GR) or global set/reset (GSR) is active. GR for XC3000 is active-Low. GR for XC5200 and GSR (XC4000, Spartans, Virtex) default to active-High but can be inverted by adding an inverter in front of the GR/GSR input of the STARTUP or STARTUP_VIRTEX symbol.

Inputs						**Outputs**	
CLR	**L**	**CE**	**SLI**	**Dn – D0**	**C**	**Q0**	**Qz – Q1**
1	X	X	X	X	X	0	0
0	1	X	X	Dn – D0	↑	d0	dn
0	0	1	SLI	X	↑	SLI	qn-1
0	0	0	X	X	X	No Chg	No Chg

z = 3 for SR4CLE; z = 7 for SR8CLE; z = 15 for SR16CLE
dn = state of referenced input one setup time prior to active clock transition
qn-1 = state of referenced output one setup time prior to active clock transition

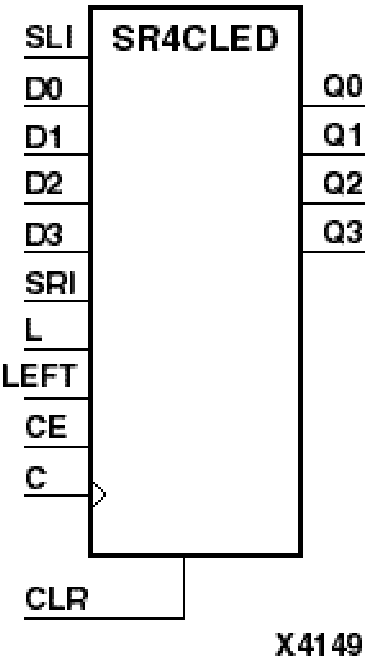
Figure 10-5SR8CLE Implementation XC3000, XC4000, XC5200, XC9000, Spartans, Virtex

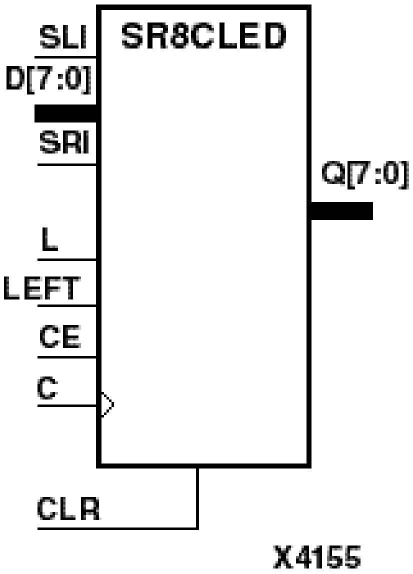


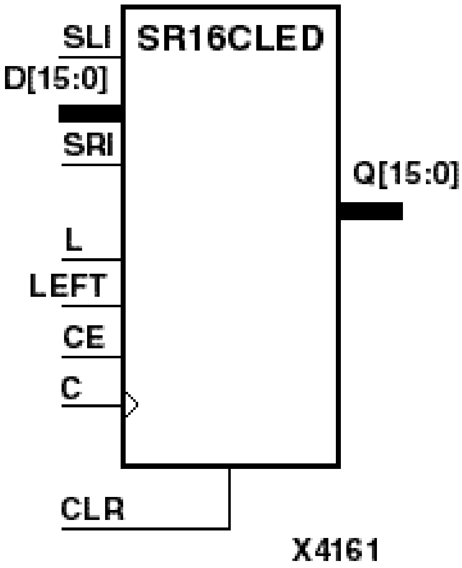
SR4CLED, SR8CLED, SR16CLED

4-, 8-, 16-Bit Shift Registers with Clock Enable and Asynchronous Clear

XC3000	XC4000 E	XC4000 X	XC5200	XC9000	Spartan	Spartan XL	Virtex
Macro	Macro	Macro	Macro	Macro	Macro	Macro	Macro


SR4CLED
SLI
D0
D1
D2
D3
SRI
L
LEFT
CE
C
CLR
Q0
Q1
Q2
Q3
X4149


SR8CLED
SLI
D[7:0]
SRI
L
LEFT
CE
C
CLR
Q[7:0]
X4155



SR4CLED, SR8CLED, and SR16CLED are 4-, 8-, and 16-bit shift registers, respectively, with shift-left (SLI) and shift-right (SRI) serial inputs, parallel inputs (D), 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 (Qn) Low. When L is High and CLR is Low, the data on the D inputs is loaded into the corresponding Q 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 the last Q output (Q3 for SR4CLED, Q7 for SR8CLED, or Q15 for SR16CLED) during the Low-to-High clock transition and shifted right (to Q2, Q1,... for SR4CLED; to Q6, Q5,... for SR8CLED; and to Q14, Q13,... for SR16CLED) during subsequent clock transitions. The truth tables for SR4CLED, SR8CLED, and SR16CLED indicate the state of the Q outputs under all input conditions for SR4CLED, SR8CLED, and SR16CLED.

The register is asynchronously cleared, outputs Low, when power is applied. For CPLDs, the power-on condition can be simulated by applying a High-level pulse on the PRLD global net. FPGAs simulate power-on when global reset (GR) or global set/reset (GSR) is active. GR for XC3000 is active-Low. GR for XC5200 and GSR (XC4000, Spartans, Virtex) default to active-High but can be inverted by adding an inverter in front of the GR/GSR input of the STARTUP or STARTUP_VIRTEX symbol.

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 Chg	No Chg	No Chg
0	0	1	1	SLI	X	X	↑	SLI	q2	qn-1
0	0	1	0	X	SRI	X	↑	q1	SRI	qn+1

dn = state of referenced input one setup time prior to active clock transition
qn-1 and qn+1 = state of referenced output one setup time prior to active clock transition

SR4CLED Truth Table

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 Chg	No Chg	No Chg
0	0	1	1	SLI	X	X	↑	SLI	q6	qn-1
0	0	1	0	X	SRI	X	↑	q1	SRI	qn+1

dn = state of referenced input one setup time prior to active clock transition
qn-1 or qn+1 = state of referenced output one setup time prior to active clock transition

SR8CLED Truth Table

Inputs								**Outputs**		
C	**L**	**C**	**LEF**	**S**	**S**	**D15 –**	**C**	**Q0**	**Q15**	**Q14 –**

L R		E	T	LI	RI	D0				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 Chg	No Chg	No Chg
0	0	1	1	SLI	X	X	↑	SLI	q14	qn-1
0	0	1	0	X	SRI	X	↑	q1	SRI	qn+1

dn = state of referenced input one setup time prior to active clock transition
qn-1 or qn+1 = state of referenced output one setup time prior to active clock transition

SR16CLED Truth Table

Figure 10-6SR8CLED Implementation XC3000, XC4000, XC5200, Spartans, Virtex

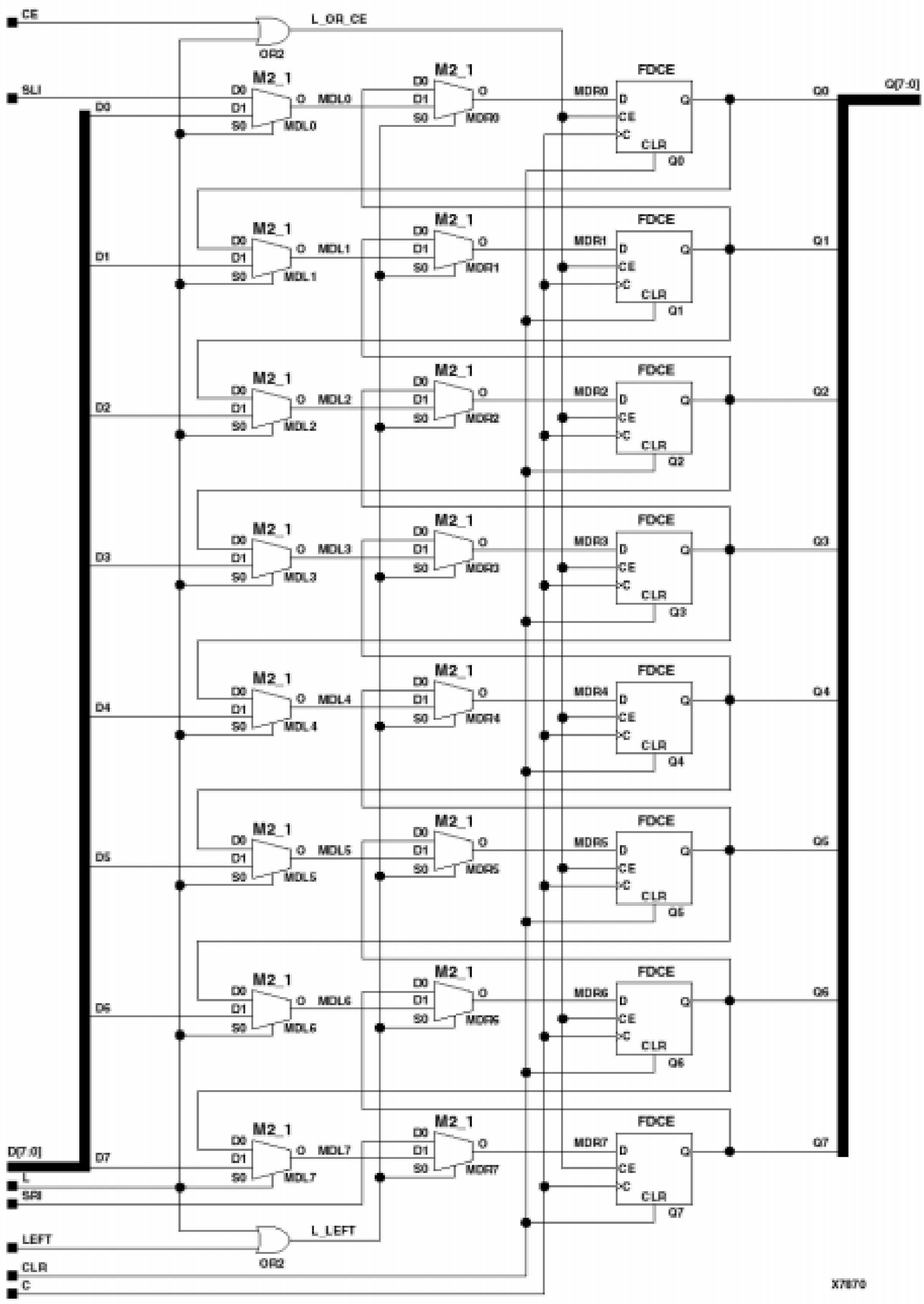
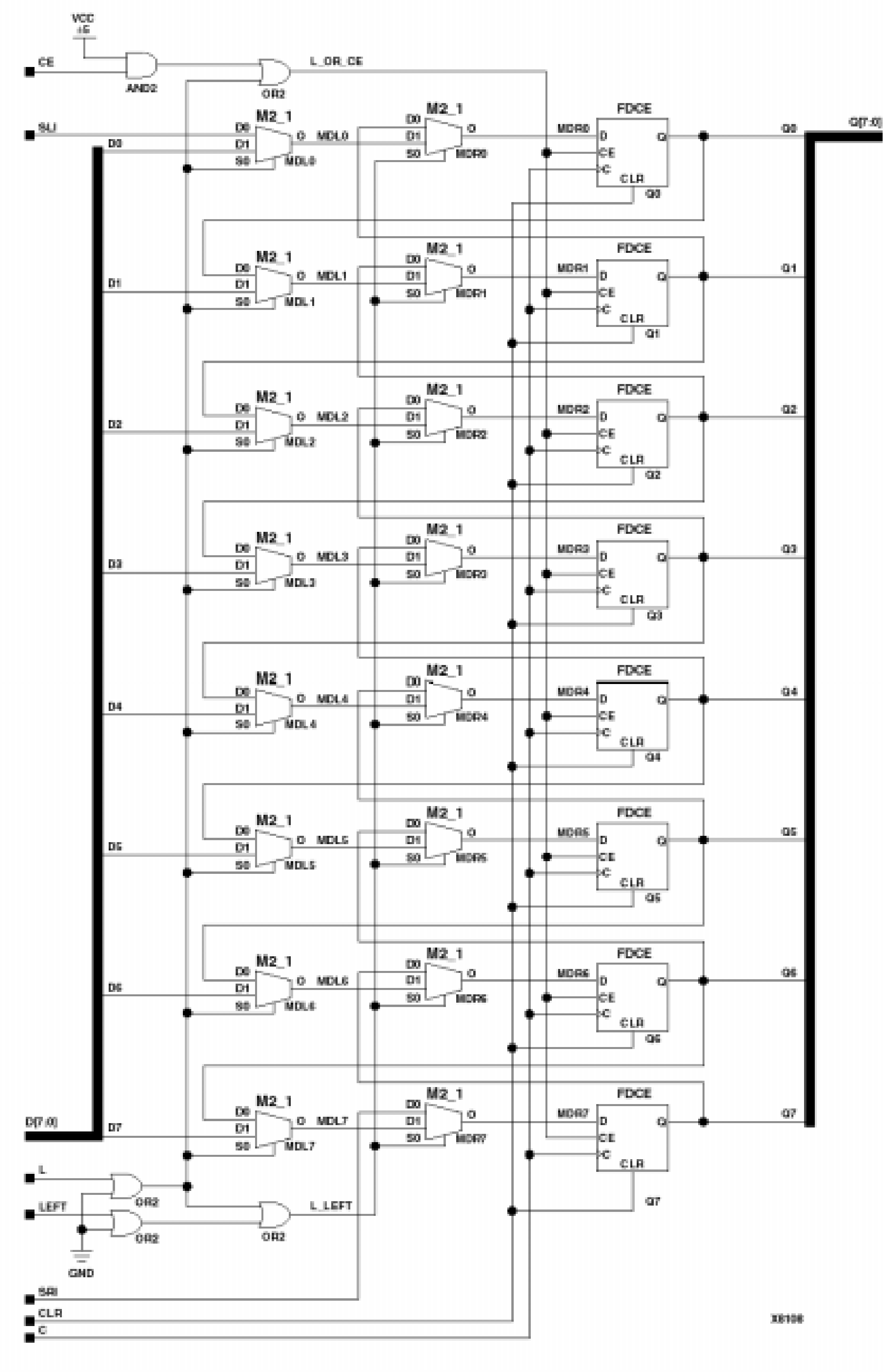

CE
L_OR_CE
OR2
SLI
D0
M2_1
D0
D1
S0
O
MDL0
MDL0
M2_1
D0
D1
S0
O
MDR0
MDR0
FDCE
D
CE
C
CLR
Q
Q0
Q0
Q[7:0]
D1
MDL1
MDR1
Q1
D2
MDL2
MDR2
Q2
D3
MDL3
MDR3
Q3
D4
MDL4
MDR4
Q4
D5
MDL5
MDR5
Q5
D6
MDL6
MDR6
Q6
D7
MDL7
MDR7
Q7
D[7:0]
L
SRI
LEFT
L_LEFT
OR2
CLR
C
X7870

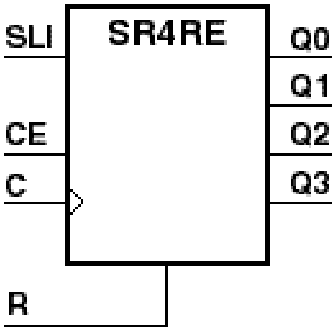
Figure 10-7SR8CLED Implementation XC9000


VCC
+5
CE
AND2
OR2
L_OR_CE
SLI
D0
D1
S0
O
M2_1
MDL0
MDR0
FDCE
D
CE
C
CLR
Q
Q0
Q[7:0]
D1
MDL1
MDR1
Q1
D2
MDL2
MDR2
Q2
D3
MDL3
MDR3
Q3
D4
MDL4
MDR4
Q4
D5
MDL5
MDR5
Q5
D6
MDL6
MDR6
Q6
D7
MDL7
MDR7
Q7
D[7:0]
L
LEFT
GND
L_LEFT
SRI
CLR
C
X8108

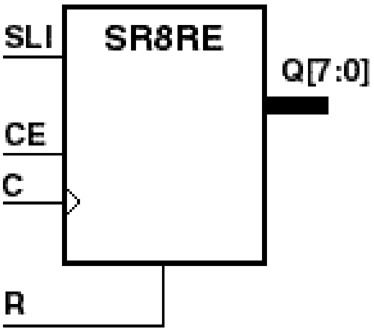
SR4RE, SR8RE, SR16RE

4-, 8-, 16-Bit Serial-In Parallel-Out Shift Registers with Clock Enable and Synchronous Reset

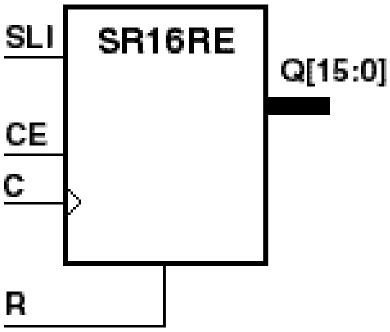
XC3000	XC4000 E	XC4000 X	XC5200	XC9000	Spartan	Spartan XL	Virtex
Macro	Macro	Macro	Macro	Macro	Macro	Macro	Macro



X4144



X4150



X4156

SR4RE, SR8RE, and SR16RE are 4-, 8-, and 16-bit shift registers, respectively, with shift-left serial input (SLI), parallel outputs (Qn), clock enable (CE), and synchronous reset (R) inputs. The R input, when High, overrides all other inputs during the Low-to-High clock (C) transition and resets the data outputs (Q) 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.

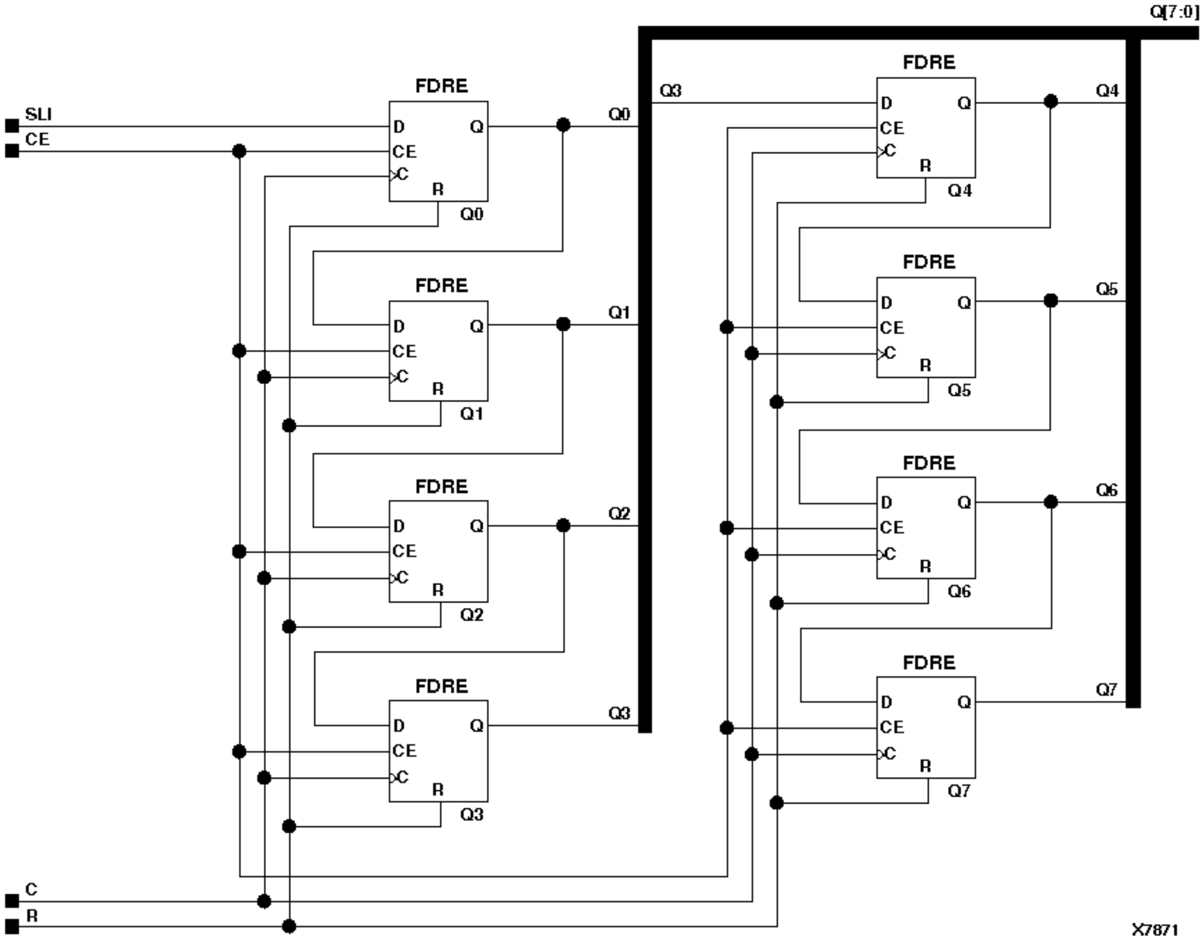
Registers can be cascaded by connecting the last Q output (Q3 for SR4RE, Q7 for SR8RE, or Q15 for SR16RE) of one stage to the SLI input of the next stage and connecting clock, CE, and R in parallel.

The register is asynchronously cleared, outputs Low, when power is applied. For CPLDs, the power-on condition can be simulated by applying a High-level pulse on the PRLD global net. FPGAs simulate power-on when global reset (GR) or global set/reset (GSR) is active. GR for XC3000 is active-Low. GR for XC5200 and GSR (XC4000, Spartans, Virtex) default to active-High but can be inverted by adding an inverter in front of the GR/GSR input of the STARTUP or STARTUP_VIRTEX symbol.

Inputs				**Outputs**	
R	**CE**	**SLI**	**C**	**Q0**	**Qz – Q1**
1	X	X	↑	0	0
0	0	X	X	No Chg	No Chg
0	1	1	↑	1	qn-1
0	1	0	↑	0	qn-1

z = 3 for SR4RE; z = 7 for SR8RE; z = 15 for SR16RE
qn-1 = state of referenced output one setup time prior to active clock transition

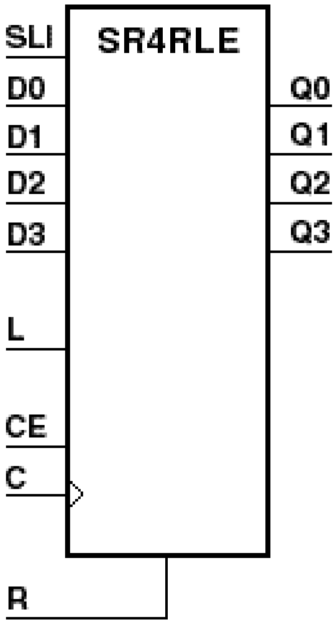
Figure 10-8SR8RE Implementation XC3000, XC4000, XC5200, XC9000, Spartans, Virtex



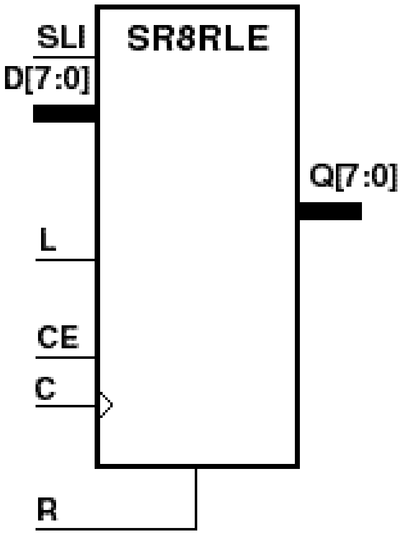
SR4RLE, SR8RLE, SR16RLE

4-, 8-, 16-Bit Loadable Serial/Parallel-In Parallel-Out Shift Registers with Clock Enable and Synchronous Reset

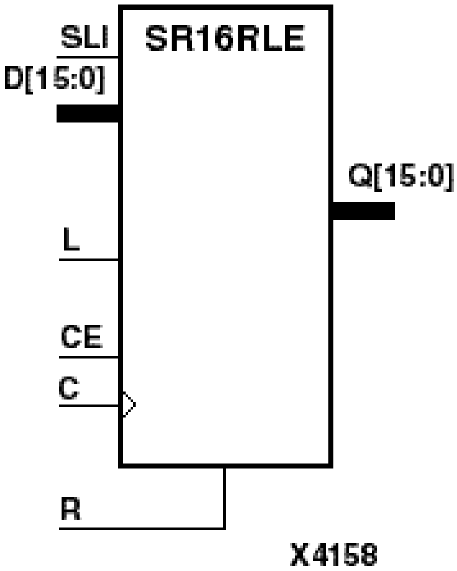
XC3000	XC4000 E	XC4000 X	XC5200	XC9000	Spartan	Spartan XL	Virtex
Macro	Macro	Macro	Macro	Macro	Macro	Macro	Macro


SLI
SR4RLE
D0
D1
D2
D3
L
CE
C
R
Q0
Q1
Q2
Q3

X4146


SLI
SR8RLE
D[7:0]
Q[7:0]
L
CE
C
R

X4152



SR4RLE, SR8RLE, and SR16RLE are 4-, 8-, and 16-bit shift registers, respectively, with shift-left serial input (SLI), parallel inputs (D), parallel outputs (Q), 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 during the Low-to-High clock (C) transition and resets the data outputs (Q) Low. When L is High and R is Low, data on the D inputs is loaded into the corresponding Q 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 last Q output (Q3 for SR4RLE, Q7 for SR8RLE, or 15 for SR16RLE) of one stage to the SLI input of the next stage and connecting clock, CE, L, and R inputs in
parallel.

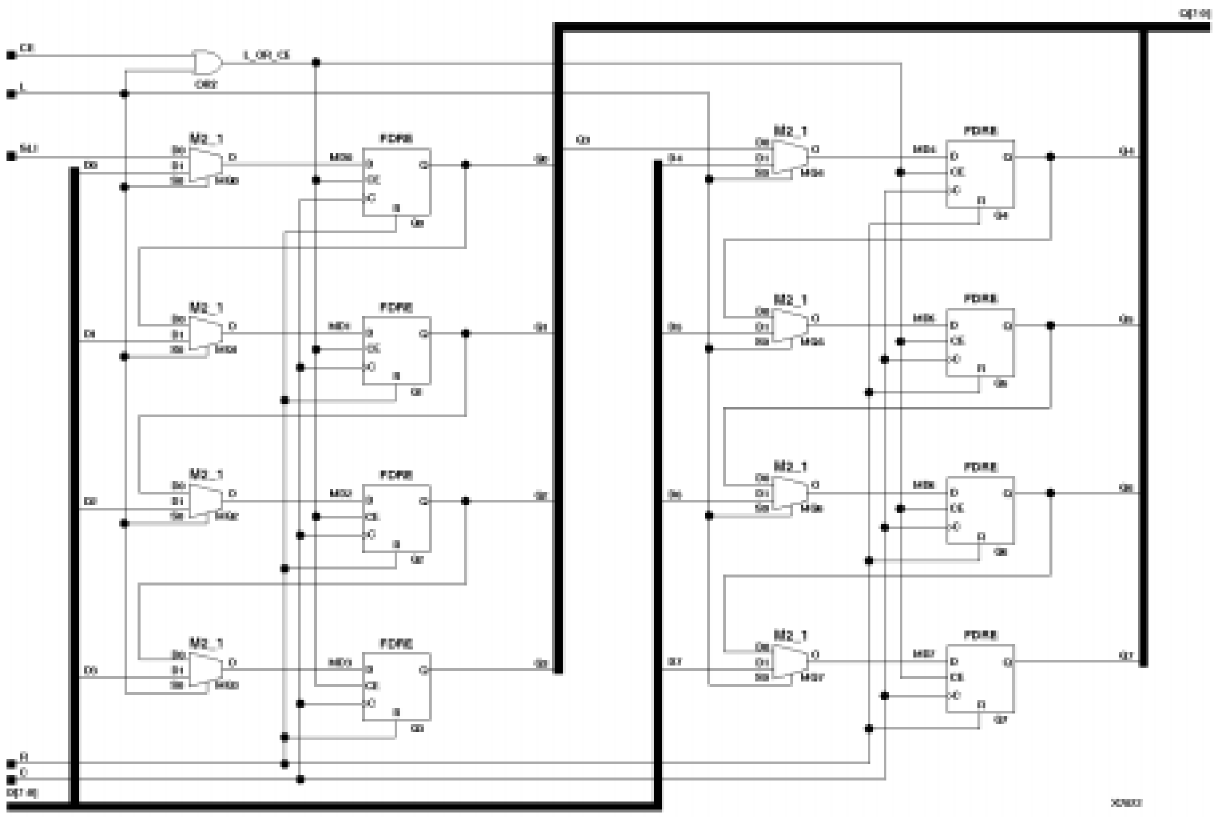
The register is asynchronously cleared, outputs Low, when power is applied. For CPLDs, the power-on condition can be simulated by applying a High-level pulse on the PRLD global net. FPGAs simulate power-on when global reset (GR) or global set/reset (GSR) is active. GR for XC3000 is active-Low. GR for XC5200 and GSR (XC4000, Spartans, Virtex) default to active-High but can be inverted by adding an inverter in front of the GR/GSR input of the STARTUP or STARTUP_VIRTEX symbol.

Inputs						**Outputs**	
R	**L**	**CE**	**SLI**	**Dz – D0**	**C**	**Q0**	**Qz – Q1**
1	X	X	X	X	↑	0	0
0	1	X	X	Dz – D0	↑	d0	dn
0	0	1	SLI	X	↑	SLI	qn-1

0	0	0	X	X	X	No Chg	No Chg

z = 3 for SR4RLE; z = 7 for SR8RLE; z = 15 for SR16RLE
dn = state of referenced input one setup time prior to active clock transition
qn-1 = state of referenced output one setup time prior to active clock transition

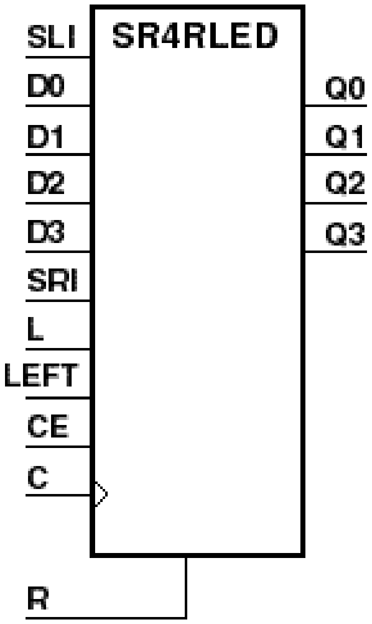
Figure 10-9SR8RLE Implementation XC3000, XC4000, XC5200, XC9000, Spartans, Virtex



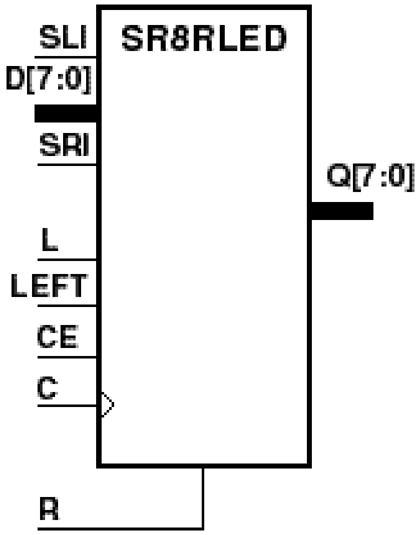
SR4RLED, SR8RLED, SR16RLED

4-, 8-, 16-Bit Shift Registers with Clock Enable and Synchronous Reset

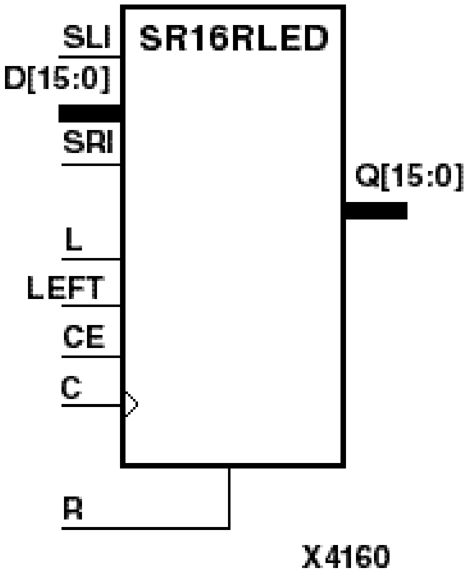
XC3000	XC4000 E	XC4000 X	XC5200	XC9000	Spartan	Spartan XL	Virtex
Macro	Macro	Macro	Macro	Macro	Macro	Macro	Macro


SLI
SR4RLED
D0
Q0
D1
Q1
D2
Q2
D3
Q3
SRI
L
LEFT
CE
C
R

X4148


SLI
SR8RLED
D[7:0]
SRI
Q[7:0]
L
LEFT
CE
C
R

X4154



SR4RLED, SR8RLED, and SR16RLED are 4-, 8-, and 16-bit shift registers, respectively, with shift-left (SLI) and shift-right (SRI) serial inputs, parallel inputs (D), 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 during the Low-to-High clock (C) transition and resets the data outputs (Q) Low. When L is High and R is Low, the data on the D inputs is loaded into the corresponding Q 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 the last Q output (Q3 for SR4RLED, Q7 for SR8RLED, or Q15 for SR16RLED) during the Low-to-High clock transition and shifted right (to Q2, Q1,... for SR4RLED; to Q6, Q5,... for SR8RLED; or to Q14, Q13,... for SR16RLED) during subsequent clock transitions. The truth table indicates the state of the Q outputs under all input conditions.

The register is asynchronously cleared, outputs Low, when power is applied. For CPLDs, the power-on condition can be simulated by applying a High-level pulse on the PRLD global net. FPGAs simulate power-on when global reset (GR) or global set/reset (GSR) is active. GR for XC3000 is active-Low. GR for XC5200 and GSR (XC4000, Spartans, Virtex) default to active-High but can be inverted by adding an inverter in front of the GR/GSR input of the STARTUP or STARTUP_VIRTEX symbol.

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 Chg	No Chg	No Chg

0	0	1	1	SLI	X	X	↑	SLI	q2	qn-1
0	0	1	0	X	SRI	X	↑	q1	SRI	qn+1

dn = state of referenced input one setup time prior to active clock transition

qn-1 or qn+1 = state of referenced output one setup time prior to active clock transition

SR4RLED Truth Table

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 Chg	No Chg	No Chg
0	0	1	1	SLI	X	X	↑	SLI	q6	qn-1
0	0	1	0	X	SRI	X	↑	q1	SRI	qn+1

dn = state of referenced input one setup time prior to active clock transition

qn-1 or qn+1 = state of referenced output one setup time prior to active clock transition

SR8RLED Truth Table

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

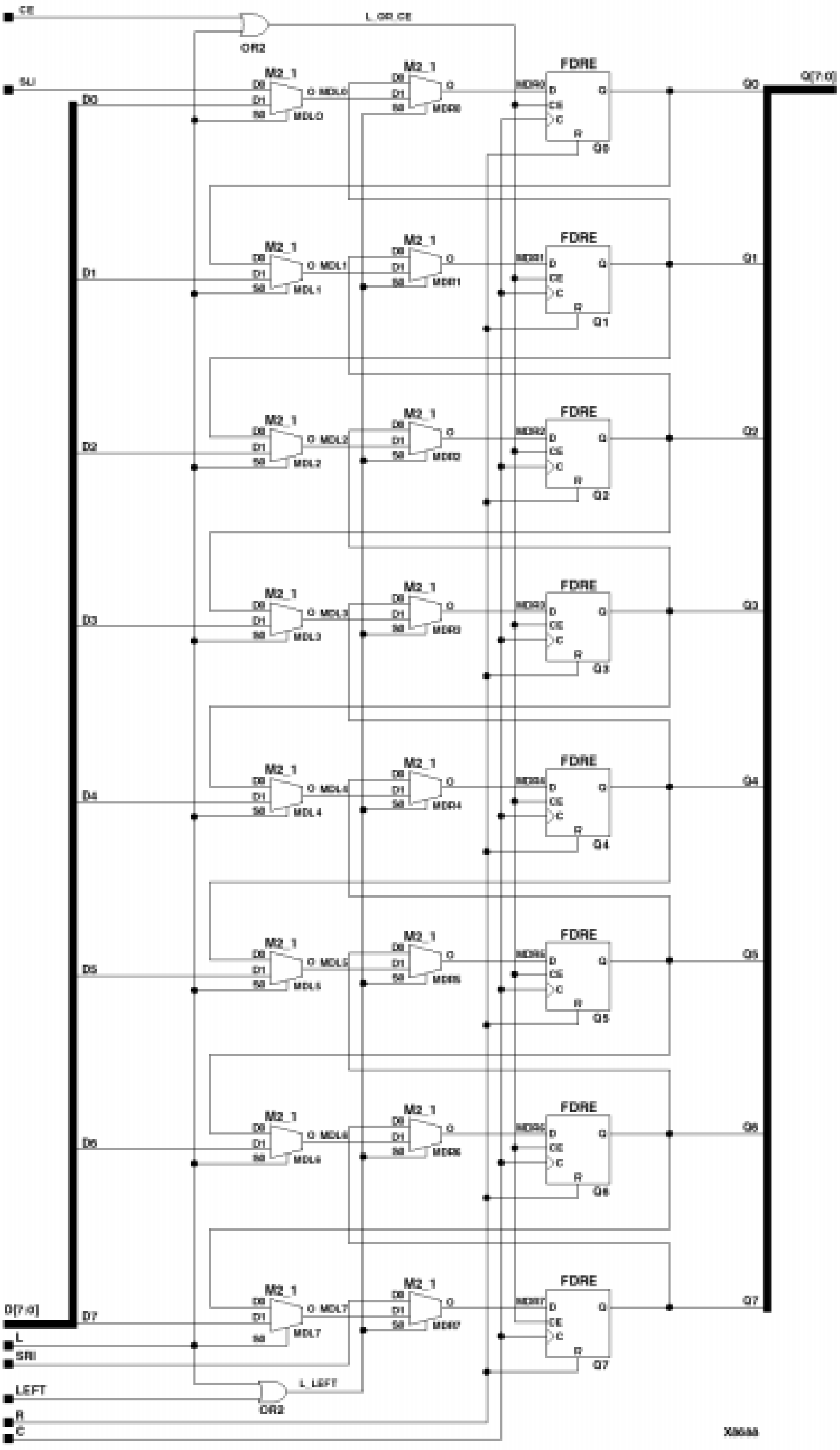
0	1	X	X	X	X	D15 – D0	↑	d0	d15	dn
0	0	0	X	X	X	X	X	No Chg	No Chg	No Chg
0	0	1	1	SLI	X	X	↑	SLI	q14	qn-1
0	0	1	0	X	SRI	X	↑	q1	SRI	qn+1

dn = state of referenced input one setup time prior to active clock transition

qn-1 or qn+1 = state of referenced output one setup time prior to active clock transition

SR16RLED Truth Table

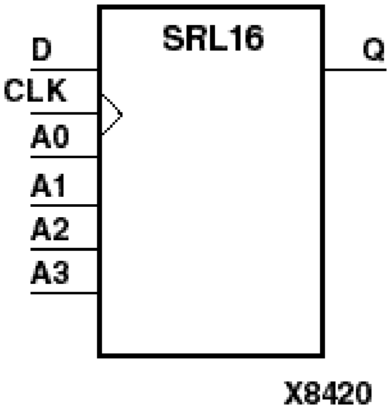
Figure 10-10SR8RLED Implementation XC3000, XC4000, XC5200, XC9000, Spartans, Virtex


CE
OR2
L_OR_CE
SLI
D0
D1
D2
D3
D4
D5
D6
D7
M2_1
D0
D1
S0
O
MDL0
MDL1
MDL2
MDL3
MDL4
MDL5
MDL6
MDL7
MDR0
MDR1
MDR2
MDR3
MDR4
MDR5
MDR6
MDR7
FDRE
D
CE
C
R
Q
Q0
Q1
Q2
Q3
Q4
Q5
Q6
Q7
Q[7:0]
D[7:0]
L
SRI
LEFT
L_LEFT
R
C

SRL16

16-Bit Shift Register Look-Up-Table (LUT)

XC3000	XC4000 E	XC4000 X	XC5200	XC9000	Spartan	Spartan XL	Virtex
N/A	N/A	N/A	N/A	N/A	N/A	N/A	Primitive



SRL16 is a shift register look up table (LUT). The inputs A3, A2, A1, and A0 select the output length of the shift register. The shift register may be of a fixed, static length or dynamically adjusted.

The shift register LUT contents are initialized by assigning a four-digit hexadecimal number to an INIT attribute. The first, or the left-most, hexadecimal digit is the most significant bit. If an INIT value is not specified, it defaults to a value of four zeros (0000) so that the shift register LUT is cleared during configuration.

The data (D) is loaded into the first bit of the shift register during the Low-to-High clock (CLK) transition and appears on the Q output. During subsequent Low-to-High clock transitions data is shifted to the next highest bit position as new data is loaded into Q.

Static Length Mode

To get a fixed length shift register, drive the A3 through A0 inputs with static values. The length of the shift register can vary from 1 bit to 16 bits as determined from the following formula:

Length = (8*A3) +(4*A2) + (2*A1) + A0 +1

If A3, A2, A1, and A0 are all zeros (0000), the shift register is one bit long. If they are all ones (1111), it is 16 bits long.

Dynamic Length Mode

The length of the shift register can be changed dynamically by changing the values driving the A3 through A0 inputs. For example, if A2, A1, and A0 are all ones (111) and A3 toggles between a one (1) and a zero (0), the length of the shift register changes from 16 bits to 8 bits.

Internally, the length of the shift register is always 16 bits and the input lines A3 through A0 select which of the 16 bits reach the output.

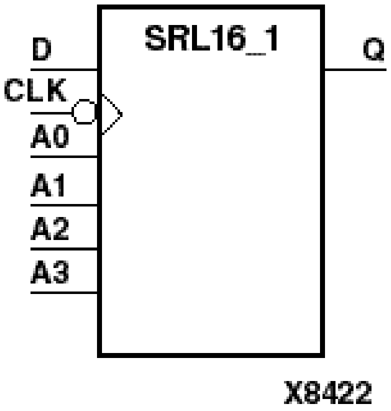
Inputs				**Output**
CLK	**D**	**<SR(1)>**	**<SR(i)>**	**Q**
1	X	No Chg	No Chg	No Chg
0	X	No Chg	No Chg	No Chg
↑	D	D	SR(i-1)	SR(L)

SR(1) = contents of first shift register
SR(i) = contents of the i'th shift register stage (2<= n <= L)
L = shift register length (1 through 16 determined by (8*A3) +(4*A2) + (2*A1) + A0 +1)

SRL16_1

16-Bit Shift Register Look-Up-Table (LUT) with Negative-Clock Edge

XC3000	XC4000 E	XC4000 X	XC5200	XC9000	Spartan	Spartan XL	Virtex
N/A	N/A	N/A	N/A	N/A	N/A	N/A	Primitive



SRL16_1 is a shift register look up table (LUT). The inputs A3, A2, A1, and A0 select the output length of the shift register. The shift register may be of a fixed, static length or dynamically adjusted. Refer to **"Static Length Mode"** and

"Dynamic Length Mode" in the SRL16 section.

The shift register LUT contents are initialized by assigning a four-digit hexadecimal number to an INIT attribute. The first, or the left-most, hexadecimal digit is the most significant bit. If an INIT value is not specified, it defaults to a value of four zeros (0000) so that the shift register LUT is cleared during configuration.

The data (D) is loaded into the first bit of the shift register during the High-to-Low clock (CLK) transition and appears on the Q output. During subsequent High-to-Low clock transitions data is shifted to the next highest bit position as new data is loaded into Q.

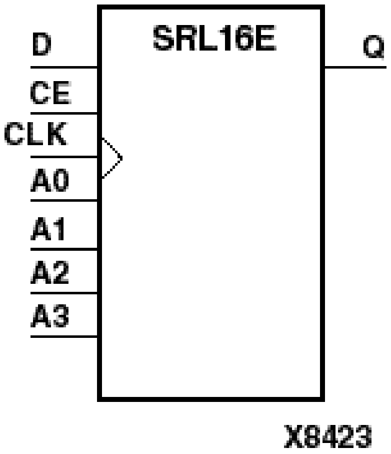
Inputs				**Output**
CLK	**D**	**<SR(1)>**	**<SR(i)>**	**Q**
1	X	No Chg	No Chg	No Chg
0	X	No Chg	No Chg	No Chg
↓	D	D	SR(i-1)	SR(L)

SR(1) = contents of first shift register
SR(i) = contents of the i'th shift register stage (2<= n <= L)
L = shift register length (1 through 16 determined by (8*A3) +(4*A2) + (2*A1) + A0 +1)

SRL16E

16-Bit Shift Register Look-Up-Table (LUT) with Clock Enable

XC3000	**XC4000 E**	**XC4000 X**	**XC5200**	**XC9000**	**Spartan**	**Spartan XL**	**Virtex**
N/A	N/A	N/A	N/A	N/A	N/A	N/A	Primitive



SRL16E is a shift register look up table (LUT). The inputs A3, A2, A1, and A0 select the output length of the shift register. The shift register may be of a fixed, static length or dynamically adjusted. Refer to **"Static Length Mode"** and **"Dynamic Length Mode"** in the SRL16 section.

The shift register LUT contents are initialized by assigning a four-digit hexadecimal number to an INIT attribute. The first, or the left-most, hexadecimal digit is the most significant bit. If an INIT value is not specified, it defaults to a value of four zeros (0000) so that the shift register LUT is cleared during configuration.

When CE is High, the data (D) is loaded into the first bit of the shift register during the Low-to-High clock (CLK) transition and appears on the Q output. During subsequent Low-to-High clock transitions, when CE is High, data is shifted to the next highest bit position as new data is loaded into Q. When CE is Low, the register ignores clock transitions.

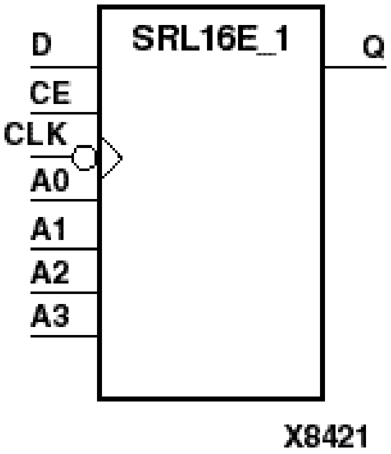
Inputs					**Output**
CE	**CLK**	**D**	**<SR(1)>**	**<SR(i)>**	**Q**
0	X	X	No Chg	No Chg	No Chg
1	1	X	No Chg	No Chg	No Chg
1	0	X	No Chg	No Chg	No Chg
1	↑	D	D	SR(i-1)	SR(L)

SR(1) = contents of first shift register
SR(i) = contents of the i'th shift register stage (2<= n <= L)
L = shift register length (1 through 16 determined by (8*A3) +(4*A2) + (2*A1) + A0 +1)

SRL16E_1

16-Bit Shift Register Look-Up-Table (LUT) with Negative-Edge Clock and Clock Enable

XC3000	XC4000E	XC4000X	XC5200	XC9000	Spartan	SpartanXL	Virtex
N/A	N/A	N/A	N/A	N/A	N/A	N/A	Primitive



SRL16E_1 is a shift register look up table (LUT) with clock enable (CE). The inputs A3, A2, A1, and A0 select the output length of the shift register. The shift register may be of a fixed, static length or dynamically adjusted. Refer to **"Static Length Mode"** and **"Dynamic Length Mode"** in the SRL16 section.

The shift register LUT contents are initialized by assigning a four-digit hexadecimal number to an INIT attribute. The first, or the left-most, hexadecimal digit is the most significant bit. If an INIT value is not specified, it defaults to a value of four zeros (0000) so that the shift register LUT is cleared during configuration.

When CE is High, the data (D) is loaded into the first bit of the shift register during the High-to-Low clock (CLK) transition and appears on the Q output. During subsequent High-to-Low clock transitions, when CE is High, data is shifted to the next highest bit position as new data is loaded into Q. When CE is Low, the register ignores clock transitions.

Inputs					Output
CE	CLK	D	<SR(1)>	<SR(i)>	Q
0	X	X	No Chg	No Chg	No Chg
1	1	X	No Chg	No Chg	No Chg

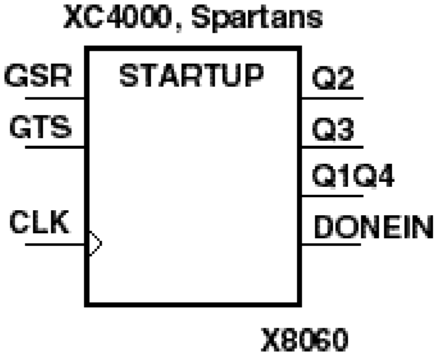
1	0	X	No Chg	No Chg	No Chg
1	↓	D	D	SR(i-1)	SR(L)

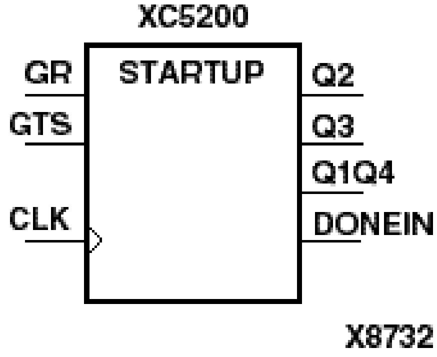
SR(1) = contents of first shift register
SR(i) = contents of the i'th shift register stage (2<= n <= L)
L = shift register length (1 through 16 determined by (8*A3) +(4*A2) + (2*A1) + A0 +1)

STARTUP

User Interface to Global Clock, Reset, and 3-State Controls

XC3000	XC4000 E	XC4000 X	XC5200	XC9000	Spartan	Spartan XL	Virtex
N/A	Primitive	Primitive	Primitive	N/A	Primitive	Primitive	N/A

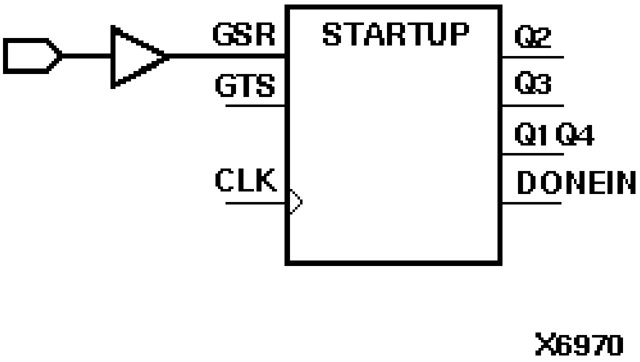




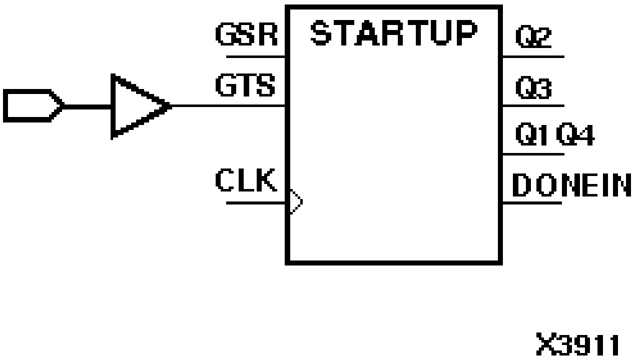
The STARTUP primitive is used for Global Set/Reset, global 3-state control, and the user configuration clock. The Global Set/Reset (GSR) input, when High, sets or resets every flip-flop in the device, depending on the initialization state (S or R) of the flip-flop. Following configuration, the global 3-state control (GTS), when High, forces all the IOB outputs into high impedance mode, which isolates the device outputs from the circuit but leaves the inputs active.

Including the STARTUP symbol in a design is optional. You must include the symbol under the following conditions.

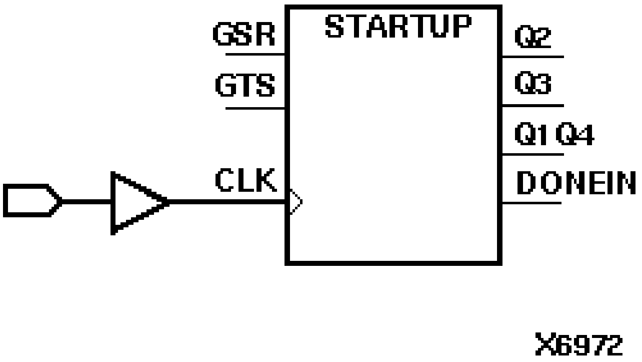
- If you intend to exert external control over global set/reset, you must connect the GSR pin to an IPAD and an IBUF, as shown here. (For the XC5200, connect the GR pin to an IPAD and an IBUF.)



- If you intend to exert external control over global tristate, you must connect the GTS pin to an IPAD and IBUF, as shown here.



- If you wish to synchronize startup to a user clock, you must connect the user clock signal to the CLK input, as shown here. Furthermore, "user clock" must be selected in the BitGen program.



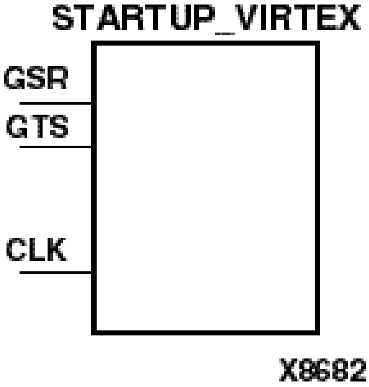
You can use location constraints to specify the pin from which GSR or GTS (or both) is accessed.

The STARTUP outputs (Q2, Q3, Q1Q4, and DONEIN) display the progress/status of the start-up process following the configuration. Refer to *The Programmable Logic Data Book* for additional details.

STARTUP_VIRTEX

Virtex User Interface to Global Clock, Reset, and 3-State Controls

XC3000	XC4000 E	XC4000 X	XC5200	XC9000	Spartan	Spartan XL	Virtex
N/A	N/A	N/A	N/A	N/A	N/A	N/A	Primitive



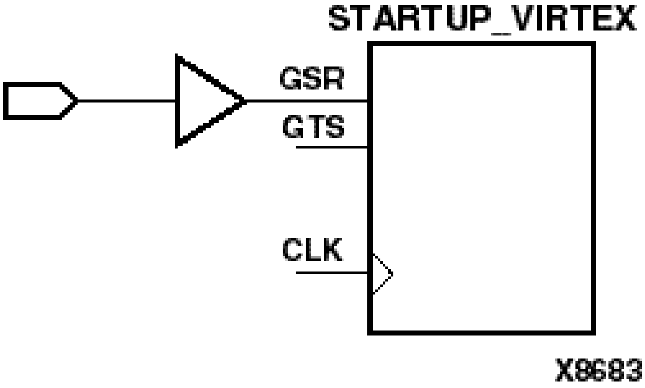
The STARTUP_VIRTEX primitive is used for Global Set/Reset, global 3-state control, and the user configuration clock. The Global Set/Reset (GSR) input, when High, sets or resets all flip-flops, all latches, and every block RAM (RAMB4) output register in the device, depending on the initialization state (S or R) of the component.

Note: Block RAMB4 content, LUT RAMs, delay locked loop elements (CLKDLL, CLKDLLHF, BUFGDLL), and shift register LUTs (SRL16, SRL16_1, SRL16E, SRL16E_1) are not set/reset.

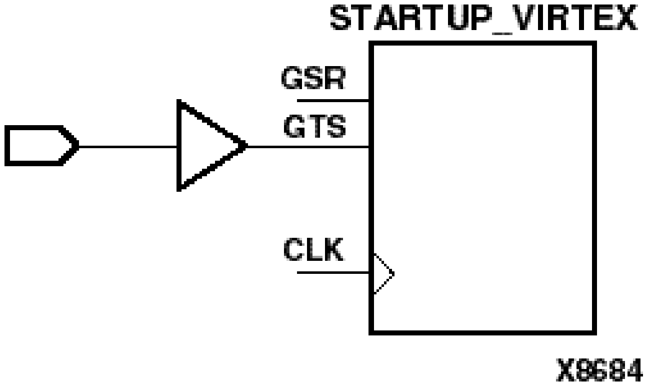
Following configuration, the global 3-state control (GTS), when High—and BSCAN is not enabled and executing an EXTEST instruction—forces all the IOB outputs into high impedance mode, which isolates the device outputs from the circuit but leaves the inputs active.

Including the STARTUP symbol in a design is optional. You must include the symbol under the following conditions.

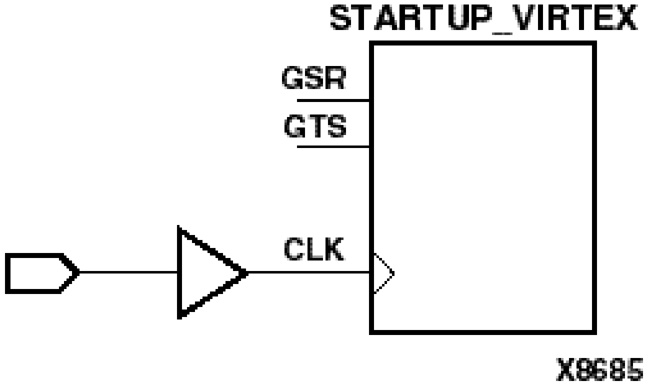
- If you intend to exert external control over global set/reset, you must connect the GSR pin to a top level port and an IBUF, as shown here.



- If you intend to exert external control over global tristate, you must connect the GTS pin to a top level port and IBUF, as shown here.



- If you wish to synchronize startup to a user clock, you must connect the user clock signal to the CLK input, as shown here. Furthermore, "user clock" must be selected in the BitGen program.

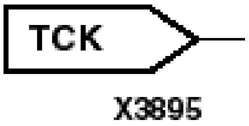


You can use location constraints to specify the pin from which GSR or GTS (or both) is accessed.

TCK

Boundary Scan Test Clock Input Pad

XC3000	XC4000 E	XC4000 X	XC5200	XC9000	Spartan	Spartan XL	Virtex
N/A	Primitive	Primitive	Primitive	N/A	Primitive	Primitive	N/A



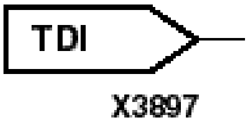
The TCK input pad is connected to the boundary scan test clock, which shifts the serial data and instructions into and out of the boundary scan data registers. The function of the TCK pad is device configuration dependent and can be used as follows.

- During configuration TCK is connected to the boundary scan logic.
- After configuration, if boundary scan is not used, the TCK pad is unrestricted and can be used by the routing tool as an input/output pad.
- After configuration, if boundary scan is used, the TCK pad can be used for user-logic input by connecting it directly to the user logic.

TDI

Boundary Scan Test Data Input Pad

XC3000	XC4000 E	XC4000 X	XC5200	XC9000	Spartan	Spartan XL	Virtex
N/A	Primitive	Primitive	Primitive	N/A	Primitive	Primitive	N/A



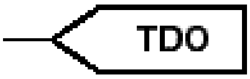
The TDI input pad is connected to the boundary scan TDI input. It loads instructions and data on the Low-to-High TCK transition. The function of the TDI pad is device configuration dependent and can be used as follows.

- During configuration, TDI is connected to the boundary scan logic.
- After configuration, if boundary scan is not used, the TDI pad is unrestricted and can be used by the routing tools as an input/output pad.
- After configuration, if boundary scan is used, the TDI pad can be used for user-logic input by connecting the TDI pad directly to the user logic.

TDO

Boundary Scan Data Output Pad

XC3000	XC4000 E	XC4000 X	XC5200	XC9000	Spartan	Spartan XL	Virtex
N/A	Primitive	Primitive	Primitive	N/A	Primitive	Primitive	N/A



X3899

The TDO data output pad is connected to the boundary scan TDO output. It is connected to the external circuit to provide the boundary scan data for each Low-to-High TCK transition. The function of the TDO pad is device configuration dependent and can be used as follows.

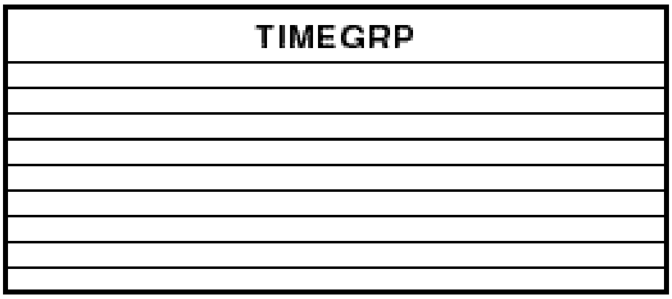
- During configuration, TDO is connected to the boundary scan logic.
- After configuration, if boundary scan is not used, the TDO pad can be used as a bidirectional 3-state I/O pad by the routing tool.
- After configuration, if boundary scan is used, the TDO pad is still used as an output from the boundary scan logic.

TIMEGRP

Schematic-Level Table of Basic Timing Specification Groups

XC3000	XC4000 E	XC4000 X	XC5200	XC9000	Spartan	Spartan XL	Virtex
Primitive	Primitive	Primitive	Primitive	Primitive	Primitive	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.



X4699

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 sample statement defines groups in a TIMEGRP symbol.

```
TIMEGRP=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.

Note: When entering timegroup properties into a TIMEGRP symbol, some property names should not be used because they cause a conflict with the predefined (reserved) property names of the TIMEGRP primitive.

The standard procedure for adding a property to a symbol is to use the following command.

```
PROPERTY = property_name VALUE=value
```

For *property_name* you must not use any of the system reserved names LIBVER, INST, COMP, MODEL, or any other names reserved by your schematic capture program. Please consult your schematic capture documentation to familiarize yourself with reserved property names.

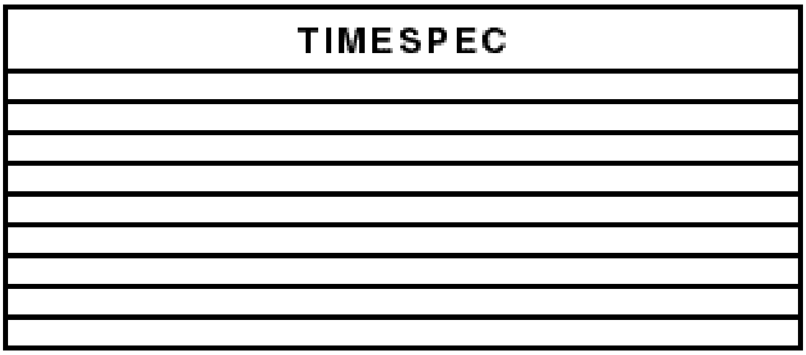
For more on time group attributes, see the **"Time Group Attributes" section of the "Attributes, Constraints, and Carry Logic" chapter**.

TIMESPEC

Schematic-Level Timing Requirement Table

XC3000	XC4000E	XC4000X	XC5200	XC9000	Spartan	Spartan XL	Virtex
Primitive	Primitive	Primitive	Primitive	Primitive	Primitive	Primitive	Primitive

The TIMESPEC primitive is a table that you can use to specify 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 following figure.



X3866

For more information on "TS" timing attributes refer to the **"TSidentifier" section of the "Attributes, Constraints, and Carry Logic" chapter**.

TMS

Boundary Scan Test Mode Select Input Pad

XC3000	XC4000 E	XC4000 X	XC5200	XC9000	Spartan	Spartan XL	Virtex
N/A	Primitive	Primitive	Primitive	N/A	Primitive	Primitive	N/A

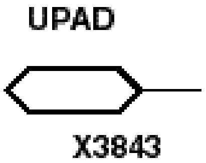
The TMS input pad is connected to the boundary scan TMS input. It determines which boundary scan operation is performed. The function of the TMS pad is device configuration dependent and can be used as follows.

- During configuration, TMS is connected to the boundary scan logic.
- After configuration, if boundary scan is not used, the TMS pad is unrestricted and can be used by the routing tools as an input/output pad.
- After configuration, if boundary scan is used, the TMS pad can be used for user-logic input by connecting the TMS pad directly to the user logic.

UPAD

Connects the I/O Node of an IOB to the Internal PLD Circuit

XC3000	XC4000 E	XC4000 X	XC5200	XC9000	Spartan	Spartan XL	Virtex
Primitive	Primitive	Primitive	Primitive	N/A	Primitive	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

XC3000	XC4000	XC4000	XC5200	XC9000	Spartan	Spartan	Virtex

	E	X				XL	
Primitive	Primitive	Primitive	Primitive	Primitive	Primitive	Primitive	Primitive

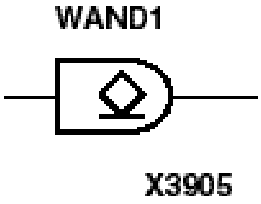
The VCC signal tag or parameter forces a net or input function to a logic High level. A net tied to VCC cannot have any other source.

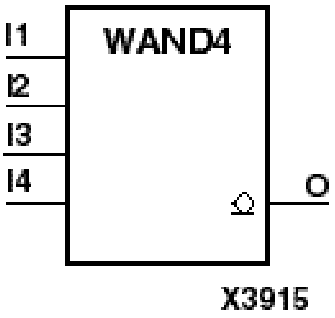
When the placement and routing software encounters a net or input function tied to VCC, it removes any logic that is disabled by the VCC signal. The VCC signal is only implemented when the disabled logic cannot be removed.

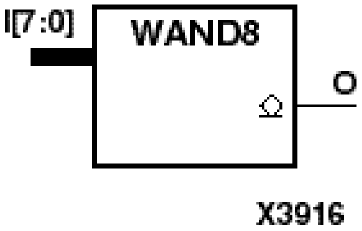
WAND1, 4, 8, 16

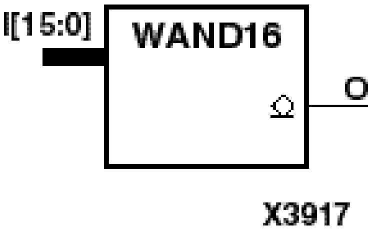
Open-Drain Buffers

Element	XC3000	XC4000E	XC4000X	XC5200	XC9000	Spartan	Spartan XL	Virtex
WAND 1	N/A	Primitive	Primitive	N/A	N/A	N/A	N/A	N/A
WAND 4, WAND 8. WAND 16	N/A	Macro	Macro	N/A	N/A	N/A	N/A	N/A





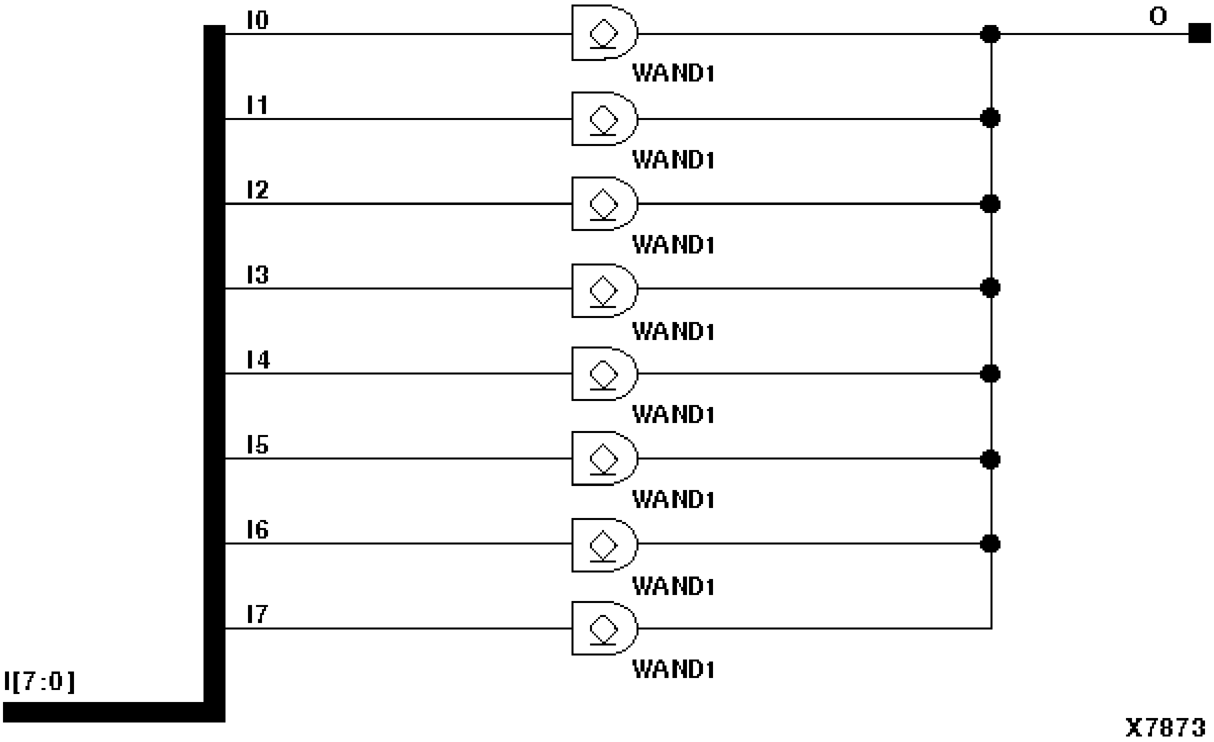




WAND1, WAND4, WAND8, and WAND16 are single and multiple open-drain buffers. Each buffer has an input (I) and an open-drain output (O). When any of the inputs is Low, the output is Low. When all the inputs are High, the output is off. To obtain a High output, add pull-up resistors to the output (O). One pull-up resistor uses the least power, and two pull-up resistors achieve the fastest Low-to-High transition.

To indicate two pull-up resistors, add a DOUBLE parameter to the pull-up symbol attached to the output (O) node. Refer to the appropriate CAE tool interface user guide for details.

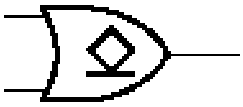
Figure 10-11WAND8 Implementation XC4000



WOR2AND

2-Input OR Gate with Wired-AND Open-Drain Buffer Output

XC3000	XC4000 E	XC4000 X	XC5200	XC9000	Spartan	Spartan XL	Virtex
N/A	Primitive	Primitive	N/A	N/A	N/A	N/A	N/A



X3906

WOR2AND is a 2-input (I1 and I2) OR gate/buffer with an open-drain output (O). It is used in bus applications by tying multiple open-drain outputs together. When both inputs (I1 and I2) are Low, the output (O) is Low. When either input is High, the output is off; wor2and cannot source or sink current. To establish an output High level, tie a pull-up resistor(s) to the output (O). One pull-up resistor uses the least power, two pull-up resistors achieve the fastest Low-to-High speed.

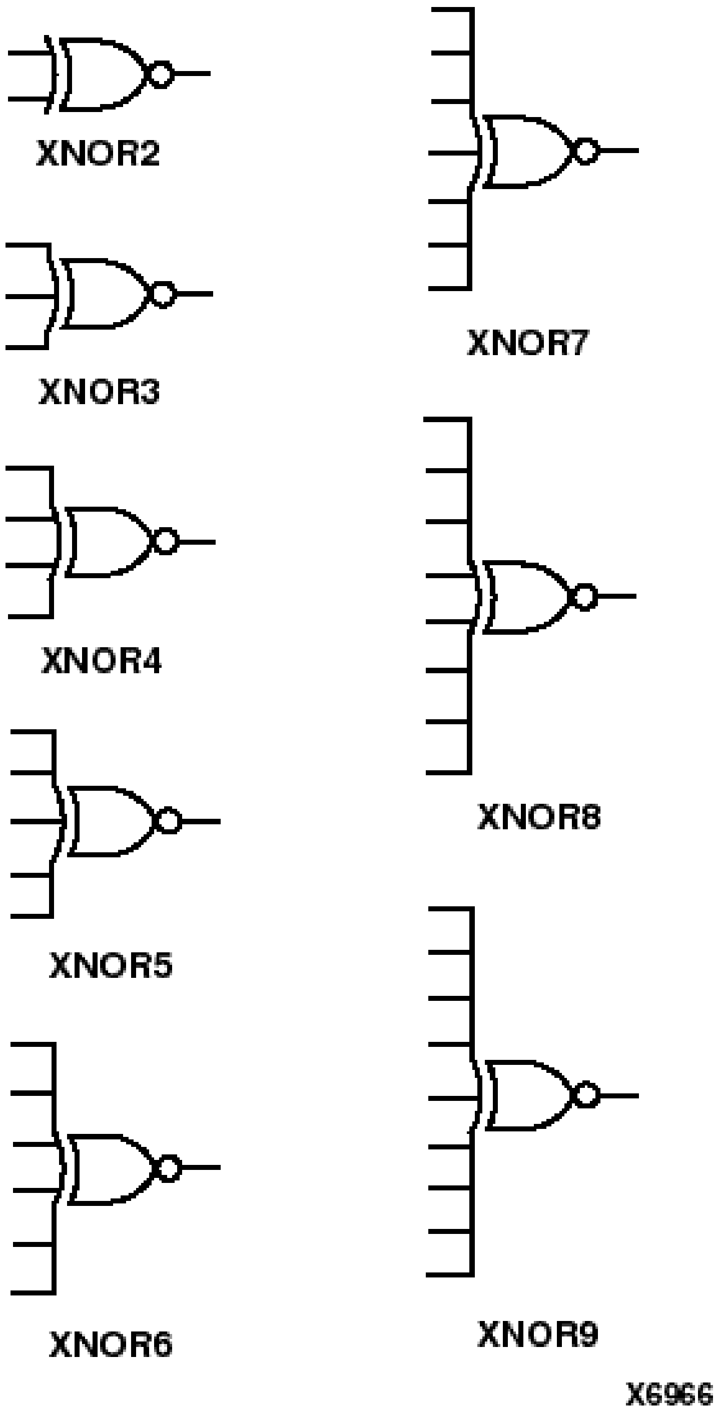
To indicate two pull-up resistors, append a DOUBLE parameter to the pull-up symbol attached to the output (O) node. Refer to the appropriate CAE tool interface user guide for details.

XNOR2-9

2- to 9-Input XNOR Gates with Non-Inverted Inputs

Element	XC3000	XC4000E	XC4000X	XC5200	XC9000	Spartan	Spartan XL	Virtex
XNOR2, XNOR3, XNOR4	Primitive	Primitive	Primitive	Primitive	Primitive	Primitive	Primitive	Primitive
XNOR5	Primitive	Primitive	Primitive	Macro	Macro	Primitive	Primitive	Primitive
XNOR6, XNOR7, XNOR8, XNOR9	Macro	Macro	Macro	Macro	Macro	Macro	Macro	Macro

Figure 10-12XNOR Gate Representations



The XNOR function is performed in the Configurable Logic Block (CLB) function generators in XC3000, XC4000, and Spartans. XNOR functions of up to nine inputs are available. All inputs are non-inverting. Because each input uses a CLB resource, replace functions with unused inputs with functions having the necessary number of
inputs.

Figure 10-13XNOR5 Implementation XC5200

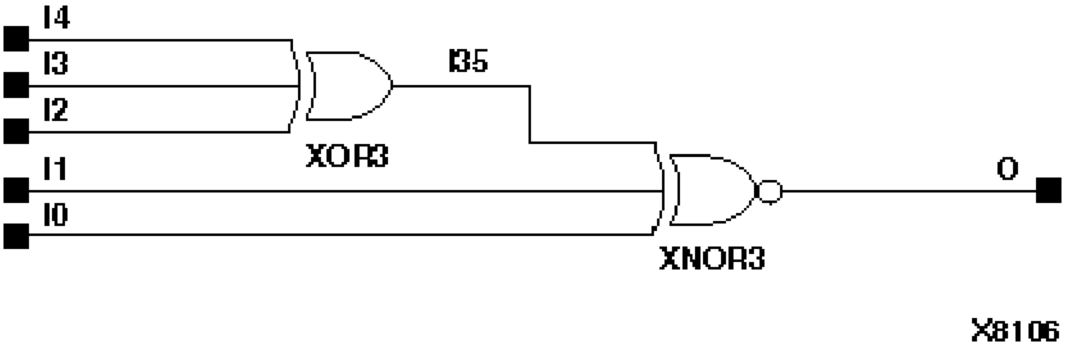


Figure 10-14XNOR5 Implementation XC9000

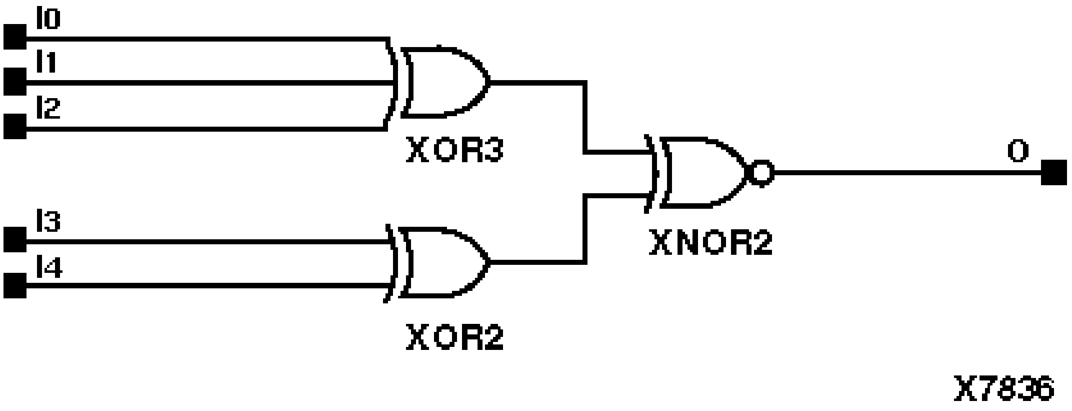


Figure 10-15XNOR6 Implementation XC9000

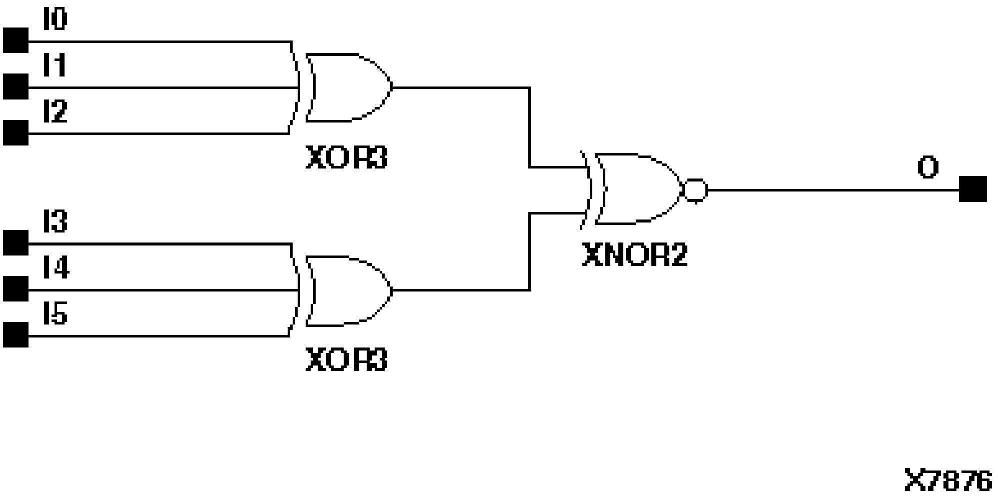


Figure 10-16XNOR7 Implementation XC3000

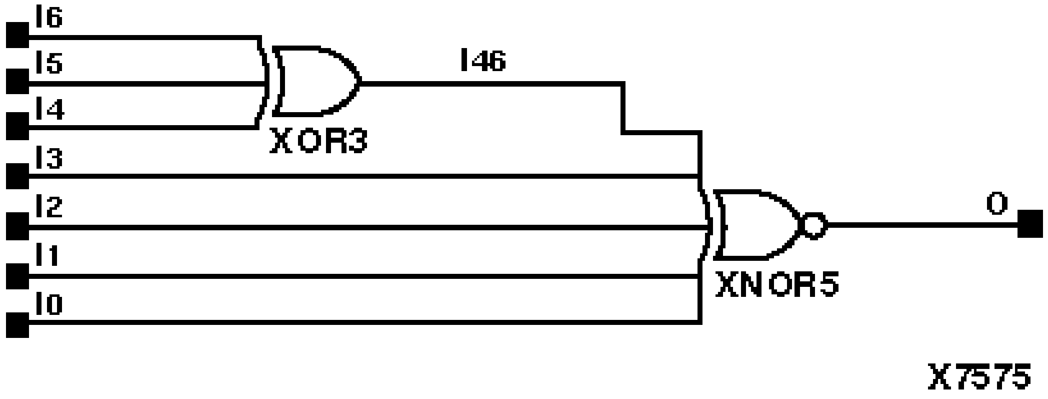


Figure 10-17XNOR7 Implementation XC4000, XC5200, Spartans

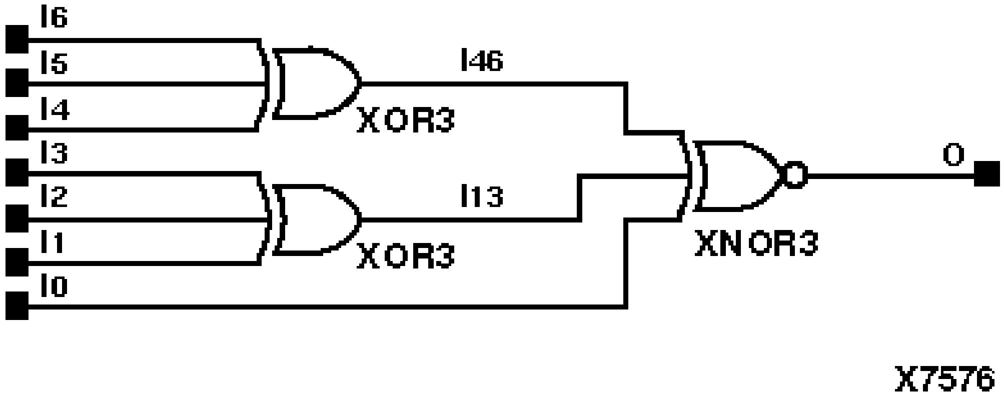


Figure 10-18XNOR7 Implementation XC9000

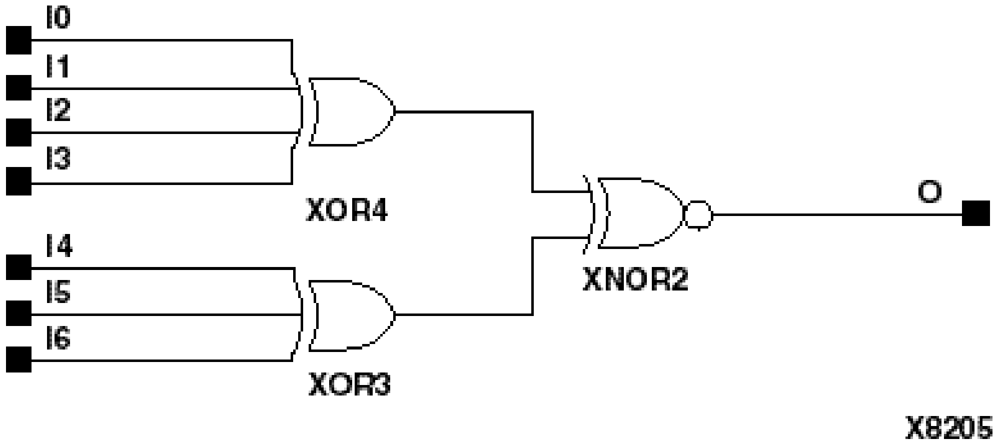


Figure 10-19XNOR7 Implementation Virtex

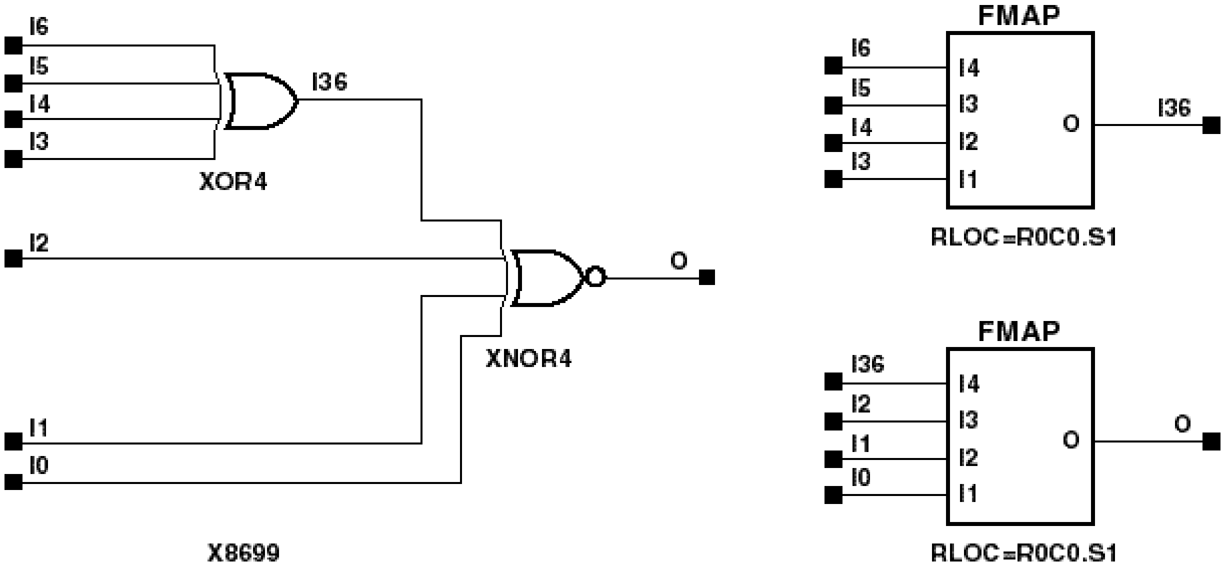


Figure 10-20XNOR8 Implementation XC3000

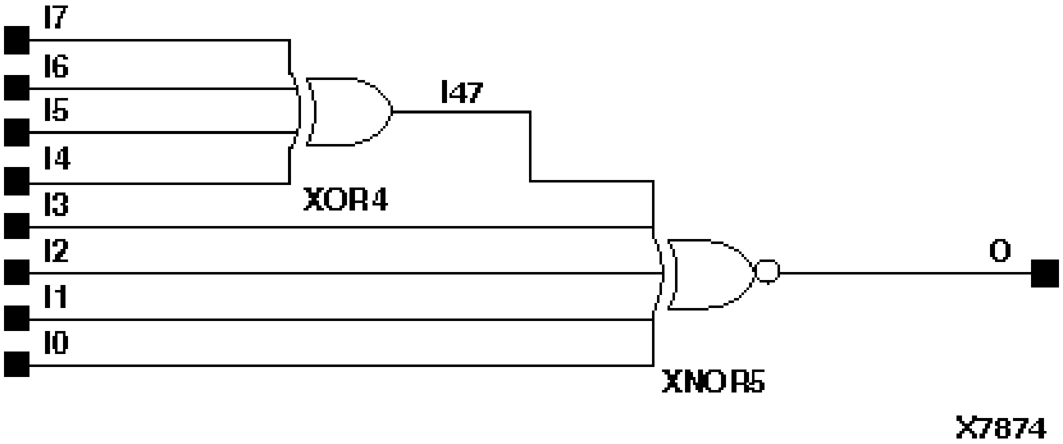


Figure 10-21XNOR8 Implementation XC4000, XC5200, Spartans

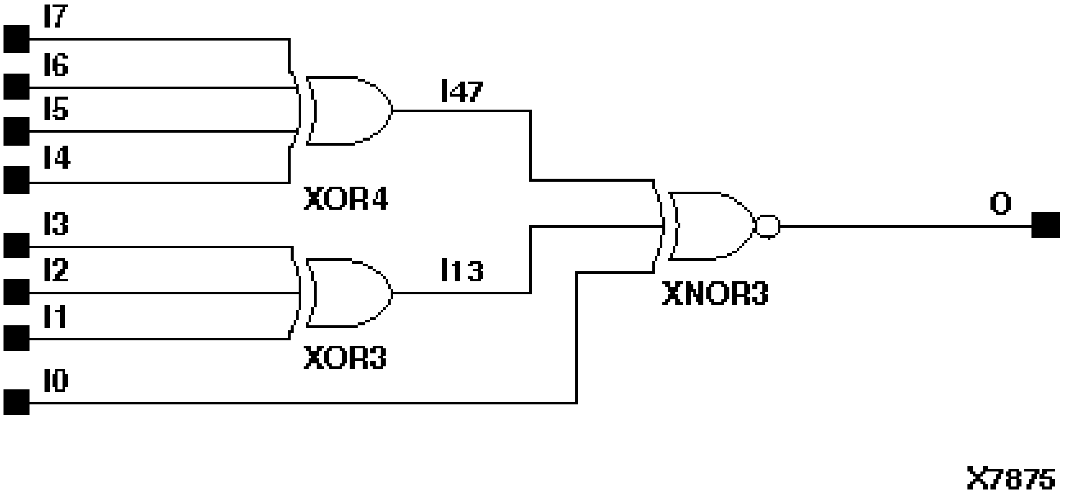


Figure 10-22XNOR8 Implementation XC9000

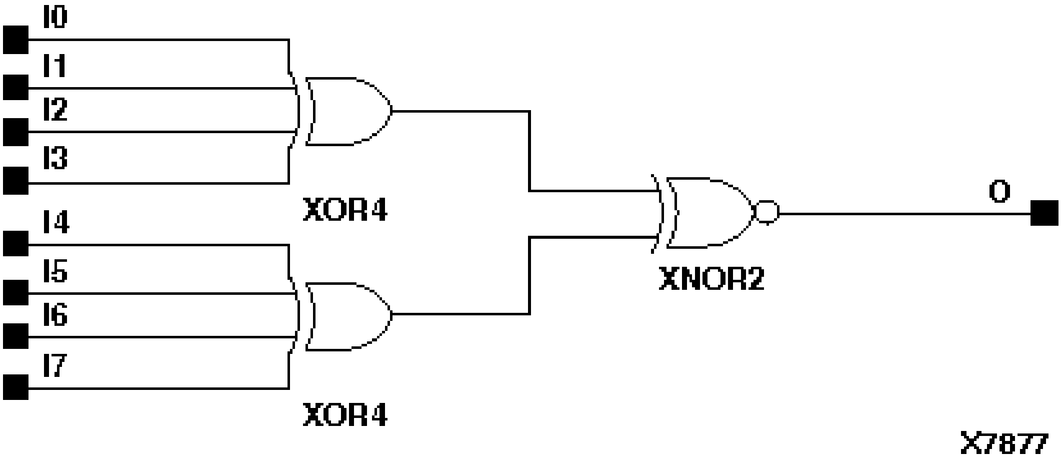


Figure 10-23XNOR8 Implementation Virtex

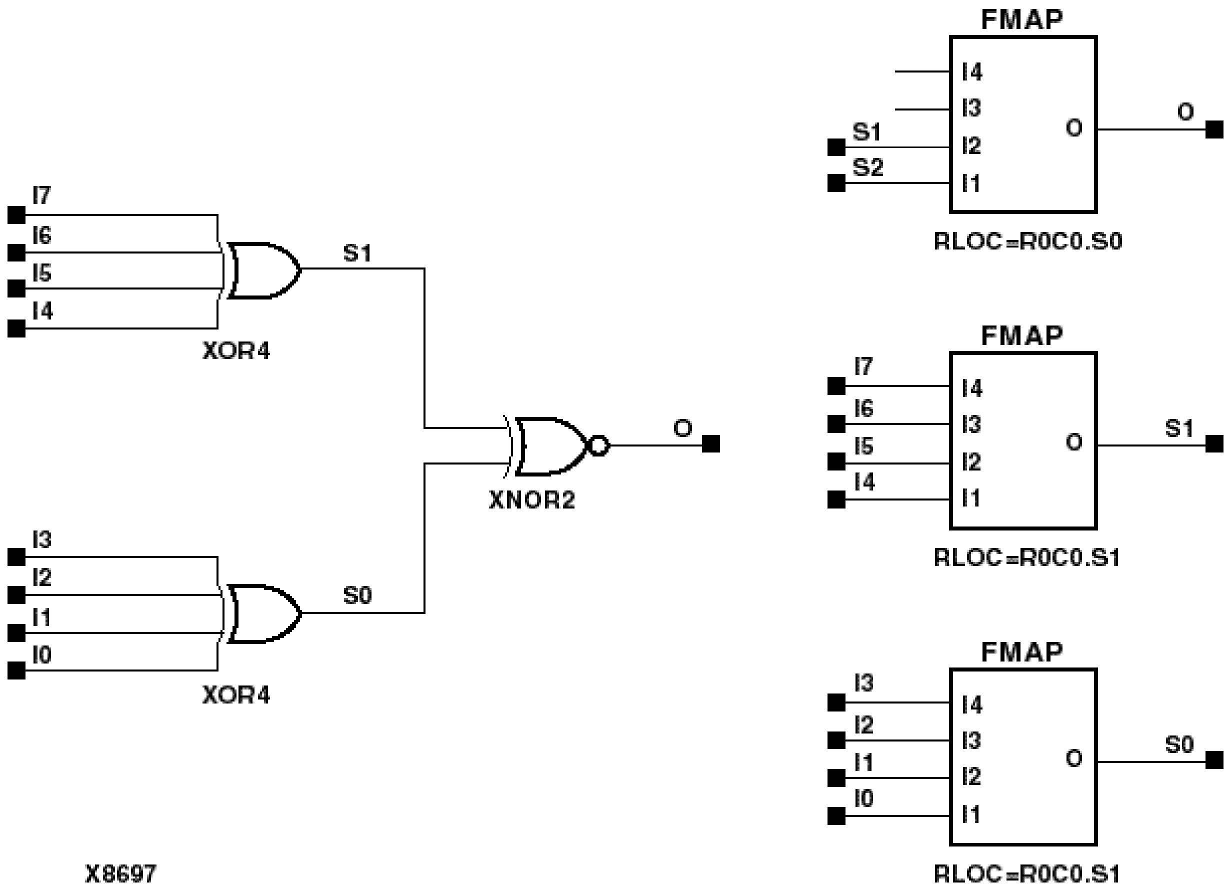


Figure 10-24XNOR9 Implementation XC3000

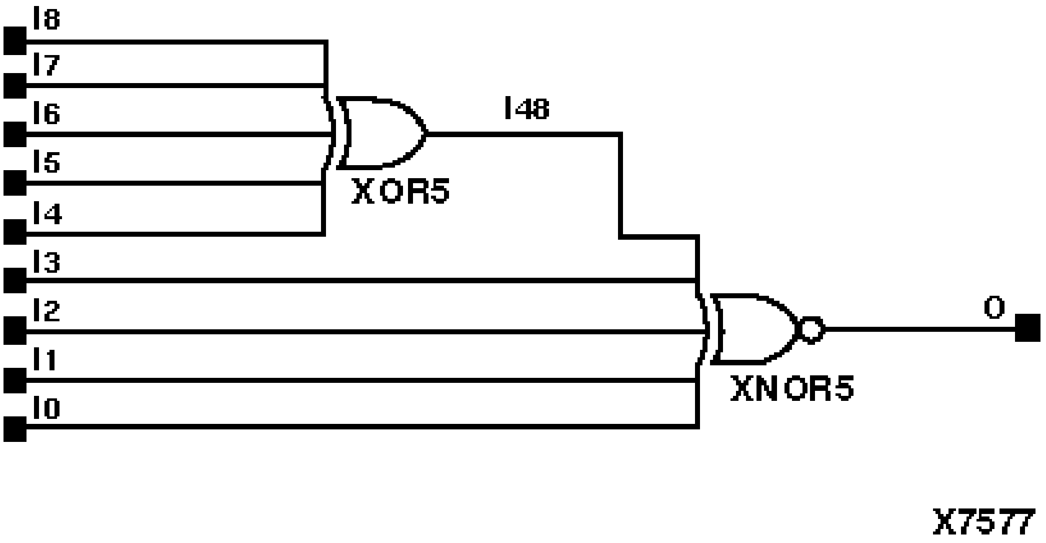


Figure 10-25XNOR9 Implementation XC4000, XC5200, Spartans

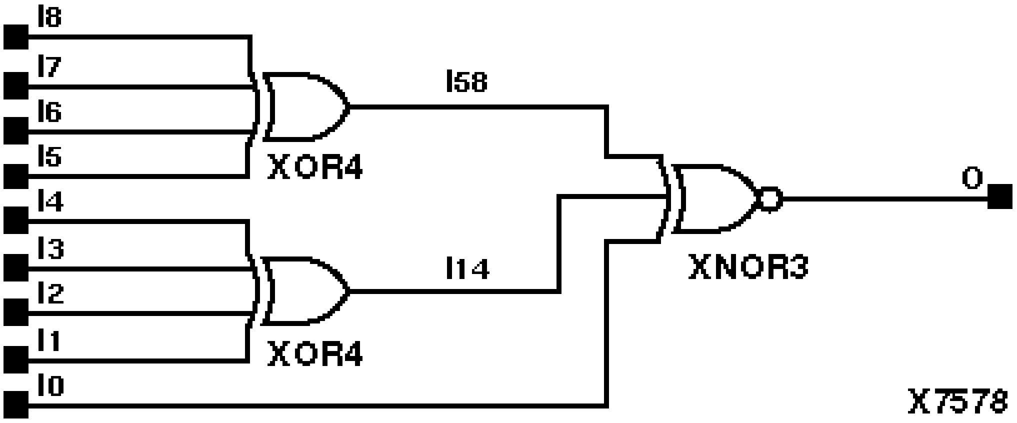


Figure 10-26XNOR9 Implementation XC9000

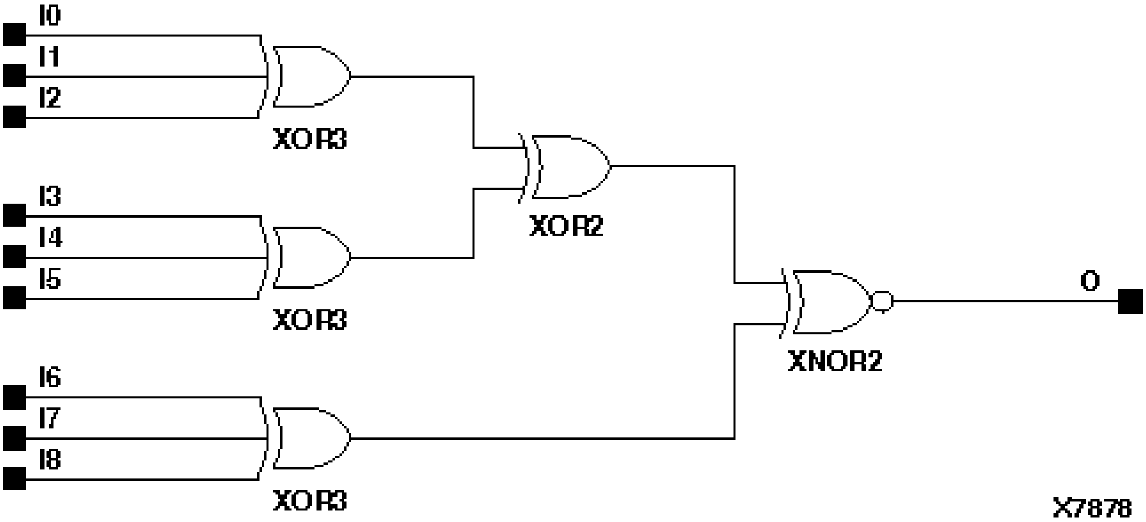
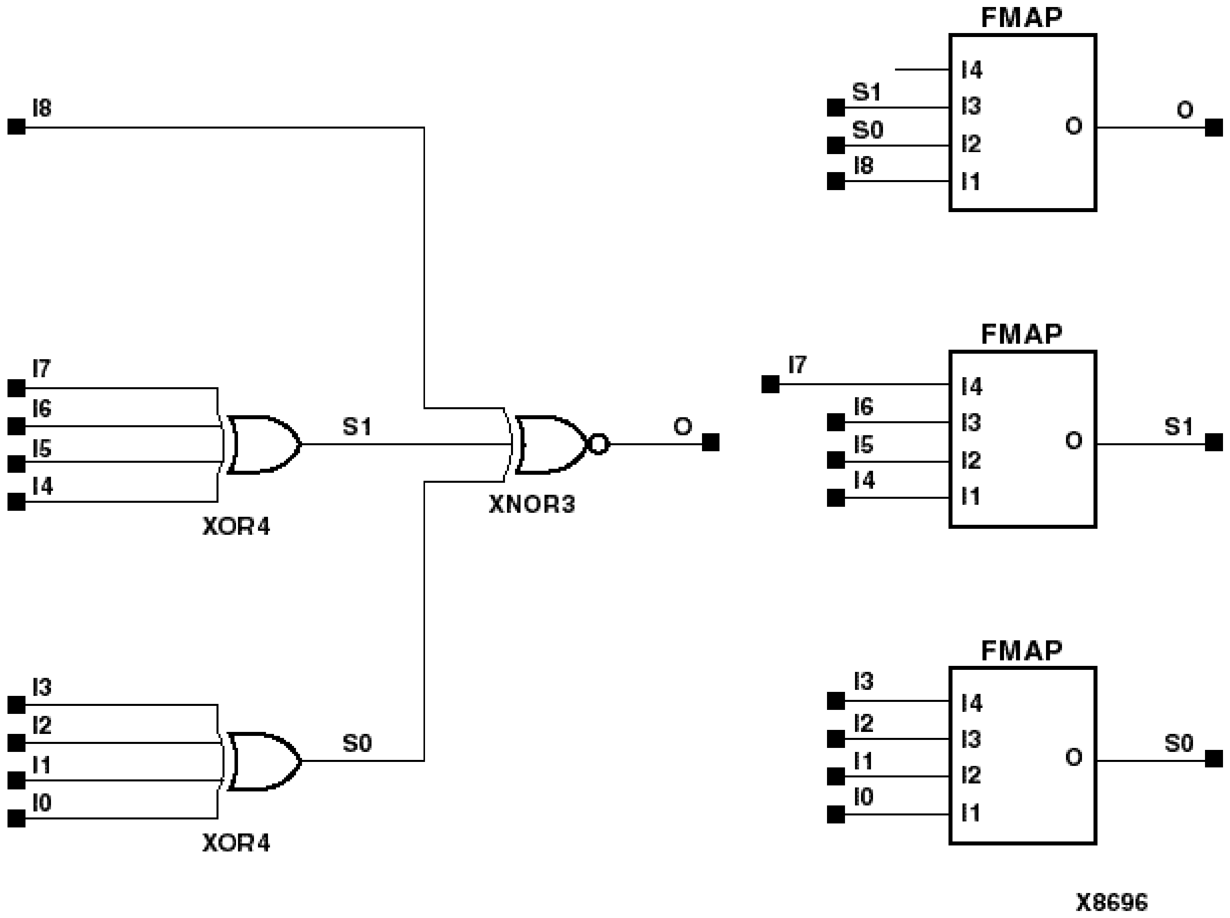


Figure 10-27XNOR9 Implementation Virtex

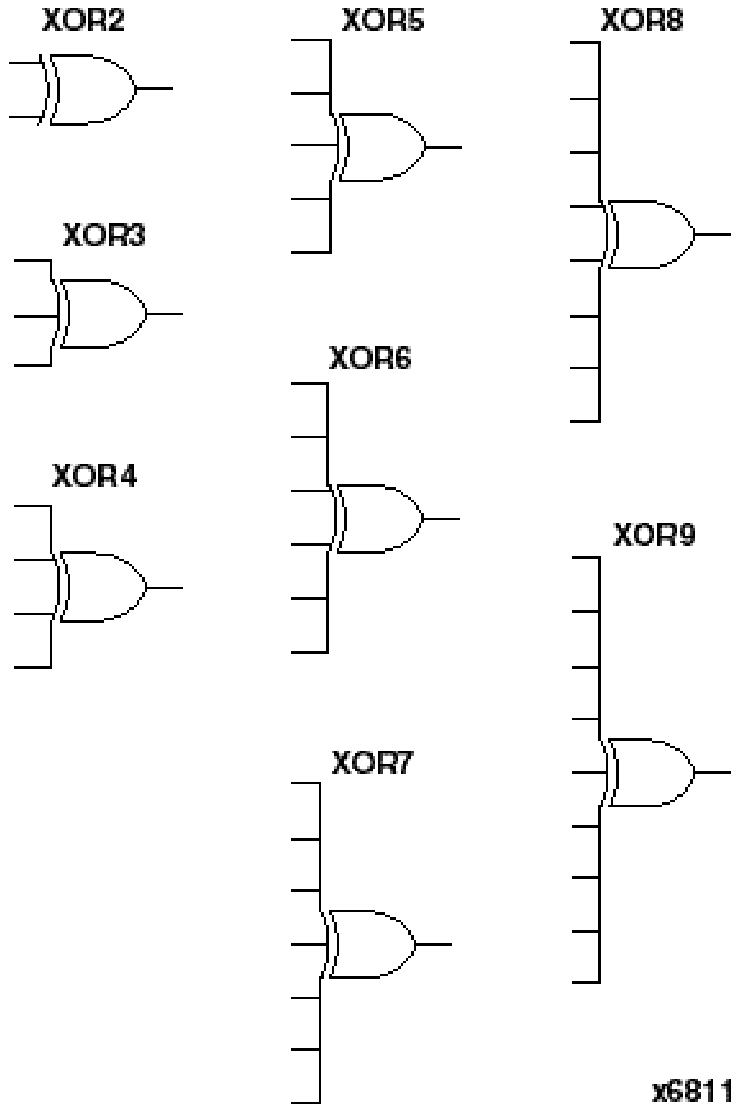


XOR2-9

2- to 9-Input XOR Gates with Non-Inverted Inputs

Element	XC3000	XC4000E	XC4000X	XC5200	XC9000	Spartan	Spartan XL	Virtex
XOR2, XOR3, XOR4	Primitive	Primitive	Primitive	Primitive	Primitive	Primitive	Primitive	Primitive
XOR5	Primitive	Primitive	Primitive	Macro	Macro	Primitive	Primitive	Primitive
XOR6, XOR7, XOR8, XOR9	Macro	Macro	Macro	Macro	Macro	Macro	Macro	Macro

Figure 10-28XOR Gate Representations



The XOR function is performed in the Configurable Logic Block (CLB) function generators in XC3000, XC4000, and Spartans. XOR functions of up to nine inputs are available. All inputs are non-inverting. Because each input uses a CLB resource, replace functions with unused inputs with functions having the necessary number of
inputs.

Figure 10-29XOR5 Implementation XC5200

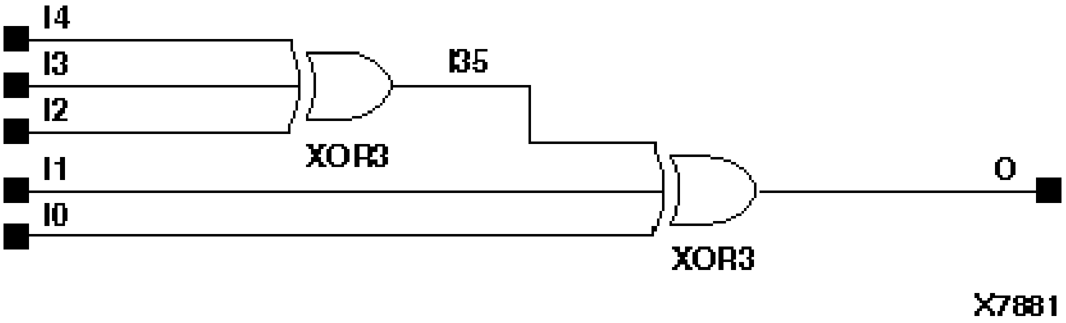


Figure 10-30XOR5 Implementation XC9000

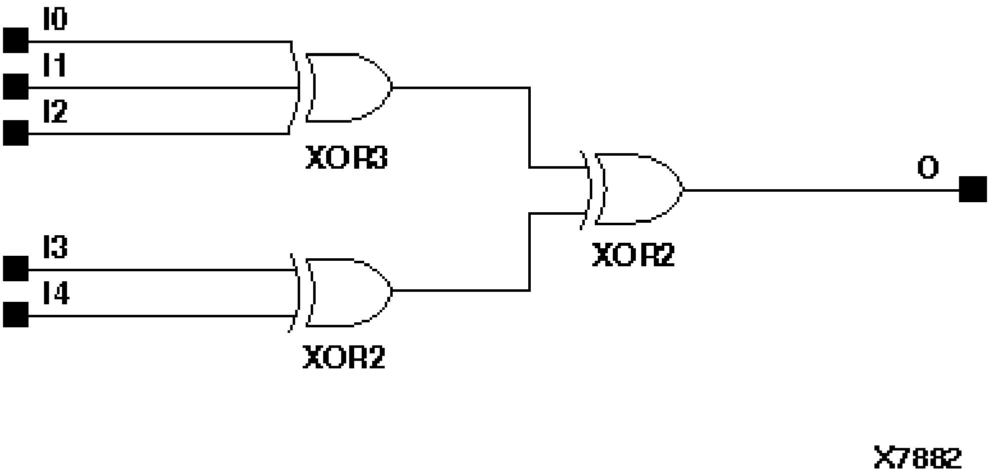


Figure 10-31XOR6 Implementation XC9000

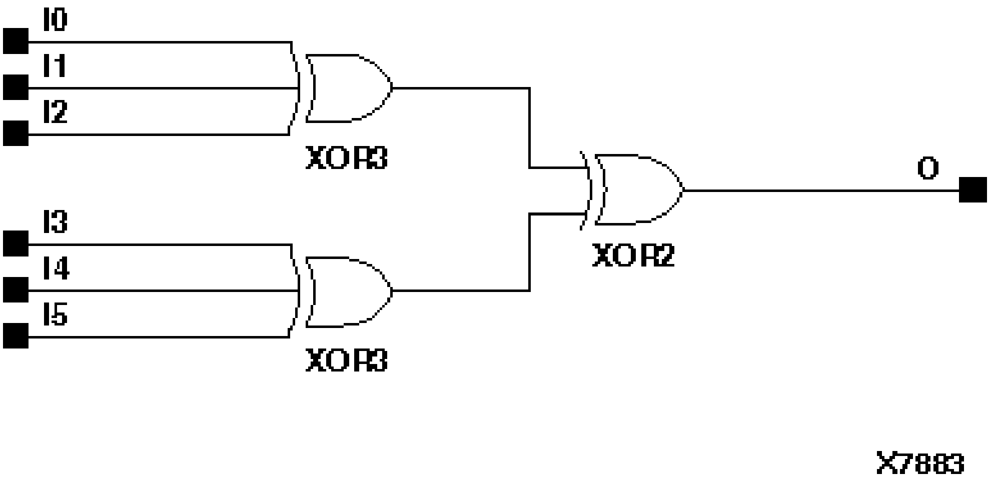


Figure 10-32XOR7 Implementation XC9000

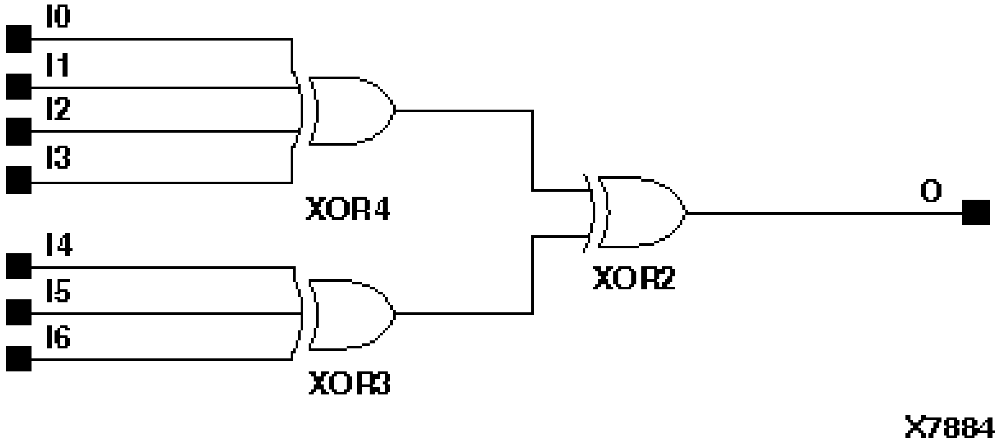


Figure 10-33XOR8 Implementation XC3000

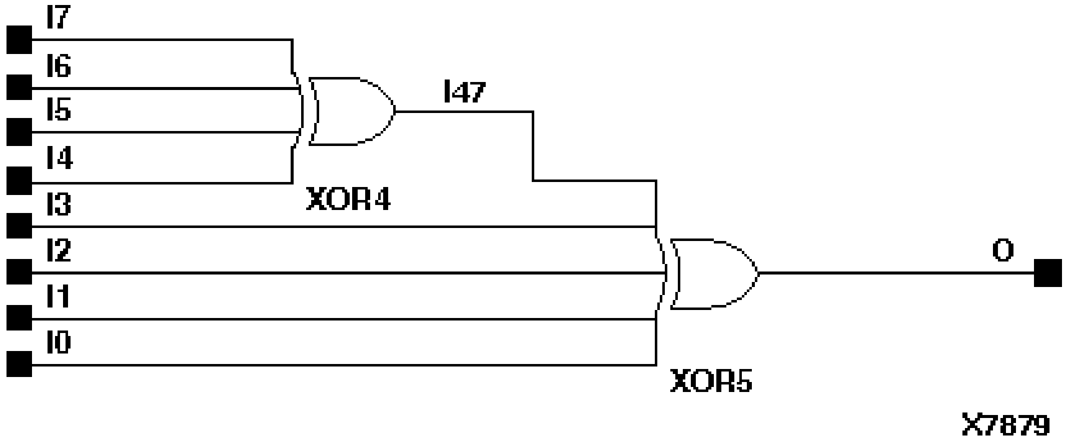


Figure 10-34XOR8 Implementation XC4000, XC5200, Spartans

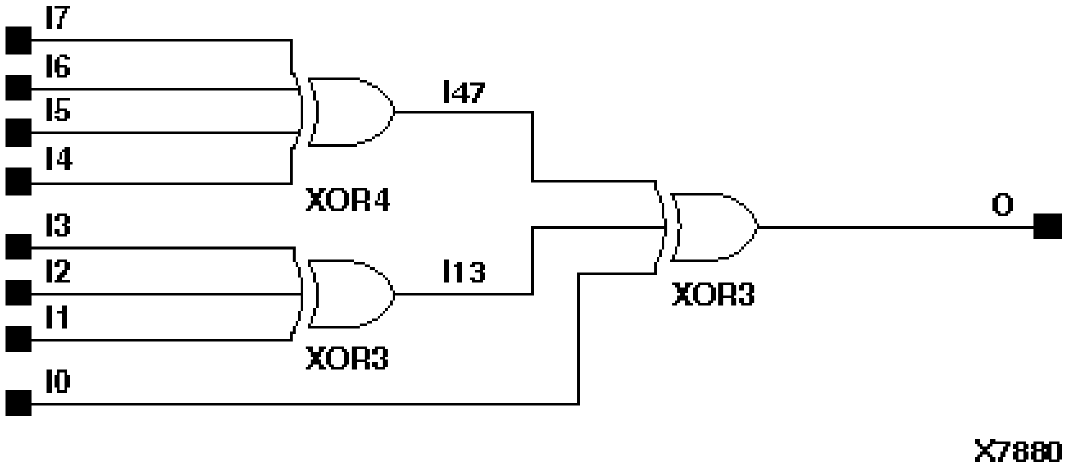


Figure 10-35XOR8 Implementation XC9000

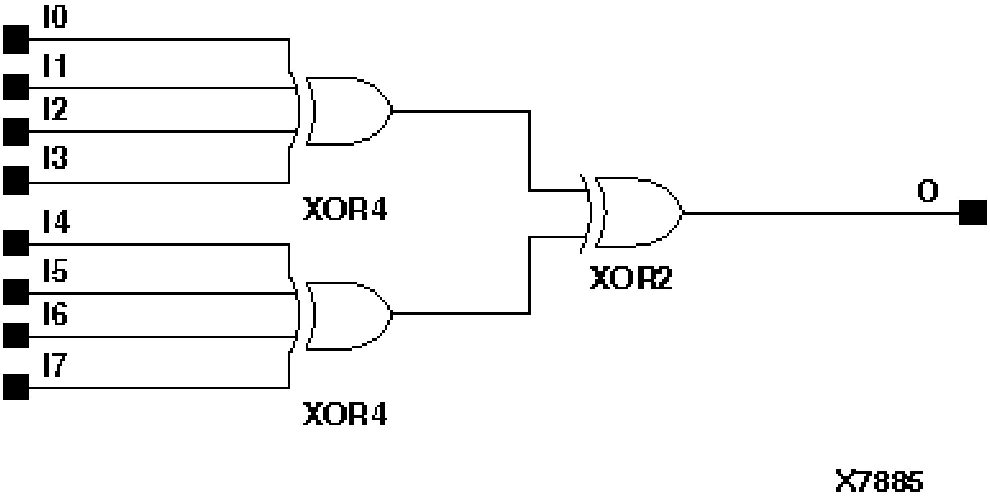


Figure 10-36XOR8 Implementation Virtex

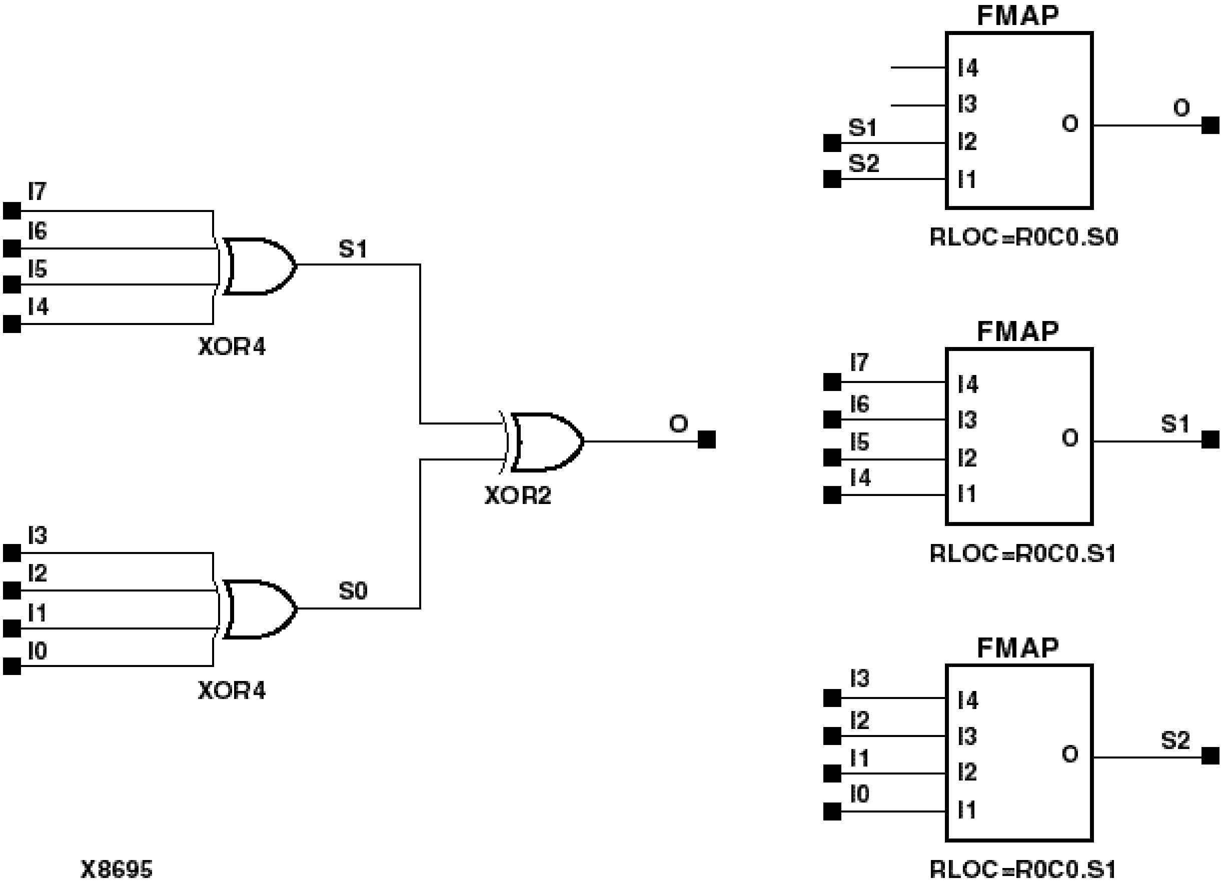
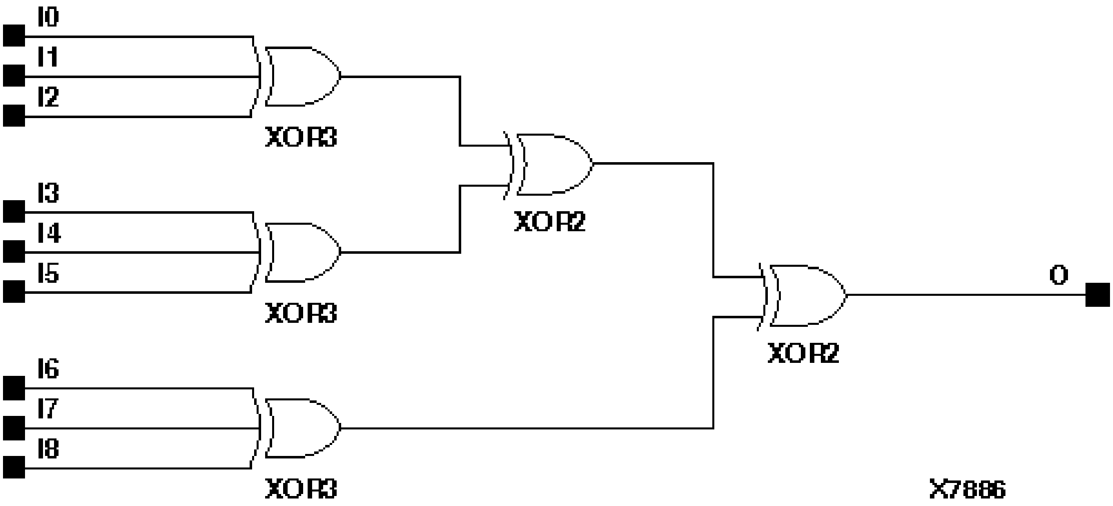


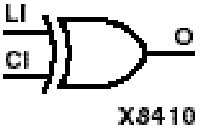
Figure 10-37XOR9 Implementation XC9000



XORCY

XOR for Carry Logic with General Output

XC3000	XC4000 E	XC4000 X	XC5200	XC9000	Spartan	Spartan XL	Virtex
N/A	N/A	N/A	N/A	N/A	N/A	N/A	Primitive



XORCY is a special XOR with general O output used for generating faster and smaller arithmetic functions.

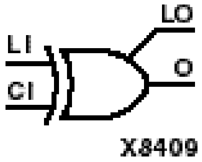
Its O output is a general interconnect. See also**"XORCY_D"** and **"XORCY_L"**.

XORCY_D

XOR for Carry Logic with Dual Output

XC3000	XC4000 E	XC4000 X	XC5200	XC9000	Spartan	Spartan XL	Virtex

N/A	N/A	N/A	N/A	N/A	N/A	N/A	Primitive



XORCY_D is a special XOR used for generating faster and smaller arithmetic functions.

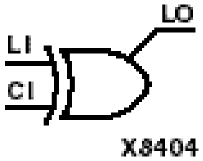
XORCY_D has two functionally identical outputs, O and LO. The O output is a general interconnect. The LO output is used to connect to another output within the same CLB slice.

See also **"XORCY"** and **"XORCY_L"**.

XORCY_L

XOR for Carry Logic with Local Output

XC3000	XC4000 E	XC4000 X	XC5200	XC9000	Spartan	Spartan XL	Virtex
N/A	N/A	N/A	N/A	N/A	N/A	N/A	Primitive



XORCY_L is a special XOR with local LO output used for generating faster and smaller arithmetic functions. The LO output is used to connect to another output within the same CLB slice.

See also **"XORCY"** an d **"XORCY_D"**.