Stratix III Device Handbook, Volume 1

101 Innovation Drive
San Jose, CA 95134
www.altera.com

Software Version: 10.0
Document Version: 2.2
Document Date: © March 2011

Copyright © 2010 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

SIII5V1-2.2

I.S. EN ISO 9001

Contents

Chapter Revision Dates

Additional Information
How to Contact Altera 1-xxv
Typographic Conventions 1-xxv

Chapter I. Device Core
Revision History I-1

Chapter 1. Stratix III Device Family Overview
Features Summary 1-1
Architecture Features 1-6
Logic Array Blocks and Adaptive Logic Modules 1-6
MultiTrack Interconnect 1-7
TriMatrix Embedded Memory Blocks 1-7
DSP Blocks 1-7
Clock Networks and PLLs 1-8
I/O Banks and I/O Structure 1-9
External Memory Interfaces 1-9
High-Speed Differential I/O Interfaces with DPA 1-9
Hot Socketing and Power-On Reset 1-10
Configuration 1-10
Remote System Upgrades 1-11
IEEE 1149.1 (JTAG) Boundary-Scan Testing 1-11
Design Security 1-11
SEU Mitigation 1-12
Programmable Power 1-12
Signal Integrity 1-13
Reference and Ordering Information 1-13
Software Support 1-13
Ordering Information 1-14
Chapter Revision History 1-14

Chapter 2. Logic Array Blocks and Adaptive Logic Modules in Stratix III Devices
Introduction 2-1
Logic Array Blocks 2-1
LAB Interconnects 2-3
LAB Control Signals 2-4

Adaptive Logic Modules 2-5
ALM Operating Modes 2-8
Normal Mode 2-8
Extended LUT Mode 2-11
Arithmetic Mode 2-11
Carry Chain 2-14
Shared Arithmetic Mode 2-14
Shared Arithmetic Chain 2-16
LUT-Register Mode 2-17
Register Chain 2-18
ALM Interconnects 2-20
Clear and Preset Logic Control 2-20
LAB Power Management Techniques 2-21
Conclusion 2-21
Chapter Revision History 2-21

Chapter 3. MultiTrack Interconnect in Stratix III Devices

Introduction 3-1
Row Interconnects 3-1
Column Interconnects 3-2
Memory Block Interface 3-6
DSP Block Interface 3-9
I/O Block Connections to Interconnect 3-11
Conclusion 3-13
Chapter Revision History 3-14

Chapter 4. TriMatrix Embedded Memory Blocks in Stratix III Devices

Introduction 4-1
Overview 4-1
TriMatrix Memory Block Types 4-3
Parity Bit Support 4-3
Byte-Enable Support 4-3
Packed Mode Support 4-5
Address Clock Enable Support 4-5
Mixed Width Support 4-8
Asynchronous Clear 4-8
Error Correction Code Support 4-9
Single Port RAM 4-10
Simple Dual-Port Mode 4-12
True Dual-Port Mode 4-15
Shift-Register Mode 4-17
ROM Mode 4-18
FIFO Mode 4-18
Clocking Modes 4-19
Independent Clock Mode 4-19
Input/Output Clock Mode 4-19
Read/Write Clock Mode 4-20
Single Clock Mode 4-20

Design Considerations 4-20
Selecting TriMatrix Memory Blocks 4-20
Conflict Resolution 4-20
Read During Write 4-21
Same-Port Read-During-Write Mode 4-21
Mixed-Port Read-During-Write Mode 4-22
Power-Up Conditions and Memory Initialization 4-23
Power Management 4-24
Programming File Compatibility 4-24
Conclusion 4-24
Chapter Revision History 4-25

Chapter 5. DSP Blocks in Stratix III Devices

Introduction 5-1
DSP Block Overview 5-1
Simplified DSP Operation 5-3
Operational Modes Overview 5-8
DSP Block Resource Descriptions 5-9
Input Registers 5-10
Multiplier and First-Stage Adder 5-12
Pipeline Register Stage 5-13
Second-Stage Adder 5-13
Round and Saturation Stage 5-14
Second Adder and Output Registers 5-14
Operational Mode Descriptions 5-15
Independent Multiplier Modes 5-15
9-, 12-, and 18-Bit Multiplier 5-15
36-Bit Multiplier 5-18
Double Multiplier 5-19
Two-Multiplier Adder Sum Mode 5-21
18 × 18 Complex Multiply 5-24
Four-Multiplier Adder 5-25
High Precision Multiplier Adder 5-26
Multiply Accumulate Mode 5-28
Shift Modes 5-29
Rounding and Saturation Mode 5-31
DSP Block Control Signals 5-34
Application Examples 5-35
FIR Example 5-35
FFT Example 5-40
Software Support 5-41
Chapter Revision History 5-42

Chapter 6. Clock Networks and PLLs in Stratix III Devices

Clock Networks in Stratix III Devices 6-1
Global Clock Networks 6-2
Regional Clock Networks 6-3
Periphery Clock Networks 6-4
Clocking Regions 6-6
Clock Network Sources 6-7
Clock Output Connections 6-12
Clock Source Control for PLLs 6-13
Clock Control Block 6-14
Clock Enable Signals 6-17
PLLs in Stratix III Devices 6-18
Stratix III PLL Hardware Overview 6-20
PLL Clock I/O Pins 6-21
Stratix III PLL Software Overview 6-24
Clock Feedback Modes 6-26
Source Synchronous Mode 6-27
Source-Synchronous Mode for LVDS Compensation 6-28
No-Compensation Mode 6-28
Normal Mode 6-28
Zero-Delay Buffer Mode 6-29
External Feedback Mode 6-30
Clock Multiplication and Division 6-31
Post-Scale Counter Cascading 6-32
Programmable Duty Cycle 6-32
PLL Control Signals 6-33
pfdena 6-33
areset 6-33
locked 6-33
Clock Switchover 6-33
Automatic Clock Switchover 6-34
Manual Clock Switchover 6-37
Guidelines 6-37
Programmable Bandwidth 6-39
Background 6-39
Implementation 6-40
Phase-Shift Implementation 6-41
PLL Reconfiguration 6-42
PLL Reconfiguration Hardware Implementation 6-43
Post-Scale Counters (C0 to C9) 6-45
Scan Chain Description 6-46
Charge Pump and Loop Filter 6-48
Bypassing PLL 6-49
Dynamic Phase-Shifting 6-49
PLL Cascading and Clock Network Guidelines 6-52
Spread-Spectrum Tracking 6-52
PLL Specifications 6-52
Chapter Revision History 6-53

Chapter II. I/O Interfaces

Revision History II-1

Chapter 7. Stratix III Device I/O Features

Stratix III I/O Standards Support 7-2
I/O Standards and Voltage Levels 7-3
Stratix III I/O Banks 7-5
Modular I/O Banks 7-7
Stratix III I/O Structure 7-12
3.3-V I/O Interface 7-14
External Memory Interfaces 7-14
High-Speed Differential I/O with DPA Support 7-15
Programmable Current Strength 7-15
Programmable Slew Rate Control 7-16
Programmable Delay 7-17
Programmable Output Buffer Delay 7-17
Open-Drain Output 7-17
Bus Hold 7-18
Programmable Pull-Up Resistor 7-18
Programmable Pre-Emphasis 7-18
Programmable Differential Output Voltage 7-19
MultiVolt I/O Interface 7-19
OCT Support 7-20
On-Chip Series Termination without Calibration 7-20
On-Chip Series Termination with Calibration 7-21
Expanded On-Chip Series Termination with Calibration 7-22
Left Shift Series Termination Control 7-23
On-Chip Parallel Termination with Calibration 7-24
Dynamic OCT 7-25
LVDS Input On-Chip Termination (R_D) 7-26
OCT Calibration 7-27
OCT Calibration Block Location 7-27
Sharing an OCT Calibration Block in Multiple I/O Banks 7-29
OCT Calibration Block Modes of Operation 7-29
Power-Up Mode 7-29
User Mode 7-30
OCT Calibration 7-31
Serial Data Transfer 7-31
Example of Using Multiple OCT Calibration Blocks 7-31
RS Calibration 7-32
Termination Schemes for I/O Standards 7-32
Single-Ended I/O Standards Termination 7-32
Differential I/O Standards Termination 7-34
LVDS 7-36
Differential LVPECL 7-37
RSDS 7-37
Mini-LVDS 7-38
Design Considerations 7-39
I/O Termination 7-40
Single-Ended I/O Standards 7-40
Differential I/O Standards 7-40
I/O Banks Restrictions 7-40
Non-Voltage-Referenced Standards 7-40
Voltage-Referenced Standards 7-41
Mixing Voltage-Referenced and Non-Voltage-Referenced Standards 7-41
Chapter Revision History 7-42

Chapter 8. External Memory Interfaces in Stratix III Devices

Memory Interfaces Pin Support 8-3
Data and Data-Strobe/Clock Pins 8-4
Using R_{UP}/R_{DN} Pins in a DQS/DQ Group Used for Memory Interfaces 8-5
Combining ×16/×18 DQS/DQ groups for ×36 QDR II+/QDR II SRAM Interface 8-15
Rules to Combine Groups 8-15
Optional Parity, DM, BWSn, NWSn, ECC and QVLD Pins 8-17
Address and Control/Command Pins 8-17
Memory Clock Pins 8-18
Stratix III External Memory Interface Features 8-18
DQS Phase-Shift Circuitry 8-19
DLL 8-21
Phase Offset Control 8-27
DQS Logic Block 8-29
DQS Delay Chain 8-29
Update Enable Circuitry 8-30
DQS Postamble Circuitry 8-30
Leveling Circuitry 8-31
Dynamic OCT Control 8-33
IOE Registers 8-34
Delay Chain 8-39
I/O Configuration Block and DQS Configuration Block 8-41
IOE Features 8-42
OCT 8-43
Programmable IOE Delay Chains 8-43
Programmable Output Buffer Delay 8-43
Programmable Slew Rate Control 8-44
Programmable Drive Strength 8-44
PLL 8-44
Chapter Revision History 8-45

Chapter 9. High-Speed Differential I/O Interfaces and DPA in Stratix III Devices

I/O Banks 9-2
LVDS Channels 9-3
Differential Transmitter 9-4
Differential Receiver 9-6
Receiver Data Realignment Circuit (Bit Slip) 9-7
Dynamic Phase Aligner (DPA) 9-8
Soft-CDR Mode 9-9
Synchronizer 9-10
Programmable Pre-Emphasis and Programmable V_{OD} 9-11
Differential I/O Termination 9-12
Left/Right PLLs (PLL_Lx/ PLL_Rx) 9-12
Clocking 9-14
Source-Synchronous Timing Budget 9-15
Differential Data Orientation 9-15
Differential I/O Bit Position 9-15
Receiver Skew Margin for Non-DPA 9-17

Differential Pin Placement Guidelines 9-19
Guidelines for DPA-Enabled Differential Channels 9-19
DPA-Enabled Channels and Single-Ended I/Os 9-19
DPA-Enabled Channel Driving Distance 9-19
Using Corner and Center Left/Right PLLs 9-19
Using Both Center Left/Right PLLs 9-21
Guidelines for DPA-Disabled Differential Channels 9-23
DPA-Disabled Channels and Single-Ended I/Os 9-23
DPA-Disabled Channel Driving Distance 9-23
Using Corner and Center Left/Right PLLs 9-23
Using Both Center Left/Right PLLs 9-26
Chapter Revision History 9-27

Chapter III. Hot Socketing, Configuration, Remote Upgrades, and Testing
Revision History III-1

Chapter 10. Hot Socketing and Power-On Reset in Stratix III Devices
Stratix III Hot-Socketing Specifications 10-1
Stratix III Devices Can Be Driven Before Power Up 10-1
I/O Pins Remain Tri-Stated During Power Up 10-2
Insertion or Removal of a Stratix III Device from a Powered-Up System 10-2
Hot-Socketing Feature Implementation in Stratix III Devices 10-2
Power-On Reset Circuitry 10-4
Power-On Reset Specifications 10-5
Chapter Revision History 10-7

Chapter 11. Configuring Stratix III Devices
Configuration Devices 11-1
Configuration Schemes 11-1
Configuration Features 11-3
Configuration Data Decompression 11-4
Design Security Using Configuration Bitstream Encryption 11-6
Remote System Upgrade 11-7
Power-On Reset Circuit 11-7
V_{CCPGM} Pins 11-7
V_{CCPD} Pins 11-7
Fast Passive Parallel Configuration 11-8
FPP Configuration Using a MAX II Device as an External Host 11-8
FPP Configuration Timing 11-14
FPP Configuration Using a Microprocessor 11-16
Fast Active Serial Configuration (Serial Configuration Devices) 11-17
Estimating Active Serial Configuration Time 11-23
Programming Serial Configuration Devices 11-25
Passive Serial Configuration 11-27
PS Configuration Using a MAX II Device as an External Host 11-27
PS Configuration Timing 11-32
PS Configuration Using a Microprocessor 11-33
PS Configuration Using a Download Cable 11-33
JTAG Configuration 11-37
Jam STAPL 11-43
Device Configuration Pins 11-43
Chapter Revision History 11-51

Chapter 12. Remote System Upgrades with Stratix III Devices

Functional Description 12-1
Enabling Remote Update 12-3
Configuration Image Types 12-4
Remote System Upgrade Mode 12-5
Remote Update Mode 12-5
Dedicated Remote System Upgrade Circuitry 12-7
Remote System Upgrade Registers 12-8
Remote System Upgrade Control Register 12-9
Remote System Upgrade Status Register 12-10
Remote System Upgrade State Machine 12-11
User Watchdog Timer 12-11
Quartus II Software Support 12-12
ALTREMOTE_UPDATE Megafunction 12-13
Chapter Revision History 12-14

Chapter 13. IEEE 1149.1 (JTAG) Boundary-Scan Testing in Stratix III Devices

IEEE Std. 1149.1 BST Architecture 13-1
IEEE Std. 1149.1 Boundary-Scan Register 13-4
Boundary-Scan Cells of a Stratix III Device I/O Pin 13-5
IEEE Std. 1149.1 BST Operation Control 13-7
SAMPLE/PRELOAD Instruction Mode 13-11
EXTEST Instruction Mode 13-13
BYPASS Instruction Mode 13-15
IDCODE Instruction Mode 13-16
USERCODE Instruction Mode 13-16
CLAMP Instruction Mode 13-16
HIGHZ Instruction Mode 13-17
I/O Voltage Support in JTAG Chain 13-17
IEEE Std. 1149.1 BST Circuitry 13-18
IEEE Std. 1149.1 BST Circuitry (Disabling) 13-19
IEEE Std. 1149.1 BST Guidelines 13-19
Boundary-Scan Description Language (BSDL) Support 13-20
Chapter Revision History 13-21

Chapter IV. Design Security and Single Event Upset (SEU) Mitigation

Revision History IV-1

Chapter 14. Design Security in Stratix III Devices

Introduction 14-1
Stratix III Security Protection 14-1
Security Against Copying 14-1
Security Against Reverse Engineering 14-2
Security Against Tampering 14-2
AES Decryption Block 14-2
Flexible Security Key Storage 14-2
Stratix III Design Security Solution 14-3
Security Modes Available 14-4
Volatile Key 14-4
Non-Volatile Key 14-4
Non-Volatile Key with Tamper Protection Bit Set 14-4
No Key Operation 14-5
Supported Configuration Schemes 14-5

Conclusion 14-8
Chapter Revision History 14-8

Chapter 15. SEU Mitigation in Stratix III Devices
Error Detection Fundamentals 15-2
Configuration Error Detection 15-2
User Mode Error Detection 15-2
Automated Single Event Upset Detection 15-5
Error Detection Pin Description 15-6
CRC_ERROR Pin 15-6
Error Detection Block 15-7
Error Detection Registers 15-7
Error Detection Timing 15-9
Software Support 15-11
Recovering From CRC Errors 15-12
Chapter Revision History 15-12

Chapter V. Power and Thermal Management
Revision History V-1

Chapter 16. Programmable Power and Temperature-Sensing Diodes in Stratix III Devices
Introduction 16-1
Stratix III Power Technology 16-1
Selectable Core Voltage 16-1
Programmable Power Technology 16-2
Relationship Between Selectable Core Voltage and Programmable Power Technology 16-3
Stratix III External Power Supply Requirements 16-3
Temperature Sensing Diode 16-5
External Pin Connections 16-6
Conclusion 16-6
Chapter Revision History 16-7

Chapter VI. Packaging Information
Revision History VI-1

Chapter 17. Stratix III Device Packaging Information
Thermal Resistance 17-2
Package Outlines 17-2
Chapter Revision History 17-2

Chapter Revision Dates

The chapters in this book were revised on the following dates. Where chapters or groups of chapters are available separately, part numbers are listed.

Chapter 1 Stratix III Device Family Overview
Revised: *March 2010*
Part Number: *SIII51001-1.8*

Chapter 2 Logic Array Blocks and Adaptive Logic Modules in Stratix III Devices
Revised: *February 2009*
Part Number: *SIII51002-1.5*

Chapter 3 MultiTrack Interconnect in Stratix III Devices
Revised: *October 2008*
Part Number: *SIII51003-1.2*

Chapter 4 TriMatrix Embedded Memory Blocks in Stratix III Devices
Revised: *May 2009*
Part Number: *SIII51004-1.8*

Chapter 5 DSP Blocks in Stratix III Devices
Revised: *March 2010*
Part Number: *SIII51005-1.7*

Chapter 6 Clock Networks and PLLs in Stratix III Devices
Revised: *July 2010*
Part Number: *SIII51006-2.0*

Chapter 7 Stratix III Device I/O Features
Revised: *July 2010*
Part Number: *SIII51007-1.9*

Chapter 8 External Memory Interfaces in Stratix III Devices
Revised: *March 2010*
Part Number: *SIII51008-1.9*

Chapter 9 High-Speed Differential I/O Interfaces and DPA in Stratix III Devices
Revised: *July 2010*
Part Number: *SIII51009-1.9*

Chapter 10 Hot Socketing and Power-On Reset in Stratix III Devices
Revised: *March 2010*
Part Number: *SIII51010-1.7*

Chapter 11 Configuring Stratix III Devices
Revised: *March 2011*
Part Number: *SIII51011-2.0*

Chapter 12 Remote System Upgrades with Stratix III Devices
Revised: *March 2010*
Part Number: *SIII51012-1.5*

Chapter 13 IEEE 1149.1 (JTAG) Boundary-Scan Testing in Stratix III Devices
Revised: *July 2010*
Part Number: *SIII51013-1.9*

Chapter 14 Design Security in Stratix III Devices
Revised: *May 2009*
Part Number: *SIII51014-1.5*

Chapter 15 SEU Mitigation in Stratix III Devices
Revised: *March 2010*
Part Number: *SIII51015-1.7*

Chapter 16 Programmable Power and Temperature-Sensing Diodes in Stratix III Devices
Revised: *February 2009*
Part Number: *SIII51016-1.5*

Chapter 17 Stratix III Device Packaging Information
Revised: *March 2010*
Part Number: *SIII51017-1.7*

List of Figures

Figure 1–1: Stratix III Device Packaging Ordering Information 1-14
Figure 2–1: Stratix III LAB Structure 2-2
Figure 2–2: Stratix III LAB and MLAB Structure 2-3
Figure 2–3: Direct Link Connection 2-4
Figure 2–4: LAB-Wide Control Signals 2-5
Figure 2–5: High-Level Block Diagram of the Stratix III ALM 2-6
Figure 2–6: Stratix III ALM Details 2-7
Figure 2–7: ALM in Normal Mode (Note 1) 2-9
Figure 2–8: 4 × 2 Crossbar Switch Example 2-10
Figure 2–9: Input Function in Normal Mode (Note 1) 2-11
Figure 2–10: Template for Supported Seven-Input Functions in Extended LUT Mode 2-11
Figure 2–11: ALM in Arithmetic Mode 2-12
Figure 2–12: Conditional Operation Example 2-13
Figure 2–13: ALM in Shared Arithmetic Mode 2-15
Figure 2–14: Example of a 3-Bit Add Utilizing Shared Arithmetic Mode 2-16
Figure 2–15: LUT Register from Two Combinational Blocks 2-17
Figure 2–16: ALM in LUT-Register Mode with 3-Register Capability 2-18
Figure 2–17: Register Chain within an LAB (Note 1) 2-19
Figure 2–18: Shared Arithmetic Chain, Carry Chain, and Register Chain Interconnects 2-20
Figure 3–1: R4 Interconnect Connections (Note 1), (2) 3-2
Figure 3–2: Shared Arithmetic Chain, Carry Chain, and Register Chain Interconnects 3-3
Figure 3–3: C4 Interconnect Connections (Note 1) 3-4
Figure 3–4: MLAB RAM Block LAB Row Interface 3-7
Figure 3–5: M9K RAM Block LAB Row Interface 3-8
Figure 3–6: M144K Row Unit Interface to Interconnect 3-9
Figure 3–7: High-Level View, DSP Block Interface to Interconnect 3-10
Figure 3–8: Detailed View, DSP Block Interface to Interconnect 3-11
Figure 3–9: Row I/O Block Connection to Interconnect 3-12
Figure 3–10: Column I/O Block Connection to Interconnect 3-13
Figure 4–1: Stratix III Byte-Enable Functional Waveform for M9K and M144K 4-4
Figure 4–2: Stratix III Byte-Enable Functional Waveform for MLABs 4-5
Figure 4–3: Stratix III Address Clock Enable Block Diagram 4-6
Figure 4–4: Stratix III Address Clock Enable during Read Cycle Waveform 4-6
Figure 4–5: Stratix III Address Clock Enable during Write Cycle Waveform for M9K and M144K 4-7
Figure 4–6: Stratix III Address Clock Enable during Write Cycle Waveform for MLABs 4-8
Figure 4–7: Output Latch Asynchronous Clear Waveform 4-9
Figure 4–8: ECC Block Diagram of the M144K 4-10
Figure 4–9: Single-Port Memory (Note 1) 4-11
Figure 4–10: Timing Waveform for Read-Write Operations (Single-Port Mode) for M9K and M144K . 4-12
Figure 4–11: Timing Waveform for Read-Write Operations (Single-Port Mode) for MLABs 4-12
Figure 4–12: Stratix III Simple Dual-Port Memory (Note 1) 4-13
Figure 4–13: Stratix III Simple Dual-Port Timing Waveforms for M9K and M144K 4-14
Figure 4–14: Stratix III Simple Dual-Port Timing Waveforms for MLABs 4-14
Figure 4–15: Stratix III True Dual-Port Memory (Note 1) 4-15
Figure 4–16: Stratix III True Dual-Port Timing Waveform 4-17
Figure 4–17: Stratix III Shift-Register Memory Configuration 4-18
Figure 4–18: Stratix III Read-During-Write Data Flow 4-21

Figure 4–19: Same Port Read-During-Write: New Data Mode (Note 1) 4-22
Figure 4–20: Same Port Read-During-Write: Old Data Mode (Note 1) 4-22
Figure 4–21: Mixed Port Read During Write: Old Data Mode (Note 1) 4-23
Figure 5–1: Overview of DSP Block Signals 5-3
Figure 5–2: Basic Two-Multiplier Adder Building Block 5-3
Figure 5–3: Four-Multiplier Adder and Accumulation Capability 5-5
Figure 5–4: Output Cascading Feature for FIR Structures 5-6
Figure 5–5: Stratix III Full DSP Block Summary 5-7
Figure 5–6: Half-DSP Block Architecture 5-9
Figure 5–7: Input Register of Half-DSP Block 5-11
Figure 5–8: 18-Bit Independent Multiplier Mode for Half-DSP Block 5-15
Figure 5–9: 12-Bit Independent Multiplier Mode for Half-DSP Block 5-16
Figure 5–10: 9-Bit Independent Multiplier Mode for Half-Block 5-17
Figure 5–11: 36-Bit Independent Multiplier Mode for Half-DSP Block 5-18
Figure 5–12: Double Mode for Half-DSP Block 5-19
Figure 5–13: Unsigned 54 × 54 Multiplier 5-20
Figure 5–14: Two-Multiplier Adder Mode for Half-DSP Block 5-22
Figure 5–15: Loopback Mode for Half-DSP Block 5-23
Figure 5–16: Complex Multiplier Using Two-Multiplier Adder Mode 5-24
Figure 5–17: Four-Multiplier Adder Mode for Half-DSP Block 5-25
Figure 5–18: Four-Multiplier Adder Mode for Half-DSP Block 5-27
Figure 5–19: Multiply Accumulate Mode for Half-DSP Block 5-28
Figure 5–20: Shift Operation Mode for Half-DSP Block 5-30
Figure 5–21: Round and Saturation Locations 5-33
Figure 5–22: FIR Filter Using Tap-Delay Line Input and Tree Summation of Final Result 5-37
Figure 5–23: FIR Filter using Tap-Delay Line Input and Chained Cascade Summation of Final Result 5-38
Figure 5–24: Semi-Parallel FIR Structure Using Chained Cascaded Summation 5-40
Figure 5–25: Radix-4 Butterfly 5-41
Figure 6–1: Global Clock Networks 6-2
Figure 6–2: Regional Clock Networks (EP3SL50, EP3SL70, and EP3SE50 Devices) 6-3
Figure 6–3: Regional Clock Networks (EP3SL110, EP3SL150, EP3SE80, and EP3SE110 Devices) 6-3
Figure 6–4: Regional Clock Networks (EP3SL200, EP3SE260, and EP3SL340 Devices) (Note 1) 6-4
Figure 6–5: Periphery Clock Networks (EP3SL50, EP3SL70, and EP3SE50 Devices) 6-4
Figure 6–6: Periphery Clock Networks (EP3SL110, EP3SL150, EP3SE80, and EP3SE110 Devices) 6-5
Figure 6–7: Periphery Clock Networks (EP3SL200 Devices) 6-5
Figure 6–8: Periphery Clock Networks (EP3SE260 Devices) 6-6
Figure 6–9: Periphery Clock Networks (EP3SL340 Devices) 6-6
Figure 6–10: Stratix III Dual-Regional Clock Region 6-7
Figure 6–11: Clock Input Multiplexer Logic for L2, L3, T1, T2, B1, B2, R2, and R3 PLLs 6-14
Figure 6–12: Clock Input Multiplexer Logic for L1, L4, R1, and R4 PLLs 6-14
Figure 6–13: Stratix III Global Clock Control Block 6-15
Figure 6–14: Regional Clock Control Block 6-15
Figure 6–15: Stratix III External PLL Output Clock Control Block 6-16
Figure 6–16: clkena Implementation 6-17
Figure 6–17: clkena Signals 6-17
Figure 6–18: Stratix III PLL Locations 6-20
Figure 6–19: Stratix III PLL Block Diagram 6-21
Figure 6–20: External Clock Outputs for Top/Bottom PLLs 6-22
Figure 6–21: External Clock Outputs for Left/Right PLLs 6-23
Figure 6–22: Stratix III PLL Ports 6-24

Figure 6–23: Phase Relationship Between Clock and Data in Source-Synchronous Mode 6-27
Figure 6–24: Phase Relationship Between Clock and Data LVDS Modes 6-27
Figure 6–25: Phase Relationship Between PLL Clocks in No Compensation Mode 6-28
Figure 6–26: Phase Relationship Between PLL Clocks in Normal Mode 6-29
Figure 6–27: Zero-Delay Buffer Mode in Stratix III PLLs 6-29
Figure 6–28: Phase Relationship Between PLL Clocks in Zero Delay Buffer Mode 6-30
Figure 6–29: External Feedback Mode in Stratix III Devices 6-30
Figure 6–30: Phase Relationship Between PLL Clocks in External-Feedback Mode 6-31
Figure 6–31: Counter Cascading 6-32
Figure 6–32: Automatic Clock Switchover Circuit Block Diagram 6-34
Figure 6–33: Automatic Switchover Upon Loss of Clock Detection 6-35
Figure 6–34: Clock Switchover Using the clkswitch (Manual) Control (Note 1) 6-36
Figure 6–35: Manual Clock Switchover Circuitry in Stratix III PLLs 6-37
Figure 6–36: VCO Switchover Operating Frequency 6-38
Figure 6–37: Open- and Closed-Loop Response Bode Plots 6-39
Figure 6–38: Loop Filter Programmable Components 6-40
Figure 6–39: Delay Insertion Using VCO Phase Output and Counter Delay Time 6-42
Figure 6–40: PLL Reconfiguration Scan Chain 6-43
Figure 6–41: PLL Reconfiguration Waveform 6-45
Figure 6–42: Scan-Chain Order of PLL Components for Top/Bottom PLLs (Note 1) 6-47
Figure 6–43: Scan-Chain Bit-Order Sequence for Post-Scale Counters in Stratix III PLLs 6-47
Figure 6–44: Dynamic Phase Shifting Waveform 6-51
Figure 7–1: I/O Banks for Stratix III Devices (Note 1), (2), (3), (4), (5), (6), (7), (8), (9) 7-6
Figure 7–2: Number of I/Os in Each Bank in EP3SL50, EP3SL70, and EP3SE50 Devices in 484-Pin FineLine BGA Package (Note 1), (2), ... 7-8
Figure 7–3: Number of I/Os in Each Bank in the 780-pin FineLine BGA Package (Note 1), (2), (3), (4) . 7-9
Figure 7–4: Number of I/Os in Each Bank in the 1152-pin FineLine BGA Package (Note 1), (2), (3), (4) . . . 7-10
Figure 7–5: Number of I/Os in Each Bank in EP2SL200, EP3SE260, and EP3SL340 Devices in the 1517-Pin FineLine BGA Package (Note 1), (2) 7-11
Figure 7–6: Number of I/Os in Each Bank in EP3SL340 Devices in the 1760-pin FineLine BGA Package (Note 1), (2) 7-12
Figure 7–7: IOE Structure for Stratix III Devices (Note 1), (2) 7-13
Figure 7–8: On-Chip Series Termination without Calibration for Stratix III Devices 7-21
Figure 7–9: On-Chip Series Termination with Calibration for Stratix III Devices 7-21
Figure 7–10: On-Chip Parallel Termination with Calibration for Stratix III Devices 7-24
Figure 7–11: Dynamic Parallel OCT in Stratix III Devices 7-25
Figure 7–12: Differential Input On-Chip Termination 7-26
Figure 7–13: OCT Calibration Block (CB) Location in EP3SL50, EP3SL70, and EP3SE50 Devices (Note 1) . 7-27
Figure 7–14: OCT Calibration Block (CB) Location in EP3SL110, EP3SL150, EP3SE80, and EP3SE110 Devices (Note 1) 7-28
Figure 7–15: OCT Calibration Block (CB) Location in EP3SL200, EP3SE260 and EP3SL340 (Note 1) . . 7-28
Figure 7–16: Example of Sharing Multiple I/O Banks with One OCT Calibration Block (Note 1) 7-29
Figure 7–17: Signals Used for User Mode Calibration (Note 1) 7-30
Figure 7–18: OCT User-Mode Signal Timing Waveform for One OCT Block 7-31
Figure 7–19: OCT User-Mode Signal Timing Waveform for Two OCT Blocks 7-32
Figure 7–20: SSTL I/O Standard Termination for Stratix III Devices 7-33
Figure 7–21: HSTL I/O Standard Termination for Stratix III Devices 7-34
Figure 7–22: Differential SSTL I/O Standard Termination for Stratix III Devices 7-35

Figure 7–23: Differential HSTL I/O Standard Termination for Stratix III Devices 7-35
Figure 7–24: LVDS I/O Standard Termination for Stratix III Devices (Note 1) 7-36
Figure 7–25: LVPECL AC Coupled Termination (Note 1) 7-37
Figure 7–26: LVPECL DC Coupled Termination (Note 1) 7-37
Figure 7–27: RSDS I/O Standard Termination for Stratix III Devices (Note 1), (2) 7-38
Figure 7–28: Mini-LVDS I/O Standard Termination for Stratix III Devices (Note 1), (2) 7-39
Figure 8–1: Package Bottom View for Stratix III Devices (Note 1), (2) 8-2
Figure 8–2: External Memory Interface Data Path Overview (Note 1), (2), (3) 8-3
Figure 8–3: Number of DQS/DQ Groups per Bank in EP3SE50, EP3SL50, and EP3SL70 Devices in the 484-pin FineLine BGA Package (Note 1) 8-8
Figure 8–4: Number of DQS/DQ Groups per Bank in EP3SE50, EP3SL50, EP3SL70, EP3SE80, EP3SE110, EP3SL110, EP3SL150, EP3SL200, and EP3SE260 Devices in the 780-pin FineLine BGA Package (Note 1) . 8-9
Figure 8–5: Number of DQS/DQ Groups in EP3SE80, EP3SE110, EP3SL110, EP3SL150, EP3SL200, EP3SE260, and EP3SL340 Devices in the 1152-pin FineLine BGA Package (Note 1) 8-10
Figure 8–6: Number of DQS/DQ Groups per Bank in EP3SL200, EP3SE260 and EP3SL340 Devices in the 1517-pin FineLine BGA Package 8-11
Figure 8–7: DQS/DQ Bus Mode Support per Bank in EP3SL340 Devices in the 1760-pin FineLine BGA Package 8-12
Figure 8–8: DQS Pins in Stratix III I/O Banks 8-14
Figure 8–9: Memory Clock Generation Block Diagram (Note 1) 8-18
Figure 8–10: DQS and CQn Pins and DQS Phase-Shift Circuitry (Note 1) 8-20
Figure 8–11: Stratix III DLL and I/O Bank Locations (Package Bottom View) 8-22
Figure 8–12: Simplified Diagram of the DQS Phase Shift Circuitry (Note 1) 8-26
Figure 8–13: Stratix III DQS Logic Block 8-29
Figure 8–14: Example of a DQS Update Enable Waveform 8-30
Figure 8–15: Avoiding a Glitch on a Non-Consecutive Read Burst Waveform 8-31
Figure 8–16: DDR3 SDRAM Unbuffered Module Clock Topology 8-32
Figure 8–17: Stratix III Write Leveling Delay Chains and Multiplexers (Note 1) 8-32
Figure 8–18: Stratix III Read Leveling Delay Chains and Multiplexers (Note 1) 8-33
Figure 8–19: Stratix III Dynamic OCT Control Block 8-34
Figure 8–20: Stratix III IOE Input Registers (Note 1) 8-35
Figure 8–21: Stratix III IOE Output and Output-Enable Path Registers (Note 1) 8-38
Figure 8–22: Delay Chain 8-39
Figure 8–23: Delay Chains in an I/O Block 8-40
Figure 8–24: Delay Chains in the DQS Input Path 8-40
Figure 8–25: I/O Configuration Block and DQS Configuration Block 8-41
Figure 9–1: I/O Banks in Stratix III Devices (Note 1), (2), (3), (4), (5), (6) 9-2
Figure 9–2: Transmitter Block Diagram for Stratix III Devices 9-5
Figure 9–3: Transmitter in Clock Output Mode for Stratix III Devices 9-5
Figure 9–4: Serializer Bypass for Stratix III Devices 9-6
Figure 9–5: Receiver Block Diagram for Stratix III Devices 9-7
Figure 9–6: Deserializer Bypass for Stratix III Devices 9-7
Figure 9–7: Data Realignment Timing 9-8
Figure 9–8: DPA Clock Phase-to-Serial Data Timing Relationship 9-8
Figure 9–9: Soft-CDR Data and Clock Path for a Stratix III Devices 9-10
Figure 9–10: Programmable V_{OD} 9-11
Figure 9–11: On-Chip Differential I/O Termination for Stratix III Devices 9-12
Figure 9–12: PLL Block Diagram for Stratix III Devices 9-13
Figure 9–13: LVDS/DPA Clocks with Center PLLs for Stratix III Devices 9-14

Figure 9–14: LVDS/DPA Clocks with Center and Corner PLLs for Stratix III Devices 9-14
Figure 9–15: Bit Orientation in Quartus II Software 9-15
Figure 9–16: Bit-Order and Word Boundary for One Differential Channel (Note 1) 9-16
Figure 9–17: Differential High-Speed Timing Diagram and Timing Budget for Non-DPA 9-18
Figure 9–18: Corner and Center Left/Right PLLs Driving DPA-Enabled Differential I/Os in the Same Bank 9-20
Figure 9–19: Center Left/Right PLLs Driving DPA-Enabled Differential I/Os 9-21
Figure 9–20: Invalid Placement of DPA-Enabled Differential I/Os Driven by Both Center Left/Right PLLs 9-22
Figure 9–21: Corner and Center Left/Right PLLs Driving DPA-Disabled Differential I/Os in the Same Bank 9-24
Figure 9–22: Invalid Placement of DPA-Disabled Differential I/Os Due to Interleaving of Channels Driven by the Corner and Center Left/Right PLLs 9-25
Figure 9–23: Both Center Left/Right PLLs Driving Cross-Bank DPA-Disabled Channels Simultaneously . 9-26
Figure 10–1: Hot-Socketing Circuitry for Stratix III Devices 10-3
Figure 10–2: Transistor Level Diagram of a Stratix III Device I/O Buffers 10-4
Figure 10–3: Simplified POR Block Diagram 10-5
Figure 11–1: Enabling Compression for Stratix III Bitstreams in Compiler Settings 11-5
Figure 11–2: Compressed and Uncompressed Configuration Data in the Same Configuration File . . . 11-6
Figure 11–3: Single Device FPP Configuration Using an External Host 11-9
Figure 11–4: Multi-Device FPP Configuration Using an External Host 11-12
Figure 11–5: Multiple-Device FPP Configuration Using an External Host When Both Devices Receive the Same Data 11-13
Figure 11–6: FPP Configuration Timing Waveform (Note 1), (2) 11-14
Figure 11–7: FPP Configuration Timing Waveform with Decompression or Design Security Feature Enabled (Note 1), (2) 11-15
Figure 11–8: Single Device Fast AS Configuration 11-18
Figure 11–9: Multi-Device Fast AS Configuration 11-21
Figure 11–10: Multi-Device Fast AS Configuration When the Devices Receive the Same Data Using a Single SOF 11-23
Figure 11–11: Fast AS Configuration Timing 11-24
Figure 11–12: In-System Programming of Serial Configuration Devices 11-26
Figure 11–13: Single Device PS Configuration Using an External Host 11-27
Figure 11–14: Multi-Device PS Configuration Using an External Host 11-30
Figure 11–15: Multiple-Device PS Configuration When Both Devices Receive the Same Data 11-31
Figure 11–16: PS Configuration Timing Waveform (Note 1) 11-32
Figure 11–17: PS Configuration Using a Download Cable 11-35
Figure 11–18: Multi-Device PS Configuration using a Download Cable 11-36
Figure 11–19: JTAG Configuration of a Single Device Using a Download Cable 11-39
Figure 11–20: JTAG Configuration of Multiple Devices Using a Download Cable 11-41
Figure 11–21: JTAG Configuration of a Single Device Using a Microprocessor 11-42
Figure 12–1: Functional Diagram of Stratix III Remote System Upgrade 12-2
Figure 12–2: Remote System Upgrade Block Diagram for Stratix III Fast AS Configuration Scheme . . 12-2
Figure 12–3: Enabling Remote Update for Stratix III Devices in Compiler Settings 12-4
Figure 12–4: Transitions Between Configurations in Remote Update Mode 12-6
Figure 12–5: Remote System Upgrade Circuit Data Path (Note 1) 12-8
Figure 12–6: Remote System Upgrade Control Register 12-9
Figure 12–7: Remote System Upgrade Status Register 12-10
Figure 12–8: Interface Signals Between the ALTREMOTE_UPDATE Megafunction and the Nios II Proces-

sor 12-13
Figure 13–1: IEEE Std. 1149.1 Boundary-Scan Testing 13-1
Figure 13–2: IEEE Std. 1149.1 Circuitry 13-3
Figure 13–3: Boundary-Scan Register 13-4
Figure 13–4: Stratix III Device's User I/O BSC with IEEE Std. 1149.1 BST Circuitry 13-5
Figure 13–5: IEEE Std. 1149.1 TAP Controller State Machine 13-8
Figure 13–6: IEEE Std. 1149.1 Timing Waveforms 13-9
Figure 13–7: Selecting the Instruction Mode 13-9
Figure 13–8: IEEE Std. 1149.1 BST SAMPLE/PRELOAD Mode 13-11
Figure 13–9: SAMPLE/PRELOAD Shift Data Register Waveforms 13-12
Figure 13–10: IEEE Std. 1149.1 BST EXTEST Mode 13-13
Figure 13–11: EXTEST Shift Data Register Waveforms 13-14
Figure 13–12: BYPASS Shift Data Register Waveforms 13-15
Figure 13–13: JTAG Chain of Mixed Voltages 13-18
Figure 14–1: Design Security (Note 1) 14-4
Figure 14–2: Stratix III Security Modes - Sequence and Restrictions 14-6
Figure 15–1: Error Detection Block Diagram 15-8
Figure 15–2: Enabling the Error Detection CRC Feature in the Quartus II Software 15-11
Figure 16–1: Stratix III Power Management Example (Note 1), (2) 16-5
Figure 16–2: TEMPDIODEP and TEMPDIODEN External Pin Connections 16-6
Figure 16–3: TSD Connections 16-6

List of Tables

Table 1–1: FPGA Family Features for Stratix III Devices 1-3
Table 1–2: Package Options and I/O Pin Counts (Note 1) 1-4
Table 1–3: FineLine BGA Package Sizes 1-4
Table 1–4: Hybrid FineLine BGA Package Sizes 1-5
Table 1–5: Speed Grades for Stratix III Devices 1-5
Table 1–6: Chapter Revision History 1-14
Table 2–1: Chapter Revision History 2-21
Table 3–1: Stratix III Device Routing Scheme 3-5
Table 3–2: Number of LABs reachable using C4 and R4 interconnects 3-6
Table 3–3: Chapter Revision History 3-14
Table 4–1: Summary of TriMatrix Memory Features 4-1
Table 4–2: TriMatrix Memory Capacity and Distribution in Stratix III Devices 4-3
Table 4–3: Truth Table for ECC Status Flags 4-9
Table 4–4: Stratix III Port Width Configurations for MLABs, M9K Blocks, and M144K Blocks (Single-Port Mode) 4-11
Table 4–5: Stratix III M9K Block Mixed-Width Configurations (Simple Dual-Port Mode) 4-13
Table 4–6: Stratix III M144K Block Mixed-Width Configurations (Simple Dual-Port Mode) 4-13
Table 4–7: Stratix III M9K Block Mixed-Width Configuration (True Dual-Port Mode) 4-15
Table 4–8: Stratix III M144K Block Mixed-Width Configurations (True Dual-Port Mode) 4-16
Table 4–9: Stratix III TriMatrix Memory Clock Modes 4-19
Table 4–10: Chapter Revision History 4-25
Table 5–1: Number of DSP Blocks in Stratix III Devices 5-2
Table 5–2: Stratix III DSP Block Operation Modes 5-8
Table 5–3: Input Register Modes 5-12
Table 5–4: Multiplier Sign Representation 5-12
Table 5–5: Examples of Shift Operations (Note 1) 5-31
Table 5–6: Example of Round-To-Nearest-Even Mode 5-31
Table 5–7: Comparison of Round-to-Nearest-Integer and Round-to-Nearest-Even 5-32
Table 5–8: Examples of Saturation 5-32
Table 5–10: DSP Block Dynamic Signals 5-34
Table 5–10: Document Revision History 5-42
Table 6–1: Clock Resources in Stratix III Devices 6-1
Table 6–2: Clock Input Pin Connectivity to Global Clock Networks 6-8
Table 6–3: Clock Input Pin Connectivity to Regional Clock Networks (Quadrant 1) 6-9
Table 6–4: Clock Input Pin Connectivity to Regional Clock Networks (Quadrant 2) 6-9
Table 6–5: Clock Input Pin Connectivity to Regional Clock Networks (Quadrant 3) 6-10
Table 6–6: Clock Input Pin Connectivity to Regional Clock Networks (Quadrant 4) 6-11
Table 6–7: Stratix III Device PLLs and PLL Clock Pin Drivers (Note 1) 6-11
Table 6–8: PLL Connectivity to GCLKs on Stratix III Devices (Note 1) 6-12
Table 6–9: Regional Clock Outputs From PLLs on Stratix III Devices (Note 1) 6-13
Table 6–10: Stratix III Device PLL Availability 6-18
Table 6–11: Stratix III PLL Features 6-19
Table 6–12: PLL Input Signals 6-24
Table 6–13: PLL Output Signals 6-25
Table 6–14: Clock Feedback Mode Availability 6-26
Table 6–15: Real-Time PLL Reconfiguration Ports 6-44
Table 6–16: Top/Bottom PLL Reprogramming Bits 6-46

Table 6–17: charge_pump_current Bit Settings 6-48
Table 6–18: loop_filter_r Bit Settings 6-48
Table 6–19: loop_filter_c Bit Settings 6-48
Table 6–20: PLL Counter Settings 6-49
Table 6–21: Dynamic Phase-Shifting Control Signals 6-49
Table 6–22: Phase Counter Select Mapping 6-50
Table 6–23: Chapter Revision History 6-53
Table 7–1: I/O Standard Applications for Stratix III Devices 7-2
Table 7–2: I/O Standards and Voltage Levels for Stratix III Devices (Note 1), (3) 7-3
Table 7–3: Bank Migration Path with Increasing Device Size (Note 1) 7-7
Table 7–4: Memory Interface Standards Supported 7-14
Table 7–5: Programmable Current Strength (Note 1) 7-16
Table 7–6: Default Programmable Slew Rate 7-17
Table 7–7: MultiVolt I/O Support for Stratix III Devices (Note 1), (2) 7-19
Table 7–8: Selectable I/O Standards with On-Chip Series Termination With or Without Calibration . 7-22
Table 7–9: Selectable I/O Standards with Expanded On-Chip Series Termination with Calibration Range 7-23
Table 7–10: Selectable I/O Standards that Support On-Chip Parallel Termination with Calibration . . 7-24
Table 7–11: On-Chip Differential Termination in Quartus II Software Assignment Editor 7-26
Table 7–12: OCT Calibration Block Ports for User Control and Description 7-30
Table 7–13: Chapter Revision History 7-42
Table 8–1: DQS and DQ Bus Mode Pins for Stratix III Devices 8-5
Table 8–2: Number of DQS/DQ Groups in Stratix III Devices per Side 8-6
Table 8–3: DQ/DQS Group in Stratix III Modular I/O Banks 8-15
Table 8–4: I/O Sub-Bank Combinations for Stratix III Devices that do not have ×36 Groups to form two ×36 Groups. 8-16
Table 8–5: DLL Location and Supported I/O Banks 8-23
Table 8–6: DLL Reference Clock Input for EP3SE50, EP3SL50, and EP3SL70 Devices 8-23
Table 8–8: DLL Reference Clock Input for EP3SE80, EP3SE110, EP3SL110, and EP3SL150 Devices in the 1152-pin Package 8-24
Table 8–7: DLL Reference Clock Input for EP3SE80, EP3SE110, and EP3SL150 Devices in the 780-pin Package 8-24
Table 8–9: DLL Reference Clock Input for EP3SL200, EP3SE260 and EP3SL340 Devices (Note 1), (2) . 8-25
Table 8–10: Stratix III DLL Frequency Modes 8-27
Table 8–11: I/O Configuration Block Bit Sequence 8-41
Table 8–12: DQS Configuration Block Bit Sequence 8-42
Table 8–13: Chapter Revision History 8-45
Table 9–1: LVDS Channels Supported in Stratix III Device Side I/O Banks (Note 1), (2), (3) 9-3
Table 9–2: LVDS Channels (Emulated) Supported in Stratix III Device Column I/O Banks (Note 1), (2) . . 9-4
Table 9–3: Differential Bit Naming 9-16
Table 9–4: Chapter Revision History 9-27
Table 10–1: Power Supplies Ramp-Up Time (t_{RAMP}) Requirements 10-5
Table 10–2: Power Supplies Monitored by the POR Circuitry 10-5
Table 10–3: Power Supplies That Are Not Monitored by the POR Circuitry 10-6
Table 10–4: Chapter Revision History 10-7
Table 11–1: Stratix III Configuration Schemes 11-2
Table 11–2: Stratix III Uncompressed Raw Binary File (**.rbf**) Sizes 11-2
Table 11–3: Stratix III Configuration Features 11-3
Table 11–4: Stratix III MSEL Pin Settings for FPP Configuration Schemes 11-8

Table 11–5: FPP Timing Parameters for Stratix III Devices (Note 1) 11-14
Table 11–6: FPP Timing Parameters for Stratix III Devices with Decompression or Design Security Feature Enabled (Note 1) 11-16
Table 11–7: Stratix III MSEL Pin Settings for AS Configuration Schemes (Note 1) 11-17
Table 11–8: Fast AS Timing Parameters for Stratix III Devices 11-24
Table 11–9: Stratix III MSEL Pin Settings for PS Configuration Scheme 11-27
Table 11–10: PS Timing Parameters for Stratix III Devices 11-32
Table 11–11: Dedicated JTAG Pins 11-38
Table 11–12: Dedicated Configuration Pin Connections During JTAG Configuration 11-40
Table 11–13: Stratix III Configuration Pin Summary (Note 1) 11-43
Table 11–14: Dedicated Configuration Pins on the Stratix III Device 11-44
Table 11–15: Optional Configuration Pins 11-49
Table 11–16: Dedicated JTAG Pins 11-50
Table 11–17: Chapter Revision History 11-51
Table 12–1: Stratix III Remote System Upgrade Modes 12-3
Table 12–2: Remote System Upgrade Registers 12-8
Table 12–3: Remote System Upgrade Control Register Contents 12-9
Table 12–4: Remote System Upgrade Status Register Contents 12-10
Table 12–5: Control Register Contents After an Error or Reconfiguration Trigger Condition 12-11
Table 12–6: 10-MHz Internal Oscillator Specifications 12-12
Table 12–7: Chapter Revision History 12-14
Table 13–1: IEEE Std. 1149.1 Pin Descriptions 13-2
Table 13–2: Stratix III Boundary-Scan Register Length 13-4
Table 13–3: Stratix III Device Boundary Scan Cell Descriptions (Note 1) 13-6
Table 13–4: Stratix III JTAG Instructions 13-7
Table 13–5: 32-Bit Stratix III Device IDCODE 13-16
Table 13–6: Supported TDO/TDI Voltage Combinations 13-17
Table 13–7: Disabling IEEE Std. 1149.1 Circuitry 13-19
Table 13–8: Chapter Revision History 13-21
Table 14–1: Security Keys Options 14-2
Table 14–2: Key Retention Time of Coin-Cell Type Batteries used for Volatile Key Storage 14-3
Table 14–3: Security Modes Supported 14-5
Table 14–4: Allowed Configuration Modes for Various Security Modes (Note 1) 14-6
Table 14–5: Design Security Configuration Schemes Availability 14-7
Table 14–6: Chapter Revision History 14-8
Table 15–1: EDERROR_INJECT JTAG Instruction 15-4
Table 15–2: Fault Injection Register and Error Injection 15-5
Table 15–3: CRC_ERROR Pin Description 15-6
Table 15–4: Error Detection Registers 15-8
Table 15–5: Minimum and Maximum Error Detection Frequencies 15-9
Table 15–6: Minimum Update Interval for Error Message Register (Note 1) 15-10
Table 15–7: CRC Calculation Time 15-10
Table 15–8: Chapter Revision History 15-12
Table 16–1: Stratix III Programmable Power Capabilities 16-3
Table 16–2: Stratix III Power Supply Requirements 16-4
Table 16–3: Chapter Revision History 16-7
Table 17–1: FineLine and Hybrid FineLine BGA Packages for Stratix III Devices 17-1
Table 17–2: Chapter Revision History 17-2

This handbook provides comprehensive information about the Altera® Stratix® III family of devices.

How to Contact Altera

For the most up-to-date information about Altera products, see the following table.

Contact *(Note 1)*	Contact Method	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
	Email	custrain@altera.com
Product literature	Website	www.altera.com/literature
Non-technical support (General)	Email	nacomp@altera.com
(Software Licensing)	Email	authorization@altera.com

Note:

(1) You can also contact your local Altera sales office or sales representative.

Typographic Conventions

The following table shows the typographic conventions that this document uses.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Command names, dialog box titles, checkbox options, and dialog box options are shown in bold, initial capital letters. Example: **Save As** dialog box.
bold type	External timing parameters, directory names, project names, disk drive names, file names, file name extensions, and software utility names are shown in bold type. Examples: $\mathbf{f_{MAX}}$, **\qdesigns** directory, **d:** drive, **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Document titles are shown in italic type with initial capital letters. Example: *AN 75: High-Speed Board Design.*
Italic type	Internal timing parameters and variables are shown in italic type. Examples: t_{PIA}, $n + 1$. Variable names are enclosed in angle brackets (< >) and shown in italic type. Example: *<file name>*, *<project name>***.pof** file.
Initial Capital Letters	Keyboard keys and menu names are shown with initial capital letters. Examples: Delete key, the Options menu.
“Subheading Title”	References to sections within a document and titles of on-line help topics are shown in quotation marks. Example: “Typographic Conventions.”

Visual Cue	Meaning
`Courier type`	Signal and port names are shown in lowercase Courier type. Examples: `data1`, `tdi`, `input`. Active-low signals are denoted by suffix `n`, e.g., `resetn`. Anything that must be typed exactly as it appears is shown in Courier type. For example: `c:\qdesigns\tutorial\chiptrip.gdf`. Also, sections of an actual file, such as a Report File, references to parts of files (e.g., the AHDL keyword `SUBDESIGN`), as well as logic function names (e.g., `TRI`) are shown in Courier.
1., 2., 3., and a., b., c., etc.	Numbered steps are used in a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ■	Bullets are used in a list of items when the sequence of the items is not important.
✓	The checkmark indicates a procedure that consists of one step only.
☞	The hand points to information that requires special attention.
CAUTION	A caution calls attention to a condition or possible situation that can damage or destroy the product or the user's work.
WARNING	A warning calls attention to a condition or possible situation that can cause injury to the user.
↵	The angled arrow indicates you should press the Enter key.
	The feet direct you to more information on a particular topic.

Section I. Device Core

This section provides a complete overview of all features relating to the Stratix® III device family, which is the most architecturally advanced, high performance, low power FPGA in the market place. This section includes the following chapters:

- Chapter 1, Stratix III Device Family Overview
- Chapter 2, Logic Array Blocks and Adaptive Logic Modules in Stratix III Devices
- Chapter 3, MultiTrack Interconnect in Stratix III Devices
- Chapter 4, TriMatrix Embedded Memory Blocks in Stratix III Devices
- Chapter 5, DSP Blocks in Stratix III Devices
- Chapter 6, Clock Networks and PLLs in Stratix III Devices

Revision History

Refer to each chapter for its own specific revision history. For information on when each chapter was updated, refer to the Chapter Revision Dates section, which appears in the full handbook.

1. Stratix III Device Family Overview

SIII51001-1.8

The Stratix® III family provides one of the most architecturally advanced, high-performance, low-power FPGAs in the marketplace.

Stratix III FPGAs lower power consumption through Altera's innovative Programmable Power Technology, which provides the ability to turn on the performance where needed and turn down the power consumption for blocks not in use. Selectable Core Voltage and the latest in silicon process optimizations are also employed to deliver the industry's lowest power, high-performance FPGAs.

Specifically designed for ease of use and rapid system integration, the Stratix III FPGA family offers two variants optimized to meet different application needs:

- The Stratix III *L* family provides balanced logic, memory, and multiplier ratios for mainstream applications.
- The Stratix III *E* family is memory- and multiplier-rich for data-centric applications.

Modular I/O banks with a common bank structure for vertical migration lend efficiency and flexibility to the high-speed I/O. Package and die enhancements with dynamic on-chip termination, output delay, and current strength control provide best-in-class signal integrity.

Based on a 1.1-V, 65-nm all-layer copper SRAM process, the Stratix III family is a programmable alternative to custom ASICs and programmable processors for high-performance logic, digital signal processing (DSP), and embedded designs.

Stratix III devices include optional configuration bit stream security through volatile or non-volatile 256-bit Advanced Encryption Standard (AES) encryption. Where ultra-high reliability is required, Stratix III devices include automatic error detection circuitry to detect data corruption by soft errors in the configuration random-access memory (CRAM) and user memory cells.

Features Summary

Stratix III devices offer the following features:

- 48,000 to 338,000 equivalent logic elements (LEs) (refer to Table 1–1)
- 2,430 to 20,497 Kbits of enhanced TriMatrix memory consisting of three RAM block sizes to implement true dual-port memory and FIFO buffers
- High-speed DSP blocks provide dedicated implementation of 9×9, 12×12, 18×18, and 36×36 multipliers (at up to 550 MHz), multiply-accumulate functions, and finite impulse response (FIR) filters
- I/O:GND:PWR ratio of 8:1:1 along with on-die and on-package decoupling for robust signal integrity
- Programmable Power Technology, which minimizes power while maximizing device performance

- Selectable Core Voltage, available in low-voltage devices (L ordering code suffix), enables selection of lowest power or highest performance operation
- Up to 16 global clocks, 88 regional clocks, and 116 peripheral clocks per device
- Up to 12 phase-locked loops (PLLs) per device that support PLL reconfiguration, clock switchover, programmable bandwidth, clock synthesis, and dynamic phase shifting
- Memory interface support with dedicated DQS logic on all I/O banks
- Support for high-speed external memory interfaces including DDR, DDR2, DDR3 SDRAM, RLDRAM II, QDR II, and QDR II+ SRAM on up to 24 modular I/O banks
- Up to 1,104 user I/O pins arranged in 24 modular I/O banks that support a wide range of industry I/O standards
- Dynamic On-Chip Termination (OCT) with auto calibration support on all I/O banks
- High-speed differential I/O support with serializer/deserializer (SERDES) and dynamic phase alignment (DPA) circuitry for 1.6 Gbps performance
- Support for high-speed networking and communications bus standards including SPI-4.2, SFI-4, SGMII, Utopia IV, 10 Gigabit Ethernet XSBI, Rapid I/O, and NPSI
- The only high-density, high-performance FPGA with support for 256-bit AES volatile and non-volatile security key to protect designs
- Robust on-chip hot socketing and power sequencing support
- Integrated cyclical redundancy check (CRC) for configuration memory error detection with critical error determination for high availability systems support
- Built-in error correction coding (ECC) circuitry to detect and correct data errors in M144K TriMatrix memory blocks
- Nios® II embedded processor support
- Support for multiple intellectual property megafunctions from Altera® MegaCore® functions and Altera Megafunction Partners Program (AMPP℠)

Table 1–1 lists the Stratix III FPGA family features.

Table 1–1. FPGA Family Features for Stratix III Devices

	Device/ Feature	ALMs	LEs	M9K Blocks	M144K Blocks	MLAB Blocks	Total Embedded RAM Kbits	MLAB RAM Kbits *(1)*	Total RAM Kbits *(2)*	18×18-bit Multipliers (FIR Mode)	PLLs *(3)*
Stratix III Logic Family	EP3SL50	19K	47.5K	108	6	950	1,836	297	2,133	216	4
	EP3SL70	27K	67.5K	150	6	1,350	2,214	422	2,636	288	4
	EP3SL110	43K	107.5K	275	12	2,150	4,203	672	4,875	288	8
	EP3SL150	57K	142.5K	355	16	2,850	5,499	891	6,390	384	8
	EP3SL200	80K	200K	468	36	4,000	9,396	1,250	10,646	576	12
	EP3SL340	135K	337.5K	1,040	48	6,750	16,272	2,109	18,381	576	12
Stratix III Enhanced Family	EP3SE50	19K	47.5K	400	12	950	5,328	297	5,625	384	4
	EP3SE80	32K	80K	495	12	1,600	6,183	500	6,683	672	8
	EP3SE110	43K	107.5K	639	16	2,150	8,055	672	8,727	896	8
	EP3SE260	102K	255K	864	48	5,100	14,688	1,594	16,282	768	12

Notes to Table 1–1:

(1) MLAB ROM mode supports twice the number of MLAB RAM Kbits.

(2) For total ROM Kbits, use this equation to calculate:
Total ROM Kbits = Total Embedded RAM Kbits + [(# of MLAB blocks × 640)/1024]

(3) The availability of the PLLs shown in this column is based on the device with the largest package. Refer to the *Clock Networks and PLLs in Stratix III Devices* chapter in volume 1 of the *Stratix III Device Handbook* for the availability of the PLLs for each device.

The Stratix III logic family (*L*) offers balanced logic, memory, and multipliers to address a wide range of applications, while the enhanced family (*E*) offers more memory and multipliers per logic and is ideal for wireless, medical imaging, and military applications.

Stratix III devices are available in space-saving FineLine BGA (FBGA) packages (refer to Table 1–2 and Table 1–3).

Table 1–2 lists the Stratix III FPGA package options and I/O pin counts.

Table 1–2. Package Options and I/O Pin Counts *(Note 1)*

Device	484-Pin FineLine BGA *(2)*	780-Pin FineLine BGA *(2)*	1152-Pin FineLine BGA *(2)*	1517-Pin FineLine BGA *(3)*	1760-Pin FineLine BGA *(3)*
EP3SL50	296	488	—	—	—
EP3SL70	296	488	—	—	—
EP3SL110	—	488	744	—	—
EP3SL150	—	488	744	—	—
EP3SL200	—	488 *(5)*	744	976	—
EP3SL340	—	—	744 *(4)*	976	1,120
EP3SE50	296	488	—	—	—
EP3SE80	—	488	744	—	—
EP3SE110	—	488	744	—	—
EP3SE260	—	488 *(5)*	744	976	—

Notes to Table 1–2:

(1) The arrows indicate vertical migration.

(2) All I/O pin counts include eight dedicated clock inputs (`CLK1p`, `CLK1n`, `CLK3p`, `CLK3n`, `CLK8p`, `CLK8n`, `CLK10p`, and `CLK10n`) that can be used for data inputs.

(3) All I/O pin counts include eight dedicated clock inputs (`CLK1p`, `CLK1n`, `CLK3p`, `CLK3n`, `CLK8p`, `CLK8n`, `CLK10p`, and `CLK10n`) and eight dedicated corner PLL clock inputs (`PLL_L1_CLKp`, `PLL_L1_CLKn`, `PLL_L4_CLKp`, `PLL_L4_CLKn`, `PLL_R4_CLKp`, `PLL_R4_CLKn`, `PLL_R1_CLKp`, and `PLL_R1_CLKn`) that can be used for data inputs.

(4) The EP3SL340 FPGA is offered only in the H1152 package, but not offered in the F1152 package.

(5) The EP3SE260 and EP3SL200 FPGAs are offered only in the H780 package, but not offered in the F780 package.

All Stratix III devices support vertical migration within the same package (for example, you can migrate between the EP3SL50 and EP3SL70 devices in the 780-pin FineLine BGA package). Vertical migration allows you to migrate to devices whose dedicated pins, configuration pins, and power pins are the same for a given package across device densities.

To ensure that a board layout supports migratable densities within one package offering, enable the applicable vertical migration path within the Quartus® II software. On the Assignments menu, point to **Device** and click **Migration Devices**. You can migrate from the *L* family to the *E* family without increasing the number of LEs available. This minimizes the cost of vertical migration.

Table 1–3 lists the Stratix III FineLine BGA (FBGA) package sizes.

Table 1–3. FineLine BGA Package Sizes

Dimension	484 Pin	780 Pin	1152 Pin	1517 Pin	1760 Pin
Pitch (mm)	1.00	1.00	1.00	1.00	1.00
Area (mm^2)	529	841	1,225	1,600	1,849
Length/Width (mm/ mm)	23/23	29/29	35/35	40/40	43/43

Table 1–4 lists the Stratix III Hybrid FineLine BGA (HBGA) package sizes.

Table 1–4. Hybrid FineLine BGA Package Sizes

Dimension	780 Pin	1152 Pin
Pitch (mm)	1.00	1.00
Area (mm^2)	1,089	1,600
Length/Width (mm/ mm)	33/33	40/40

Stratix III devices are available in up to three speed grades: –2, –3, and –4, with –2 being the fastest. Stratix III devices are offered in both commercial and industrial temperature range ratings with leaded and lead-free packages. Selectable Core Voltage is available in specially marked low-voltage devices (*L* ordering code suffix).

Table 1–5 lists the Stratix III device speed grades.

Table 1–5. Speed Grades for Stratix III Devices (Part 1 of 2)

Device	Temperature Grade	484 -Pin FineLine BGA	780-Pin FineLine BGA	780-Pin Hybrid FineLine BGA	1152-Pin FineLine BGA	1152-Pin Hybrid FineLine BGA	1517-Pin FineLine BGA	1760-Pin FineLine BGA
EP3SL50	Commercial	–2, –3, –4, –4L	–2, –3,–4, –4L	—	—	—	—	—
	Industrial	–3, –4, –4L	–3, –4, –4L	—	—	—	—	—
EP3SL70	Commercial	–2, –3, –4, –4L	–2, –3, –4, –4L	—	—	—	—	—
	Industrial	–3, –4, –4L	–3, –4, –4L	—	—	—	—	—
EP3SL110	Commercial	—	–2, –3, –4, –4L	—	–2, –3, –4, –4L	—	—	—
	Industrial	—	–3, –4, –4L	—	–3, –4, –4L	—	—	—
EP3SL150	Commercial	—	–2,–3, –4, –4L	—	–2, –3, –4, –4L	—	—	—
	Industrial	—	–3, –4, –4L	—	–3, –4, –4L	—	—	—
EP3SL200	Commercial	—	—	–2,–3, –4, –4L	–2,–3, –4, –4L	—	–2,–3, –4, –4L	—
	Industrial *(1)*	—	—	–3, –4, –4L	–3, –4, –4L	—	–3, –4, –4L	—
EP3SL340	Commercial	—	—	—	—	–2, –3, –4	–2, –3, –4	–2, –3, –4
	Industrial *(1)*	—	—	—	—	–3, –4, –4L	–3, –4, –4L	–3, –4, –4L
EP3SE50	Commercial	–2, –3, –4, –4L	–2, –3, –4, –4L	—	—	—	—	—
	Industrial	–3, –4, –4L	–3, –4, –4L	—	—	—	—	—
EP3SE80	Commercial	—	–2, –3, –4, –4L	—	–2, –3, –4, –4L	—	—	—
	Industrial	—	–3, –4, –4L	—	–3, –4, –4L	—	—	—
EP3SE110	Commercial	—	–2,–3, –4, –4L	—	–2, –3, –4, –4L	—	—	—
	Industrial	—	–3, –4, –4L	—	–3, –4, –4L	—	—	—

Table 1–5. Speed Grades for Stratix III Devices (Part 2 of 2)

Device	Temperature Grade	484 -Pin FineLine BGA	780-Pin FineLine BGA	780-Pin Hybrid FineLine BGA	1152-Pin FineLine BGA	1152-Pin Hybrid FineLine BGA	1517-Pin FineLine BGA	1760-Pin FineLine BGA
EP3SE260	Commercial	—	—	–2, –3, –4, –4L	–2,– 3, –4, –4L	—	–2, –3, –4, –4L	—
	Industrial *(1)*	—	—	–3, –4, –4L	–3, –4, –4L	—	–3, –4,–4L	—

Note to Table 1–5:

(1) For EP3SL340, EP3SL200, and EP3SE260 devices, the industrial junction temperature range for –4L is 0–100°C, regardless of supply voltage.

Architecture Features

The following section describes the various features of the Stratix III family FPGAs.

Logic Array Blocks and Adaptive Logic Modules

The Logic Array Block (LAB) is composed of basic building blocks known as Adaptive Logic Modules (ALMs) that can be configured to implement logic, arithmetic, and register functions. Each LAB consists of ten ALMs, carry chains, shared arithmetic chains, LAB control signals, local interconnect, and register chain connection lines. ALMs are part of a unique, innovative logic structure that delivers faster performance, minimizes area, and reduces power consumption. ALMs expand the traditional 4-input look-up table architecture to 7 inputs, increasing performance by reducing LEs, logic levels, and associated routing. In addition, ALMs maximize DSP performance with dedicated functionality to efficiently implement adder trees and other complex arithmetic functions. The Quartus II Compiler places associated logic in an LAB or adjacent LABs, allowing the use of local, shared arithmetic chain, and register chain connections for performance and area efficiency.

The Stratix III LAB has a new derivative called Memory LAB (or MLAB), which adds SRAM memory capability to the LAB. MLAB is a superset of the LAB and includes all LAB features. MLABs support a maximum of 320 bits of simple dual-port Static Random Access Memory (SRAM). Each ALM in an MLAB can be configured as a 16×2 block, resulting in a configuration of 16×20 simple dual port SRAM block. MLAB and LAB blocks always co-exist as pairs in all Stratix III families, allowing up to 50% of the logic (LABs) to be traded for memory (MLABs).

For more information about LABs and ALMs, refer to the *Logic Array Blocks and Adaptive Logic Modules in Stratix III Devices* chapter.

For more information about MLAB modes, features and design considerations, refer to the *TriMatrix Embedded Memory Blocks in Stratix III Devices* chapter.

MultiTrack Interconnect

In the Stratix III architecture, connections between ALMs, TriMatrix memory, DSP blocks, and device I/O pins are provided by the MultiTrack interconnect structure with DirectDrive technology. The MultiTrack interconnect consists of continuous, performance-optimized row and column interconnects that span fixed distances. A routing structure with fixed length resources for all devices allows predictable and repeatable performance when migrating through different device densities. The MultiTrack interconnect provides 1-hop connection to 34 adjacent LABs, 2-hop connections to 96 adjacent LABs and 3-hop connections to 160 adjacent LABs.

DirectDrive technology is a deterministic routing technology that ensures identical routing resource usage for any function regardless of placement in the device. The MultiTrack interconnect and DirectDrive technology simplify the integration stage of block-based designing by eliminating the reoptimization cycles that typically follow design changes and additions. The Quartus II Compiler also automatically places critical design paths on faster interconnects to improve design performance.

For more information, refer to the *MultiTrack Interconnect in Stratix III Devices* chapter.

TriMatrix Embedded Memory Blocks

TriMatrix embedded memory blocks provide three different sizes of embedded SRAM to efficiently address the needs of Stratix III FPGA designs. TriMatrix memory includes the following blocks:

- 320-bit MLAB blocks optimized to implement filter delay lines, small FIFO buffers, and shift registers
- 9-Kbit M9K blocks that can be used for general purpose memory applications
- 144-Kbit M144K blocks that are ideal for processor code storage, packet and video frame buffering

Each embedded memory block can be independently configured to be a single- or dual-port RAM, ROM, or shift register via the Quartus II MegaWizard™ Plug-In Manager. Multiple blocks of the same type can also be stitched together to produce larger memories with minimal timing penalty. TriMatrix memory provides up to 16,272 Kbits of embedded SRAM at up to 600 MHz operation.

For more information about TriMatrix memory blocks, modes, features, and design considerations, refer to the *TriMatrix Embedded Memory Blocks in Stratix III Devices* chapter.

DSP Blocks

Stratix III devices have dedicated high-performance digital signal processing (DSP) blocks optimized for DSP applications requiring high data throughput. Stratix III devices provide you with the ability to implement various high-performance DSP functions easily. Complex systems such as WiMAX, 3GPP WCDMA, CDMA2000, voice over Internet Protocol (VoIP), H.264 video compression, and high-definition television (HDTV) require high-performance DSP blocks to process data. These system designs typically use DSP blocks to implement finite impulse response (FIR) filters, complex FIR filters, infinite impulse response (IIR) filters, fast Fourier transform (FFT) functions, and discrete cosine transform (DCT) functions.

Stratix III devices have up to 112 DSP blocks. The architectural highlights of the Stratix III DSP block are the following:

- High-performance, power optimized, fully pipelined multiplication operations
- Native support for 9-bit, 12-bit, 18-bit, and 36-bit word lengths
- Native support for 18-bit complex multiplications
- Efficient support for floating point arithmetic formats (24-bit for Single Precision and 53-bit for Double Precision)
- Signed and unsigned input support
- Built-in addition, subtraction, and accumulation units to efficiently combine multiplication results
- Cascading 18-bit input bus to form tap-delay lines
- Cascading 44-bit output bus to propagate output results from one block to the next block
- Rich and flexible arithmetic rounding and saturation units
- Efficient barrel shifter support
- Loopback capability to support adaptive filtering

DSP block multipliers can optionally feed an adder/subtractor or accumulator in the block depending on user configuration. This option saves ALM routing resources and increases performance, because all connections and blocks are inside the DSP block. Additionally, the DSP Block input registers can efficiently implement shift registers for FIR filter applications, and the Stratix III DSP blocks support rounding and saturation. The Quartus II software includes megafunctions that control the mode of operation of the DSP blocks based on user parameter settings.

For more information, refer to the *DSP Blocks in Stratix III Devices* chapter.

Clock Networks and PLLs

Stratix III devices provide dedicated Global Clock Networks (GCLKs), Regional Clock Networks (RCLKs), and Periphery Clock Networks (PCLKs). These clocks are organized into a hierarchical clock structure that provides up to 104 unique clock domains (16 GCLK + 88 RCLK) within the Stratix III device and allows for up to 38 (16 GCLK + 22 RCLK) unique GCLK/RCLK clock sources per device quadrant.

Stratix III devices deliver abundant PLL resources with up to 12 PLLs per device and up to 10 outputs per PLL. Every output can be independently programmed, creating a unique, customizable clock frequency. Inherent jitter filtration and fine granularity control over multiply, divide ratios, and dynamic phase-shift reconfiguration provide the high-performance precision required in today's high-speed applications. Stratix III PLLs are feature rich, supporting advanced capabilities such as clock switchover, reconfigurable phase shift, PLL reconfiguration, and reconfigurable bandwidth. PLLs can be used for general-purpose clock management supporting multiplication, phase shifting, and programmable duty cycle. Stratix III PLLs also support external feedback mode, spread-spectrum input clock tracking, and post-scale counter cascading.

For more information, refer to the *Clock Networks and PLLs in Stratix III Devices* chapter.

I/O Banks and I/O Structure

Stratix III devices contain up to 24 modular I/O banks, each of which contains 24, 32, 36, 40, or 48 I/Os. This modular bank structure improves pin efficiency and eases device migration. The I/O banks contain circuitry to support external memory interfaces at speeds up to 533 MHz and high-speed differential I/O interfaces meeting up to 1.6 Gbps performance. It also supports high-speed differential inputs and outputs running at speeds up to 800 MHz.

Stratix III devices support a wide range of industry I/O standards, including single-ended, voltage referenced single-ended, and differential I/O standards. The Stratix III I/O supports programmable bus hold, programmable pull-up resistor, programmable slew rate, programmable drive strength, programmable output delay control, and open-drain output. Stratix III devices also support on-chip series (R_S) and on-chip parallel (R_T) termination with auto calibration for single-ended I/O standards and on-chip differential termination (R_D) for LVDS I/O standards on Left/Right I/O banks. Dynamic OCT is also supported on bi-directional I/O pins in all I/O banks.

For more information, refer to the *Stratix III Device I/O Features* chapter.

External Memory Interfaces

The Stratix III I/O structure has been completely redesigned to provide flexibility and enable high-performance support for existing and emerging external memory standards such as DDR, DDR2, DDR3, QDR II, QDR II+, and RLDRAM II at frequencies of up to 533 MHz.

Packed with features such as dynamic on-chip termination, trace mismatch compensation, read/write leveling, half-rate registers, and 4-to 36-bit programmable DQ group widths, Stratix III I/Os supply the built-in functionality required for rapid and robust implementation of external memory interfaces. Double data-rate support is found on all sides of the Stratix III device. Stratix III devices provide an efficient architecture to quickly and easily fit wide external memory interfaces exactly where you want them.

A self-calibrating soft IP core (ALTMEMPHY), optimized to take advantage of the Stratix III device I/O, along with the Quartus II timing analysis tool (TimeQuest), provide the total solution for the highest reliable frequency of operation across process voltage and temperature.

For more information about external memory interfaces, refer to the *External Memory Interfaces in Stratix III Devices* chapter.

High-Speed Differential I/O Interfaces with DPA

Stratix III devices contain dedicated circuitry for supporting differential standards at speeds up to 1.6 Gbps. The high-speed differential I/O circuitry supports the following high-speed I/O interconnect standards and applications: Utopia IV, SPI-4.2, SFI-4, 10 Gigabit Ethernet XSBI, Rapid I/O, and NPSI. Stratix III devices support 2×, 4×, 6×, 7×, 8×, and 10× SERDES modes for high-speed differential I/O interfaces and

4×, 6×, 7×, 8×, and 10× SERDES modes when using the dedicated DPA circuitry. DPA minimizes bit errors, simplifies PCB layout and timing management for high-speed data transfer, and eliminates channel-to-channel and channel-to-clock skew in high-speed data transmission systems. Soft CDR can also be implemented, enabling low-cost 1.6-Gbps clock embedded serial links.

Stratix III devices have the following dedicated circuitry for high-speed differential I/O support:

- Differential I/O buffer
- Transmitter serializer
- Receiver deserializer
- Data realignment
- Dynamic phase aligner (DPA)
- Soft CDR functionality
- Synchronizer (FIFO buffer)
- PLLs

For more information, refer to the *High Speed Differential I/O Interfaces with DPA in Stratix III Devices* chapter.

Hot Socketing and Power-On Reset

Stratix III devices are hot-socketing compliant. Hot socketing is also known as hot plug-in or hot swap, and power sequencing support without the use of any external devices. Robust on-chip hot-socketing and power-sequencing support ensures proper device operation independent of the power-up sequence. You can insert or remove a Stratix III board in a system during system operation without causing undesirable effects to the running system bus or the board that was inserted into the system.

The hot-socketing feature makes it easier to use Stratix III devices on PCBs that also contain a mixture of 3.3-V, 3.0-V, 2.5-V, 1.8-V, 1.5-V, and 1.2-V devices. With the Stratix III hot socketing feature, you do not need to ensure a specific power-up sequence for each device on the board.

For more information, refer to the *Hot Socketing and Power-On Reset in Stratix III Devices* chapter.

Configuration

Stratix III devices are configured using one of the following four configuration schemes:

- Fast passive parallel (FPP)
- Fast active serial (AS)
- Passive serial (PS)
- Joint Test Action Group (JTAG)

All configuration schemes use either an external controller (for example, a MAX® II device or microprocessor), a configuration device, or a download cable.

Stratix III devices support configuration data decompression, which saves configuration memory space and time. This feature allows you to store compressed configuration data in configuration devices or other memory and transmit this compressed bitstream to Stratix III devices. During configuration, the Stratix III device decompresses the bitstream in real time and programs its SRAM cells.

Stratix III devices support decompression in the FPP when using a MAX II device/microprocessor plus flash, fast AS, and PS configuration schemes. The Stratix III decompression feature is not available in the FPP when using the enhanced configuration device and JTAG configuration schemes.

For more information, refer to the *Configuring Stratix III Devices* chapter.

Remote System Upgrades

Stratix III devices feature remote system upgrade capability, allowing error-free deployment of system upgrades from a remote location securely and reliably. Soft logic (either the Nios embedded processor or user logic) implemented in a Stratix III device can download a new configuration image from a remote location, store it in configuration memory, and direct the dedicated remote system upgrade circuitry to initiate a reconfiguration cycle. The dedicated circuitry performs error detection during and after the configuration process, and can recover from an error condition by reverting back to a safe configuration image, and provides error status information. This dedicated remote system upgrade circuitry is unique to Stratix series FPGAs and helps to avoid system downtime.

For more information, refer to the *Remote System Upgrades with Stratix III Devices* chapter.

IEEE 1149.1 (JTAG) Boundary-Scan Testing

Stratix III devices support the JTAG IEEE Std. 1149.1 specification. The Boundary-Scan Test (BST) architecture offers the capability to test pin connections without using physical test probes and capture functional data while a device is operating normally. Boundary-scan cells in the Stratix III device can force signals onto pins or capture data from pin or logic array signals. Forced test data is serially shifted into the boundary-scan cells. Captured data is serially shifted out and externally compared to expected results. In addition to BST, you can use the IEEE Std. 1149.1 controller for Stratix III device in-circuit reconfiguration (ICR).

For more information, refer to the *IEEE 1149.1 (JTAG) Boundary Scan Testing in Stratix III Devices* chapter.

Design Security

Stratix III devices are high-density, high-performance FPGAs with support for 256-bit volatile and non-volatile security keys to protect designs against copying, reverse engineering, and tampering. Stratix III devices have the ability to decrypt a configuration bitstream using the Advanced Encryption Standard (AES) algorithm, an industry standard encryption algorithm that is FIPS-197 certified and requires a 256-bit security key.

The design security feature is available when configuring Stratix III FPGAs using the fast passive parallel (FPP) configuration mode with an external host (such as a MAX II device or microprocessor), or when using fast active serial (AS) or passive serial (PS) configuration schemes.

For more information about the design security feature, refer to the *Design Security in Stratix III Devices* chapter.

SEU Mitigation

Stratix III devices have built-in error detection circuitry to detect data corruption due to soft errors in the configuration random-access memory (CRAM) cells. This feature allows all CRAM contents to be read and verified continuously during user mode operation to match a configuration-computed CRC value. The enhanced CRC circuit and frame-based configuration architecture allows detection and location of multiple, single, and adjacent bit errors which, in conjunction with a soft circuit supplied as a reference design, allows don't-care soft errors in the CRAM to be ignored during device operation. This provides a steep decrease in the effective soft error rate, increasing system reliability.

On-chip memory block SEU mitigation is also offered using the ninth bit and a configurable megafunction in the Quartus II software for MLAB and M9K blocks while the M144K memory blocks have built-in error correction code (ECC) circuitry.

For more information about the dedicated error detection circuitry, refer to the *SEU Mitigation in Stratix III Devices* chapter.

Programmable Power

Stratix III delivers Programmable Power, the only FPGA with user programmable power options balancing today's power and performance requirements. Stratix III devices utilize the most advanced power-saving techniques, including a variety of process, circuit, and architecture optimizations and innovations. In addition, user controllable power reduction techniques provide an optimal balance of performance and power reduction specific for each design configured into the Stratix III FPGA. The Quartus II software (starting from version 6.1) automatically optimizes designs to meet the performance goals while simultaneously leveraging the programmable power-saving options available in the Stratix III FPGA without the need for any changes to the design flow.

For more information about Programmable Power in Stratix III devices, refer to the following documents:

- *Programmable Power and Temperature Sensing Diode in Stratix III Devices* chapter
- *AN 437: Power Optimization in Stratix III FPGAs*
- *Stratix III Programmable Power White Paper*

Signal Integrity

Stratix III devices simplify the challenge of signal integrity through a number of chip, package, and board level enhancements to enable efficient high-speed data transfer into and out of the device. These enhancements include:

- 8:1:1 user I/O/Gnd/V_{CC} ratio to reduce the loop inductance in the package
- Dedicated power supply for each I/O bank, limit of I/Os is 24 to 48 I/Os per bank, to help limit simultaneous switching noise
- Programmable slew-rate support with up to four settings to match desired I/O standard, control noise, and overshoot
- Programmable output-current drive strength support with up to six settings to match desired I/O standard performance
- Programmable output-delay support to control rise/fall times and adjust duty cycle, compensate for skew, and reduce simultaneous switching outputs (SSO) noise
- Dynamic OCT with auto calibration support for series and parallel OCT and differential OCT support for LVDS I/O standard on the left/right banks

For more information about SI support in the Quartus II software, refer to the *Quartus II Handbook*.

For more information about how to use the various configuration, PLL, external memory interfaces, I/O, high-speed differential I/O, power, and JTAG pins, refer to the *Stratix III Device Family Pin Connection Guidelines*.

Reference and Ordering Information

The following section describes Stratix III device software support and ordering information.

Software Support

Stratix III devices are supported by the Altera Quartus II design software, version 6.1 and later, which provides a comprehensive environment for system-on-a-programmable-chip (SOPC) design. The Quartus II software includes HDL and schematic design entry, compilation and logic synthesis, full simulation and advanced timing analysis, SignalTap® II logic analyzer, and device configuration.

For more information about the Quartus II software features, refer to the *Quartus II Handbook*.

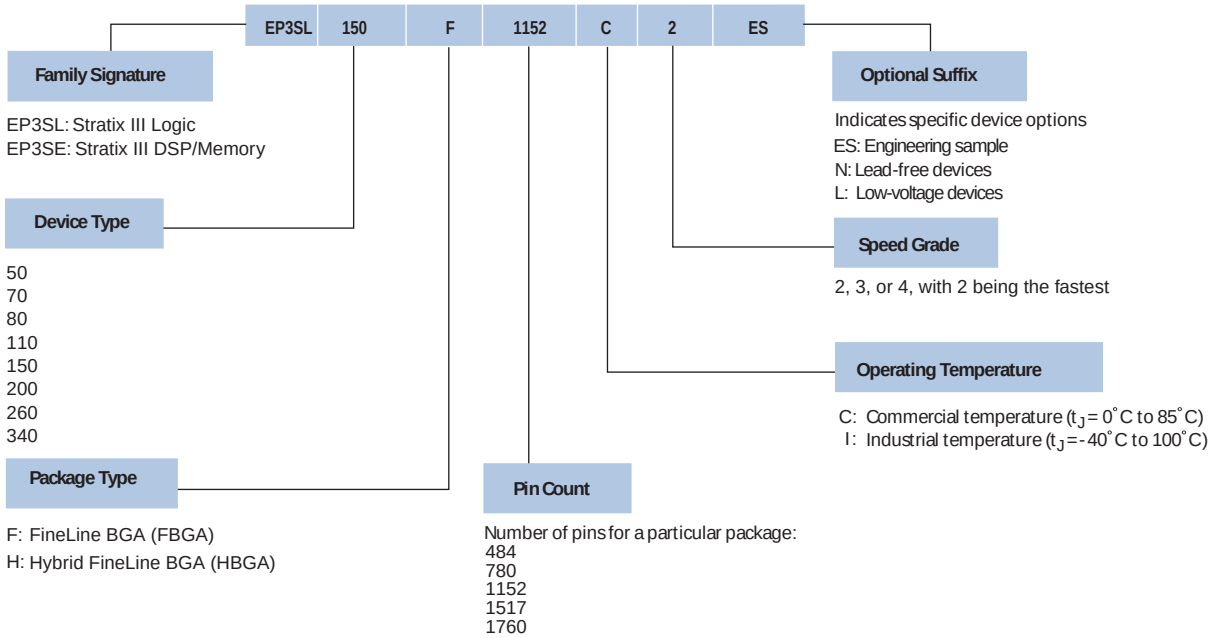
The Quartus II software supports a variety of operating systems. The specific operating system for the Quartus II software can be obtained from the Quartus II **Readme.txt** file or the *Operating System Support* section of the Altera website. It also supports seamless integration with industry-leading EDA tools through the NativeLink® interface.

Ordering Information

Figure 1–1 shows the ordering codes for Stratix III devices.

For more information about a specific package, refer to the *Stratix III Device Package Information* chapter.

Figure 1–1. Stratix III Device Packaging Ordering Information



Chapter Revision History

Table 1–6 lists the revision history for this chapter.

Table 1–6. Chapter Revision History (Part 1 of 2)

Date	Version	Changes Made
March 2010	1.8	Updated for the Quartus II software version 9.1 SP2 release: ■ Updated Table 1–2. ■ Updated "I/O Banks and I/O Structure" section.
May 2009	1.7	Updated "Software" and "Signal Integrity" sections.
February 2009	1.6	■ Updated "Features" section. ■ Updated Table 1–1. ■ Removed "Referenced Documents" section.
October 2008	1.5	■ Updated "Features" section. ■ Updated Table 1–1 and Table 1–5. ■ Updated New Document Format.

Table 1–6. Chapter Revision History (Part 2 of 2)

Date	Version	Changes Made
May 2008	1.4	■ Updated “Introduction”. ■ Updated Table 1–1. ■ Updated Table 1–2. ■ Added Table 1–5. ■ Updated “Reference and Ordering Information”. ■ Updated package type information in Figure 1–1.
November 2007	1.3	■ Updated Table 1–1. ■ Updated Table 1–2.
October 2007	1.2	■ Minor typo fixes. ■ Added Table 1–4. ■ Added section “Referenced Documents”. ■ Added live links for references.
May 2007	1.1	Minor formatting changes, fixed PLL numbers and ALM, LE and MLAB bit counts in Table 1–1.
November 2006	1.0	Initial Release.

2. Logic Array Blocks and Adaptive Logic Modules in Stratix III Devices

SIII51002-1.5

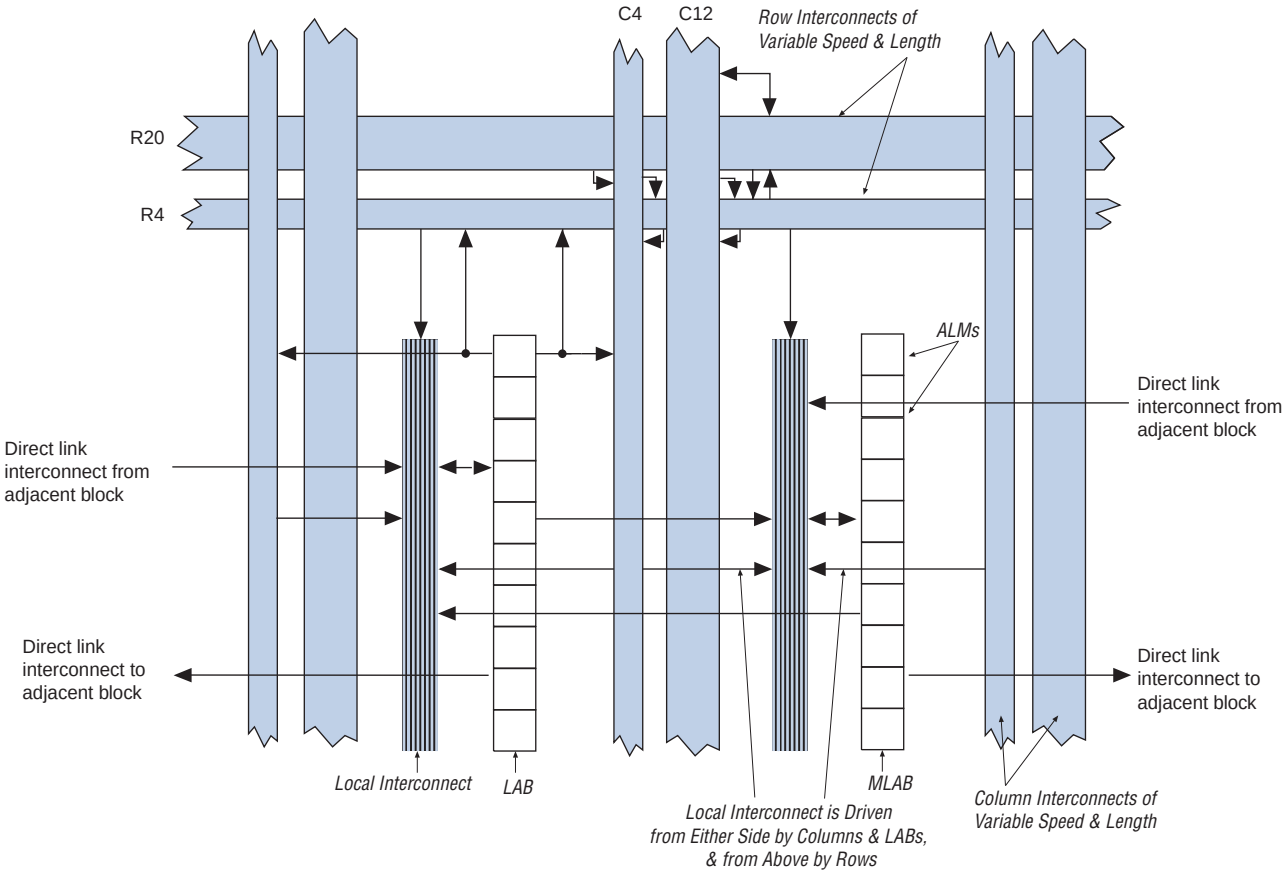
Introduction

This chapter describes the features of the logic array block (LAB) in the Stratix® III core fabric. The logic array block is composed of basic building blocks known as adaptive logic modules (ALMs) that can be configured to implement logic functions, arithmetic functions, and register functions.

Logic Array Blocks

Each LAB consists of ten ALMs, carry chains, shared arithmetic chains, LAB control signals, local interconnect, and register chain connection lines. The local interconnect transfers signals between ALMs in the same LAB. The direct link interconnect allows a LAB to drive into the local interconnect of its left and right neighbors. Register chain connections transfer the output of the ALM register to the adjacent ALM register in an LAB. The Quartus® II Compiler places associated logic in an LAB or adjacent LABs, allowing the use of local, shared arithmetic chain, and register chain connections for performance and area efficiency. Figure 2–1 shows the Stratix III LAB structure and the LAB interconnects.

Figure 2–1. Stratix III LAB Structure



The LAB of Stratix III has a new derivative called Memory LAB (MLAB), which adds look-up table (LUT)-based SRAM capability to the LAB as shown in Figure 2–2. The MLAB supports a maximum of 320-bits of simple dual-port static random access memory (SRAM). You can configure each ALM in an MLAB as a 16 × 2 block, resulting in a configuration of 16 × 20 simple dual port SRAM block. MLAB and LAB blocks always co-exist as pairs in all Stratix III families. MLAB is a superset of the LAB and includes all LAB features. Figure 2–2 shows an overview of LAB and MLAB topology.

The MLAB is described in detail in the *TriMatrix Embedded Memory Blocks in Stratix III Devices* chapter in volume 1 of the *Stratix III Device Handbook*.

Figure 2–2. Stratix III LAB and MLAB Structure

LUT-based-16 x 2 Simple dual port SRAM (1)	ALM
LUT-based-16 x 2 Simple dual port SRAM (1)	ALM
LUT-based-16 x 2 Simple dual port SRAM (1)	ALM
LUT-based-16 x 2 Simple dual port SRAM (1)	ALM
LUT-based-16 x 2 Simple dual port SRAM (1)	ALM
LAB Control Block	LAB Control Block
LUT-based-16 x 2 Simple dual port SRAM (1)	ALM
LUT-based-16 x 2 Simple dual port SRAM (1)	ALM
LUT-based-16 x 2 Simple dual port SRAM (1)	ALM
LUT-based-16 x 2 Simple dual port SRAM (1)	ALM
LUT-based-16 x 2 Simple dual port SRAM (1)	ALM
MLAB	**LAB**

Note to Figure 2–2:

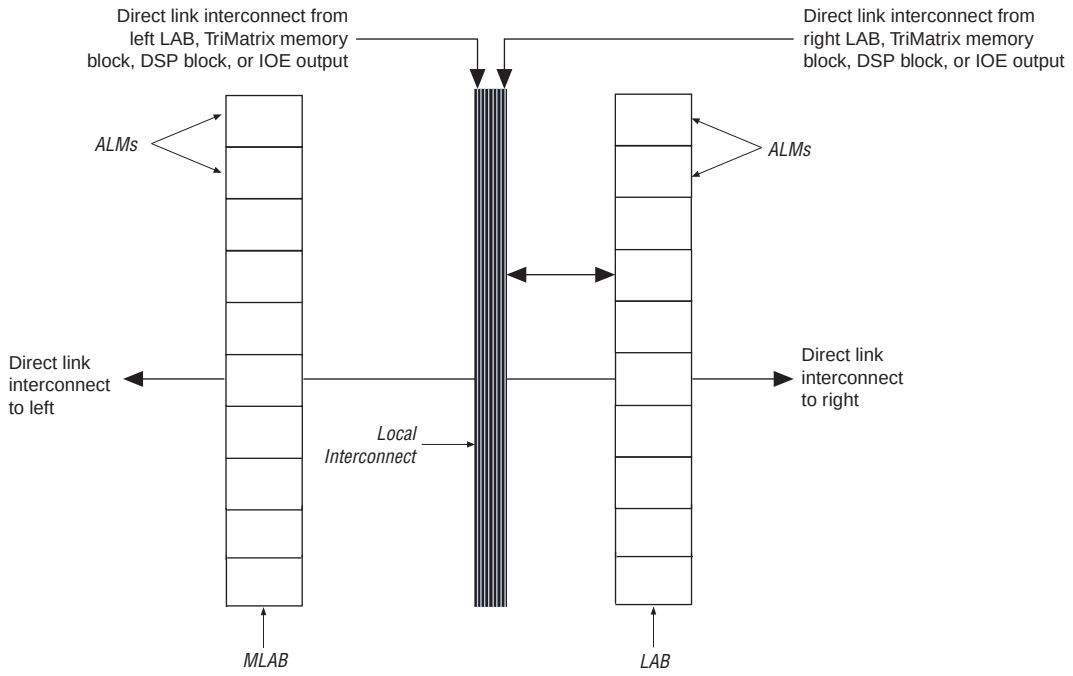
(1) You can use MLAB ALM as a regular LAB ALM or configure it as a dual-port SRAM, as shown.

LAB Interconnects

The LAB local interconnect can drive ALMs in the same LAB. It is driven by column and row interconnects and ALM outputs in the same LAB. Neighboring LABs/MLABs, M9K RAM blocks, M144K blocks, or DSP blocks from the left and right can also drive a LAB's local interconnect through the direct link connection. The direct link connection feature minimizes the use of row and column interconnects, providing higher performance and flexibility. Each ALM can drive 30 ALMs through fast local and direct link interconnects.

Figure 2–3shows the direct link connection.

Figure 2–3. Direct Link Connection



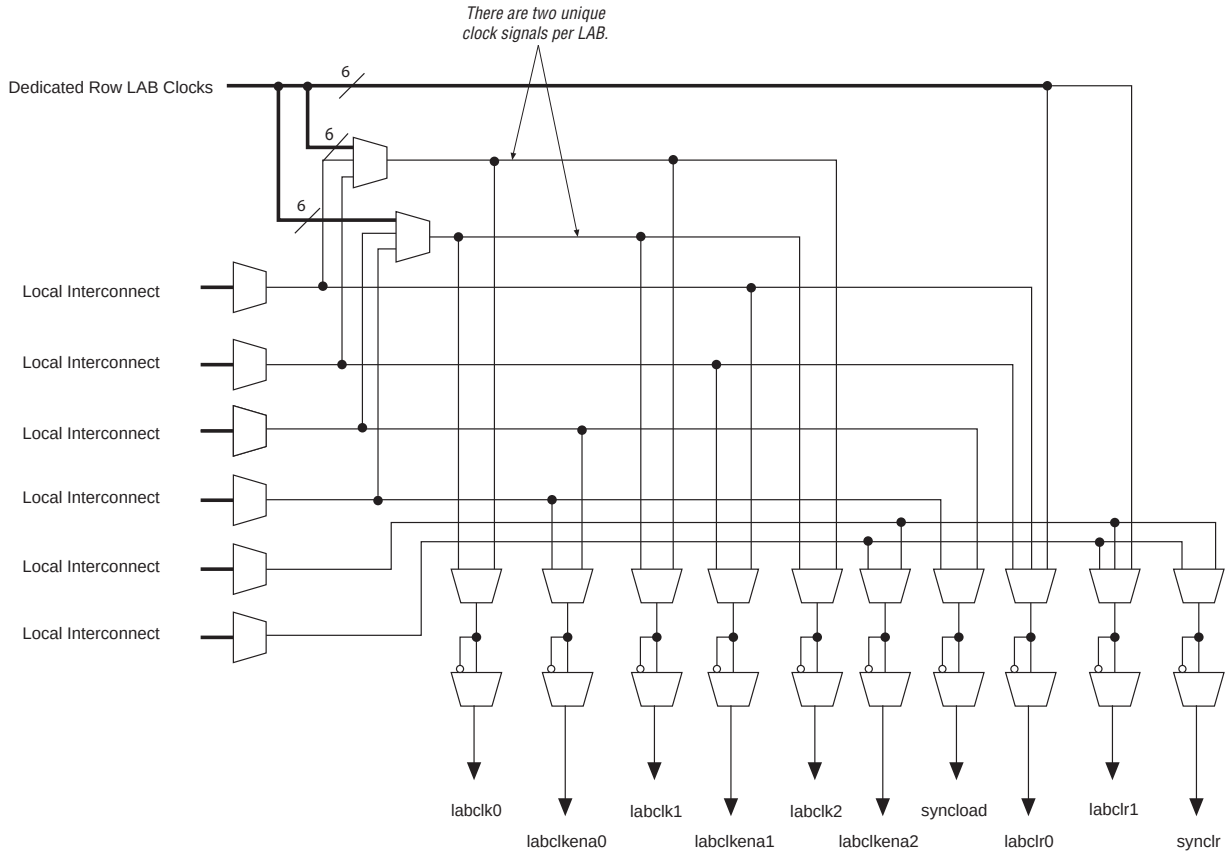
LAB Control Signals

Each LAB contains dedicated logic for driving control signals to its ALMs. The control signals include three clocks, three clock enables, two asynchronous clears, a synchronous clear, and synchronous load control signals. This gives a maximum of 10 control signals at a time. Although you generally use synchronous load and clear signals when implementing counters, you can also use them with other functions.

Each LAB has two unique clock sources and three clock enable signals, as shown in Figure 2–4. The LAB control block can generate up to three clocks using the two clock sources and three clock enable signals. Each LAB's clock and clock enable signals are linked. For example, any ALM in a particular LAB using the `labclk1` signal also uses `labclkena1` signal. If the LAB uses both the rising and falling edges of a clock, it also uses two LAB-wide clock signals. De-asserting the clock enable signal turns off the corresponding LAB-wide clock.

The LAB row clocks [5..0] and LAB local interconnect generate the LAB-wide control signals. The MultiTrack™ interconnect's inherent low skew allows clock and control signal distribution in addition to data. Figure 2–4shows the LAB control signal generation circuit.

Figure 2–4. LAB-Wide Control Signals



Adaptive Logic Modules

The basic building block of logic in the Stratix III architecture, the adaptive logic module (ALM), provides advanced features with efficient logic utilization. Each ALM contains a variety of look-up table (LUT)-based resources that can be divided between two combinational adaptive LUTs (ALUTs) and two registers. With up to eight inputs to the two combinational ALUTs, one ALM can implement various combinations of two functions. This adaptability allows an ALM to be completely backward-compatible with four-input LUT architectures. One ALM can also implement any function of up to six inputs and certain seven-input functions.

In addition to the adaptive LUT-based resources, each ALM contains two programmable registers, two dedicated full adders, a carry chain, a shared arithmetic chain, and a register chain. Through these dedicated resources, an ALM can efficiently implement various arithmetic functions and shift registers. Each ALM drives all types of interconnects: local, row, column, carry chain, shared arithmetic chain, register chain, and direct link interconnects. Figure 2–5 shows a high-level block diagram of the Stratix III ALM while Figure 2–6 shows a detailed view of all the connections in an ALM.

Figure 2–5. High-Level Block Diagram of the Stratix III ALM

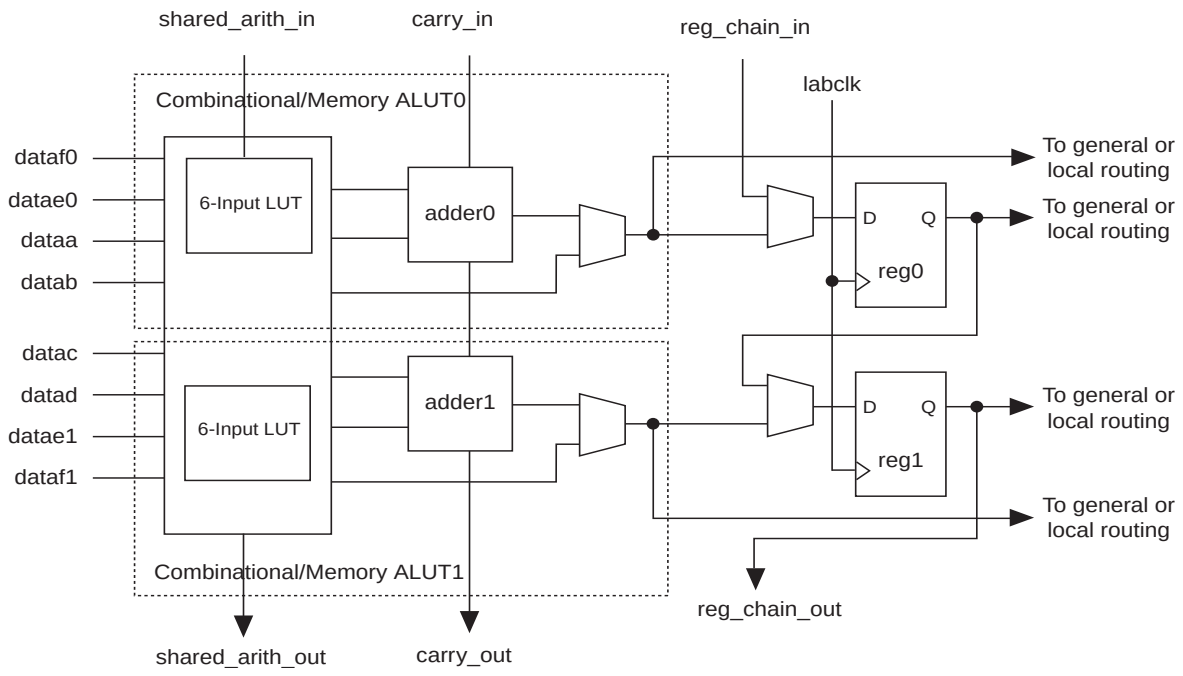
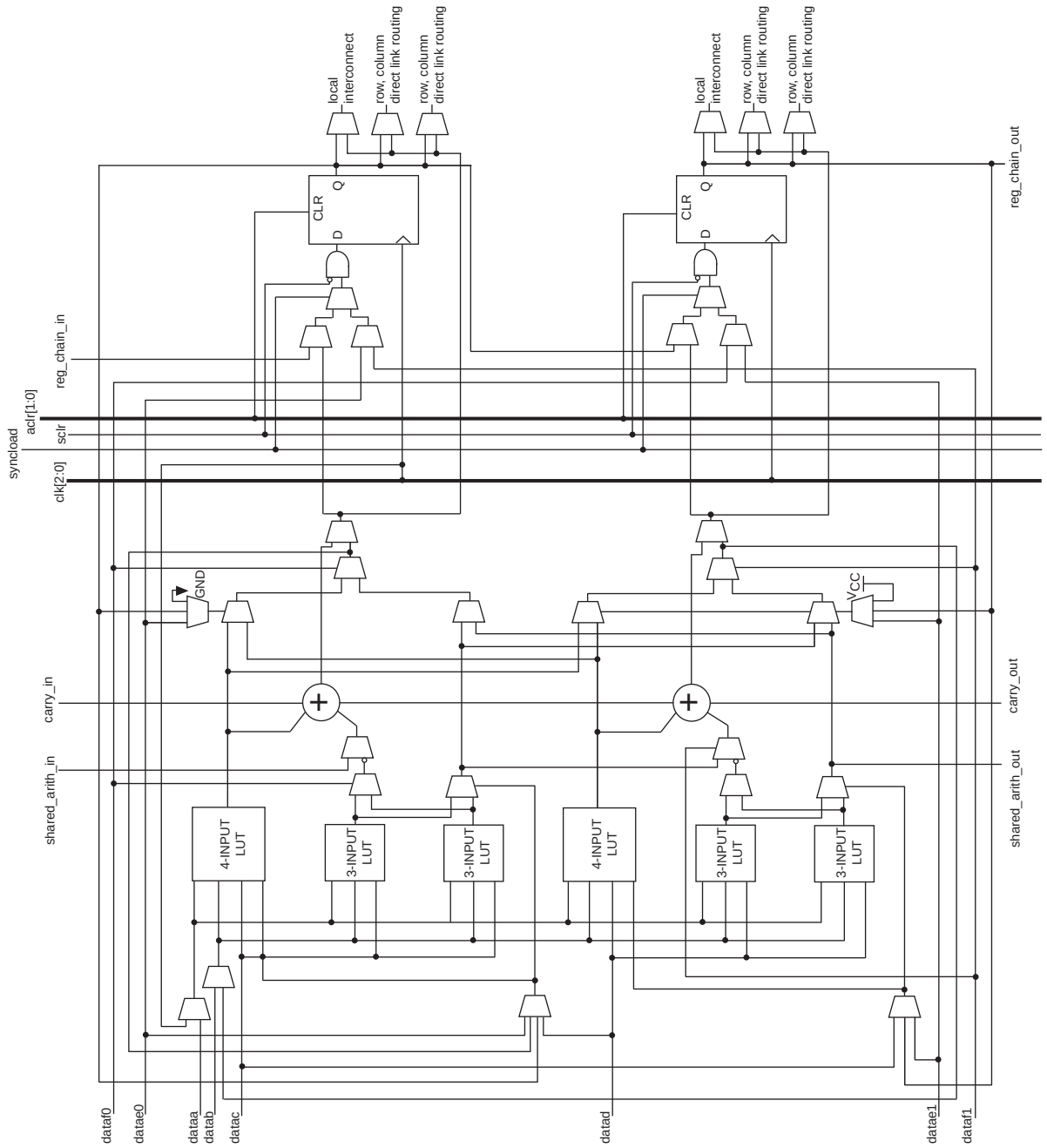


Figure 2–6. Stratix III ALM Details



One ALM contains two programmable registers. Each register has data, clock, clock enable, synchronous and asynchronous clear, and synchronous load/clear inputs. Global signals, general-purpose I/O pins, or any internal logic can drive the register's clock and clear control signals. Either general-purpose I/O pins or internal logic can drive the clock enable. For combinational functions, the register is bypassed and the output of the LUT drives directly to the outputs of an ALM.

Each ALM has two sets of outputs that drive the local, row, and column routing resources. The LUT, adder, or register output can drive these output drivers (refer to Figure 2–6). For each set of output drivers, two ALM outputs can drive column, row, or direct link routing connections, and one of these ALM outputs can also drive local interconnect resources. This allows the LUT or adder to drive one output while the register drives another output.

This feature, called register packing, improves device utilization because the device can use the register and the combinational logic for unrelated functions. Another special packing mode allows the register output to feed back into the LUT of the same ALM so that the register is packed with its own fan-out LUT. This provides another mechanism for improved fitting. The ALM can also drive out registered and unregistered versions of the LUT or adder output.

ALM Operating Modes

The Stratix III ALM can operate in one of the following modes:

- Normal
- Extended LUT Mode
- Arithmetic
- Shared Arithmetic
- LUT-Register

Each mode uses ALM resources differently. In each mode, eleven available inputs to an ALM—the eight data inputs from the LAB local interconnect, carry-in from the previous ALM or LAB, the shared arithmetic chain connection from the previous ALM or LAB, and the register chain connection—are directed to different destinations to implement the desired logic function. LAB-wide signals provide clock, asynchronous clear, synchronous clear, synchronous load, and clock enable control for the register. These LAB-wide signals are available in all ALM modes.

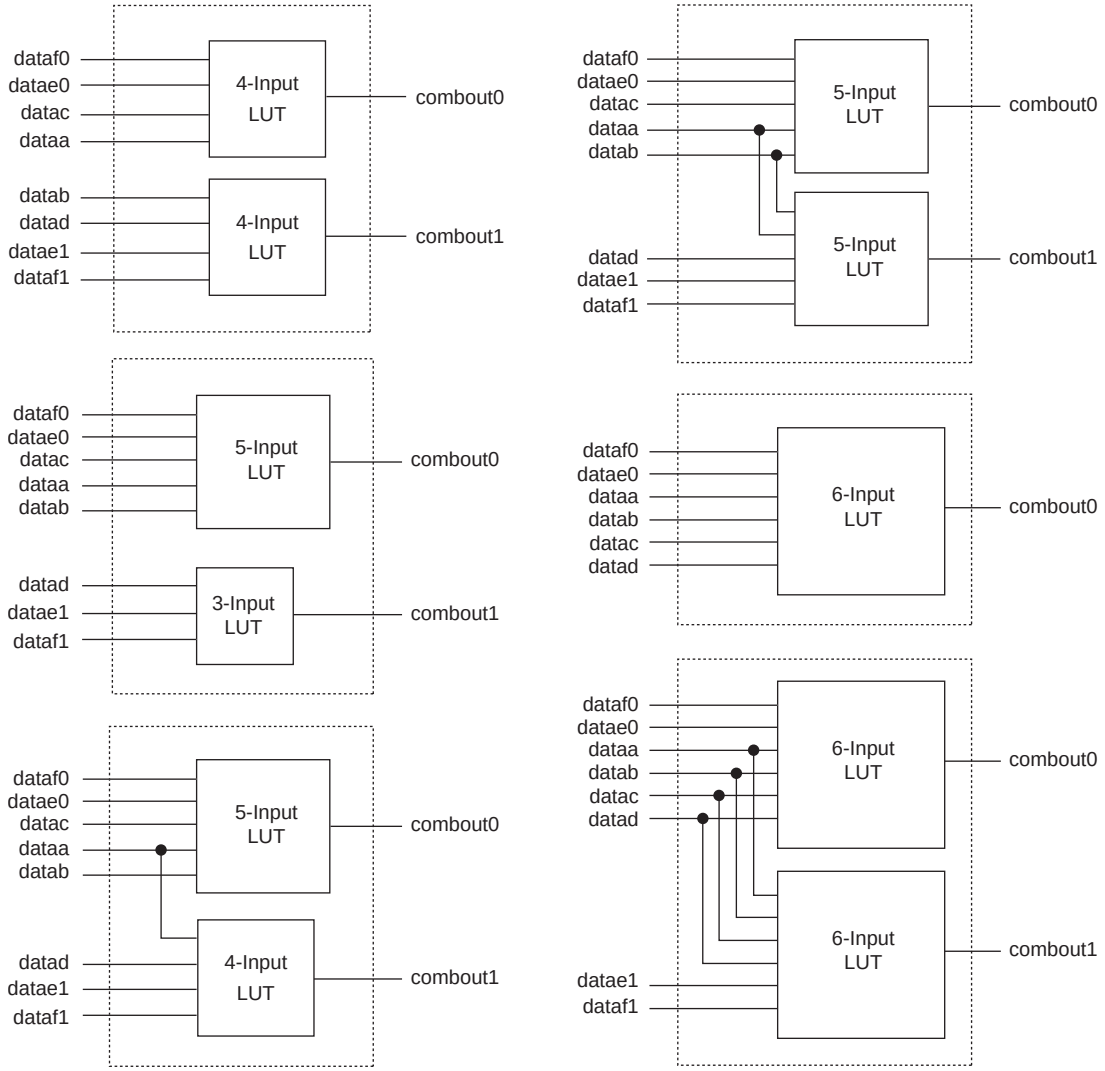
Refer to "LAB Control Signals" on page 2–4 for more information on the LAB-wide control signals.

The Quartus II software and supported third-party synthesis tools, in conjunction with parameterized functions such as the library of parameterized modules (LPM) functions, automatically choose the appropriate mode for common functions such as counters, adders, subtractors, and arithmetic functions.

Normal Mode

The normal mode is suitable for general logic applications and combinational functions. In this mode, up to eight data inputs from the LAB local interconnect are inputs to the combinational logic. The normal mode allows two functions to be implemented in one Stratix III ALM, or an ALM to implement a single function of up to six inputs. The ALM can support certain combinations of completely independent functions and various combinations of functions that have common inputs. Figure 2–7 shows the supported LUT combinations in normal mode.

Figure 2–7. ALM in Normal Mode *(Note 1)*



Note to Figure 2–7:

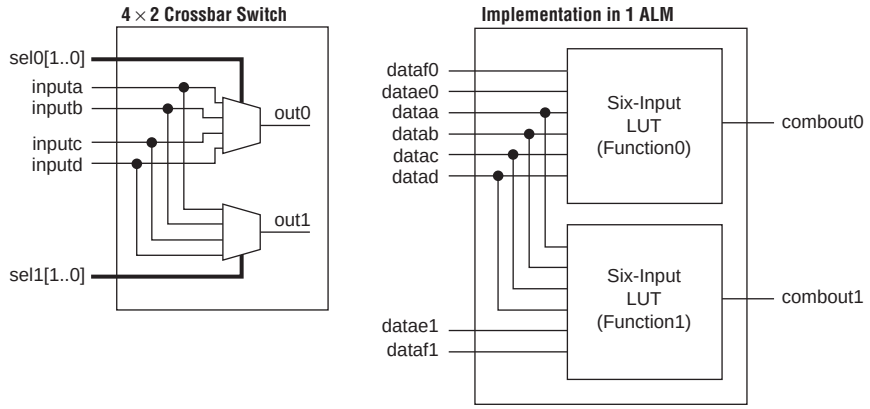
(1) Combinations of functions with fewer inputs than those shown are also supported. For example, combinations of functions with the following number of inputs are supported: 4 and 3, 3 and 3, 3 and 2, 5 and 2.

The normal mode provides complete backward compatibility with four-input LUT architectures.

For the packing of 2 five-input functions into one ALM, the functions must have at least two common inputs. The common inputs are `dataa` and `datab`. The combination of a four-input function with a five-input function requires one common input (either `dataa` or `datab`).

In the case of implementing 2 six-input functions in one ALM, four inputs must be shared and the combinational function must be the same. For example, a 4 × 2 crossbar switch (two 4-to-1 multiplexers with common inputs and unique select lines) can be implemented in one ALM, as shown in Figure 2–8. The shared inputs are `dataa`, `datab`, `datac`, and `datad`, while the unique select lines are `datae0` and `dataf0` for `function0`, and `datae1` and `dataf1` for `function1`. This crossbar switch consumes four LUTs in a four-input LUT-based architecture.

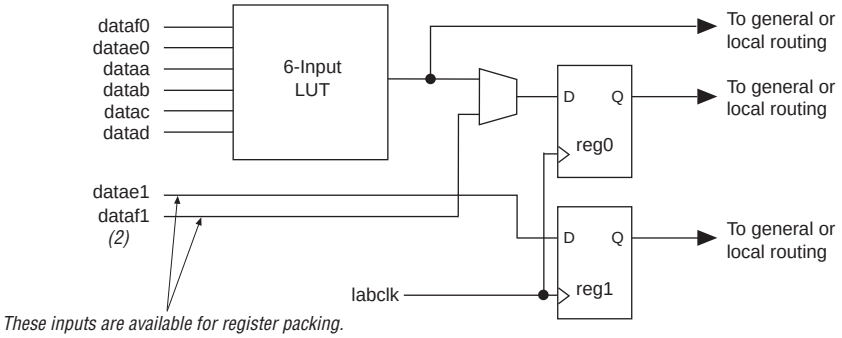
Figure 2–8. 4 × 2 Crossbar Switch Example



In a sparsely used device, functions that could be placed into one ALM may be implemented in separate ALMs by the Quartus II software in order to achieve the best possible performance. As a device begins to fill up, the Quartus II software automatically utilizes the full potential of the Stratix III ALM. The Quartus II Compiler automatically searches for functions of common inputs or completely independent functions to be placed into one ALM and to make efficient use of the device resources. In addition, you can manually control resource usage by setting location assignments.

Any six-input function can be implemented utilizing inputs `dataa`, `datab`, `datac`, `datad`, and either `datae0` and `dataf0` or `datae1` and `dataf1`. If `datae0` and `dataf0` are utilized, the output is driven to `register0`, and/or `register0` is bypassed and the data drives out to the interconnect using the top set of output drivers (refer to Figure 2–9). If `datae1` and `dataf1` are utilized, the output drives to `register1` and/or bypasses `register1` and drives to the interconnect using the bottom set of output drivers. The Quartus II Compiler automatically selects the inputs to the LUT. ALMs in normal mode support register packing.

Figure 2–9. Input Function in Normal Mode *(Note 1)*



Notes to Figure 2–9:

(1) If `datae1` and `dataf1` are used as inputs to the six-input function, then `datae0` and `dataf0` are available for register packing.

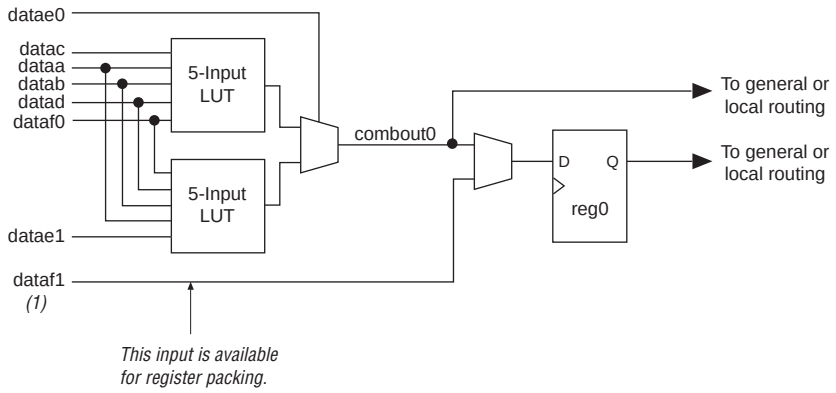
(2) The `dataf1` input is available for register packing only if the six-input function is un-registered.

Extended LUT Mode

Use the extended LUT mode to implement a specific set of seven-input functions. The set must be a 2-to-1 multiplexer fed by two arbitrary five-input functions sharing four inputs. Figure 2–10 shows the template of supported seven-input functions utilizing extended LUT mode. In this mode, if the seven-input function is unregistered, the unused eighth input is available for register packing.

Functions that fit into the template shown in Figure 2–10 occur naturally in designs. These functions often appear in designs as "if-else" statements in Verilog HDL or VHDL code.

Figure 2–10. Template for Supported Seven-Input Functions in Extended LUT Mode



Note to Figure 2–10:

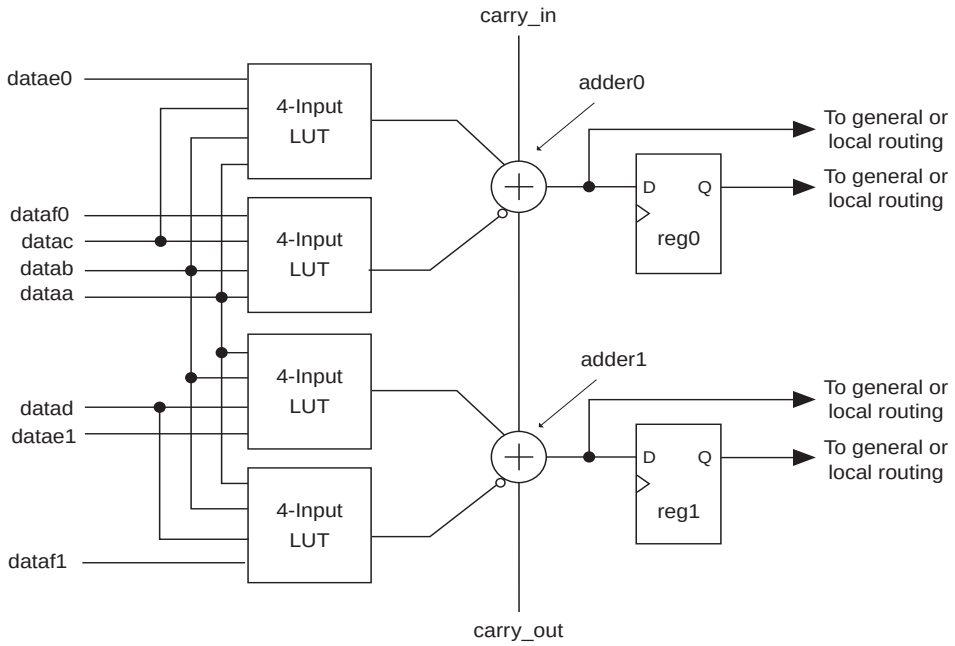
(1) If the seven-input function is unregistered, the unused eighth input is available for register packing. The second register, `reg1`, is not available.

Arithmetic Mode

The arithmetic mode is ideal for implementing adders, counters, accumulators, wide parity functions, and comparators. The ALM in arithmetic mode uses two sets of 2 four-input LUTs along with two dedicated full adders. The dedicated adders allow the LUTs to be available to perform pre-adder logic; therefore, each adder can add the output of 2 four-input functions.

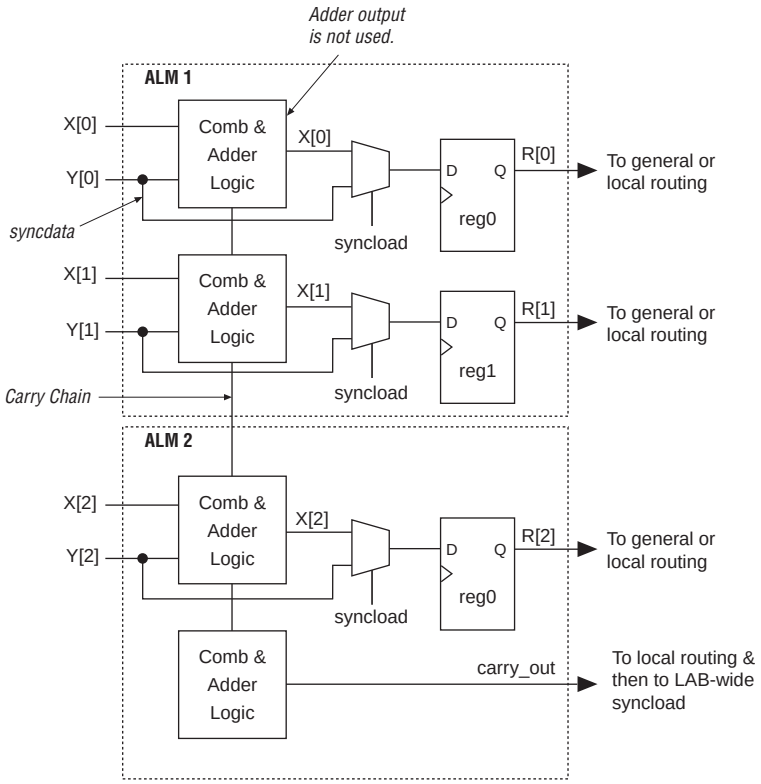
The four LUTs share the `dataa` and `datab` inputs. As shown in Figure 2–11, the carry-in signal feeds to `adder0`, and the carry-out from `adder0` feeds to carry-in of `adder1`. The carry-out from `adder1` drives to `adder0` of the next ALM in the LAB. ALMs in arithmetic mode can drive out registered and/or unregistered versions of the adder outputs.

Figure 2–11. ALM in Arithmetic Mode



While operating in arithmetic mode, the ALM can support simultaneous use of the adder's carry output along with combinational logic outputs. In this operation, the adder output is ignored. This usage of the adder with the combinational logic output provides resource savings of up to 50% for functions that can use this ability. An example of such functionality is a conditional operation, such as the one shown in Figure 2–12.

Figure 2–12. Conditional Operation Example



The equation for this example is:

$R = (X < Y)\ ?\ Y : X$

To implement this function, the adder is used to subtract *Y* from *X*. If *X* is less than *Y*, the `carry_out` signal is 1. The `carry_out` signal is fed to an adder where it drives out to the LAB local interconnect. It then feeds to the LAB-wide `syncload` signal. When asserted, `syncload` selects the `syncdata` input. In this case, the data *Y* drives the syncdata inputs to the registers. If *X* is greater than or equal to *Y*, the `syncload` signal is de-asserted and *X* drives the data port of the registers.

The arithmetic mode also offers clock enable, counter enable, synchronous up/down control, add/subtract control, synchronous clear, and synchronous load. The LAB local interconnect data inputs generate the clock enable, counter enable, synchronous up/down, and add/subtract control signals. These control signals are good candidates for the inputs that are shared between the four LUTs in the ALM. The synchronous clear and synchronous load options are LAB-wide signals that affect all registers in the LAB. These signals can also be individually disabled or enabled per register. The Quartus II software automatically places any registers that are not used by the counter into other LABs.

Carry Chain

The carry chain provides a fast carry function between the dedicated adders in arithmetic or shared arithmetic mode. The two-bit carry select feature in Stratix III devices halves the propagation delay of carry chains within the ALM. Carry chains can begin in either the first ALM or the sixth ALM in an LAB. The final carry-out signal is routed to a ALM, where it is fed to local, row, or column interconnects.

The Quartus II Compiler automatically creates carry chain logic during design processing, or you can create it manually during design entry. Parameterized functions such as LPM functions automatically take advantage of carry chains for the appropriate functions.

The Quartus II Compiler creates carry chains longer than 20 (10 ALMs in arithmetic or shared arithmetic mode) by linking LABs together automatically. For enhanced fitting, a long carry chain runs vertically allowing fast horizontal connections to TriMatrix™ memory and DSP blocks. A carry chain can continue as far as a full column.

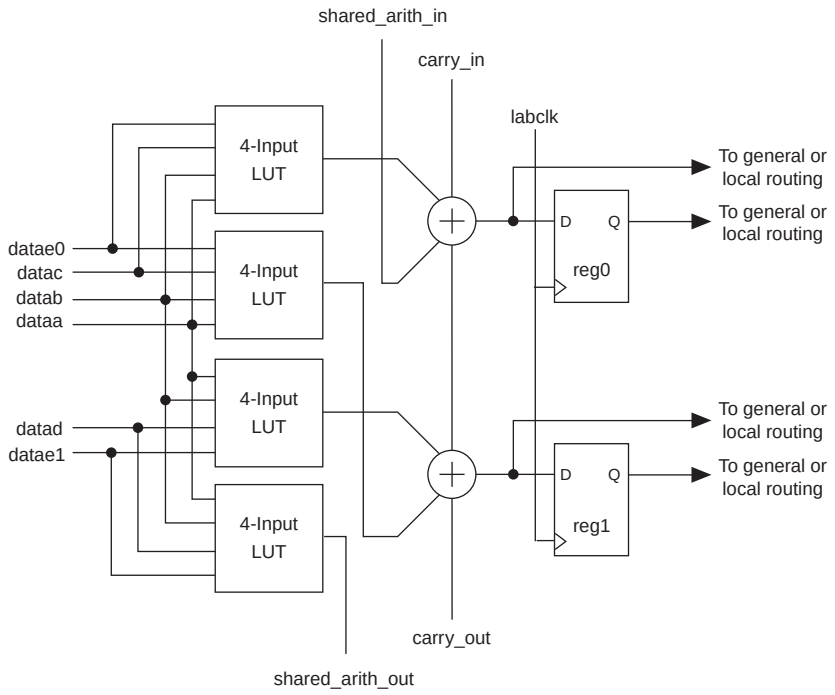
To avoid routing congestion in one small area of the device when a high fan-in arithmetic function is implemented, the LAB can support carry chains that only utilize either the top half or the bottom half of the LAB before connecting to the next LAB. This leaves the other half of the ALMs in the LAB available for implementing narrower fan-in functions in normal mode. Carry chains that use the top five ALMs in the first LAB carry into the top half of the ALMs in the next LAB within the column. Carry chains that use the bottom five ALMs in the first LAB carry into the bottom half of the ALMs in the next LAB within the column. In every alternate LAB column, the top half can be bypassed; in the other MLAB columns, the bottom half can be bypassed.

For more information on carry chain interconnect, refer to "ALM Interconnects" on page 2–20.

Shared Arithmetic Mode

In shared arithmetic mode, the ALM can implement a three-input add within an ALM. In this mode, the ALM is configured with 4 four-input LUTs. Each LUT either computes the sum of three inputs or the carry of three inputs. The output of the carry computation is fed to the next adder (either to `adder1` in the same ALM or to `adder0` of the next ALM in the LAB) via a dedicated connection called the shared arithmetic chain. This shared arithmetic chain can significantly improve the performance of an adder tree by reducing the number of summation stages required to implement an adder tree. Figure 2–13 shows the ALM using this feature.

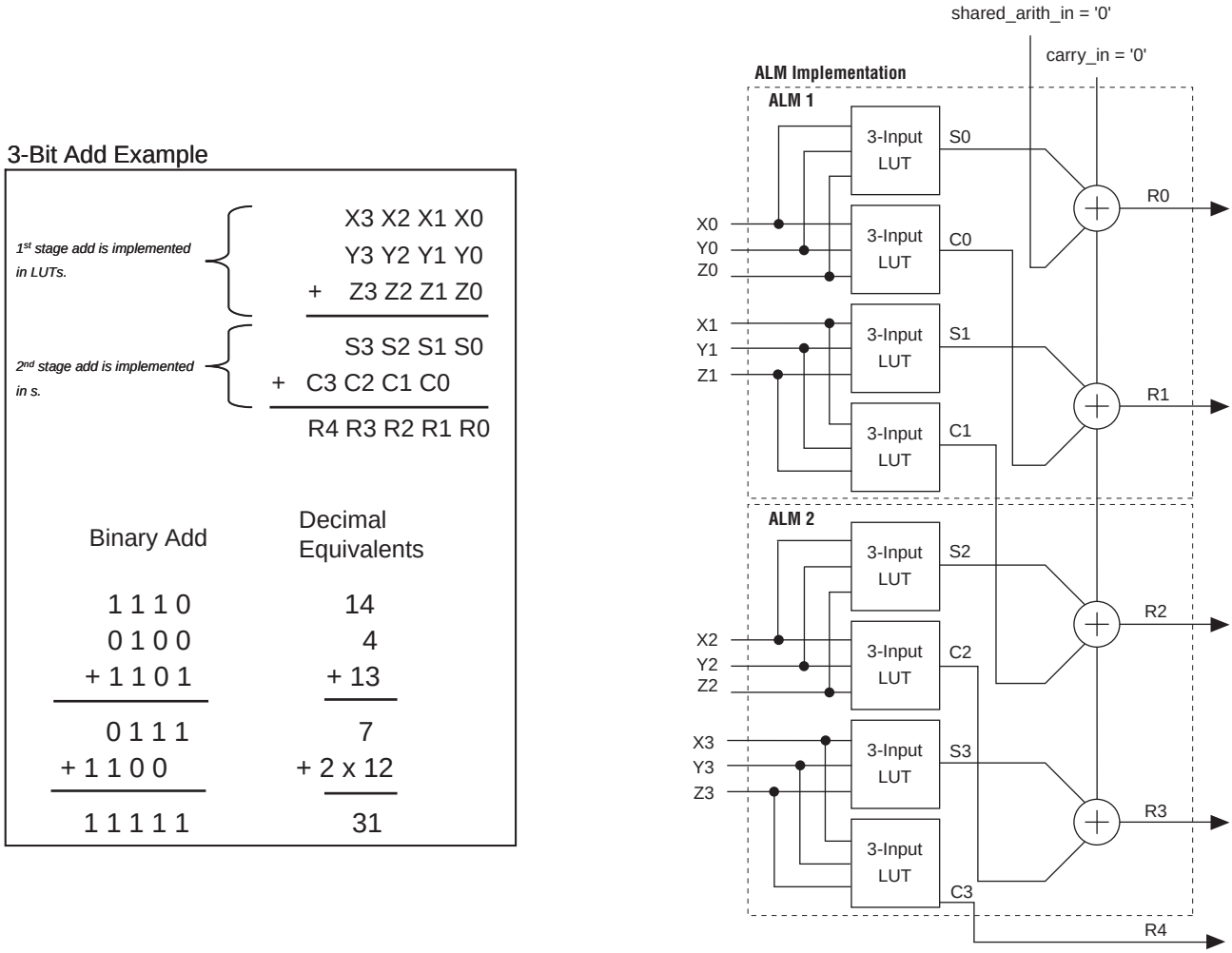
Figure 2–13. ALM in Shared Arithmetic Mode



You can find adder trees in many different applications. For example, the summation of the partial products in a logic-based multiplier can be implemented in a tree structure. Another example is a correlator function that can use a large adder tree to sum filtered data samples in a given time frame to recover or to de-spread data that was transmitted utilizing spread spectrum technology.

An example of a three-bit add operation utilizing the shared arithmetic mode is shown in Figure 2–14. The partial sum (S[3..0]) and the partial carry (C[3..0]) is obtained using the LUTs, while the result (R[3..0]) is computed using the dedicated adders.

Figure 2–14. Example of a 3-Bit Add Utilizing Shared Arithmetic Mode



Shared Arithmetic Chain

The shared arithmetic chain available in enhanced arithmetic mode allows the ALM to implement a three-input add. This significantly reduces the resources necessary to implement large adder trees or correlator functions.

The shared arithmetic chains can begin in either the first or sixth ALM in an LAB. The Quartus II Compiler creates shared arithmetic chains longer than 20 (10 ALMs in arithmetic or shared arithmetic mode) by linking LABs together automatically. For enhanced fitting, a long shared arithmetic chain runs vertically allowing fast horizontal connections to TriMatrix memory and DSP blocks. A shared arithmetic chain can continue as far as a full column.

Similar to the carry chains, the top and bottom half of shared arithmetic chains in alternate LAB columns can be bypassed. This capability allows the shared arithmetic chain to cascade through half of the ALMs in a LAB while leaving the other half available for narrower fan-in functionality. Every other LAB column is top-half bypassable, while the other LAB columns are bottom-half bypassable.

Refer to "ALM Interconnects" on page 2–20 for more information on shared arithmetic chain interconnect.

LUT-Register Mode

LUT-Register mode allows third register capability within an ALM. Two internal feedback loops allow combinational `ALUT1` to implement the master latch and combinational `ALUT0` to implement the slave latch needed for the third register. The LUT register shares its clock, clock enable, and asynchronous clear sources with the top dedicated register. Figure 2–15 shows the register constructed using two combinational blocks within the ALM. Figure 2–16 shows the ALM in LUT-Register mode.

Figure 2–15. LUT Register from Two Combinational Blocks

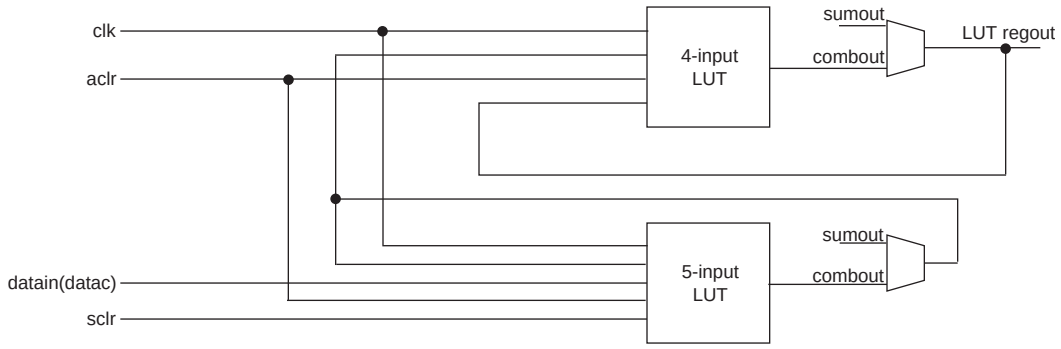
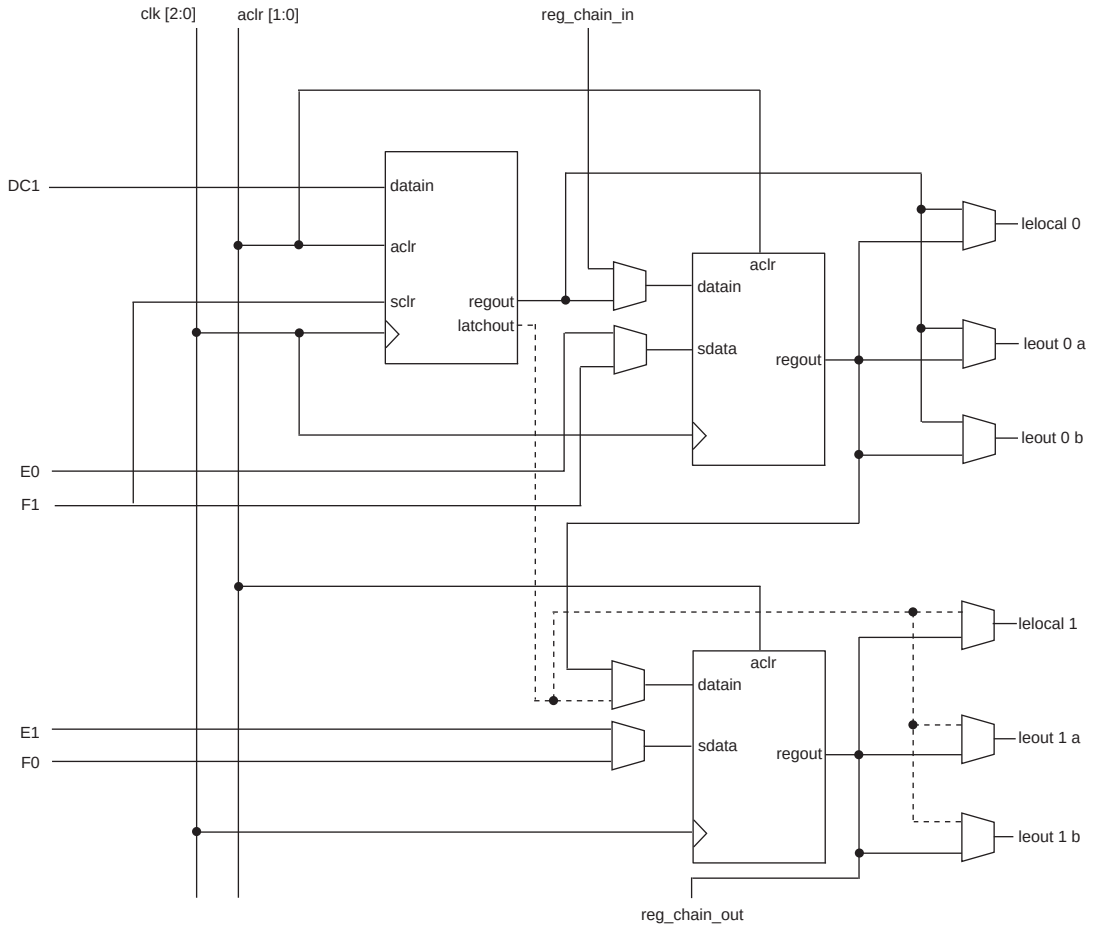


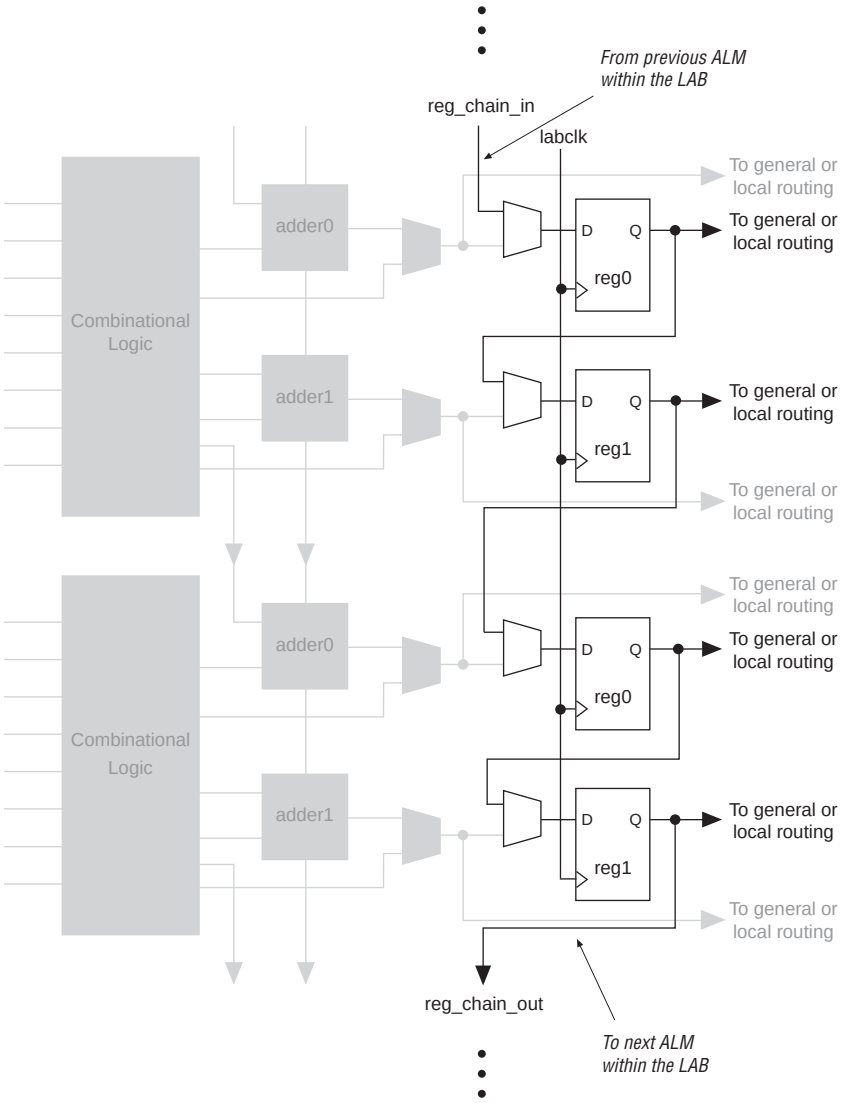
Figure 2–16. ALM in LUT-Register Mode with 3-Register Capability



Register Chain

In addition to the general routing outputs, the ALMs in an LAB have register chain outputs. The register chain routing allows registers in the same LAB to be cascaded together. The register chain interconnect allows a LAB to use LUTs for a single combinational function and the registers to be used for an unrelated shift register implementation. These resources speed up connections between ALMs while saving local interconnect resources (refer to Figure 2–17). The Quartus II Compiler automatically takes advantage of these resources to improve utilization and performance.

Figure 2–17. Register Chain within an LAB *(Note 1)*



Note to Figure 2–17:

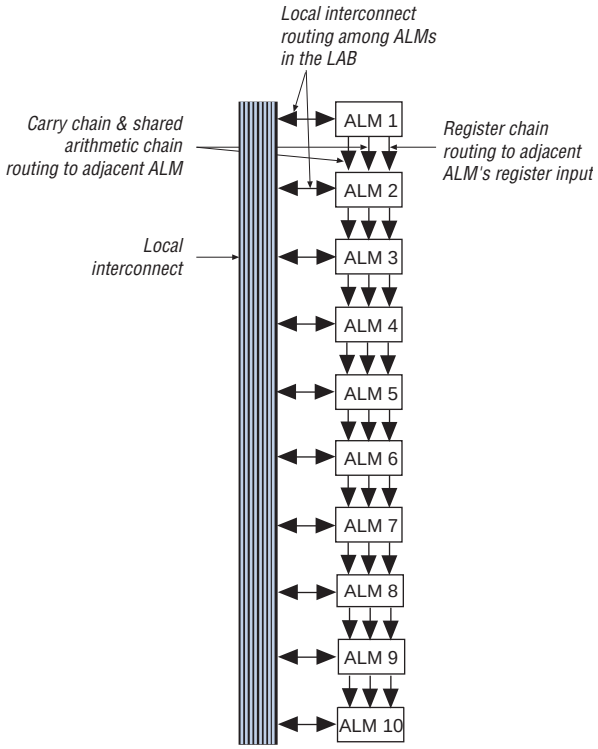
(1) You can use the combinational or adder logic to implement an unrelated, un-registered function.

For more information on register chain interconnect, refer to "ALM Interconnects" on page 2–20.

ALM Interconnects

There are three dedicated paths between ALMs: Register Cascade, Carry-chain, and Shared Arithmetic chain. Stratix III devices include an enhanced interconnect structure in LABs for routing shared arithmetic chains and carry chains for efficient arithmetic functions. The register chain connection allows the register output of one ALM to connect directly to the register input of the next ALM in the LAB for fast shift registers. These ALM-to-ALM connections bypass the local interconnect. The Quartus II Compiler automatically takes advantage of these resources to improve utilization and performance.Figure 2–18 shows the shared arithmetic chain, carry chain, and register chain interconnects.

Figure 2–18. Shared Arithmetic Chain, Carry Chain, and Register Chain Interconnects



For information about routing between LABs, refer to the *MultiTrack Interconnect in Stratix III Devices* chapter in volume 1 of the *Stratix III Device Handbook*.

Clear and Preset Logic Control

LAB-wide signals control the logic for the register's clear signal. The ALM directly supports an asynchronous clear function. You can achieve the register preset through the Quartus II software's **NOT-gate push-back logic** option. Each LAB supports up to two clears.

Stratix III devices provide a device-wide reset pin (`DEV_CLRn`) that resets all registers in the device. An option set before compilation in the Quartus II software controls this pin. This device-wide reset overrides all other control signals.

LAB Power Management Techniques

The following techniques are used to manage static and dynamic power consumption within the LAB:

- Stratix III low-voltage devices (L ordering code suffix) offer selectable core voltage to reduce both DC and AC power.
- To save AC power, Quartus II forces all adder inputs low when ALM adders are not in use.
- Stratix III LABs operate in high-performance mode or low-power mode. The Quartus II software automatically chooses the appropriate mode for an LAB based on the design to optimize speed vs. leakage trade-offs.
- Clocks represent a significant portion of dynamic power consumption due to their high switching activity and long paths. The LAB clock that distributes a clock signal to registers within a LAB is a significant contributor to overall clock power consumption. Each LAB's clock and clock enable signal are linked. For example, a combinational ALUT or register in a particular LAB using the `labclk1` signal also uses the `labclkena1` signal. To disable LAB-wide clock power consumption without disabling the entire clock tree, use the LAB-wide clock enable to gate the LAB-wide clock. The Quartus II software automatically promotes register-level clock enable signals to the LAB-level. All registers within an LAB that share a common clock and clock enable are controlled by a shared gated clock. To take advantage of these clock enables, use a clock enable construct in your HDL code for the registered logic.

Refer to the *Power Optimization* chapter in section 3 of the *Quartus II Handbook* for details on implementation.

For detailed information about Stratix III programmable power capabilities, refer to the *Programmable Power and Temperature Sensing Diode in Stratix III Devices chapter in volume 1* of the *Stratix III Device Handbook*.

Conclusion

Logic array block and adaptive logic modules are the basic building blocks of the Stratix III device. You can use these to configure logic functions, arithmetic functions, and register functions. The ALM provides advanced features with efficient logic utilization and is completely backward-compatible.

Chapter Revision History

Table 2–1shows the revision history for this document.

Table 2–1. Chapter Revision History(Sheet 1 of 2)

Date and Revision	Changes Made	Summary of Changes
February 2009, version 1.5	Removed “Referenced Documents” section.	—
October 2008, version 1.4	■ Updated “LAB Control Signals”, and “Carry Chain” Sections. ■ Updated New Document Format.	—

Table 2–1. Chapter Revision History(Sheet 2 of 2)

Date and Revision	Changes Made	Summary of Changes
May 2008, version 1.3	Updated Figure 2–2 and Figure 2–6.	—
October 2007, version 1.2	■ Added section “Referenced Documents”. ■ Added live links for references.	Minor changes.
May 2007, version 1.1	■ Minor formatting changes. ■ Updated Figure 2–6 to include a missing connection.	Minor changes.
November 2006, version 1.0	Initial Release.	—

3. MultiTrack Interconnect in Stratix III Devices

SII51003-1.2

Introduction

Stratix® III devices contain a two-dimensional row- and column-based architecture to implement custom logic. A series of column and row interconnects of varying length and speed provides signal interconnects between logic array blocks (LABs), memory block structures, digital signal processing (DSP) blocks, and input/output elements (IOE). These blocks communicate with themselves and to one another through a fabric of routing wires. This chapter provides details on the Stratix III core routing structure. It also describes how Stratix III block types interface to this fabric.

In the Stratix III architecture, connections between adaptive logic modules (ALMs), TriMatrix memory, DSP blocks, and device I/O pins are provided by the MultiTrack interconnect structure with DirectDrive technology. The MultiTrack interconnect consists of continuous, performance-optimized routing lines of different lengths and speeds used for inter- and intra-design block connectivity. The Quartus® II Compiler automatically routes critical design paths on faster interconnects to improve design performance.

DirectDrive technology is a deterministic routing technology that ensures identical routing resource usage for any function regardless of placement in the device. The MultiTrack interconnect and DirectDrive technology simplify the integration stage of block-based designing by eliminating the re-optimization cycles that typically follow design changes and additions.

The MultiTrack interconnect consists of row and column interconnects that span fixed distances. A routing structure with fixed length resources for all devices allows predictable and repeatable performance when migrating through different device densities.

Row Interconnects

Dedicated row interconnects route signals to and from LABs, DSP blocks, and TriMatrix memory blocks in the same row. These row interconnect resources include:

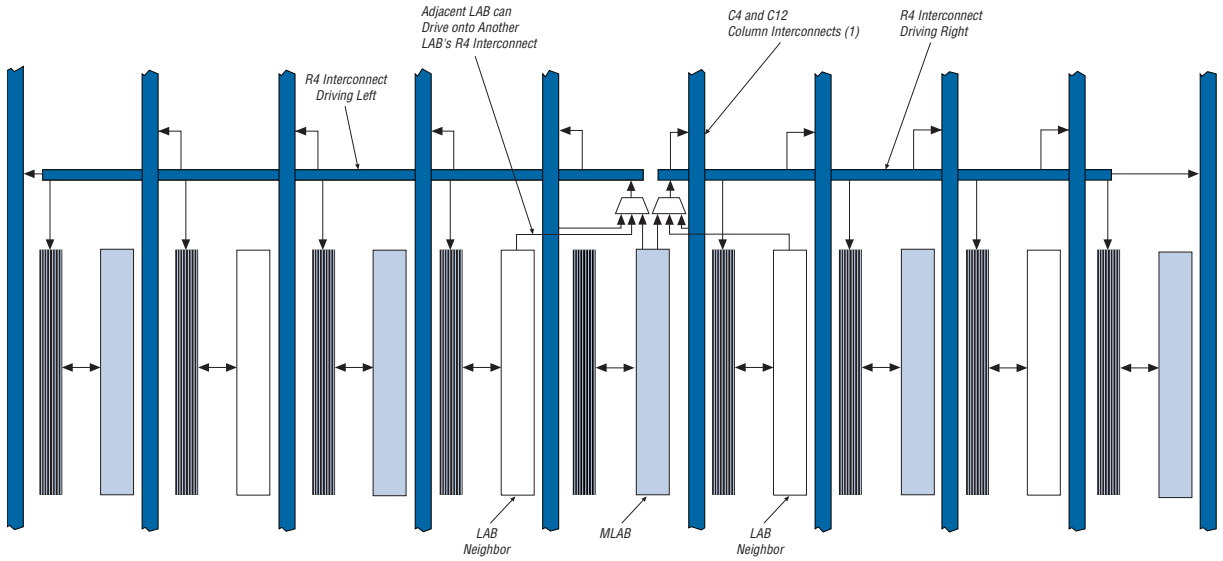
- Direct link interconnects between LABs and adjacent blocks
- R4 interconnects traversing four blocks to the right or left
- R20 row interconnects for high-speed access across the length of the device

The direct link interconnect allows a LAB, DSP block, or TriMatrix memory block to drive into the local interconnect of its left and right neighbors. This capability provides fast communication between adjacent LABs and blocks without using row interconnect resources. The direct link interconnect is the fastest way to communicate between two adjacent blocks.

The R4 interconnects span a combination of four LABs, memory logic array blocks (MLAB), DSP blocks, M9K blocks, and M144K blocks. Use these resources for fast row connections in a four-LAB region. Figure 3–1 shows R4 interconnect connections from a LAB. R4 interconnects can drive and be driven by DSP blocks and RAM blocks and row IOEs. For LAB interfacing, a primary LAB or LAB neighbor can drive a given R4

interconnect. For R4 interconnects that drive to the right, the primary LAB and right neighbor can drive to the interconnect. For R4 interconnects that drive to the left, the primary LAB and its left neighbor can drive the interconnect. R4 interconnects can drive other R4 interconnects to extend the range of LABs they drive. R4 interconnects can also drive C4 and C12 (column interconnects) for connections from one row to another. Additionally, R4 interconnects can drive R20 interconnects.

Figure 3–1. R4 Interconnect Connections *(Note 1), (2)*



Notes to Figure 3–1

(1) C4 and C12 interconnects can drive R4 interconnects.

(2) This pattern is repeated for every LAB in the LAB row.

R20 row interconnects span 20 LABs and provide the fastest resource for row connections between distant LABs, TriMatrix memory, DSP blocks, and row IOEs. R20 row interconnects drive LAB local interconnects via R4 and C4 interconnects. R20 interconnects can drive R20, R4, C12, and C4 interconnects.

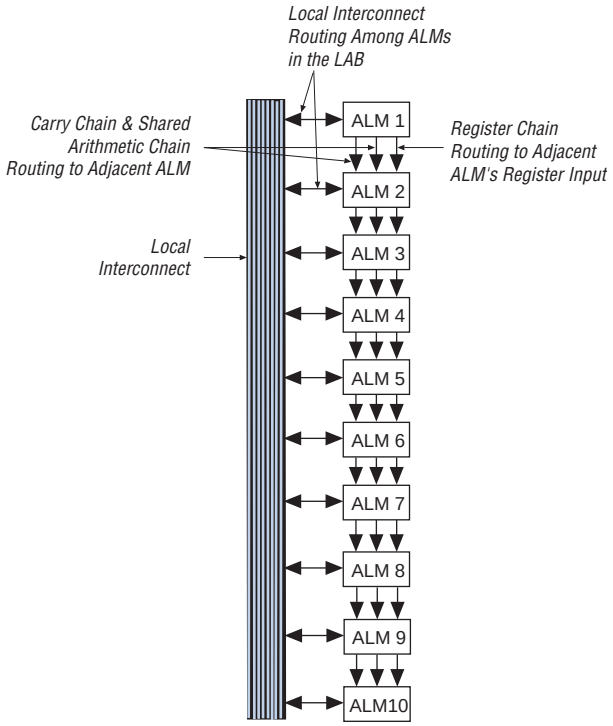
Column Interconnects

The column interconnect operates similarly to the row interconnect. It vertically routes signals to and from LABs, TriMatrix memory, DSP blocks, and IOEs. Each column of LABs is served by a dedicated column interconnect. These column interconnect resources include:

- Shared arithmetic chain interconnects in a LAB and from LAB to LAB
- Carry chain interconnects in a LAB and from LAB to LAB
- Register chain interconnects in a LAB
- C4 interconnects traversing a distance of four blocks in the same device column
- C12 column interconnects for high-speed vertical routing through the device

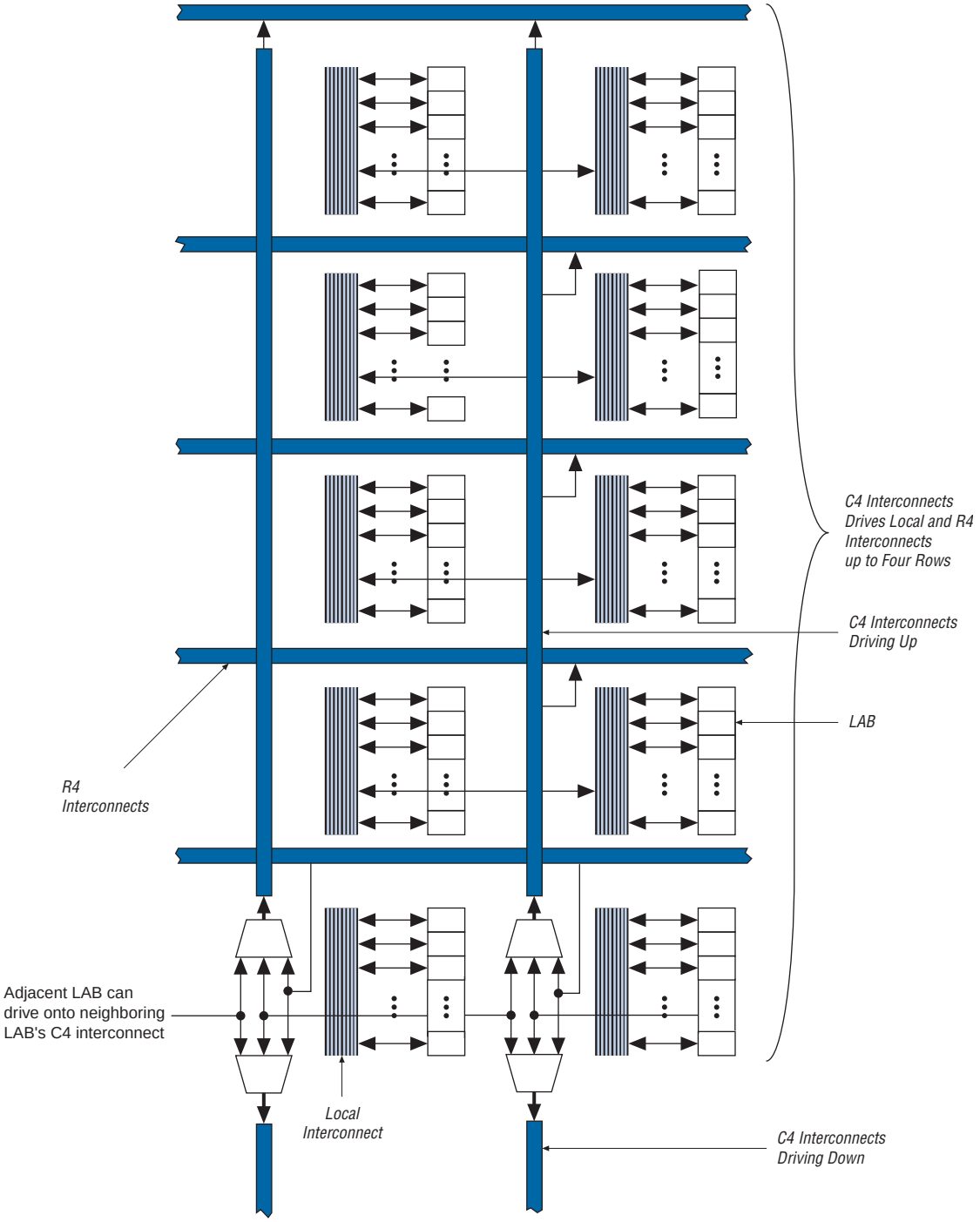
Stratix III devices include an enhanced interconnect structure in LABs for routing-shared arithmetic chains and carry chains for efficient arithmetic functions. The register chain connection allows the register output of one ALM to connect directly to the register input of the next ALM in the LAB for fast shift registers. These ALM-to-ALM connections bypass the local interconnect. The Quartus II Compiler automatically takes advantage of these resources to improve utilization and performance. Figure 3–2 shows the shared arithmetic chain, carry chain, and register chain interconnects.

Figure 3–2. Shared Arithmetic Chain, Carry Chain, and Register Chain Interconnects



The C4 interconnects span four adjacent interfaces in the same device column. C4 interconnects also pass by M144K and DSP blocks. A single M144K block utilizes eight adjacent interfaces in the same column. A DSP block utilizes four adjacent interfaces in the same column. Figure 3–3 shows the C4 interconnect connections from a LAB in a column. The C4 interconnects can drive and be driven by all types of architecture blocks, including DSP blocks, TriMatrix memory blocks, and column and row IOEs. For LAB interconnection, a primary LAB or its LAB neighbor can drive a given C4 interconnect. C4 interconnects can drive each other to extend their range as well as drive row interconnects for column-to-column connections.

Figure 3–3. C4 Interconnect Connections *(Note 1)*



Note to Figure 3–3:

(1) Each C4 interconnect can drive either up or down four rows.

C12 column interconnects span a length of 12 LABs and provide the fastest resource for column connections between distant LABs, TriMatrix memory blocks, DSP blocks, and IOEs. C12 interconnects drive LAB local interconnects via C4 and R4 interconnects and do not drive LAB local interconnects directly.

All embedded blocks communicate with the logic array through interconnects similar to LAB-to-LAB interfaces. Each block (for example, TriMatrix memory blocks and DSP blocks) connects to row and column interconnects and has local interconnect regions driven by row and column interconnects. These blocks also have direct link interconnects for fast connections to and from a neighboring LAB.

Table 3–1 shows the Stratix III device's routing scheme.

Table 3–1. Stratix III Device Routing Scheme

Source	Destination															
	Shared Arith-metic Chain	Carry Chain	Register Chain	Local Inter-connect	Direct Link Inter-connect	R4 Inter-connect	R20 Inter-connect	C4 Inter-connect	C12 Inter-connect	ALM	MLAB RAM Block	M9K RAM Block	M144K Block	DSP Blocks	Column IOE	Row IOE
Shared arithmetic chain	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—
Carry chain	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—
Register chain	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—
Local interconnect	—	—	—	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓	✓
Direct link interconnect	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—
R4 interconnect	—	—	—	✓	—	✓	✓	✓	✓	—	—	—	—	—	—	—
R20 interconnect	—	—	—	✓	—	✓	✓	✓	✓	—	—	—	—	—	—	—
C4 interconnect	—	—	—	✓	—	✓	—	✓	—	—	—	—	—	—	—	—
C12 interconnect	—	—	—	✓	—	✓	✓	✓	✓	—	—	—	—	—	—	—
ALM	✓	✓	✓	✓	✓	✓	—	✓	—	—	—	—	—	—	—	—
MLAB RAM block	—	—	—	✓	✓	✓	—	✓	—	—	—	—	—	—	—	—
M9K RAM block	—	—	—	—	✓	✓	—	✓	—	—	—	—	—	—	—	—
M144K block	—	—	—	—	✓	✓	—	✓	—	—	—	—	—	—	—	—
DSP blocks	—	—	—	—	✓	✓	—	✓	—	—	—	—	—	—	—	—
Column IOE	—	—	—	—	—	—	—	✓	✓	—	—	—	—	—	—	—
Row IOE	—	—	—	—	✓	✓	✓	✓	—	—	—	—	—	—	—	—

Notes to Table 3–1:

(1) Except column IOE local interconnects.

(2) Row IOE local interconnects.

(3) Column IOE local interconnects.

The R4 and C4 interconnects provide superior and flexible routing capabilities. Stratix III has a three-sided routing architecture which allows the interconnect wires from each LAB to reach the adjacent LABs to its right and left. A given LAB can drive 32 other LABs using one R4 or C4 interconnect, in one hop. This routing scheme improves efficiency and flexibility by placing all the critical LABs within one hop of the routing interconnects.

Table 3–2 shows how many LABs are reachable within one, two, or three hops using the R4 and C4 interconnects.

Table 3–2. Number of LABs reachable using C4 and R4 interconnects

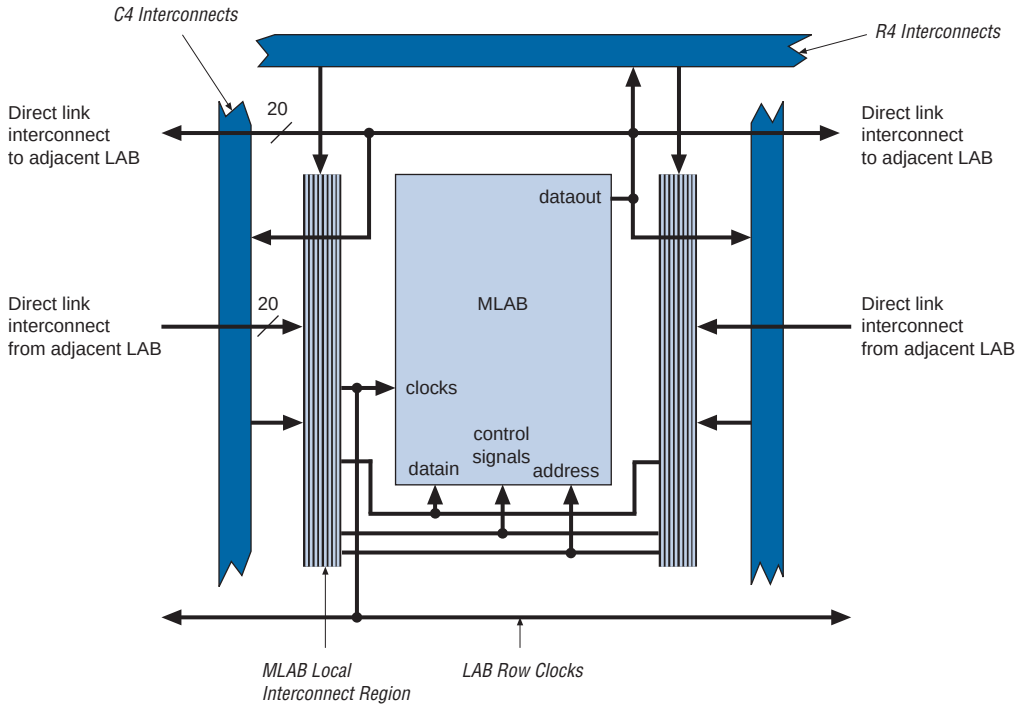
Hops	Number of LABs
1	34
2	96
3	160

Memory Block Interface

TriMatrix memory consists of three types of RAM blocks: MLAB, M9K, and M144K. This section provides a brief overview of how the different memory blocks interface to the routing structure.

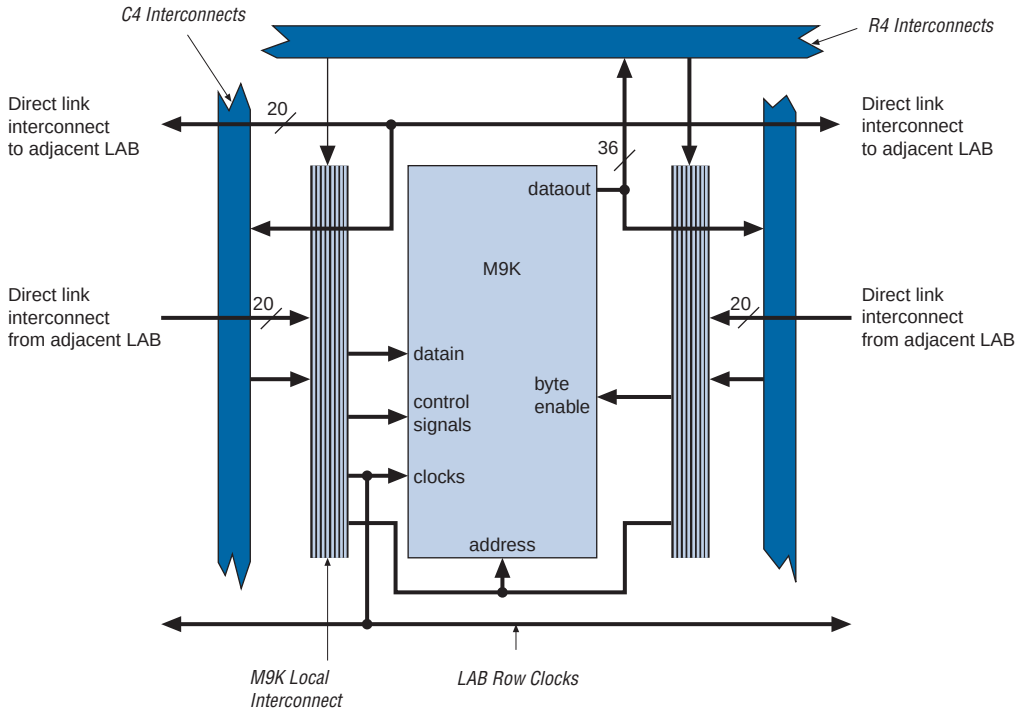
The RAM blocks in Stratix III devices have local interconnects to allow ALMs and interconnects to drive into RAM blocks. The MLAB RAM block local interconnect is driven by the R4, C4, and direct link interconnects from adjacent LABs. The MLAB RAM blocks can communicate with LABs on either the left or right side through these row interconnects or with LAB columns on the left or right side with the column interconnects. Each MLAB RAM block has up to 20 direct link input connections from the left adjacent LAB and another 20 from the right adjacent LAB. MLAB RAM outputs can also connect to left and right LABs through a direct link interconnect. The MLAB RAM block has equal opportunity for access and performance to and from LABs on either its left or right side. Figure 3–4 shows the MLAB RAM block to LAB row interface.

Figure 3–4. MLAB RAM Block LAB Row Interface



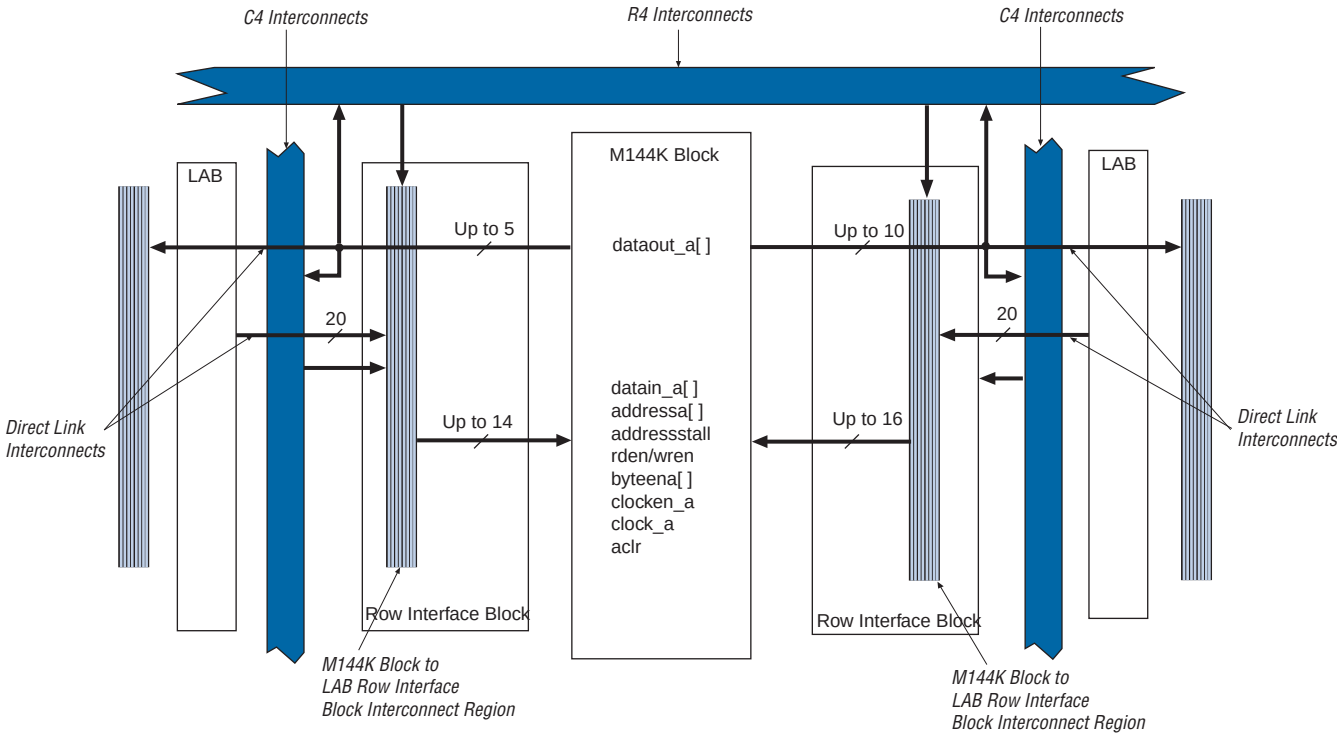
The M9K RAM block local interconnect is driven by the R4, C4, and direct link interconnects from adjacent LABs. The M9K RAM blocks can communicate with LABs on either the left or right side through these row resources or with LAB columns on either the right or left with the column resources. Up to 20 direct link input connections to the M9K RAM Block are possible from the left adjacent LABs and another 20 possible from the right adjacent LAB. M9K RAM block outputs can also connect to left and right LABs through direct link interconnect. Figure 3–5 shows the M9K RAM block to logic array interface.

Figure 3–5. M9K RAM Block LAB Row Interface



The M144K blocks use eight interfaces in the same device column. The M144K block local interconnects are driven by R4, C4, and direct link interconnects from adjacent LABs on either the right or left side of the MRAM block. Up to 20 direct link input connections to the M144K block are possible from the left adjacent LABs and another 20 possible from the right adjacent LAB. M144K block outputs can also connect to the LABs on the block's left and right sides through direct link interconnect. Figure 3–6 shows the interface between the M144K RAM block and the logic array.

Figure 3–6. M144K Row Unit Interface to Interconnect



DSP Block Interface

Stratix III device DSP block input registers can generate a shift register that cascades down in the same DSP block column. Dedicated connections between DSP blocks provide fast connections between the shift register inputs to cascade the shift register chains. You can cascade registers within multiple DSP blocks for 9-bit or 18-bit finite impulse response (FIR) filters larger than four taps, with additional adder stages implemented in ALMs. If the DSP block is configured as 36-bit blocks, the adder, subtractor, or accumulator stages are implemented in ALMs. Each DSP block can route the shift register chain out of the block to cascade multiple columns of DSP blocks.

The DSP block is divided into four block units that interface with four LAB rows on the left and right. You can consider each block unit as two 18-bit multipliers followed by an adder with 72 inputs and 36 outputs. A local interconnect region is associated with each DSP block. Like a LAB, this interconnect region can be fed with 20 direct link interconnects from the LAB to the left or right of the DSP block in the same row. R4 and C4 routing resources can access the DSP block's local interconnect region.

These outputs work similarly to LAB outputs. Eighteen outputs from the DSP block can drive to the left LAB through direct link interconnects and eighteen can drive to the right LAB though direct link interconnects. All 36 outputs can drive to R4 and C4 routing interconnects. Outputs can drive right- or left-column routing. Figure 3–7 and Figure 3–8 show the DSP block interfaces to LAB rows.

Figure 3–7. High-Level View, DSP Block Interface to Interconnect

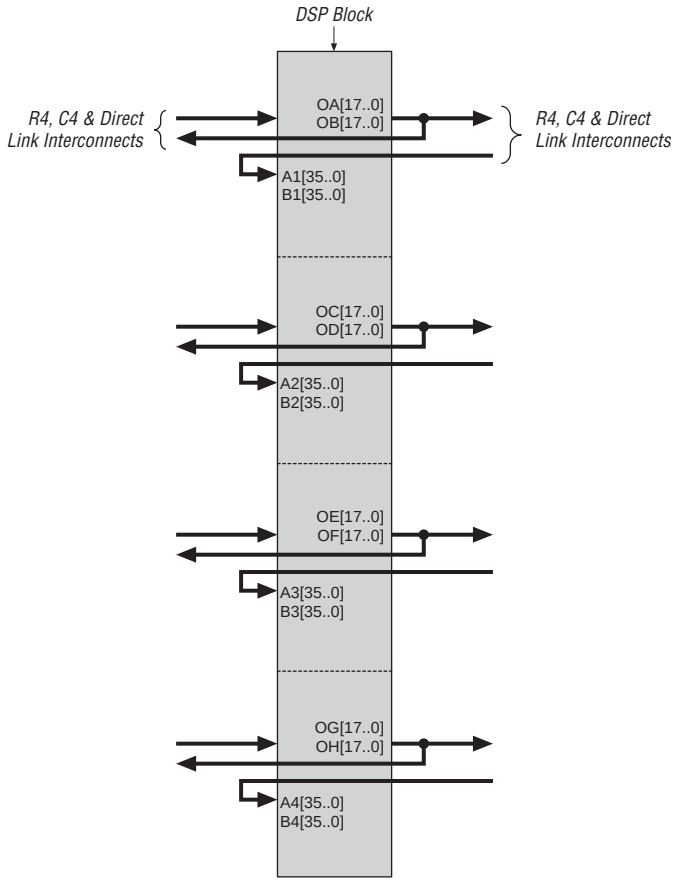
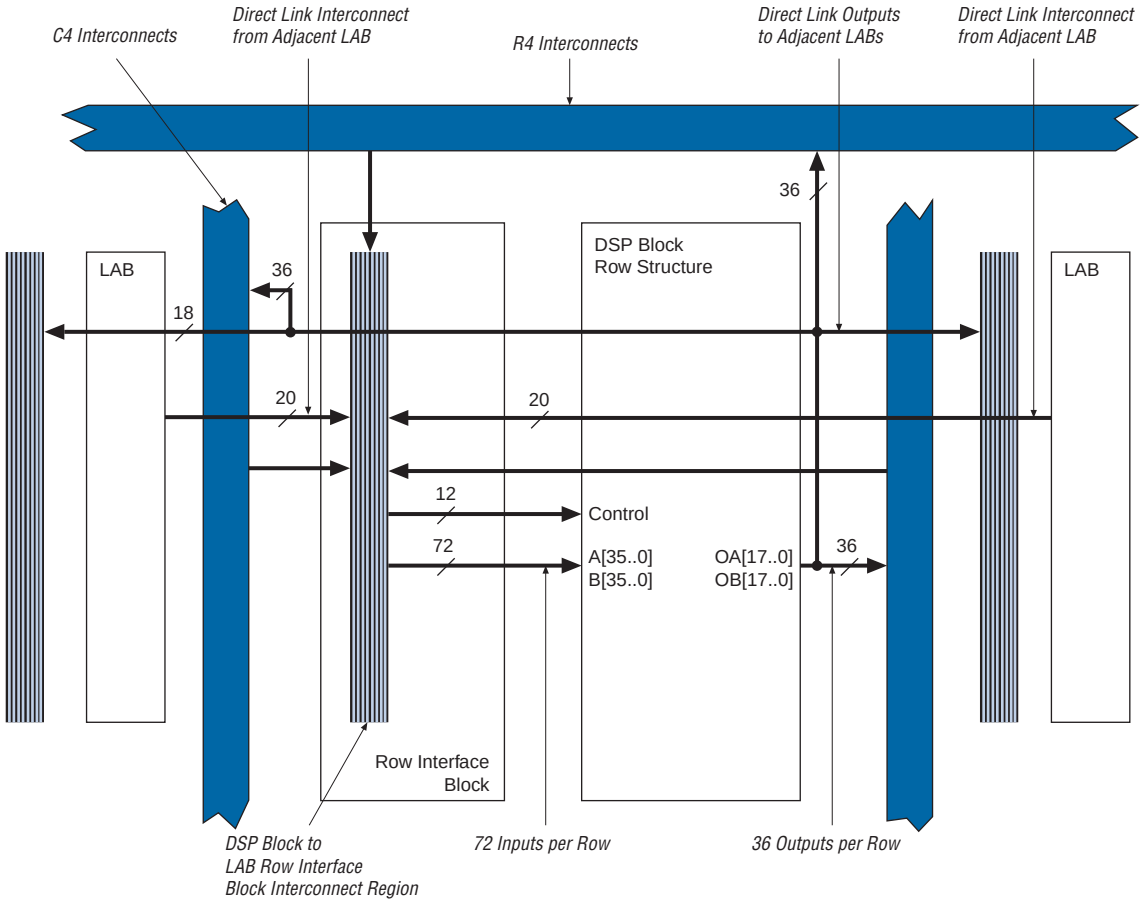


Figure 3–8. Detailed View, DSP Block Interface to Interconnect



I/O Block Connections to Interconnect

The IOEs are located in I/O blocks around the periphery of the Stratix III device. There are up to four IOEs per row I/O block and four IOEs per column I/O block. The row I/O blocks drive row, column, or direct link interconnects. The column I/O blocks drive column interconnects. Figure 3–9 shows how a row I/O block connects to the logic array. Figure 3–10 shows how a column I/O block connects to the logic array.

Figure 3–9. Row I/O Block Connection to Interconnect

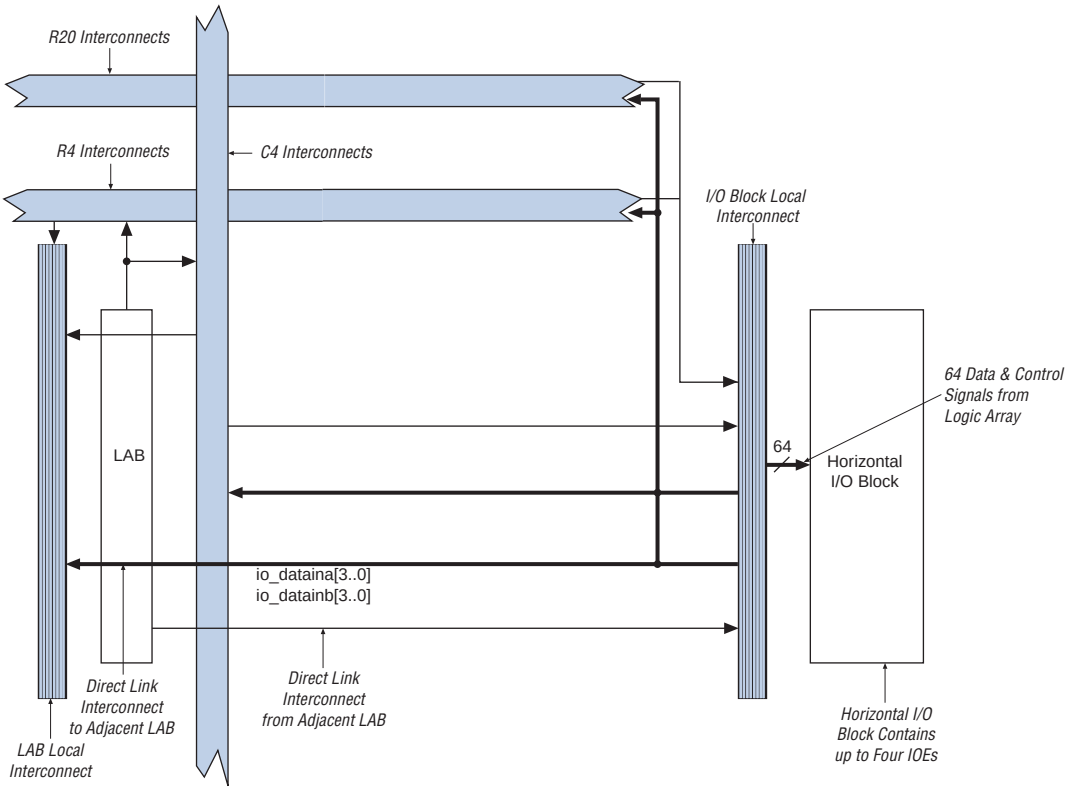
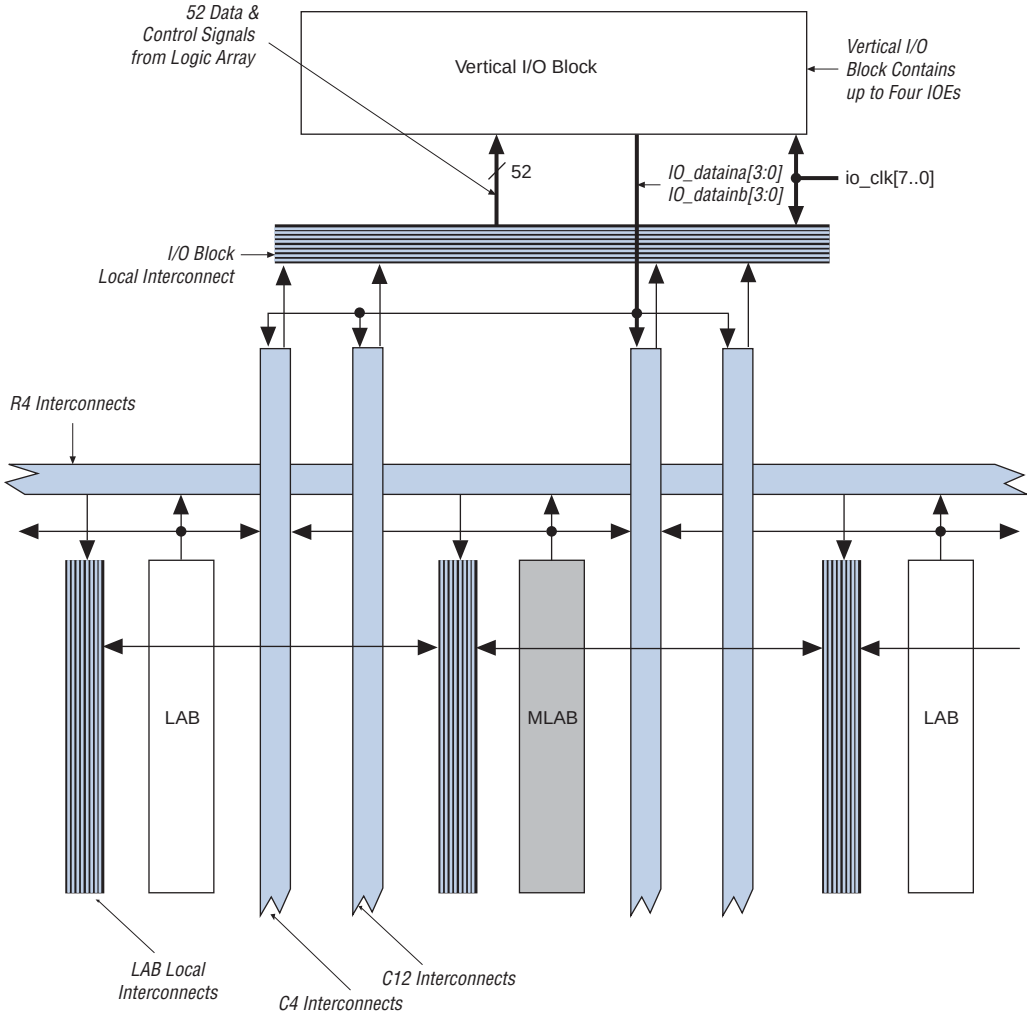


Figure 3–10. Column I/O Block Connection to Interconnect



Conclusion

Stratix III devices consist of an array of logic blocks such as LABs, TriMatrix memory, DSP blocks, and IOEs. These blocks communicate with themselves and one another through the MultiTrack interconnect structures. The Quartus II compiler automatically routes critical design paths on faster interconnects to improve design performance and optimize the device resources.

Chapter Revision History

Table 3–3 shows the revision history for this document.

Table 3–3. Chapter Revision History

Date and Revision	Changes Made	Summary of Changes
October 2008, version 1.2	Updated New Document Format.	—
October 2007, version 1.1	■ Minor formatting changes. ■ Added section “Chapter Revision History”. ■ Added live links for references.	Minor formatting changes.
November 2006, version 1.0	Initial Release.	—

4. TriMatrix Embedded Memory Blocks in Stratix III Devices

SIII51004-1.8

Introduction

TriMatrix embedded memory blocks provide three different sizes of embedded SRAM to efficiently address the needs of Stratix® III FPGA designs. TriMatrix memory includes 640- (in ROM mode only) or 320-bit memory logic array blocks (MLABs), 9-Kbit M9K blocks, and 144-Kbit M144K blocks. The MLABs have been optimized to implement filter delay lines, small first-in first-out (FIFO) buffers, and shift registers. You can use the M9K blocks for general purpose memory applications, and the M144K blocks are ideal for processor code storage, packet buffering, and video frame buffering.

You can independently configure each embedded memory block to be a single- or dual-port RAM, FIFO, ROM, or shift register via the Quartus® II MegaWizard™ Plug-In Manager. You can stitch together multiple blocks of the same type to produce larger memories with minimal timing penalty. TriMatrix memory provides up to 20,491 Kbits of embedded SRAM at up to 600 MHz operation. This chapter describes TriMatrix memory blocks, modes, features, and design considerations.

Overview

Table 4–1 summarizes the features supported by the three sizes of TriMatrix memory.

Table 4–1. Summary of TriMatrix Memory Features (Part 1 of 2)

Feature	MLABs	M9K Blocks	M144K Blocks
Maximum performance	600 MHz	580 MHz	580 MHz
Total memory bits (including parity bits)	640 (in ROM mode) or 320 (in other modes)	9,216	147,456
Configurations (depth × width) *(1)*	16 × 8 16 × 9 16 × 10 16 × 16 16 × 18 16 × 20	8 K × 1 4 K × 2 2 K × 4 1 K × 8 1 K × 9 512 × 16 512 × 18 256 × 32 256 × 36	16 K × 8 16 K × 9 8 K × 16 8 K × 18 4 K × 32 4 K × 36 2 K × 64 2 K × 72
Parity bits	✓	✓	✓

Table 4–1. Summary of TriMatrix Memory Features (Part 2 of 2)

Feature	MLABs	M9K Blocks	M144K Blocks
Byte-enable	✓	✓	✓
Packed mode	—	✓	✓
Address clock enable	✓	✓	✓
Single-port memory	✓	✓	✓
Simple dual-port memory	✓	✓	✓
True dual-port memory	—	✓	✓
Embedded shift register	✓	✓	✓
ROM	✓	✓	✓
FIFO buffer	✓	✓	✓
Simple dual-port mixed width support	—	✓	✓
True dual-port mixed width support	—	✓	✓
Memory initialization file (.mif)	✓	✓	✓
Mixed-clock mode	✓	✓	✓
Power-up condition	Outputs cleared if registered, otherwise reads memory contents	Outputs cleared	Outputs cleared
Register clears	Output registers	Output registers	Output registers
Asynchronous clear on output latch	—	✓	✓
Write/Read operation triggering	Write: Falling clock edges Read: Rising clock edges	Write and Read: Rising clock edges	Write and Read: Rising clock edges
Same-port read-during-write	Outputs set to don't care	Outputs set to old or new data	Outputs set to old or new data
Mixed-port read-during-write	Outputs set to old data or don't care	Outputs set to old data or don't care	Outputs set to old data or don't care
ECC Support	Soft IP support via Quartus II software	Soft IP support via Quartus II software	Built-in support in ×64 wide SDP mode or soft IP support via Quartus II software

Notes to Table 4–1:

(1) In ROM mode, MLABs support the (depth × width) configurations of 64×8, 64×9, 64×10, 32×16, 32×18, or 32× 20.

(2) MLABs support byte-enable via emulation.

Table 4–2 shows the capacity and distribution of the TriMatrix memory blocks for each Stratix III family member

Table 4–2. TriMatrix Memory Capacity and Distribution in Stratix III Devices

Device	MLABs	M9K Blocks	M144K Blocks	Total Dedicated RAM Bits (dedicated memory blocks only)	Total RAM Bits (including MLABs) *(1)*
EP3SL50	950	108	6	1,836 Kb	2,133 Kb
EP3SL70	1,350	150	6	2,214 Kb	2,636 Kb
EP3SL110	2,150	275	12	4,203 Kb	4,875 Kb
EP3SL150	2,850	355	16	5,499 Kb	6,390 Kb
EP3SL200	4,000	468	36	9,396 Kb	10,646 Kb
EP3SL340	6,750	1,040	48	16,272 Kb	18,381 Kb
EP3SE50	950	400	12	5,328 Kb	5,625 Kb
EP3SE80	1,600	495	12	6,183 Kb	6,683 Kb
EP3SE110	2,150	639	16	8,055 Kb	8,727 Kb
EP3SE260	5,100	864	48	14,688 Kb	16,282 Kb

Note to Table 4–2:

(1) For total ROM Kbits, use this equation to calculate:
Total ROM Kbits = Total Embedded RAM Kbits + [(number of MLAB blocks × 640)/1024]

TriMatrix Memory Block Types

While the M9K and M144K memory blocks are dedicated resources, the MLABs are dual-purpose blocks. They can be configured as regular logic array blocks (LABs) or as memory logic array blocks (MLABs). Ten adaptive logic modules (ALMs) make up one MLAB. Each ALM in an MLAB can be configured as a 16 × 2 block, resulting in a 16 × 20 simple dual-port SRAM block in a single MLAB. In ROM mode, each ALM in an MLAB can be configured as either a 64 × 1 or a 32 × 2 block, resulting in a 64 × 10 or 32 × 20 ROM block in a single MLAB.

All the ALMs share the same address bits. Therefore, you cannot combine multiple memories with different address bits and implement them in a single MLAB.

When you are using an MLAB as memory, you will not be able to use the unused ALMs in the MLAB even if you do not use the full capacity of an MLAB.

Parity Bit Support

All TriMatrix memory blocks have built-in parity-bit support. The ninth bit associated with each byte can store a parity bit or serve as an additional data bit. No parity function is actually performed on the ninth bit.

Byte-Enable Support

All TriMatrix memory blocks support byte-enables that mask the input data so that only specific bytes of data are written. The unwritten bytes retain the previous written value. The write enable (`wren`) signals, along with the byte-enable (`byteena`) signals, control the RAM blocks' write operations.

MLABs support byte-enable via emulation. There will be increased logic utilization when the byte-enables are emulated.

The default value for the byte-enable signals is high (enabled), in which case writing is controlled only by the write enable signals. The byte-enable registers have no clear port. When using parity bits on the M9K and M144K blocks, the byte-enable controls all nine bits (eight bits of data plus one parity bit). When using parity bits on the MLAB, the byte-enable controls all 10 bits in the widest mode.

Byte-enables operate in a one-hot fashion, with the LSB of the `byteena` signal corresponding to the least significant byte of the data bus. For example, if you are using a RAM block in ×18 mode, with `byteena = 01`, `data[8..0]` is enabled and `data[17..9]` is disabled. Similarly, if `byteena = 11`, both `data[8..0]` and `data[17..9]` are enabled. Byte-enables are active high.

You cannot use the byte-enable feature when using the ECC feature on M144K blocks.

Figure 4–1 shows how the write enable (`wren`) and byte-enable (`byteena`) signals control the operations of the M9K and M144K.

When a byte-enable bit is de-asserted during a write cycle, the corresponding data byte output can appear as either a "don't care" value or the current data at that location. The output value for the masked byte is controllable via the Quartus II software. When a byte-enable bit is asserted during a write cycle, the corresponding data byte output also depends on the setting chosen in the Quartus II software.

Figure 4–1. Stratix III Byte-Enable Functional Waveform for M9K and M144K

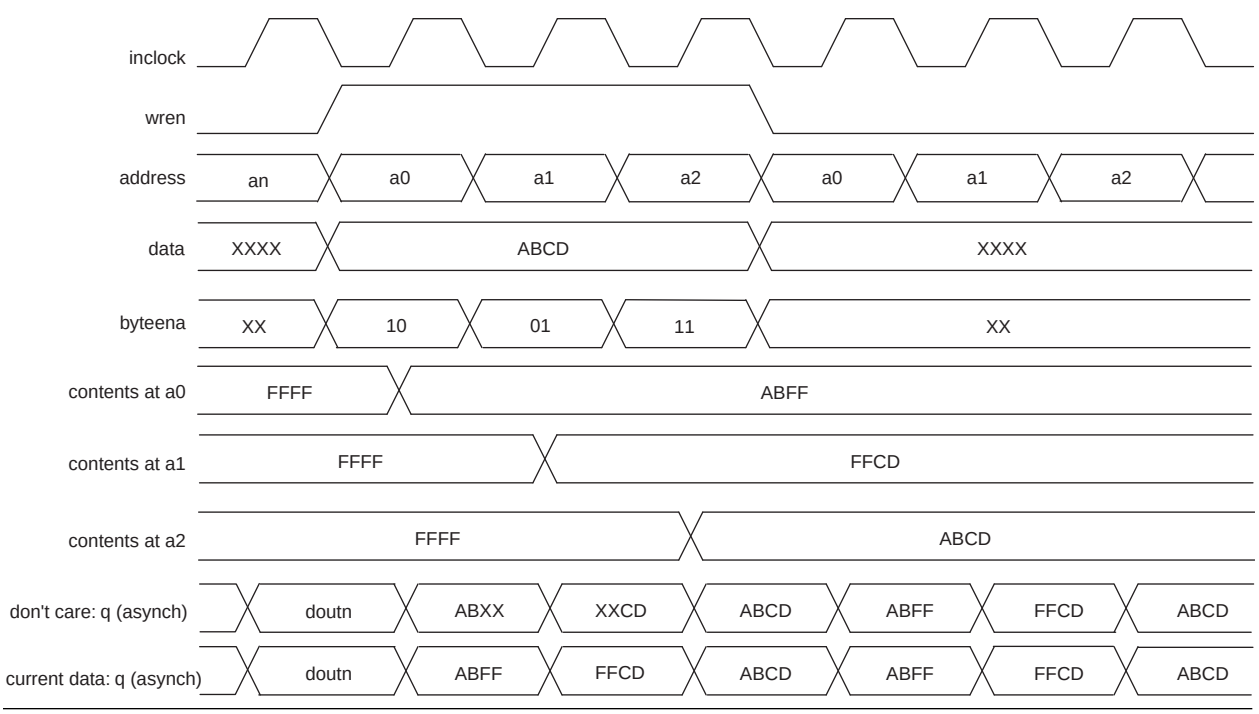
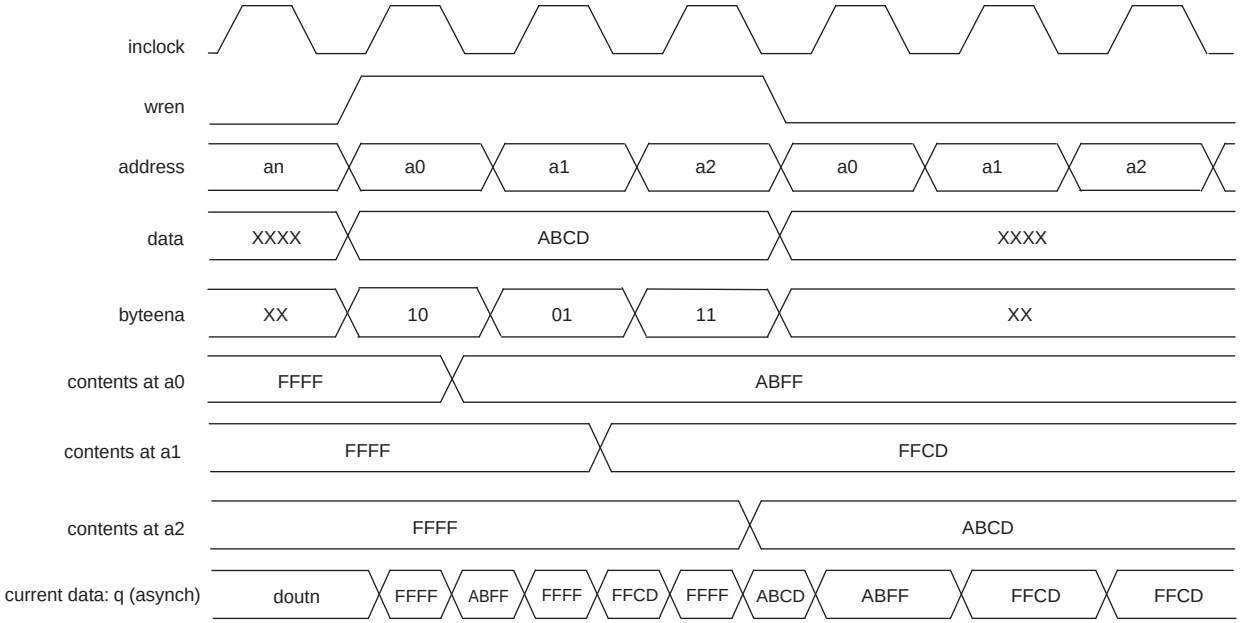


Figure 4–2 shows how the write enable (`wren`) and byte-enable (`byteena`) signals control the operations of the MLABs. The write operation in MLABs is triggered by failing clock edges.

Figure 4–2. Stratix III Byte-Enable Functional Waveform for MLABs



Packed Mode Support

Stratix III M9K and M144K blocks support packed mode. The packed mode feature packs two independent single-port RAMs into one memory block. The Quartus II software automatically implements packed mode where appropriate by placing the physical RAM block into true dual-port mode and using the MSB of the address to distinguish between the two logical RAMs. The size of each independent single-port RAM must not exceed half of the target block size.

Address Clock Enable Support

All Stratix III memory blocks support address clock enable, which holds the previous address value for as long as the signal is enabled (`addressstall` = `1`). When the memory blocks are configured in dual-port mode, each port has its own independent address clock enable. The default value for the address clock enable signals is low (disabled).

Figure 4–3 shows an address clock enable block diagram. The address clock enable is referred to by the port name `addressstall`.

Figure 4–3. Stratix III Address Clock Enable Block Diagram

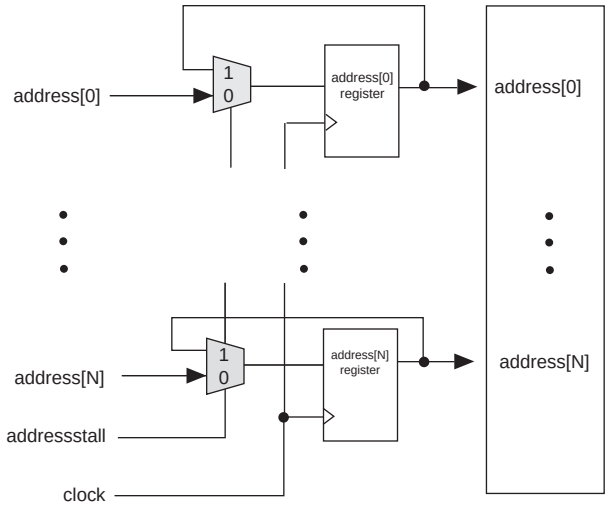


Figure 4–4 shows the address clock enable waveform during the read cycle.

Figure 4–4. Stratix III Address Clock Enable during Read Cycle Waveform

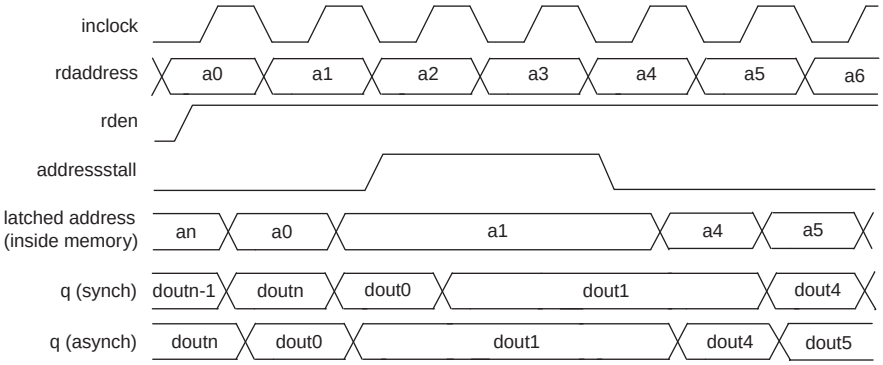


Figure 4–5 shows the address clock enable waveform during the write cycle for M9K and M144K.

Figure 4–5. Stratix III Address Clock Enable during Write Cycle Waveform for M9K and M144K

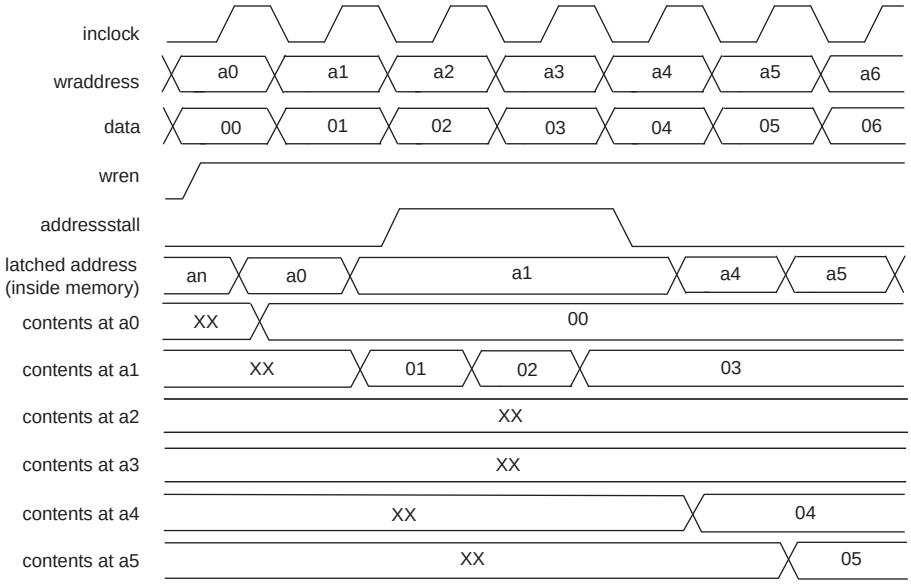
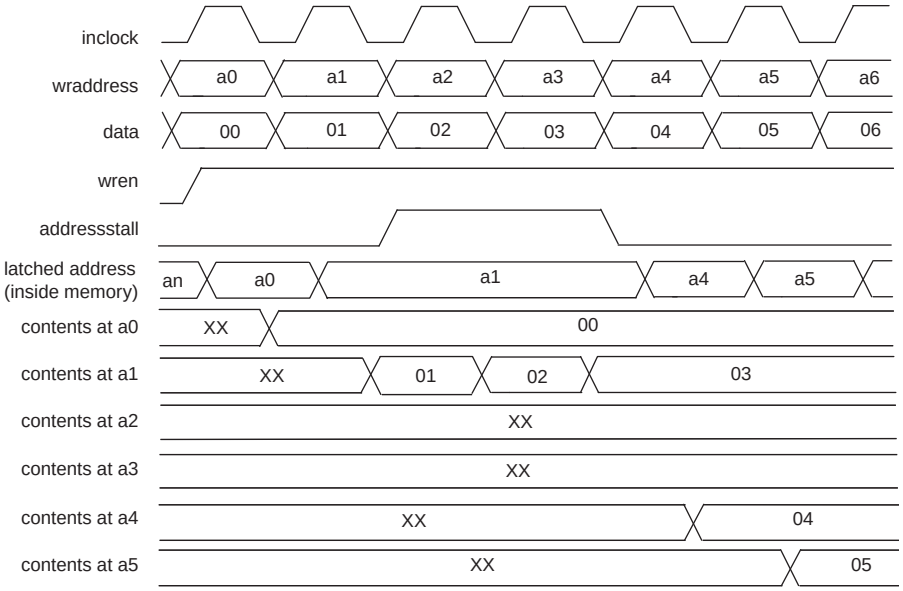


Figure 4–6 shows the address clock enable waveform during the write cycle for MLABs.

Figure 4–6. Stratix III Address Clock Enable during Write Cycle Waveform for MLABs



Mixed Width Support

M9K and M144K memory blocks inherently support mixed data widths. MLABs can support mixed data widths through emulation via the Quartus II software. When using simple dual-port or true dual-port mixed width support allows you to read and write different data widths to a memory block. Refer to "Memory Modes" on page 4–10 for details on the different widths supported per memory mode.

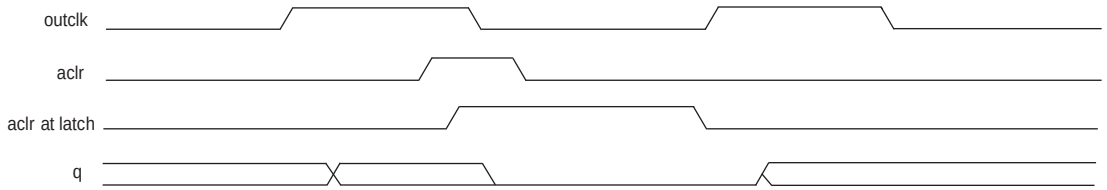
You cannot use the ECC on M144 memory blocks when using the mixed width support.

MLABs do not support mixed-width FIFO mode.

Asynchronous Clear

Stratix III M9K and M144K memory blocks support asynchronous clears on the output latches and output registers. MLABs supports asynchronous clear on the output registers only as the output is not latched. Therefore, if your M9K and M144K are not using the output registers, you can still clear the RAM outputs via the output latch asynchronous clear. The functional waveform in Figure 4–7 shows this functionality.

Figure 4–7. Output Latch Asynchronous Clear Waveform



You can selectively enable asynchronous clears per logical memory via the Quartus II RAM MegaWizard Plug-In Manager.

For more information, refer to the *RAM Megafunction User Guide*.

Error Correction Code Support

Stratix III M144K blocks have built-in support for error correction code (ECC) when in ×64-wide simple dual-port mode. ECC allows you to detect and correct data errors in the memory array. The M144K blocks have a single-error-correction double-error-detection (SECDED) implementation. SECDED can detect and fix a single-bit error in a 64-bit word or detect two-bit errors in a 64-bit word. It cannot detect three or more errors.

The M144K ECC status is communicated via a three-bit status flag `eccstatus[2..0]`. The status flag can be either registered or unregistered. When registered, it uses the same clock and asynchronous clear signals as the output registers. When not registered, it cannot be asynchronously cleared.

Table 4–3 shows the truth table for the ECC status flags.

Table 4–3. Truth Table for ECC Status Flags

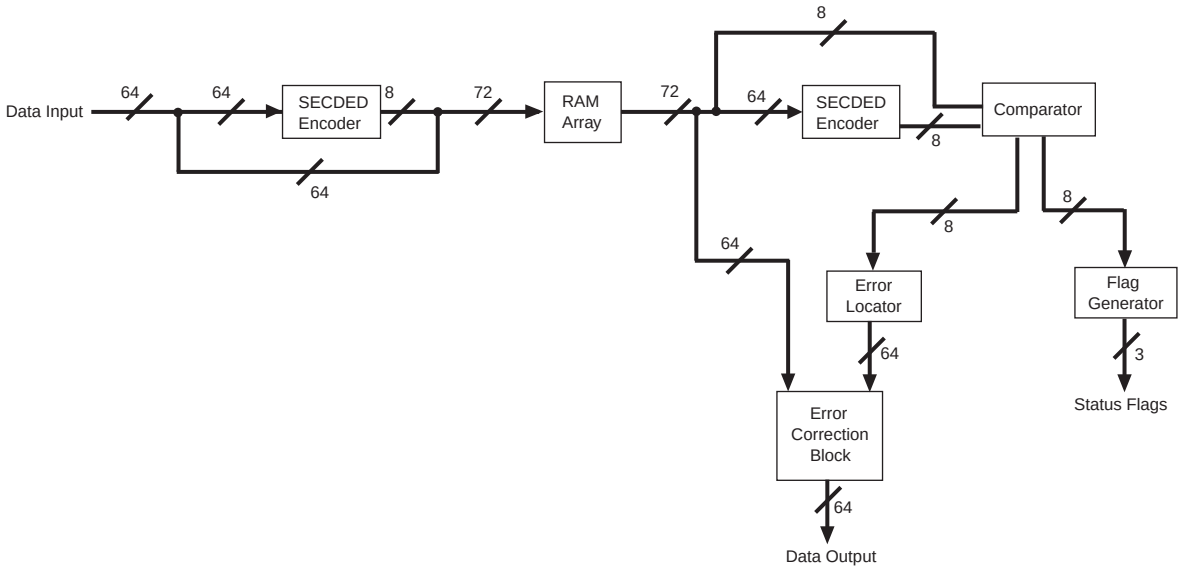
Status	eccstatus[2]	eccstatus[1]	eccstatus[0]
No error	0	0	0
Single error and fixed	0	1	1
Double error and no fix	1	0	1
Illegal	0	0	1
Illegal	0	1	0
Illegal	1	0	0
Illegal	1	1	X

You cannot use the byte-enable feature when ECC is engaged.

Read during write “old data” mode is not supported when ECC is engaged.

Figure 4–8 shows a block diagram of the ECC block of the M144K.

Figure 4–8. ECC Block Diagram of the M144K



Memory Modes

Stratix III TriMatrix memory blocks allow you to implement fully synchronous SRAM memory in multiple modes of operation. M9K and M144K blocks do not support asynchronous memory (unregistered inputs). MLABs support asynchronous (flow-through) read operations.

Depending on which TriMatrix memory block you target, the following modes may be used:

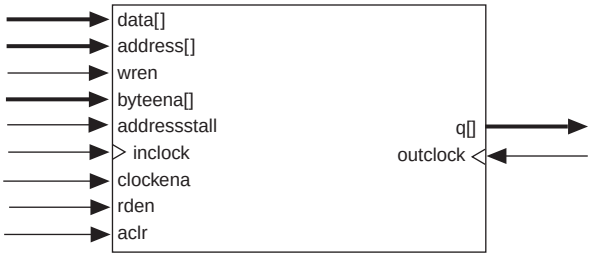
- Single-port
- Simple dual-port
- True dual-port
- Shift-register
- ROM
- FIFO

When using the memory blocks in ROM, single-port, simple dual-port, or true dual-port mode, you can corrupt the memory contents if you violate the setup or hold-time on any of the memory block input registers. This applies to both read and write operations.

Single Port RAM

All TriMatrix memory blocks support single-port mode. Single-port mode allows you to do either one read or one write operation at a time. Simultaneous reads and writes are not supported in single-port mode. Figure 4–9 shows the single-port RAM configuration.

Figure 4–9. Single-Port Memory *(Note 1)*



Note to Figure 4–9:

(1) You can implement two single-port memory blocks in a single M9K or M144K block. See "Packed Mode Support" on page 4–5 for more details.

During a write operation, behavior of the RAM outputs is configurable. If you use the read-enable signal and perform a write operation with the read enable deactivated, the RAM outputs retain the values they held during the most recent active read enable. If you activate read enable during a write operation, or if you are not using the read-enable signal at all, the RAM outputs either show the new data being written, the old data at that address, or a don't care value. To choose the desired behavior, set the read-during-write behavior to either new data, old data, or don't care in the RAM MegaWizard Plug-In Manager in the Quartus II software. See "Read During Write" on page 4–21 for more details on this behavior.

Table 4–4 shows the possible port width configurations for TriMatrix memory blocks in single-port mode.

Table 4–4. Stratix III Port Width Configurations for MLABs, M9K Blocks, and M144K Blocks (Single-Port Mode)

Port Width	MLABs *(1)*	M9K Blocks	M144K Blocks
Port Width Configurations	16 × 8	8 K × 1	16 K × 8
	16 × 9	4 K × 2	16 K × 9
	16 × 10	2 K × 4	8 K × 16
	16 × 16	1 K × 8	8 K × 18
	16 × 18	1 K × 9	4 K × 32
	16 × 20	512 × 16	4 K × 36
		512 × 18	2 K × 64
		256 × 32	2 K × 72
		256 × 36	

Note to Table 4–4:

(1) Configurations of 64 × 8, 64 × 9, 64 × 10, 32 × 16, 32 × 18, and 32 × 20 are supported by stitching multiple MLAB blocks.

Figure 4–10 shows the timing waveforms for read and write operations in single-port mode with unregistered outputs for M9K and M144K. In M9K and M144K registering the RAM's outputs would simply delay the q output by one clock cycle.

Figure 4–10. Timing Waveform for Read-Write Operations (Single-Port Mode) for M9K and M144K

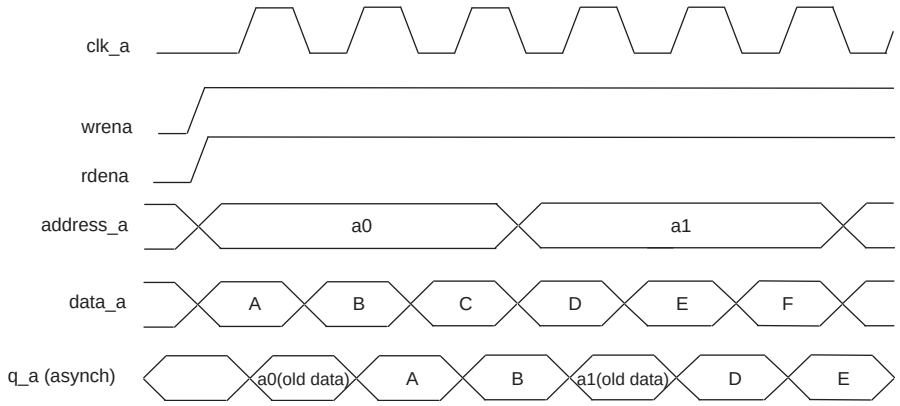
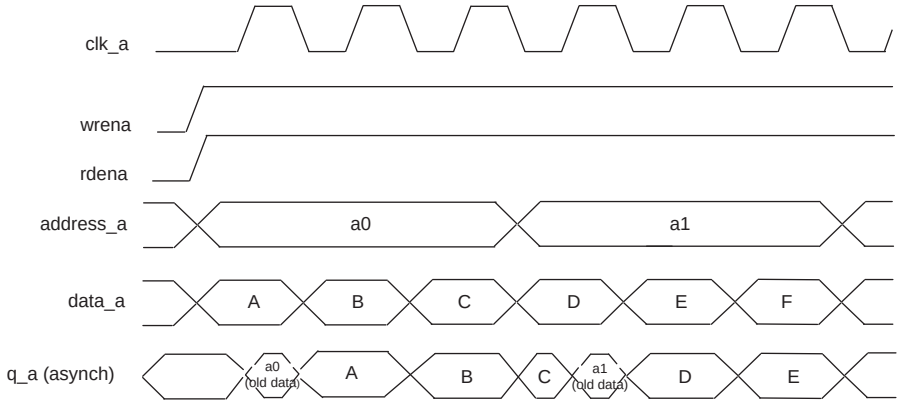


Figure 4–11 shows the timing waveforms for read and write operations in single-port mode with unregistered outputs for MLABs. For MLABs, the read operation is triggered by the rising clock edges whereas the write operation is triggered by the falling clock edges.

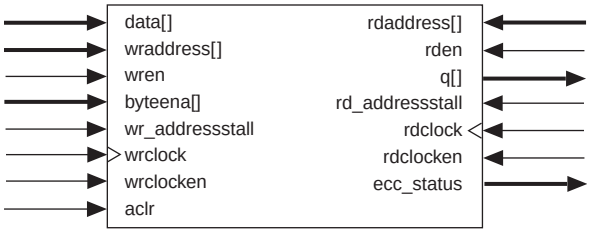
Figure 4–11. Timing Waveform for Read-Write Operations (Single-Port Mode) for MLABs



Simple Dual-Port Mode

All TriMatrix memory blocks support simple dual-port mode. Simple dual-port mode allows you to perform one-read and one-write operation to different locations at the same time. Figure 4–12 shows the simple dual-port configuration.

Figure 4–12. Stratix III Simple Dual-Port Memory *(Note 1)*



Note to Figure 4–12:

(1) Simple dual-port RAM supports input/output clock mode in addition to the read/write clock mode shown.

Simple dual-port mode supports different read and write data widths (mixed width support). Table 4–5 shows the mixed width configurations for the M9K blocks in simple dual-port mode. MLABs do not have native support for mixed width operation. The Quartus II software can implement mixed width memories in MLABs by using more than one MLAB.

Table 4–5. Stratix III M9K Block Mixed-Width Configurations (Simple Dual-Port Mode)

Read Port	Write Port								
	8K×1	**4K×2**	**2K×4**	**1K×8**	**512×16**	**256×32**	**1K×9**	**512×18**	**256×36**
8K×1	✓	✓	✓	✓	✓	✓	—	—	—
4K×2	✓	✓	✓	✓	✓	✓	—	—	—
2K×4	✓	✓	✓	✓	✓	✓	—	—	—
1K×8	✓	✓	✓	✓	✓	✓	—	—	—
512×16	✓	✓	✓	✓	✓	✓	—	—	—
256×32	✓	✓	✓	✓	✓	✓	—	—	—
1K×9	—	—	—	—	—	—	✓	✓	✓
512×18	—	—	—	—	—	—	✓	✓	✓
256×36	—	—	—	—	—	—	✓	✓	✓

Table 4–6 shows the mixed width configurations for the M144K blocks in simple dual-port mode.

Table 4–6. Stratix III M144K Block Mixed-Width Configurations (Simple Dual-Port Mode)

Read Port	Write Port							
	16K×8	**8K×16**	**4K×32**	**2K×64**	**16K×9**	**8K×18**	**4K×36**	**2K×72**
16K×8	✓	✓	✓	✓	—	—	—	—
8K×16	✓	✓	✓	✓	—	—	—	—
4K×32	✓	✓	✓	✓	—	—	—	—
2K×64	✓	✓	✓	✓	—	—	—	—
16K×9	—	—	—	—	✓	✓	✓	✓
8K×18	—	—	—	—	✓	✓	✓	✓
4K×36	—	—	—	—	✓	✓	✓	✓
2K×72	—	—	—	—	✓	✓	✓	✓

In simple dual-port mode, M9K and M144K blocks support separate write-enable and read-enable signals. You can save power by keeping the read-enable signal low (inactive) when not reading. Read-during-write operations to the same address can either output a don't care value or old data. To choose the desired behavior, set the read-during-write behavior to either don't care or old data in the RAM MegaWizard Plug-In Manager in the Quartus II software. See "Read During Write" on page 4–21 for more details about this behavior.

MLABs only support a write-enable signal. Read-during-write behavior for the MLABs can be either don't care, new data, or old data. The available choices depend on the configuration of the MLAB.

Figure 4–13 shows the timing waveforms for read and write operations in simple dual-port mode with unregistered outputs in M9K and M144K. Registering the RAM's outputs would simply delay the q output by one clock cycle in M9k and M144K.

Figure 4–13. Stratix III Simple Dual-Port Timing Waveforms for M9K and M144K

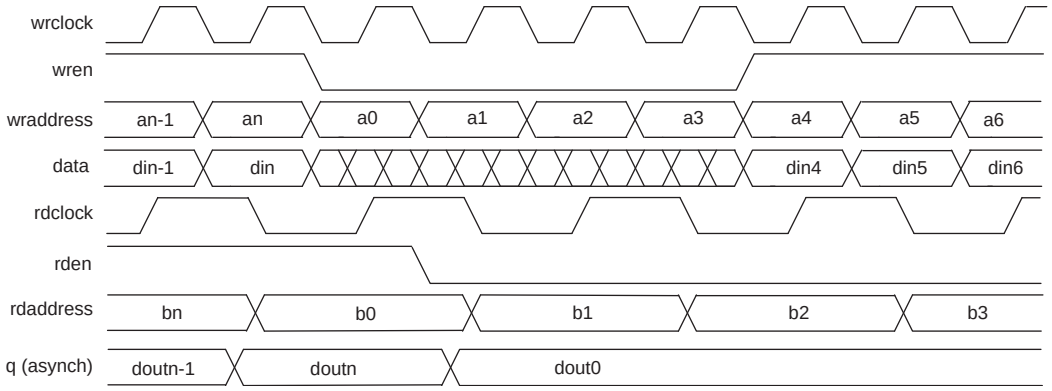
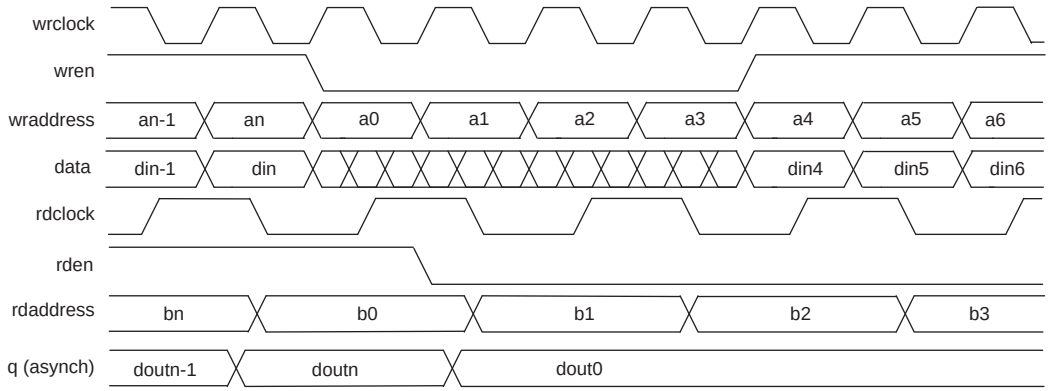


Figure 4–14 shows the timing waveforms for read and write operations in simple dual-port mode with unregistered outputs in MLABs. In MLABs, the write operation is triggered by the falling clock edges.

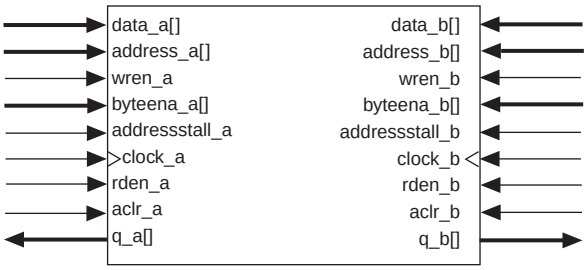
Figure 4–14. Stratix III Simple Dual-Port Timing Waveforms for MLABs



True Dual-Port Mode

Stratix III M9K and M144K blocks support true dual-port mode. Sometimes called bi-directional dual-port, this mode allows you to perform any combination of two port operations: two reads, two writes, or one read and one write at two different clock frequencies. Figure 4–15 shows the true dual-port RAM configuration.

Figure 4–15. Stratix III True Dual-Port Memory *(Note 1)*



Note to Figure 4–15:

(1) True dual-port memory supports input/output clock mode in addition to the independent clock mode shown.

The widest bit configuration of the M9K and M144K blocks in true dual-port mode is as follows:

- 512 × 16-bit (×18-bit with parity) (M9K)
- 4K × 32-bit (×36-bit with parity) (M144K)

Wider configurations are unavailable because the number of output drivers is equivalent to the maximum bit width of the respective memory block. Because true dual-port RAM has outputs on two ports, its maximum width equals half of the total number of output drivers. Table 4–7 lists the possible M9K block mixed-port width configurations in true dual-port mode.

Table 4–7. Stratix III M9K Block Mixed-Width Configuration (True Dual-Port Mode)

Read Port	Write Port						
	8K×1	4K×2	2K×4	1K×8	512×16	1K×9	512×18
8K×1	✓	✓	✓	✓	✓	—	—
4K×2	✓	✓	✓	✓	✓	—	—
2K×4	✓	✓	✓	✓	✓	—	—
1K×8	✓	✓	✓	✓	✓	—	—
512×16	✓	✓	✓	✓	✓	—	—
1K×9	—	—	—	—	—	✓	✓
512×18	—	—	—	—	—	✓	✓

Table 4–8 lists the possible M144K block mixed-port width configurations in true dual-port mode.

Table 4–8. Stratix III M144K Block Mixed-Width Configurations (True Dual-Port Mode)

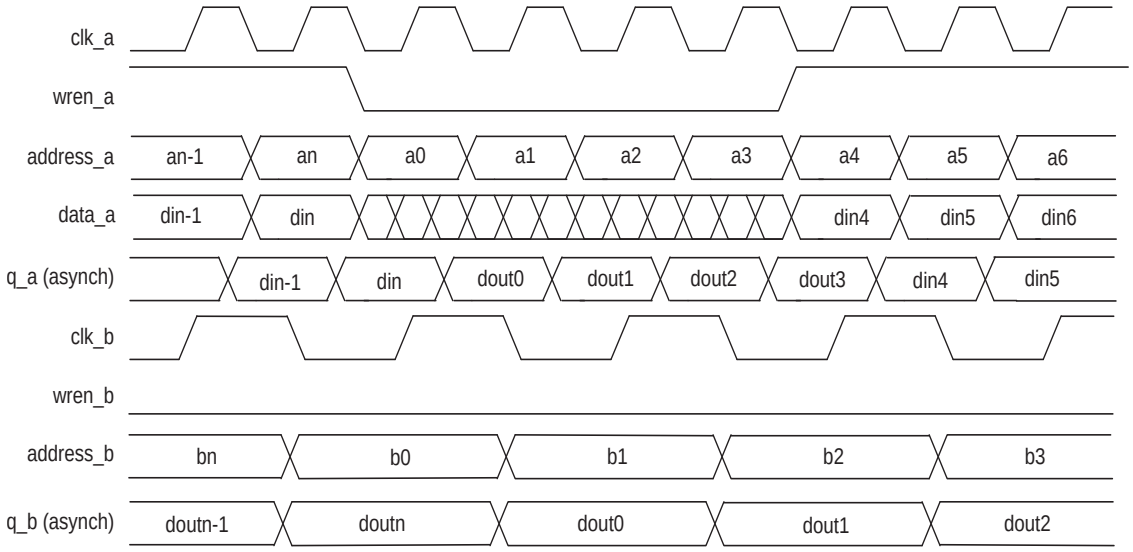
Read Port	Write Port					
	16K×8	8K×16	4K×32	16K×9	8K×18	4K×36
16K×8	✓	✓	✓	—	—	—
8K×16	✓	✓	✓	—	—	—
4K×32	✓	✓	✓	—	—	—
16K×9	—	—	—	✓	✓	✓
8K×18	—	—	—	✓	✓	✓
4K×36	—	—	—	✓	✓	✓

In true dual-port mode, M9K and M144K blocks support separate write-enable and read-enable signals. You can save power by keeping the read-enable signal low (inactive) when not reading. Read-during-write operations to the same address can either output new data at that location or old data. To choose the desired behavior, set the read-during-write behavior to either new data or old data in the RAM MegaWizard Plug-In Manager in the Quartus II software. See "Read During Write" on page 4–21 for more details about this behavior.

In true dual-port mode you can access any memory location at any time from either port. When accessing the same memory location from both ports, you must avoid possible write conflicts. A write conflict happens when you attempt to write to the same address location from both ports at the same time. This results in unknown data being stored to that address location. No conflict resolution circuitry is built into the Stratix III TriMatrix memory blocks. You must handle address conflicts external to the RAM block.

Figure 4–16 shows the true dual-port timing waveforms for the write operation at port A and read operation at port B with the Read-During-Write behavior set to new data. Registering the RAM's outputs would simply delay the `q` outputs by one clock cycle.

Figure 4–16. Stratix III True Dual-Port Timing Waveform



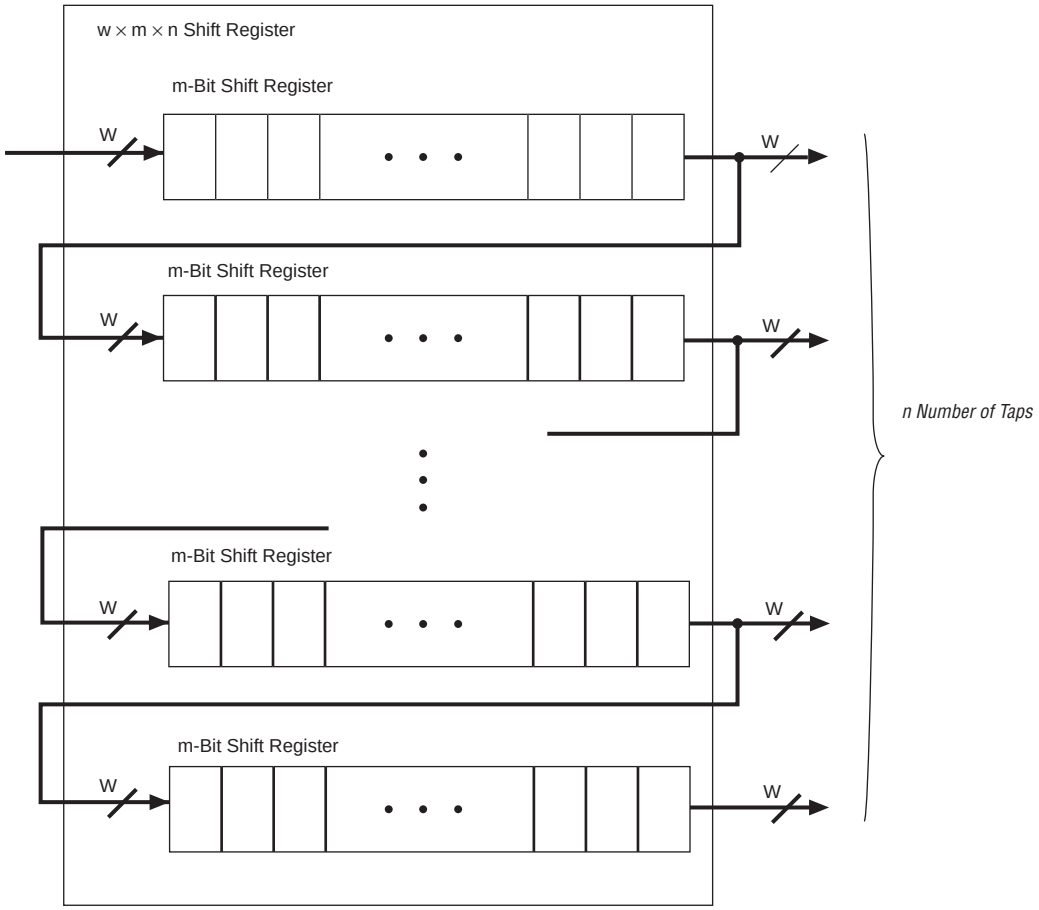
Shift-Register Mode

All Stratix III memory blocks support shift register mode. Embedded memory block configurations can implement shift registers for digital signal processing (DSP) applications, such as finite impulse response (FIR) filters, pseudo-random number generators, multi-channel filtering, and auto- and cross-correlation functions. These and other DSP applications require local data storage, traditionally implemented with standard flipflops that quickly exhaust many logic cells for large shift registers. A more efficient alternative is to use embedded memory as a shift-register block, which saves logic cell and routing resources.

The size of a shift register (`w × m × n`) is determined by the input data width (`w`), the length of the taps (`m`), and the number of taps (`n`). You can cascade memory blocks to implement larger shift registers.

Figure 4–17 shows the TriMatrix memory block in shift-register mode.

Figure 4–17. Stratix III Shift-Register Memory Configuration



ROM Mode

All Stratix III TriMatrix memory blocks support ROM mode. A **.mif** file initializes the ROM content of these blocks. The address lines of the ROM are registered on M9K and M144K blocks, but can be unregistered on MLABs. The outputs can be registered or unregistered. Output registers can be asynchronously cleared. The ROM read operation is identical to the read operation in the single-port RAM configuration.

FIFO Mode

All TriMatrix memory blocks support FIFO mode. MLABs are ideal for designs with many small, shallow FIFO buffers. To implement FIFO buffers in your design, use the Quartus II software FIFO MegaWizard Plug-In Manager. Both single and dual-clock (asynchronous) FIFOs are supported.

For more information about implementing FIFO buffers, refer to the *Single- and Dual-Clock FIFO Megafunctions User Guide*.

MLABs do not support mixed-width FIFO mode.

Clocking Modes

Stratix III TriMatrix memory blocks support the following clocking modes:

- Independent
- Input/output
- Read/write
- Single clock

Violating the setup or hold time on the memory block address registers could corrupt the memory contents. This applies to both read and write operations.

Altera recommends using a memory block clock that comes through global clock routing from an on-chip PLL set to 50% output duty cycle to achieve the maximum memory block performance. Use Quartus II to report timing for this and other memory block clocking schemes.

For more information refer to the *Stratix III Device Family Errata Sheet*.

Table 4–9 shows the clocking mode versus memory mode support matrix.

Table 4–9. Stratix III TriMatrix Memory Clock Modes

Clocking Mode	True Dual-Port Mode	Simple Dual-Port Mode	Single-Port Mode	ROM Mode	FIFO Mode
Independent	✓	—	—	✓	—
Input/output	✓	✓	✓	✓	—
Read/write	—	✓	—	—	✓
Single clock	✓	✓	✓	✓	✓

Independent Clock Mode

Stratix III TriMatrix memory blocks can implement independent clock mode for true dual-port memories. In this mode, a separate clock is available for each port (A and B). Clock A controls all registers on the port A side, while clock B controls all registers on the port B side. Each port also supports independent clock enables for port A and port B registers. Asynchronous clears are supported only for output latches and output registers on both ports.

Input/Output Clock Mode

Stratix III TriMatrix memory blocks can implement input/output clock mode for true and simple dual-port memories. In this mode, an input clock controls all registers related to the data input to the memory block, including data, address, byte-enables, read enables, and write enables. An output clock controls the data output registers. Asynchronous clears are available on output latches and output registers only.

Read/Write Clock Mode

Stratix III TriMatrix memory blocks can implement read/write clock mode for simple dual-port memories. In this mode, a write clock controls the data-input, write-address, and write-enable registers. Similarly, a read clock control the data-output, read-address, and read-enable registers. The memory blocks support independent clock enables for both the read and write clocks. Asynchronous clears are available on data output latches and registers only.

When using read/write mode, if you perform a simultaneous read/write to the same address location, the output read data will be unknown. If you require the output data to be a known value in this case, use either single-clock mode or input/output clock mode and choose the appropriate read-during-write behavior in the Megawizard.

Single Clock Mode

Stratix III TriMatrix memory blocks can implement single-clock mode for true dual-port, simple dual-port, and single-port memories. In this mode, a single clock, together with a clock enable, is used to control all registers of the memory block. Asynchronous clears are available on output latches and output registers only.

Design Considerations

This section describes guidelines for designing with TriMatrix memory blocks.

Selecting TriMatrix Memory Blocks

The Quartus II software automatically partitions user-defined memory into embedded memory blocks by taking into account both speed and size constraints placed on your design. For example, the Quartus II software may spread out a memory across multiple memory blocks when resources are available to increase the performance of the design. You can manually assign the memory to a specific block size via the RAM MegaWizard Plug-In Manager.

MLABs can implement single-port SRAM through emulation via the Quartus II software. Emulation results in minimal additional logic resources being used. Because of the dual-purpose architecture of the MLAB, it only has data input registers and output registers in the block. MLABs gain input address registers and additional optional data output registers from adjacent ALMs by using register packing.

For more information about register packing, refer to the *Logic Array Blocks and Adaptive Logic Modules in Stratix III Devices* chapter in volume 1 of the *Stratix III Device Handbook*.

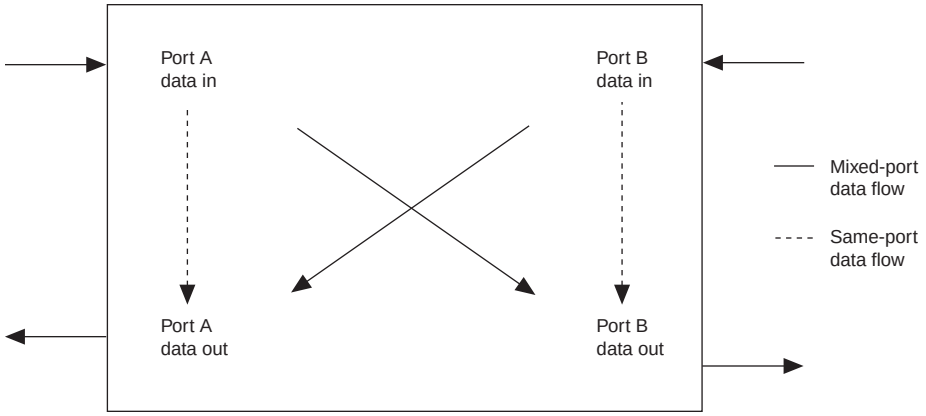
Conflict Resolution

When using the memory blocks in true dual-port mode, it is possible to attempt two write operations to the same memory location (address). Since no conflict resolution circuitry is built into the memory blocks, this results in unknown data being written to that location. Therefore, you must implement conflict resolution logic external to the memory block to avoid address conflicts.

Read During Write

You can customize the read-during-write behavior of the Stratix III TriMatrix memory blocks to suit your design needs. Two types of read-during-write operations are available: same port and mixed port. Figure 4–18 shows the difference between the two types.

Figure 4–18. Stratix III Read-During-Write Data Flow



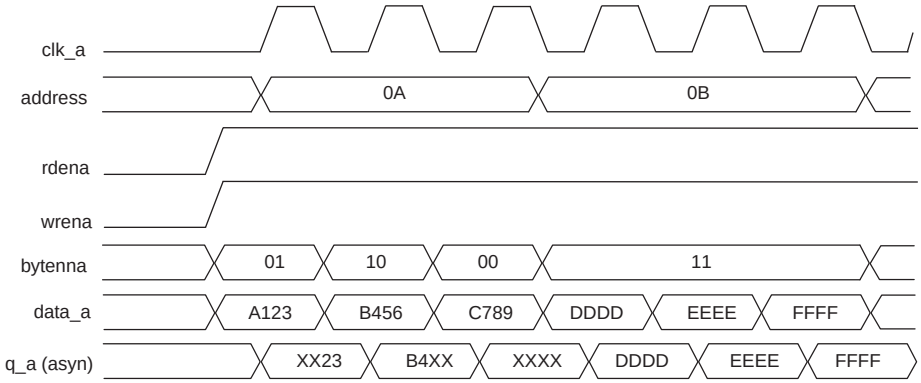
Same-Port Read-During-Write Mode

This mode applies to either a single-port RAM or the same port of a true dual-port RAM. In same-port read-during-write mode, three output choices are available: new data mode (or flow-through), old data mode, or don't care mode. In new data mode, the new data is available on the rising edge of the same clock cycle on which it was written. In old data mode, the RAM outputs reflect the old data at that address before the write operation proceeds. In don't care mode, the RAM outputs don't care values for a read-during-write operation.

If you are not using the new data mode or old data mode, you should select the don't care mode. Using the don't care mode increases the flexibility in the type of memory block used, provided you do not assign block type when instantiating a memory block. You may also get potential performance gain by selecting the don't care mode.

Figure 4–19 shows the sample functional waveforms of same-port read-during-write behavior with new data.

Figure 4–19. Same Port Read-During-Write: New Data Mode *(Note 1)*

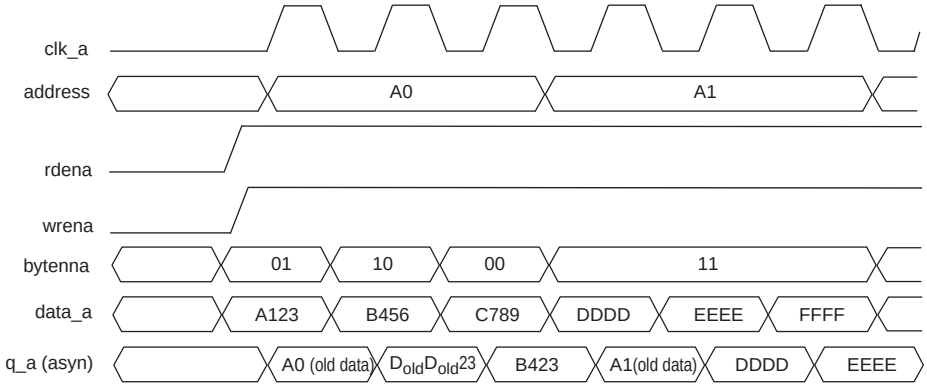


Note to Figure 4–19:

(1) "X" can be a don't care value or current data at that location, depending on the setting chosen in the Quartus II software.

Figure 4–20 shows the sample functional waveforms of same-port read-during-write behavior with old data mode.

Figure 4–20. Same Port Read-During-Write: Old Data Mode *(Note 1)*



Note to Figure 4–20:

(1) D_{old} is the old data bit at address A0, A0 (old data) is the old data at address A0, and A1 (old data) is the old data at address A1.

Mixed-Port Read-During-Write Mode

This mode applies to a RAM in simple or true dual-port mode which has one port reading and the other port writing to the same address location with the same clock.

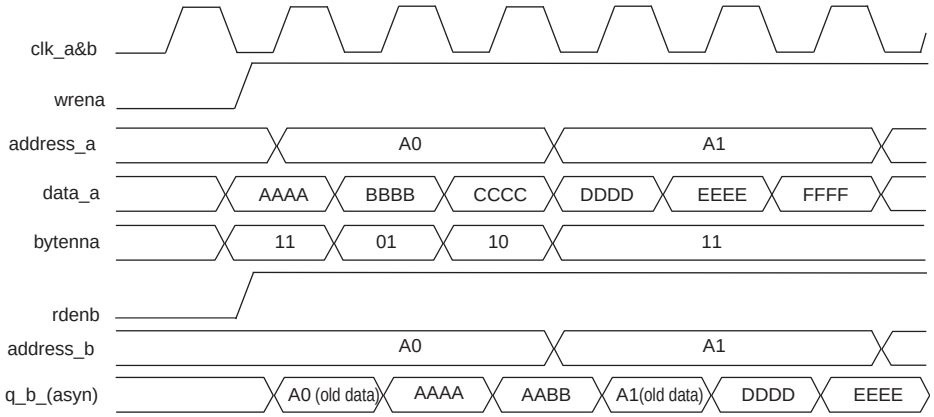
In this mode you also have two output choices: old data or don't care. In old data mode, a read-during-write operation to different ports causes the RAM outputs to reflect the old data at that address location. In don't care mode, the same operation results in a "don't care" or "unknown" value on the RAM outputs.

For more details about how to implement the desired behavior, read-during-write behavior is controlled via the RAM MegaWizard Plug-In Manager refer to the *RAM Megafunction User Guide*.

You should select don't care mode if you do not use old data mode. This increases the flexibility in the type of memory block used, if you do not assign block type when instantiating a memory block. You may also get potential performance gain by selecting don't care mode.

Figure 4–21 shows a sample functional waveform of mixed-port read-during-write behavior for the old data mode. In don't care mode, the old data shown in the figure is simply replaced with "**don't cares**".

Figure 4–21. Mixed Port Read During Write: Old Data Mode *(Note 1)*



Note to Figure 4–21:

(1) A0 (old data) is the old data at address A0 and A1 (old data) is the old data at address A1.

Mixed-port read-during-write using two different clocks in simple-dual port RAM with old data output is supported via emulation. The Quartus II software takes two memory blocks to implement the widest width mode.

Power-Up Conditions and Memory Initialization

M9K and M144K memory block outputs power up to zero (cleared), regardless of whether the output registers are used or bypassed. MLABs power up to zero if output registers are used and power up reading the memory contents if output registers are not used. However, the actual RAM cells power up to an unknown state. Therefore, after power-up, if an address is read before being written, the output from the read operation is undefined because the contents are not initialized.

All memory blocks support initialization via **.mif** file. You can create **.mif** files in the Quartus II software and specify their use with the RAM MegaWizard Plug-In Manager when instantiating a memory in your design. Even if a memory is pre-initialized (for example, by a **.mif** file), it still powers up with its outputs cleared.

For more information about **.mif** files, refer to the *RAM Megafunction User Guide* and the *Quartus II Handbook*.

Power Management

Stratix III memory block clock-enables allow you to control clocking of each memory block to reduce AC power consumption. Use the read-enable signal to ensure that read operations only occur when you need them to. If your design does not require read-during-write, you can reduce your power consumption by de-asserting the read-enable signal during write operations, or any period when no memory operations occur.

The Quartus II software automatically places any unused memory blocks in low power mode to reduce static power.

Programming File Compatibility

Beginning with version 8.1, the Quartus II software supports the logic option **STRATIXIII_MRAM_COMPATIBILITY**. When this option is set to on, the Quartus II software will generate programming files compatible with both affected and fixed silicons (for write speed decrease in M144K blocks). The default setting for this option is on.

For the list of devices that is affected by the write speed decrease for M144K blocks refer to the *Stratix III Device Family Errata Sheet*.

To set the **STRATIXIII_MRAM_COMPATIBILITY** variable, enter the following line in the Quartus Settings File:

```
set_global_assignment -name STRATIXIII_MRAM_COMPATIBILITY ON
```

When targeting fixed silicon devices, set the **STRATIXIII_MRAM_COMPATIBILITY** variable to OFF. When the **STRATIXIII_MRAM_COMPATIBILITY** option is set to OFF, you will be able to achieve the higher F_{MAX} that is published for M144K blocks in fixed silicons and the programming files will only be compatible with fixed silicons. These programming files will not configure other silicon revisions. The `nSTATUS` pin will drive out low and configuration will fail.

Conclusion

The Stratix III TriMatrix embedded memory structure provides three different on-chip RAM block sizes to address your design needs. All memory blocks are fully customizable and can be cascaded to implement wider or deeper memories with minimal speed penalty.

You can independently configure each embedded memory block to be a single- or dual-port RAM, FIFO, ROM, or shift register via the Quartus II MegaWizard Plug-In Manager software.

Chapter Revision History

Table 4–10 shows the revision history for this chapter.

Table 4–10. Chapter Revision History

Date and Revision	Changes Made	Summary of Changes
May 2009, version 1.8	■ Updated Table 4–1. ■ Updated "Read/Write Clock Mode" and "Simple Dual-Port Mode" sections.	—
February 2009, version 1.7	■ Updated Figure 4–2, Figure 4–4, and Figure 4–5. ■ Removed "Referenced Documents" section.	—
November 2008, Version 1.6	■ Updated "Byte-Enable Support", "Address Clock Enable Support", "Asynchronous Clear", "Single Port RAM", and "Simple Dual-Port Mode" sections. ■ Updated Figure 4–1, Figure 4–5, Figure 4–8, Figure 4–10, and Figure 4–15. ■ Added Figure 4–2, Figure 4–6, Figure 4–11, Figure 4–14, and Figure 4–16.	—
October 2008, version 1.5	■ Updated Table 4–1. ■ Updated "Asynchronous Clear" and "Clocking Modes" section. ■ Added "Programming File Compatibility" section. ■ Updated New Document Format.	—
May 2008, version 1.4	■ Updated "Introduction" section. ■ Updated "TriMatrix Memory Block Types" section. ■ Updated "Byte-Enable Support" section. ■ Updated "Mixed Width Support" section. ■ Updated "Same-Port Read-During-Write Mode" section. ■ Updated Figure 4–16, Figure 4–17, and Figure 4–18. ■ Updated "Mixed-Port Read-During-Write Mode" section. ■ Updated Table 4–1, Table 4–2, and Table 4–4.	—
November 2007, version 1.3	Updated Table 4–2.	—
October 2007, version 1.2	■ Updated Table 4–1. ■ Added section "Referenced Documents". ■ Added live links for references.	—
May 2007, version 1.1	Updated Table 4–2, Table 4–9.	—
November 2006, version 1.0	Initial Release.	—

5. DSP Blocks in Stratix III Devices

SII51005-1.7

Introduction

The Stratix® III family of devices have dedicated high-performance digital signal processing (DSP) blocks optimized for DSP applications. These DSP blocks of the Altera® Stratix device family are the third generation of hardwired, fixed function silicon blocks dedicated to maximizing signal processing capability, ease of use, and lowest silicon cost.

Many complex systems such as WiMAX, 3GPP WCDMA, high-performance computing (HPC), voice over Internet protocol (VoIP), H.264 video compression, medical imaging, and HDTV use sophisticated digital signal processing techniques, and this typically requires a large number of mathematical computations. Stratix III devices are ideally suited as the DSP blocks consist of a combination of dedicated elements that perform multiplication, addition, subtraction, accumulation, summation, and dynamic shift operations. Along with the high-performance Stratix III soft logic fabric and TriMatrix™ memory structures, you can configure these blocks to build sophisticated fixed-point and floating-point arithmetic functions. These can be manipulated easily to implement common larger computationally intensive subsystems such as finite impulse response (FIR) filters, complex FIR filters, infinite impulse response (IIR) filters, fast Fourier transform (FFT) functions, and discrete cosine transform (DCT) functions.

DSP Block Overview

Each Stratix III device has two to seven columns of DSP blocks that implement multiplication, multiply-add, multiply-accumulate (MAC), and dynamic shift functions efficiently. The logical functionality of the Stratix III DSP block is a superset of the previous generation of the DSP block found in Stratix and Stratix II devices.

Architectural highlights of the Stratix III DSP block include:

- High-performance, power-optimized, fully registered and pipelined multiplication operations
- Natively supported 9-bit, 12-bit, 18-bit, and 36-bit wordlengths
- Natively supported 18-bit complex multiplications
- Efficiently supported floating-point arithmetic formats (24-bit for single precision and 53-bit for double precision)
- Signed and unsigned input support
- Built-in addition, subtraction, and accumulation units to combine multiplication results efficiently
- Cascading 18-bit input bus to form tap-delay line for filtering applications
- Cascading 44-bit output bus to propagate output results from one block to the next block without external logic support
- Rich and flexible arithmetic rounding and saturation units

- Efficient barrel shifter support
- Loopback capability to support adaptive filtering

Table 5–1 lists the number of DSP blocks for the Stratix III device family.

Table 5–1. Number of DSP Blocks in Stratix III Devices

Family	Device	DSP Blocks	Independent Input and Output Multiplication Operators					Four Multiplier Adder Mode	High Precision Multiplier Adder Mode
			9 × 9 Multipliers	12 × 12 Multipliers	18 × 18 Multipliers	18 × 18 Complex	36 × 36 Multipliers	18 × 18	18 × 36
Stratix III Logic	EP3SL50	27	216	162	108	54	54	216	108
	EP3SL70	36	288	216	144	72	72	288	144
	EP3SL110	36	288	216	144	72	72	288	144
	EP3SL150	48	384	288	192	96	96	384	192
	EP3SL200	72	576	432	288	144	144	576	288
	EP3SE260	96	768	576	384	192	192	768	384
	EP3SL340	72	576	432	288	144	144	576	288
Stratix III Enhanced	EP3SE50	48	384	288	192	96	96	384	192
	EP3SE80	84	672	504	336	168	168	672	336
	EP3SE110	112	896	672	448	224	224	896	448
	EP3SE260 *(1)*	96	768	576	384	192	192	768	384

Note to Table 5–1:

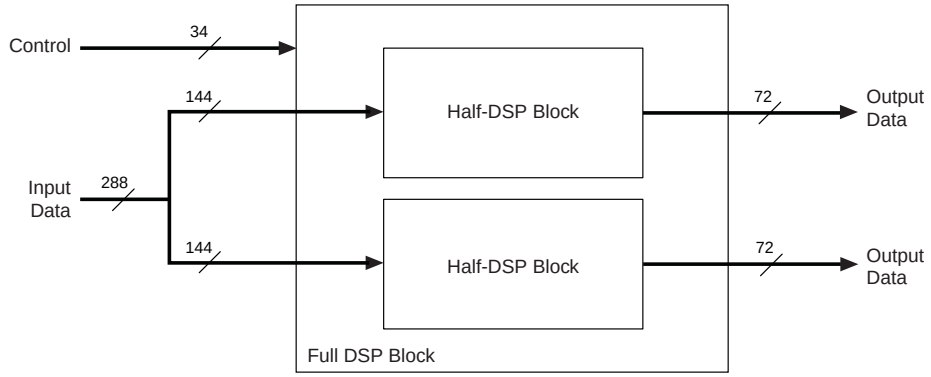
(1) The EP3SE260 device is rich in LE, memory, and multiplier resources. Hence, it aligns with both logic (*L*) and enhanced (*E*) variants.

Table 5–1 lists that the largest Stratix III DSP centric device (EP3SE110) provides up to 896 18 × 18 multiplier functionality in the 36 × 36, complex 18 × 18, and summation modes.

Each DSP block occupies four LAB blocks in height and can be divided further into two half-blocks that share some common clock signals, but are for all common purposes identical in functionality. The layout of each block is shown in Figure 5–1.

The Stratix III DSP block input data lines of 288-bits are double that of Stratix and Stratix II, but the number of output data lines remains at 144 bits.

Figure 5–1. Overview of DSP Block Signals



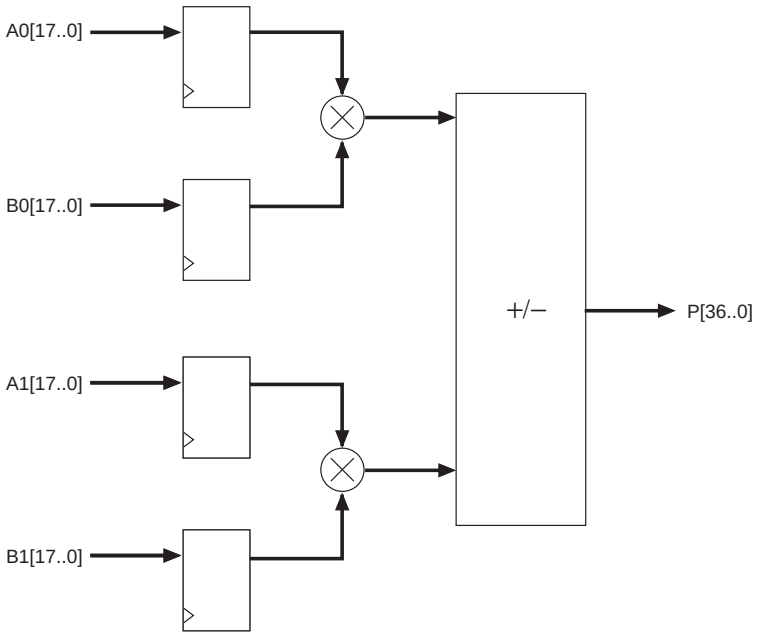
Simplified DSP Operation

In Stratix and Stratix II devices, the fundamental building block consists of an 18-bit × 18-bit multiplier that can also function as two 9-bit × 9-bit multipliers. For Stratix III, the fundamental building block is a pair of 18-bit × 18-bit multipliers followed by a first-stage 37-bit addition/subtraction unit, as shown in Equation 5–1 and Figure 5–2. Note that for all signed numbers, input and output data is represented in 2's complement format only.

Equation 5–1. Multiplier Equation

$$P[36..0] = A_0[17..0] \times B_0[17..0] \pm A_1[17..0] \times B_1[17..0]$$

Figure 5–2. Basic Two-Multiplier Adder Building Block



The structure shown in Figure 5–2 is very useful for building more complex structures, such as complex multipliers and 36 × 36 multipliers, as described in later sections.

Each Stratix III DSP block contains four Two-Multiplier Adder units (two Two-Multiplier Adder units per half-block). Therefore, there are eight 18 × 18 multiplier functionalities per DSP block.

Following the Two-Multiplier Adder units are the pipeline registers, the second-stage adders, and an output register stage. You can configure the second-stage adders to provide the following alternative functions per Half-Block:

Equation 5–2. Four-Multiplier Adder Equation

$$Z[37..0] = P_0[36..0] + P_1[36..0]$$

Equation 5–3. Four-Multiplier Adder Equation (44-Bit Accumulation)

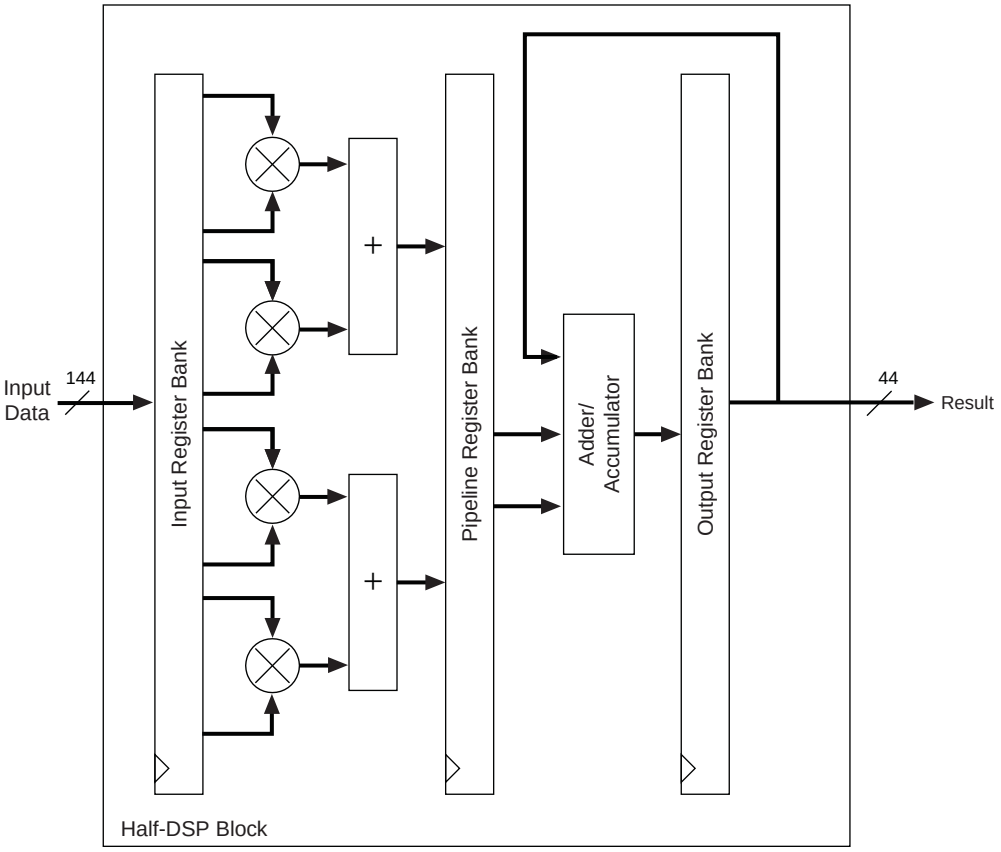
$$W_n[43..0] = W_{n-1}[43..0] \pm Z_n[37..0]$$

In these equations, *n* denotes sample time, and P[36..0] are the results from the Two-Multiplier Adder units.

Equation 5–2 provides a sum of four 18-bit × 18-bit multiplication operations (Four-Multiplier Adder), and Equation 5–3 provides a four 18-bit × 18-bit multiplication operation but with maximum of a 44-bit accumulation capability by feeding the output of the unit back to itself. This is shown in Figure 5–3.

You can bypass all register stages depending on which mode you select.

Figure 5–3. Four-Multiplier Adder and Accumulation Capability

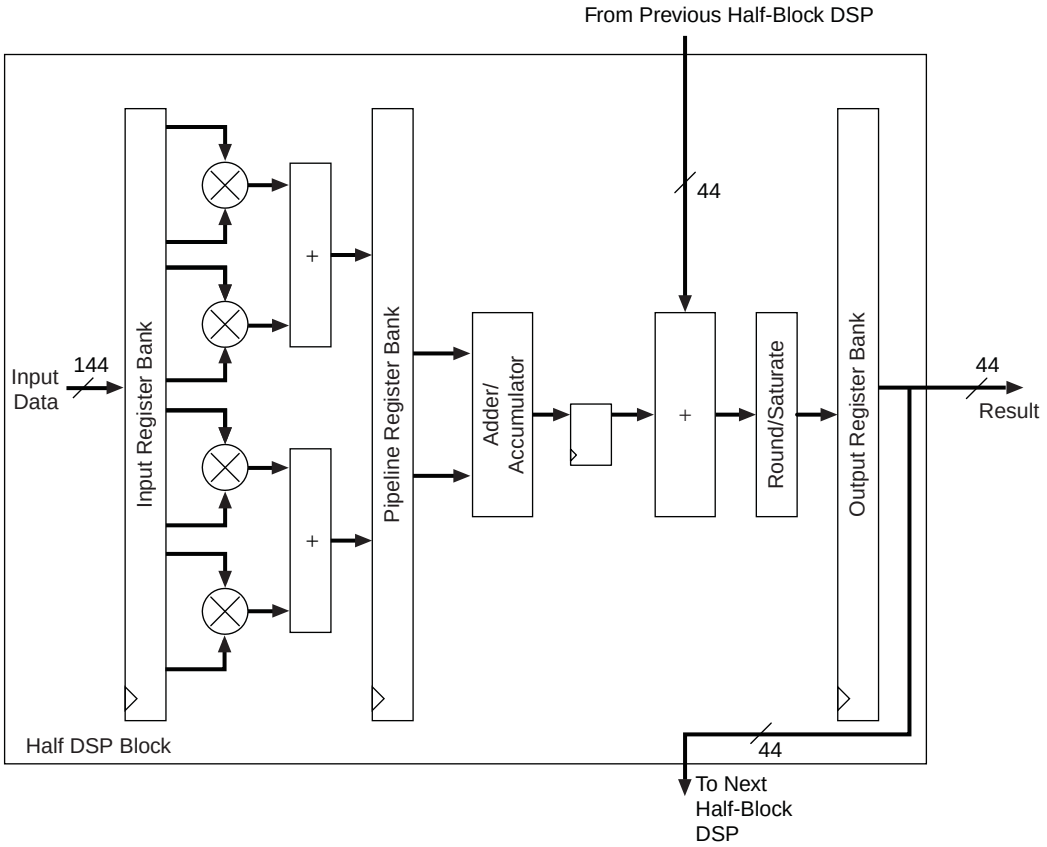


To support commonly found FIR-like structures efficiently, a major addition to the DSP block in Stratix III is the ability to propagate the result of one Half-Block to the next Half-Block completely within the DSP block without additional soft logic overhead. This is achieved by the inclusion of a dedicated addition unit and routing that adds the 44-bit result of a previous Half-Block with the 44-bit result of the current block. The 44-bit result is fed either to the next Half-Block or out of the DSP block through the output register stage. This is shown in Figure 5–4. Detailed examples are described in later sections.

The combination of a fast, low-latency Four-Multiplier Adder unit and the "chained cascade" capability of the output-chaining adder provide an optimal FIR and vector multiplication capability.

To support single-channel type FIR filters efficiently, you can configure one of the multiplier input's registers to form a tap delay line input, saving resources and providing higher system performance.

Figure 5–4. Output Cascading Feature for FIR Structures

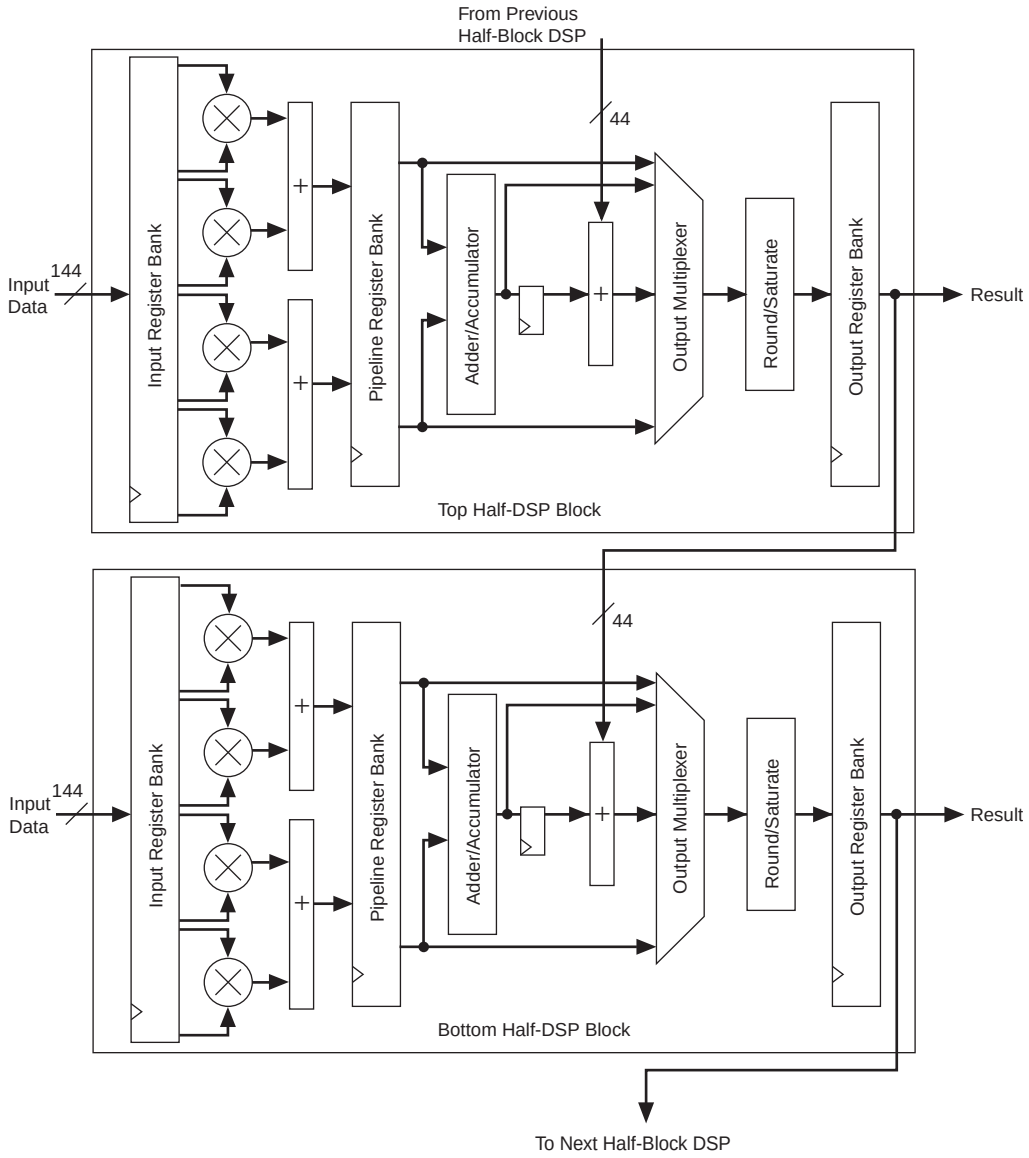


Also shown in Figure 5–4 is the optional Rounding and Saturation Unit (RSU). This unit provides a rich set of commonly found arithmetic round and saturation functions used in signal processing.

In addition to the independent multipliers and sum modes, you can use the DSP blocks to perform shift operations. The DSP block can dynamically switch between logical shift left/right, arithmetic shift left/right, and rotation operation in one clock cycle.

A top-level view of the Stratix III DSP block is shown in Figure 5–5. A more detailed diagram is shown in Figure 5–6.

Figure 5–5. Stratix III Full DSP Block Summary



Operational Modes Overview

Each Stratix III DSP block can be used in one of five basic operational modes. Table 5–2 lists the five basic operational modes and the number of multipliers that can be implemented within a single DSP block, depending on the mode.

Table 5–2. Stratix III DSP Block Operation Modes

Mode	Multiplier in Width	# of Mults	# per Block	Signed or Unsigned	RND, SAT	In Shift Register	Chainout Adder	1st Stage Add/Sub	2nd Stage Add/Acc
Independent Multiplier	9-bits	1	8	Both	No	No	No	—	—
	12-bits	1	6	Both	No	No	No	—	—
	18-bits	1	4	Both	Yes	Yes	No	—	—
	36-bits	1	2	Both	No	No	No	—	—
	Double	1	2	Both	No	No	No	—	—
Two-Multiplier Adder *(1)*	18-bits	2	4	Signed *(4)*	Yes	No	No	Both	—
Four-Multiplier Adder	18-bits	4	2	Both	Yes	Yes	Yes	Both	Add Only
High Precision Multiplier Adder	18 × 36-bits	2	2	Both	No	No	No	—	Add Only
Multiply Accumulate	18-bits	4	2	Both	Yes	Yes	Yes	Both	Both
Shift *(2)*	36-bits *(3)*	1	2	Both	No	No	—	—	—

Notes to Table 5–2:

(1) This mode also supports the loopback mode. In loopback mode, the number of loopback multipliers per DSP block is two and the remaining multipliers can be used in regular Two-Multiplier Adder mode.

(2) The dynamic shift mode supports arithmetic shift left, arithmetic shift right, logical shift left, logical shift right, and rotation operation.

(3) The dynamic shift mode operates on a 32-bit input vector but the multiplier width is configured as 36-bits.

(4) Unsigned value is also supported but you must make sure that the result can be contained within 36-bits.

The DSP block consists of two identical halves (top-half and bottom-half). Each half has four 18 × 18 multipliers.

The Quartus® II software includes megafunctions used to control the mode of operation of the multipliers. After making the appropriate parameter settings using the megafunction's MegaWizard™ Plug-In Manager, the Quartus II software automatically configures the DSP block.

Stratix III DSP blocks can operate in different modes simultaneously. Each half-block is fully independent except for the sharing of the four `clock`, `ena`, and `aclr` signals. For example, you can break down a single DSP block to operate a 9 × 9 multiplier in one Half-Block and an 18 × 18 two-multiplier adder in the other Half-Block. This increases DSP block resource efficiency and allows you to implement more multipliers within a Stratix III device. The Quartus II software automatically places multipliers that can share the same DSP block resources within the same block.

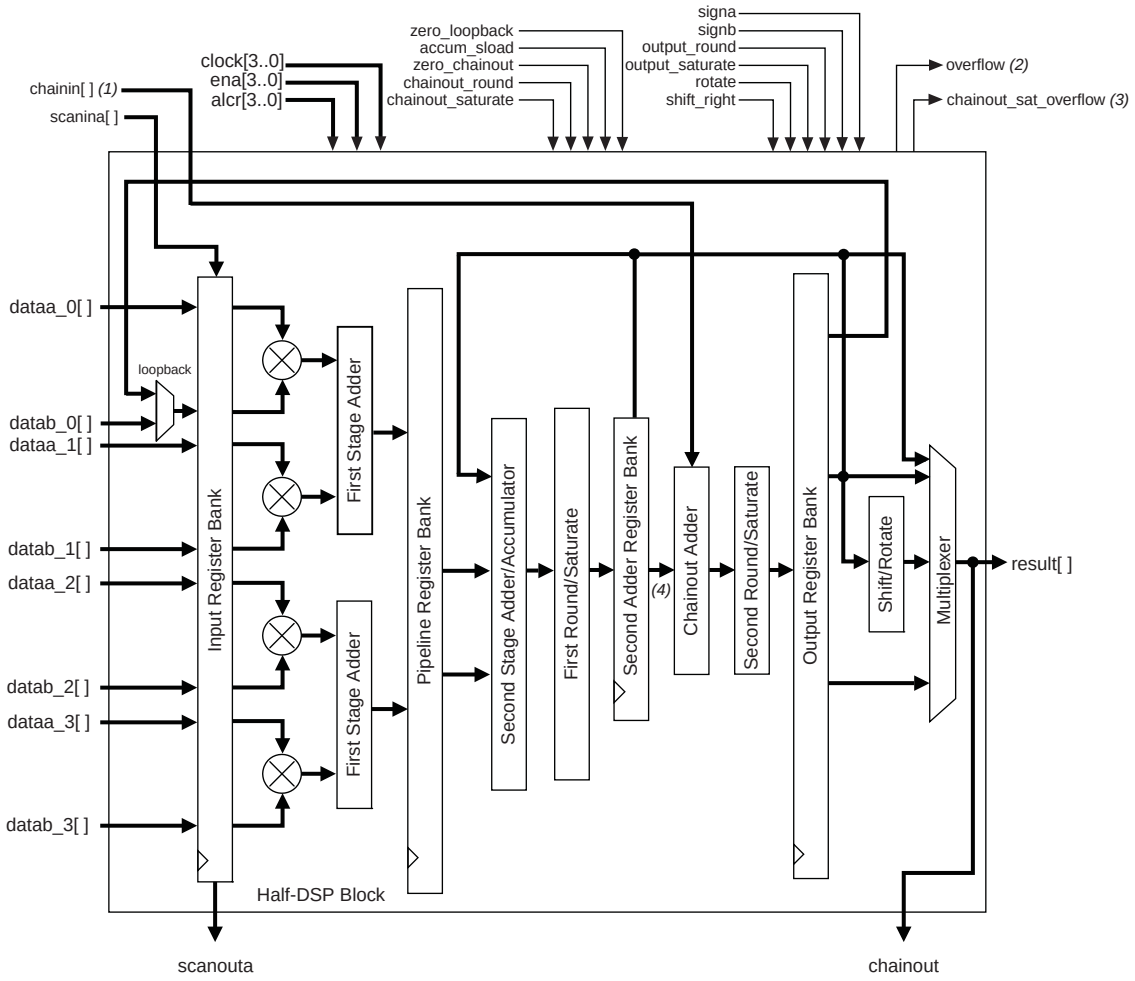
DSP Block Resource Descriptions

The DSP block consists of the following elements:

- Input register bank
- Four Two-Multiplier Adders
- Pipeline register bank
- Two second-stage adders
- Four round and saturation logic units
- Second adder register and output register bank

A detailed overall architecture of the top half of the DSP block is shown in Figure 5–6.

Figure 5–6. Half-DSP Block Architecture



Notes to Figure 5–6:

(1) `chainin[]` can only come from the `chainout` port of the previous DSP blocks and not from general routing.

(2) Block output for accumulator overflow and saturate overflow.

(3) Block output for saturation overflow of `chainout`.

(4) When the chainout adder is not in use, the second adder register banks are known as output register banks.

Input Registers

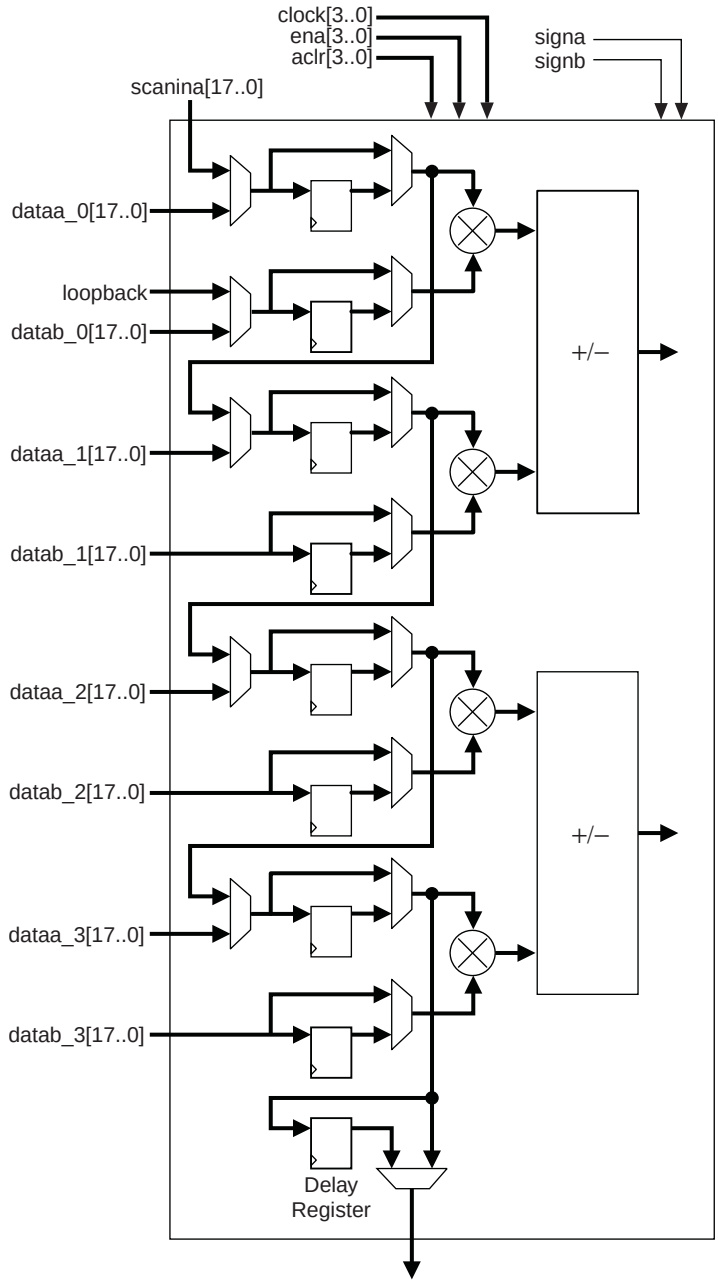
All of the DSP block registers are triggered by the positive edge of the clock signal and are cleared upon power up. Each multiplier operand can feed an input register or directly to the multiplier, bypassing the input registers. (This is configured at compile time.) The following DSP block signals control the input registers within the DSP block:

- `clock[3..0]`
- `ena[3..0]`
- `aclr[3..0]`

Every DSP block has nine 18-bit data input register banks per half DSP block. Every half DSP block has the option to use the eight data register banks as inputs to the four multipliers. The special ninth register bank is a delay register required by modes that use both the cascade and chainout features of the DSP block and is for balancing the latency requirements when using the chained cascade feature.

A feature of the input register bank is to support a tap delay line. Therefore, the top leg of the multiplier input (A) could be driven from general routing or from the cascade chain, as shown in Figure 5–7.

Figure 5–7. Input Register of Half-DSP Block



You must select whether the A-input comes from general routing or from the cascade chain at compile time. In cascade mode, the dedicated shift outputs from one multiplier block directly feeds input registers of the adjacent multiplier below it (within the same half DSP block) or the first multiplier in the next half DSP block, to form an 8-tap shift register chain per DSP Block. The DSP block can increase the length of the shift register chain by cascading to the lower DSP blocks. The dedicated shift register chain spans a single column, but you can implement longer shift register chains requiring multiple columns using the regular FPGA routing resources.

Shift registers are useful in DSP functions such as FIR filters. When implementing 18 × 18 or smaller width multipliers, you do not need external logic to create the shift register chain because the input shift registers are internal to the DSP block. This implementation significantly reduces the logical element (LE) resources required, avoids routing congestion, and results in predictable timing.

The first multiplier in every half DSP block (top- and bottom-half) in Stratix III devices has a multiplexer for the first multiplier B-input (lower-leg input) register to select between general routing and loopback, as shown in Figure 5–6. In loopback mode, the most significant 18-bit registered outputs are connected as feedback to the multiplier input of the first top multiplier in each half DSP block. Loopback modes are used by recursive filters where the previous output is needed to compute the current output.

The loopback mode is described in detail in "Two-Multiplier Adder Sum Mode" on page 5–21.

Table 5–3 lists the input register modes for the DSP block.

Table 5–3. Input Register Modes

Register Input Mode *(1)*	9 × 9	12 × 12	18 × 18	36 × 36	Double
Parallel input	✓	✓	✓	✓	✓
Shift register input *(2)*	—	—	✓	—	—
Loopback input *(3)*	—	—	✓	—	—

Notes to Table 5–3:

(1) The multiplier operand input wordlengths are statically configured at compile time.

(2) Available only on the A-operand.

(3) Only one loopback input is allowed per Half-Block. See Figure 5–15 for details.

Multiplier and First-Stage Adder

The multiplier stage natively supports 9 × 9, 12 × 12, 18 × 18, or 36 × 36 multipliers. Other wordlengths are padded up to the nearest appropriate native wordlength; for example, 16 × 16 would be padded up to use 18 × 18. Refer to "Independent Multiplier Modes" on page 5–15 for more details. Depending on the data width of the multiplier, a single DSP block can perform many multiplications in parallel.

Each multiplier operand can be a unique signed or unsigned number. Two dynamic signals, `signa` and `signb`, control the representation of each operand, respectively. A `logic 1` value on the `signa/signb` signal indicates that `data A/data B` is a signed number; a `logic 0` value indicates an unsigned number. Table 5–4 lists the sign of the multiplication result for the various operand sign representations. The result of the multiplication is signed if any one of the operands is a signed value.

Table 5–4. Multiplier Sign Representation

Data A (signa Value)	Data B (signb Value)	Result
Unsigned (`logic 0`)	Unsigned (`logic 0`)	Unsigned
Unsigned (`logic 0`)	Signed (`logic 1`)	Signed
Signed (`logic 1`)	Unsigned (`logic 0`)	Signed
Signed (`logic 1`)	Signed (`logic 1`)	Signed

Each Half Block has its own `signa` and `signb` signal. Therefore, all of the `data A` inputs feeding the same DSP Half Block must have the same sign representation. Similarly, all of the `data B` inputs feeding the same DSP Half Block must have the same sign representation. The multiplier offers full precision regardless of the sign representation in all operational modes except for full precision 18 x 18 loopback and Two-Multiplier Adder modes. Refer to "Two-Multiplier Adder Sum Mode" on page 5–21 for details.

When the `signa` and `signb` signals are unused, the Quartus II software sets the multiplier to perform unsigned multiplication by default.

The outputs of the multipliers are the only outputs that can feed into the first-stage adder, as shown in Figure 5–6. There are four first-stage adders in a DSP block (two adders per half DSP block). The first-stage adder block has the ability to perform addition and subtraction. The control signal for addition or subtraction is static and has to be configured upon compile time. The first-stage adders are used by the sum modes to compute the sum of two multipliers, 18 × 18-complex multipliers, and to perform the first stage of a 36 × 36 multiply and shift operation.

Depending on your specifications, the output of the first-stage adder has the option to feed into the pipeline registers, second-stage adder, round and saturation unit, or the output registers.

Pipeline Register Stage

The output from the first-stage adder can either feed or bypass the pipeline registers, as shown in Figure 5–6. Pipeline registers increase the DSP block's maximum performance (at the expense of extra cycles of latency), especially when using the subsequent DSP block stages. Pipeline registers split up the long signal path between the input-registers/multiplier/first-stage adder and the second-stage adder/round-and-saturation/output-registers, creating two shorter paths.

Second-Stage Adder

There are four individual 44-bit second-stage adders per DSP block (2 adders per half DSP block). You can configure the second-stage adders as follows:

- The final stage of a 36-bit multiplier
- A sum of four (18 × 18)
- An accumulator (44-bits maximum)
- A chained output summation (44-bits maximum)

The chained-output adder can be used at the same time as a second-level adder in chained output summation mode.

The output of the second-stage adder has the option to go into the round and saturation logic unit or the output register.

You cannot use the second-stage adder independently from the multiplier and first-stage adder.

Round and Saturation Stage

The round and saturation logic units are located at the output of the 44-bit second-stage adder (round logic unit followed by the saturation logic unit). There are two round and saturation logic units per half DSP block. The input to the round and saturation logic unit can come from one of the following stages:

- Output of the multiplier (independent multiply mode in 18 × 18)
- Output of the first-stage adder (Two-Multiplier Adder)
- Output of the pipeline registers
- Output of the second-stage adder (Four-Multiplier Adder, Multiply-Accumulate Mode in 18 × 18)

These stages are discussed in detail in "Operational Mode Descriptions" on page 5–15.

The round and saturation logic unit is controlled by the dynamic round and saturate signals, respectively. A `logic 1` value on the round, saturate, or both enables the round, saturate, or both logic units.

You can use the round and saturation logic units together or independently.

Second Adder and Output Registers

The second adder register and output register banks are two banks of 44-bit registers that can also be combined to form larger 72-bit banks to support 36 × 36 output results.

The outputs of the different stages in the Stratix III devices are routed to the output registers through an output selection unit. Depending on the operational mode of the DSP block, the output selection unit selects whether the outputs of the DSP blocks comes from the outputs of the multiplier block, first-stage adder, pipeline registers, second-stage adder, or the round and saturation logic unit. The output selection unit is set automatically by the software, based on the DSP block operational mode you specified, and has the option to either drive or bypass the output registers. The exception is when the block is used in shift mode, in which case the user dynamically controls the output-select multiplexer directly.

When the DSP block is configured in "chained cascaded" output mode, both of the second-stage adders are used. The first one is used for performing Four-Multiplier Adder and the second is used for the chainout adder. The outputs of the Four-Multiplier Adder are routed to the second-stage adder registers before it enters the chainout adder. The output of the chainout adder goes to the regular output register bank. Depending on the configuration, the chainout results can be routed to the input of the next half-block's chainout adder input or to the general fabric (functioning as regular output registers). Refer to "Operational Mode Descriptions" on page 5–15 for details.

The second-stage and output registers are triggered by the positive edge of the clock signal and are cleared on power up. The following DSP block signals control the output registers within the DSP block:

- `clock[3..0]`
- `ena[3..0]`
- `aclr[3..0]`

Operational Mode Descriptions

The various modes of operation are discussed below.

Independent Multiplier Modes

In independent input and output multiplier mode, the DSP block performs individual multiplication operations for general-purpose multipliers.

9-, 12-, and 18-Bit Multiplier

You can configure each DSP block multiplier for 9-, 12-, or 18-bit multiplication. A single DSP block can support up to eight individual 9 × 9 multipliers, six 12 × 12 multipliers, or up to four individual 18 × 18 multipliers. For operand widths up to 9 bits, a 9 × 9 multiplier is implemented. For operand widths from 10 to 12 bits, a 12 × 12 multiplier is implemented, and for operand widths from 13 to 18 bits, an 18 × 18 multiplier is implemented. This is done by the Quartus II software by zero-padding the LSBs. Figure 5–8, Figure 5–9, and Figure 5–10 show the DSP block in the independent multiplier operation mode.

Figure 5–8. 18-Bit Independent Multiplier Mode for Half-DSP Block

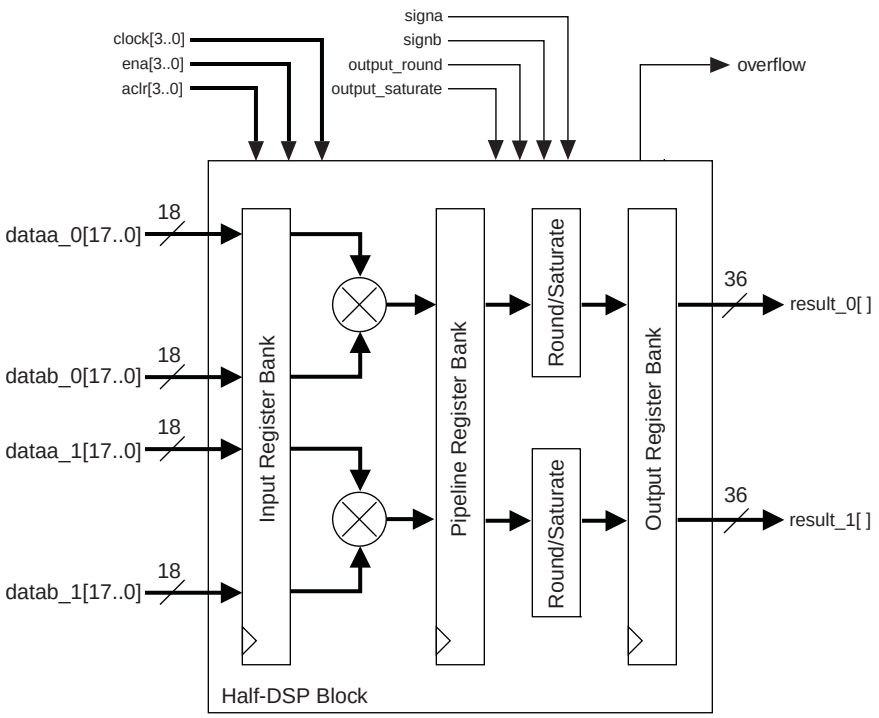


Figure 5–9. 12-Bit Independent Multiplier Mode for Half-DSP Block

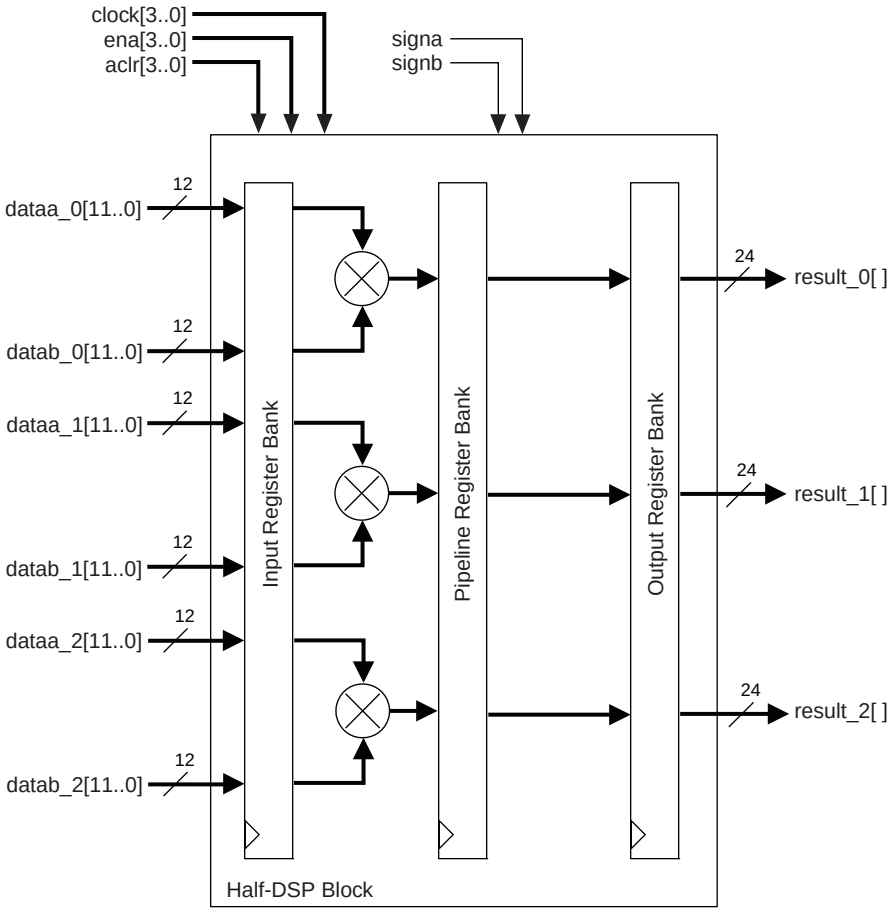
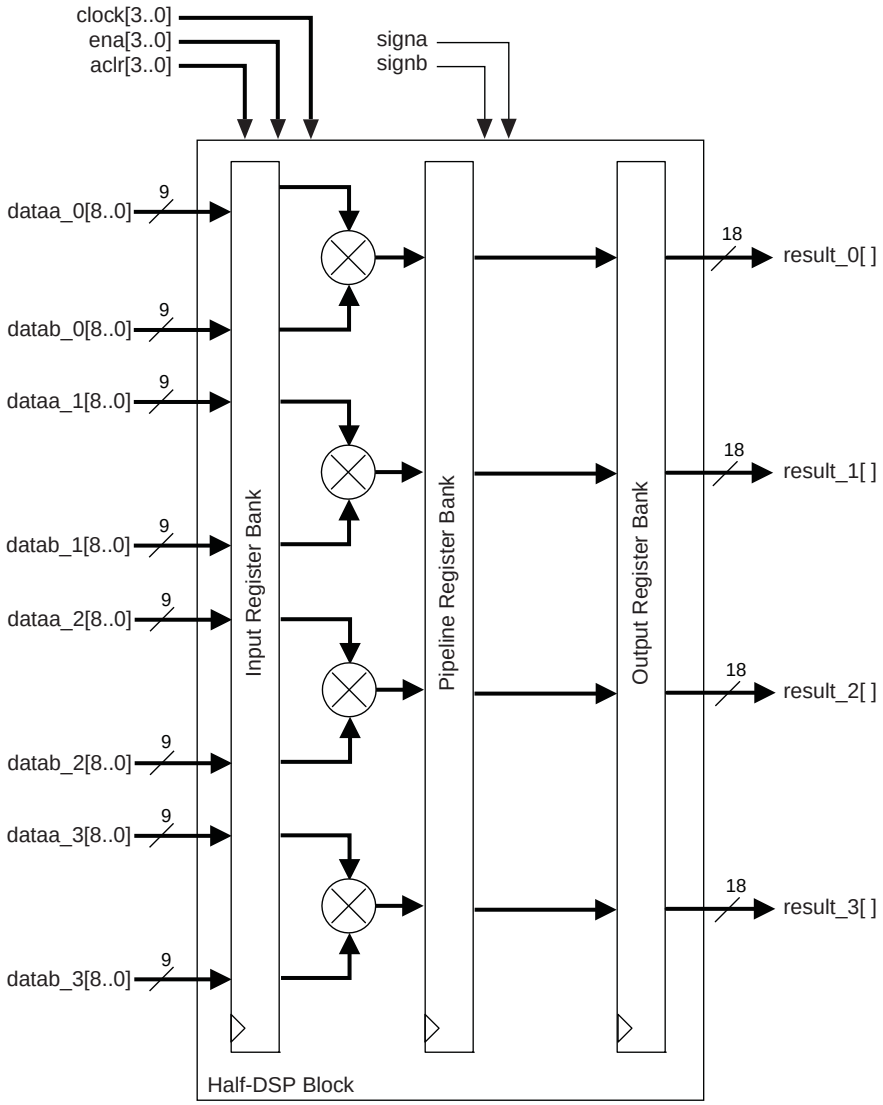


Figure 5–10. 9-Bit Independent Multiplier Mode for Half-Block



The multiplier operands can accept signed integers, unsigned integers, or a combination of both. You can change the `signa` and `signb` signals dynamically and can be registered in the DSP block. Additionally, the multiplier inputs and result can be registered independently. You can use the pipeline registers within the DSP block to pipeline the multiplier result, increasing the performance of the DSP block.

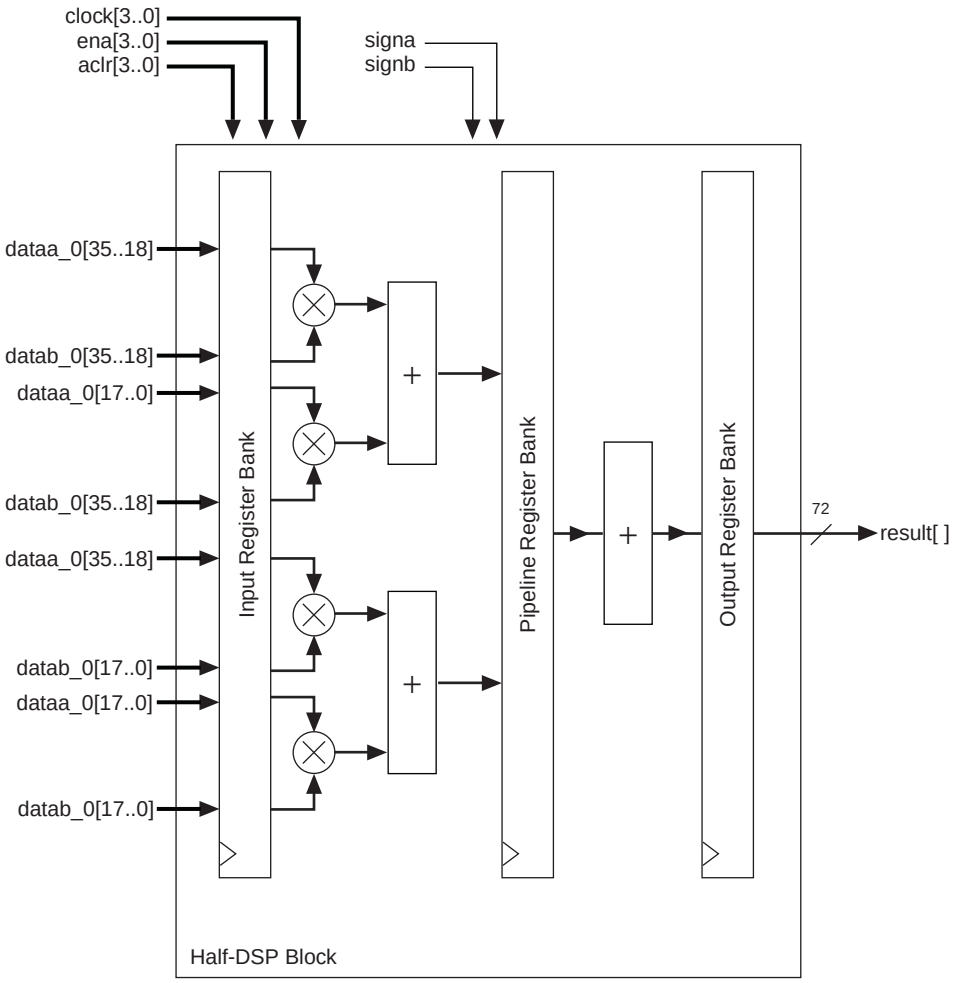
The round and saturation logic unit is supported for the 18-bit independent multiplier mode only.

36-Bit Multiplier

You can efficiently construct a 36 × 36 multiplier using four 18 × 18 multipliers. This simplification fits conveniently into one half-DSP block, and is implemented in the DSP block automatically by selecting the 36 × 36 mode. Stratix III devices can have up to two 36-bit multipliers per DSP block (one 36-bit multiplier per half DSP block). The 36-bit multiplier is also under the independent multiplier mode but uses the entire half DSP block, including the dedicated hardware logic after the pipeline registers to implement the 36 × 36 bit multiplication operation. This is shown in Figure 5–11.

The 36-bit multiplier is useful for applications requiring more than 18-bit precision; for example, for the mantissa multiplication portion of single precision and extended single precision floating-point arithmetic applications.

Figure 5–11. 36-Bit Independent Multiplier Mode for Half-DSP Block



Double Multiplier

The Stratix III DSP block can be configured to efficiently support an unsigned 54 × 54 bit multiplier that is required to compute the mantissa portion of an IEEE double precision floating point multiplication. A 54 × 54 bit multiplier can be built using basic 18 × 18 multipliers, shifters, and adders. In order to efficiently utilize the Stratix III DSP block's built in shifters and adders, a special Double mode (partial 54 × 54 multiplier) is available that is a slight modification to the basic 36 × 36 Multiplier mode. This is shown in Figure 5–12 and Figure 5–13.

Figure 5–12. Double Mode for Half-DSP Block

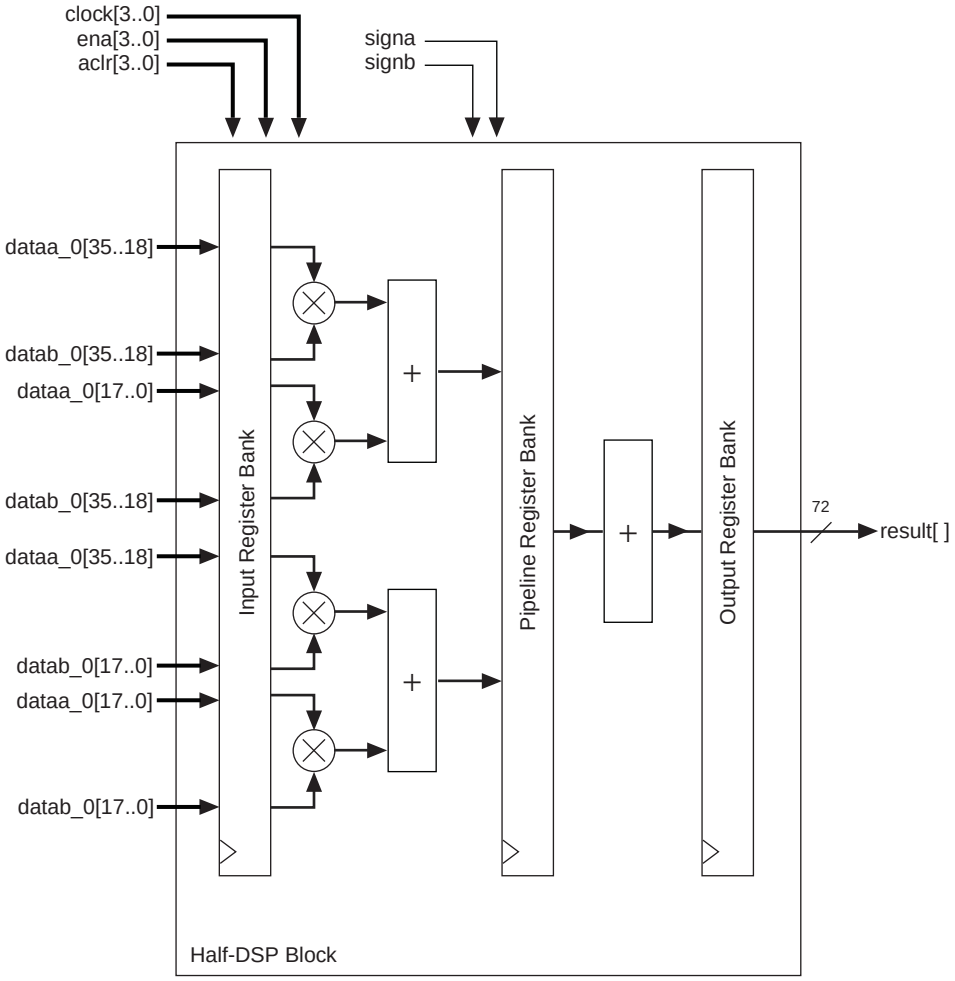
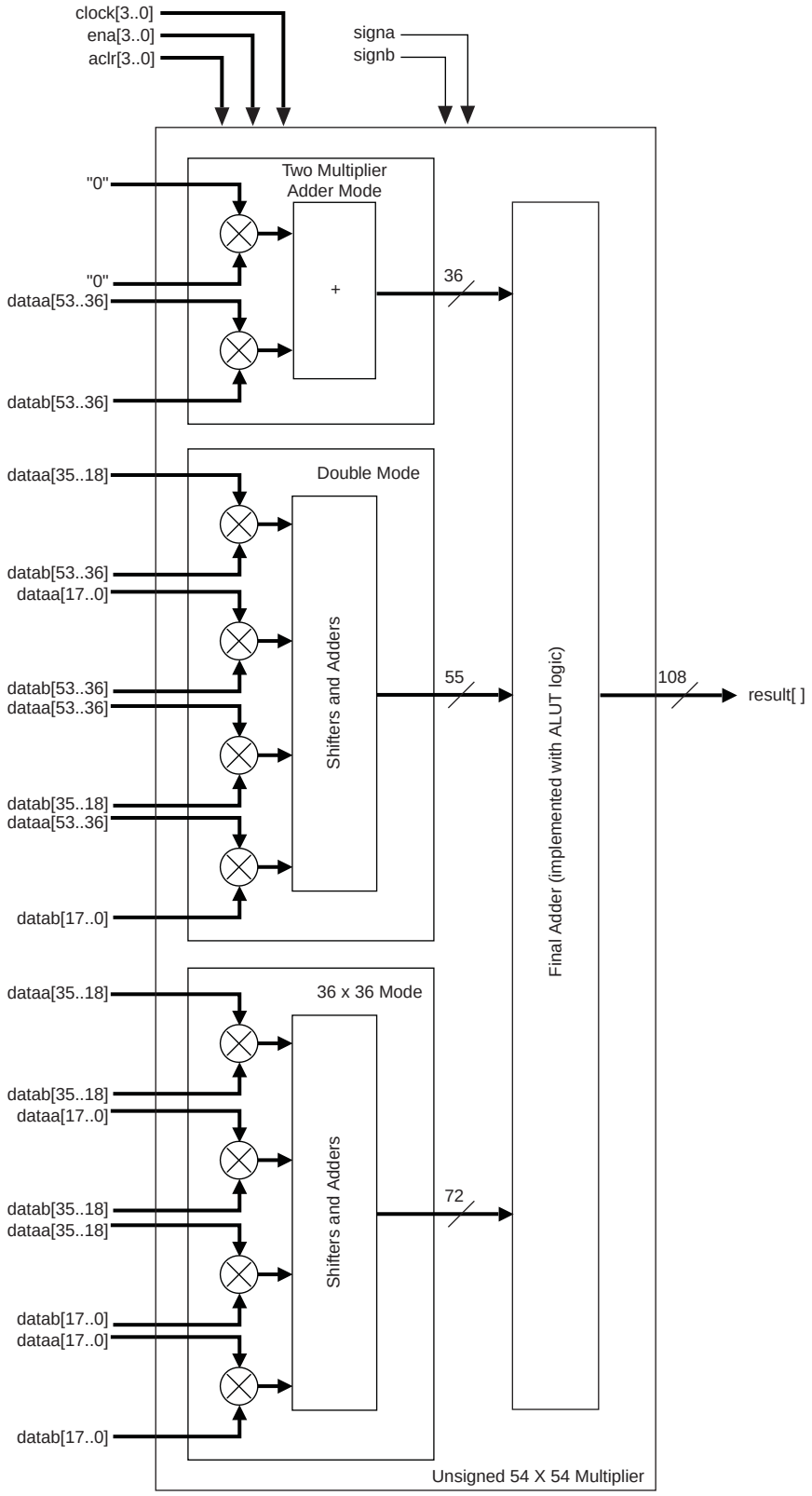


Figure 5–13. Unsigned 54 × 54 Multiplier



Two-Multiplier Adder Sum Mode

In the two-multiplier adder configuration, the DSP block can implement four 18-bit Two-Multiplier Adders (2 Two-Multiplier Adders per half DSP block). You can configure the adders to take the sum or difference of two multiplier outputs. Summation or subtraction has to be selected at compile time. The Two-Multiplier Adder function is useful for applications such as FFTs, complex FIR, and IIR filters. Figure 5–14 shows the DSP block configured in the two-multiplier adder mode.

The loopback mode is the other sub-feature of the two-multiplier adder mode. Figure 5–15 shows the DSP block configured in the loopback mode. This mode takes the 36-bit summation result of the two multipliers and feeds back the most significant 18-bits to the input. The lower 18-bits are discarded. You have the option to disable or zero-out the loopback data by using the dynamic `zero_loopback` signal. A `logic 1` value on the `zero_loopback` signal selects the `zeroed` data or disables the looped back data, while a `logic 0` selects the looped back data.

The option to use the loopback mode or the general two-multiplier adder mode must be selected at compile time.

For the Two-Multiplier Adder mode, if all the inputs are full 18-bit and unsigned, the result will require 37 bits. As the output data width in Two-Multiplier Adder mode is limited to 36 bits, this 37-bit output requirement is not allowed. Any other combination that does not violate the 36-bit maximum result is permitted; for example, two 16 × 16 signed Two-Multiplier Adders is valid.

The two-multiplier adder mode supports the round and saturation logic unit. You can use the pipeline registers and output registers within the DSP block to pipeline the multiplier-adder result, increasing the performance of the DSP block.

Figure 5–14. Two-Multiplier Adder Mode for Half-DSP Block

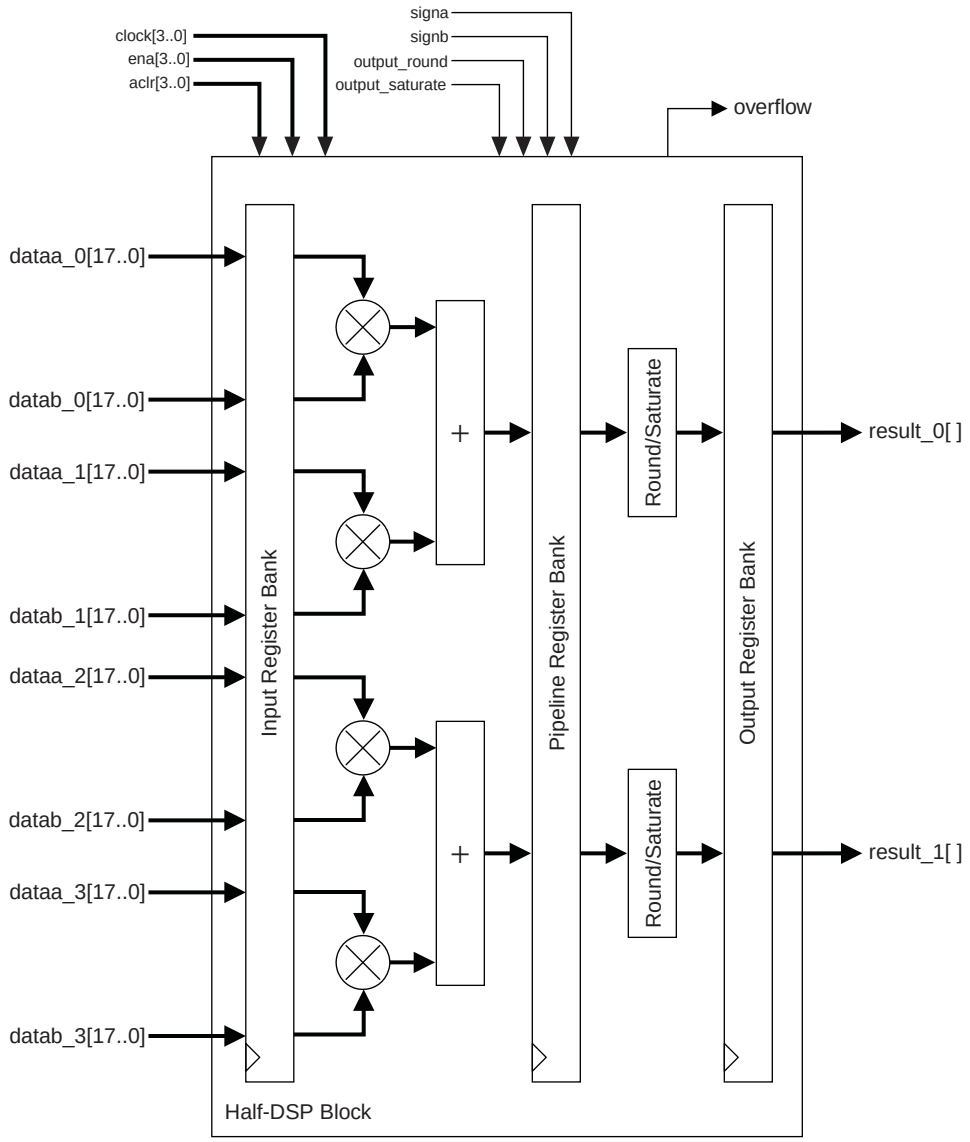
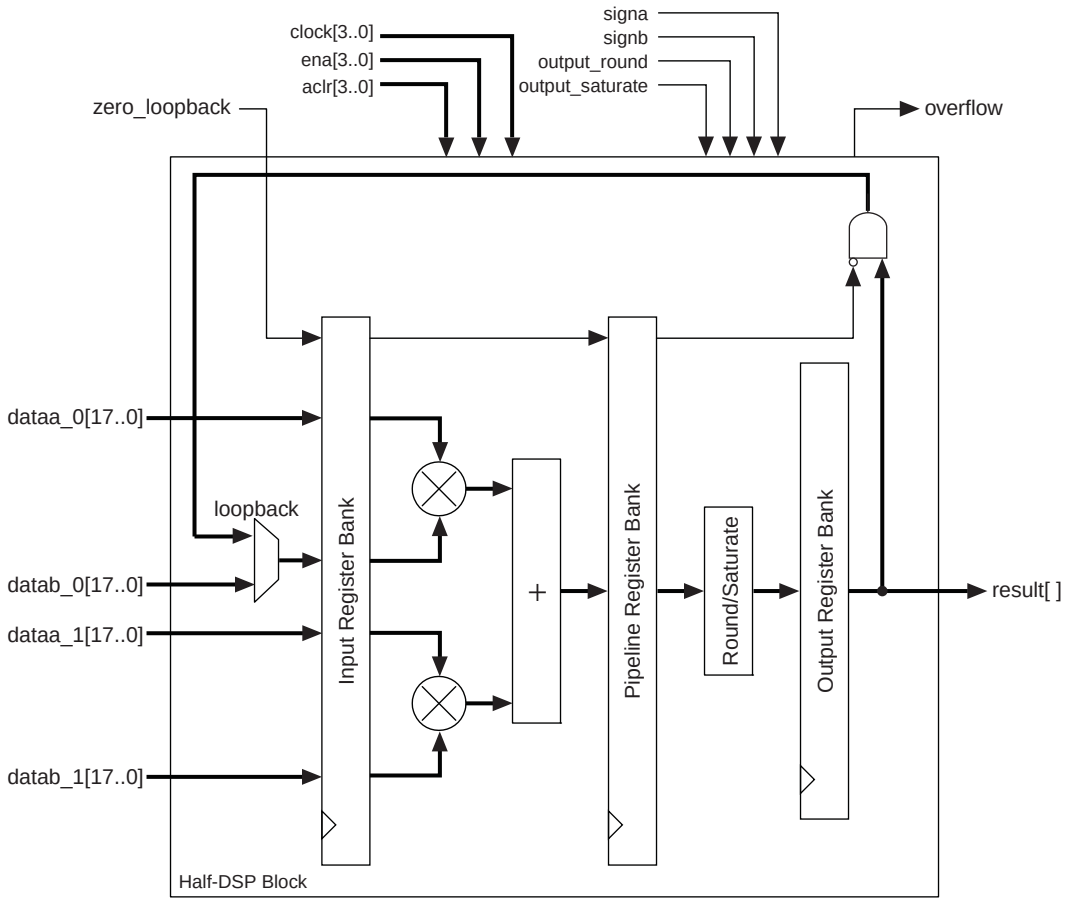


Figure 5–15. Loopback Mode for Half-DSP Block



18 × 18 Complex Multiply

You can configure the DSP block when used in Two-Multiplier Adder mode to implement complex multipliers using the two-multiplier adder mode. A single half DSP block can implement one 18-bit complex multiplier.

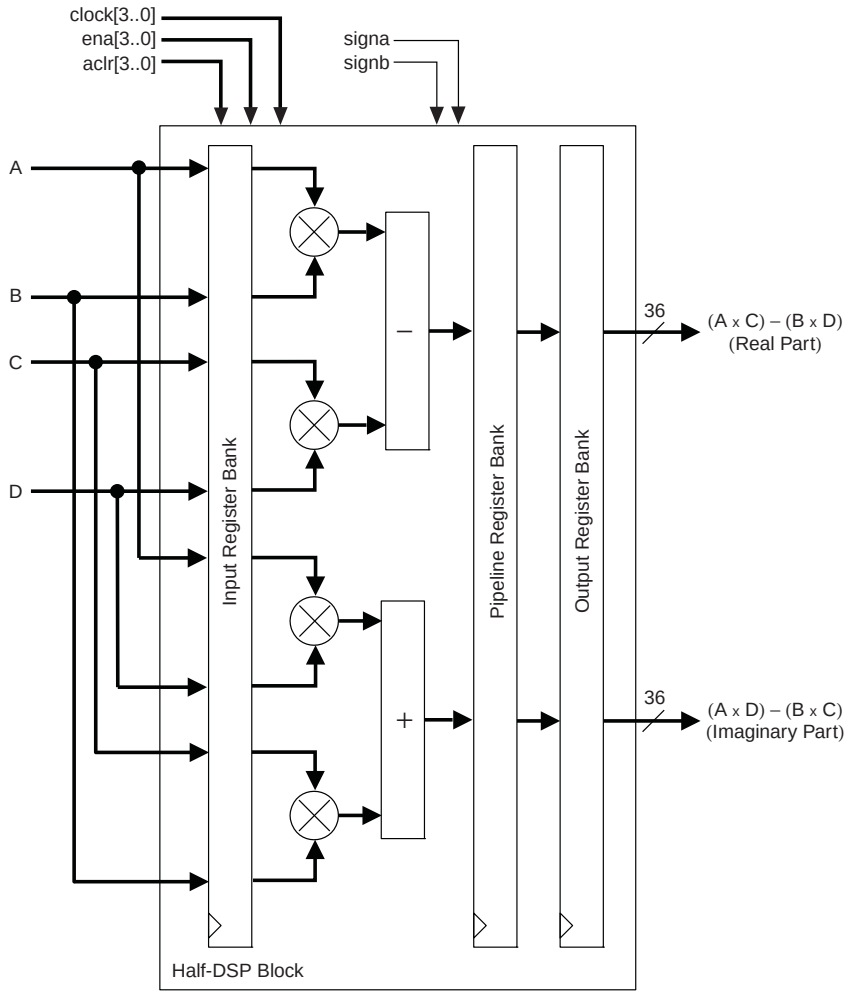
A complex multiplication can be written as shown in Equation 5–4.

Equation 5–4. Complex Multiplication Equation

$$(a + jb) \times (c + jd) = ((a \times c) - (b \times d)) + j((a \times d) + (b \times c))$$

To implement this complex multiplication within the DSP block, the real part ((a × c) – (b × d)) is implemented using two multipliers feeding one subtractor block while the imaginary part ((a × d) + (b × c)) is implemented using another two multipliers feeding an adder block. Figure 5–16 shows an 18-bit complex multiplication. This mode automatically assumes all inputs are using signed numbers.

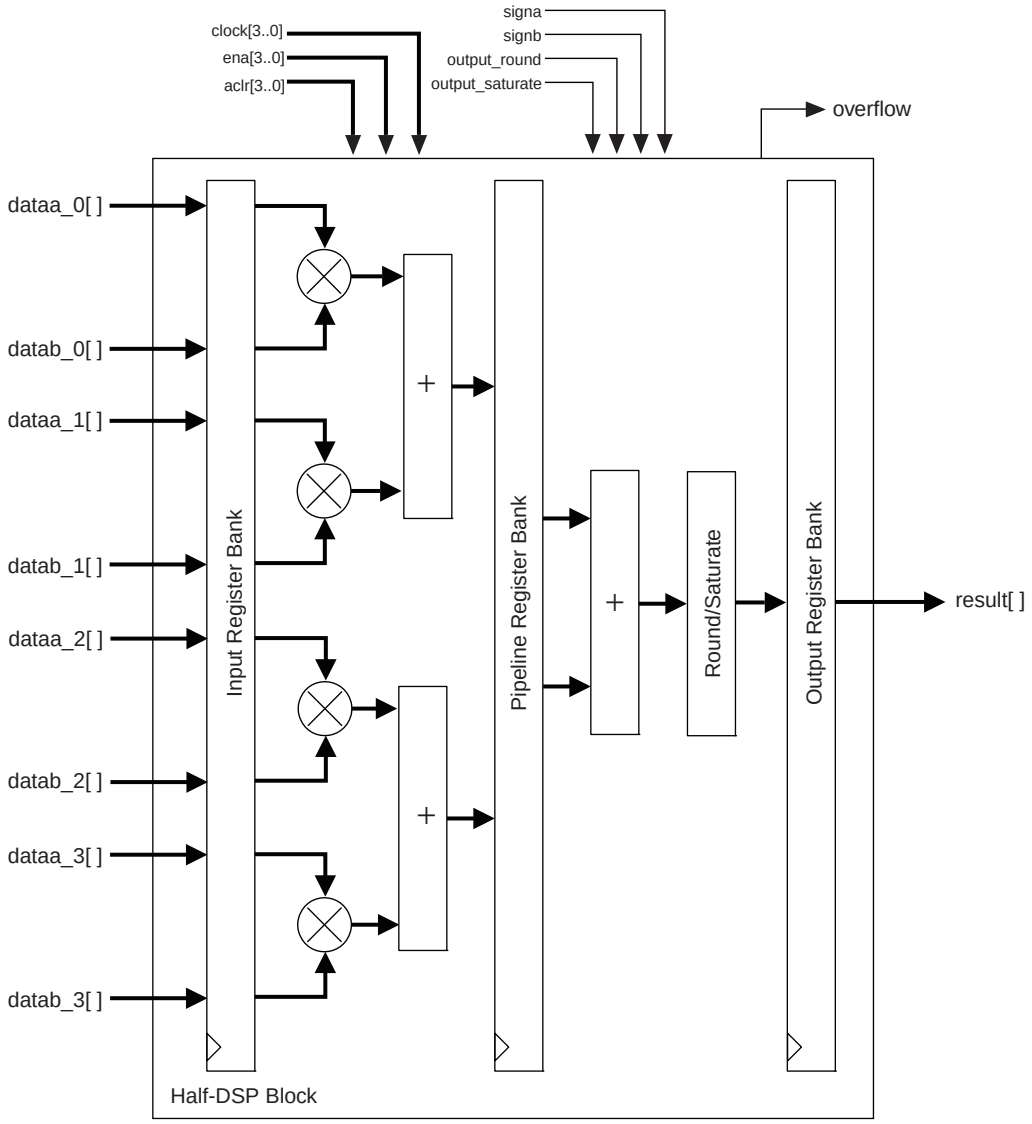
Figure 5–16. Complex Multiplier Using Two-Multiplier Adder Mode



Four-Multiplier Adder

In the four-multiplier adder configuration shown in Figure 5–17, the DSP block can implement two four-multiplier adders (one four-multiplier adder per half DSP block). These modes are useful for implementing one-dimensional and two-dimensional filtering applications. The four-multiplier adder is performed in two addition stages. The outputs of two of the four multipliers are initially summed in the two first-stage adder blocks. The results of these two adder blocks are then summed in the second-stage adder block to produce the final four-multiplier adder result, as shown by Equation 5–2 and Equation 5–3.

Figure 5–17. Four-Multiplier Adder Mode for Half-DSP Block



The four-multiplier adder mode supports the round and saturation logic unit. You can use the pipeline registers and output registers within the DSP block to pipeline the multiplier-adder result, increasing the performance of the DSP block.

High Precision Multiplier Adder

In the high precision multiplier adder configuration shown in Figure 5–18, the DSP block can implement two two-multiplier adders, with multiplier precision of 18 × 36 (one two-multiplier adder per DSP half block). This mode is useful in filtering or FFT applications where a data path greater than 18 bits is required, yet 18 bits is sufficient for the coefficient precision. This can occur in cases where that data has a high dynamic range. If the coefficients are fixed, as in FFT and most filter applications, the precision of 18 bits will provide a dynamic range over 100 dB if the largest coefficient is normalized to the maximum 18-bit representation.

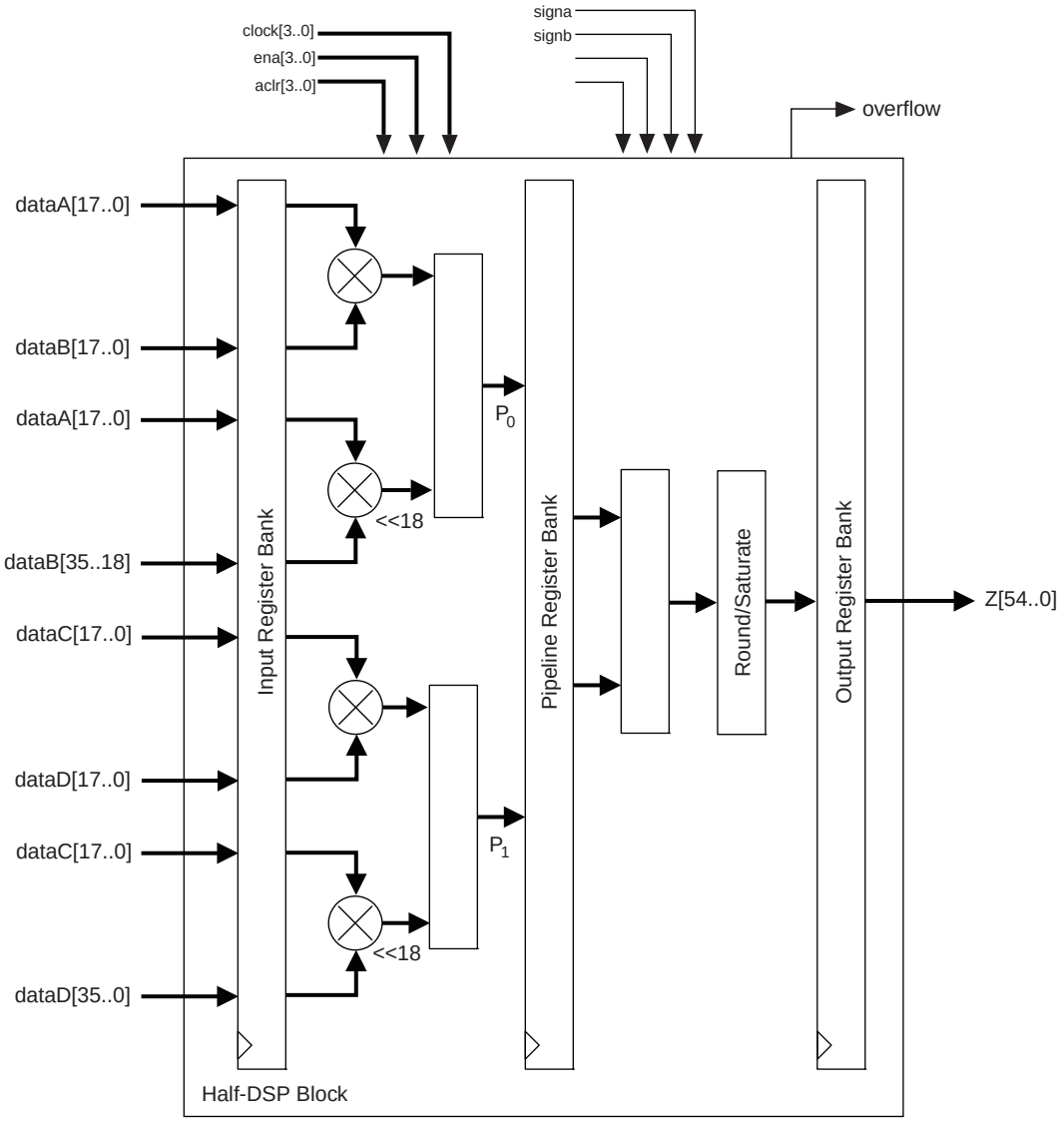
In these situations, the data path can be up to 36 bits, allowing ample headroom to bit growth, or gain changes in the signal source without loss of precision. This mode is also extremely useful in single precision block floating point applications.

The high precision multiplier adder is preformed in two stages. The 18 × 36 multiply is decomposed into two 18 × 18 multipliers. The multiplier with the LSB of the data source is performed unsigned, while the multiplier with the MSB of the data source can be signed or signed. The latter multiplier has its result left shifted by 18 bits prior to the first adder stage, creating an effective 18 × 36 multiplier. The results of these two adder blocks are then summed in the second stage adder block to produce the final result.

Equation 5–5. High Precision Multiplier Adder Equation

$$Z[54..0] = P_0[53..0] + P_1[53..0] \text{ where}$$
$$P_0 = A[17..0] \acute{} B[35..0] \text{ and } P_1 = C[17..0] \acute{} D[35..0]$$

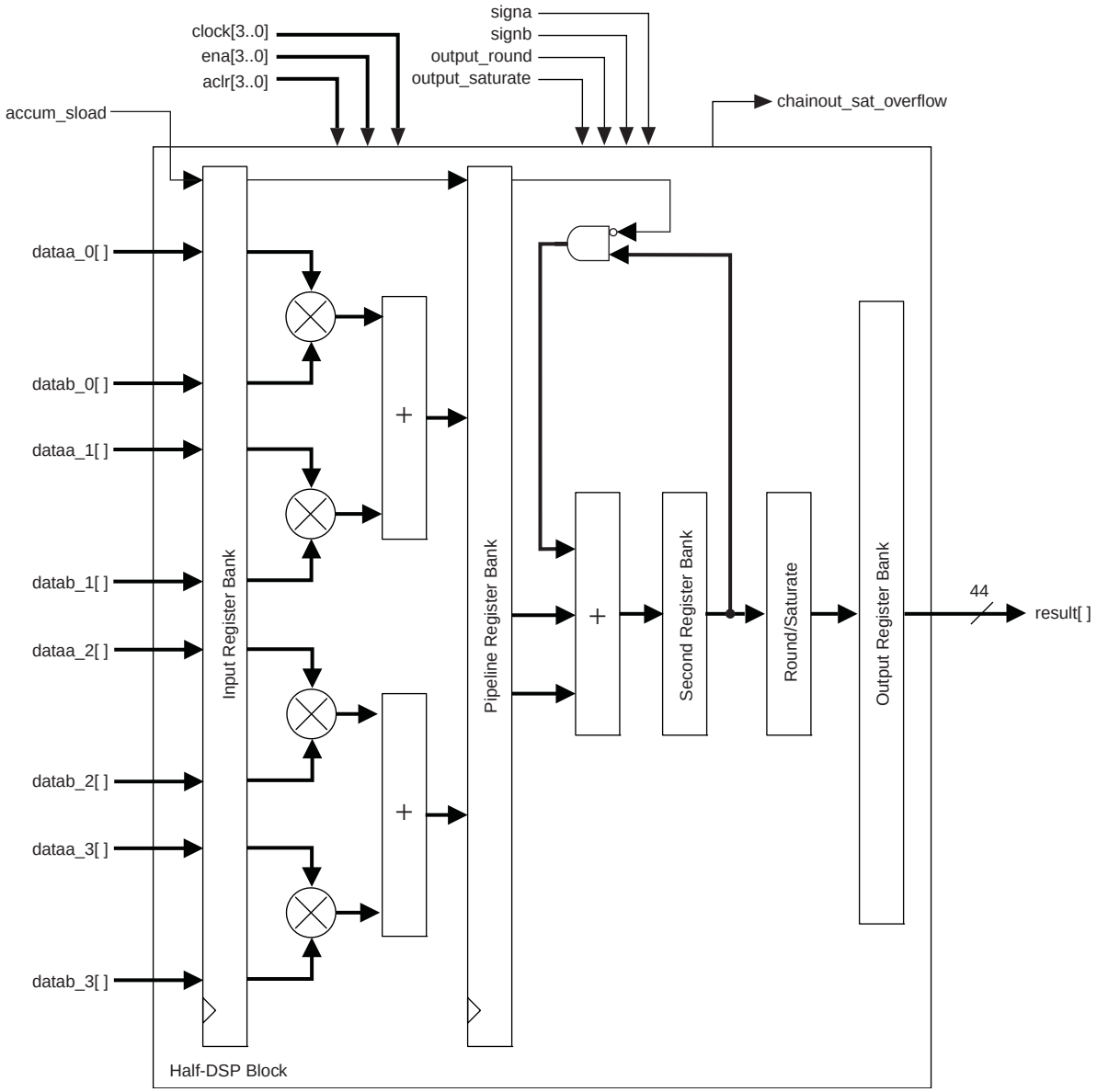
Figure 5–18. Four-Multiplier Adder Mode for Half-DSP Block



Multiply Accumulate Mode

In multiply accumulate mode, the second-stage adder is configured as a 44-bit accumulator or subtractor. The output of the DSP block is looped back to the second-stage adder and added or subtracted with the two outputs of the first-stage adder block according to Equation 5–3. Figure 5–19 shows the DSP block configured to operate in multiply accumulate mode.

Figure 5–19. Multiply Accumulate Mode for Half-DSP Block



A single DSP block can implement up to two independent 44-bit accumulators.

The dynamic `accum_sload` control signal is used to clear the accumulation. A `logic 1` value on the `accum_sload` signal synchronously loads the accumulator with the multiplier result only, while a `logic 0` enables accumulation by adding or subtracting the output of the DSP block (accumulator feedback) to the output of the multiplier and first-stage adder.

☞ The control signal for the accumulator and subtractor is static and therefore has to be configured at compile time.

This mode supports the round and saturation logic unit as it is configured as an 18-bit multiplier accumulator. You can use the pipeline registers and output registers within the DSP block to increase the performance of the DSP block.

Shift Modes

Stratix III devices support the following shift modes for 32-bit input only:

- Arithmetic shift left, `ASL[N]`
- Arithmetic shift right, `ASR[32-N]`
- Logical shift left, `LSL[N]`
- Logical shift right, `LSR[32-N]`
- 32-bit rotator or Barrel shifter, `ROT[N]`

☞ You can switch the shift mode between these modes using the dynamic rotate and shift control signals.

The shift mode in a Stratix III device can be easily used by the soft embedded processor such as Nios® II to perform the dynamic shift and rotate operation. Figure 5–20 shows the shift mode configuration.

The shift mode makes use of the available multipliers to logically or arithmetically shift left, right, or rotate the desired 32-bit data. The DSP block is configured like the independent 36-bit multiplier mode to perform the shift mode operations.

The arithmetic shift right requires signed input vector. During arithmetic shift right, the sign is extended to fill the MSB of the 32-bit vector. The logical shift right uses unsigned input vector. During logical shift right, zeros are padded in the most significant bits shifting the 32-bit vector to the right. The barrel shifter uses unsigned input vector and implements a rotation function on a 32-bit word length.

Two control signals `rotate` and `shift_right` together with the `signa` and `signb` signals, determining the shifting operation. Examples of shift operations are listed in Table 5–5 on page 5–31.

Figure 5–20. Shift Operation Mode for Half-DSP Block

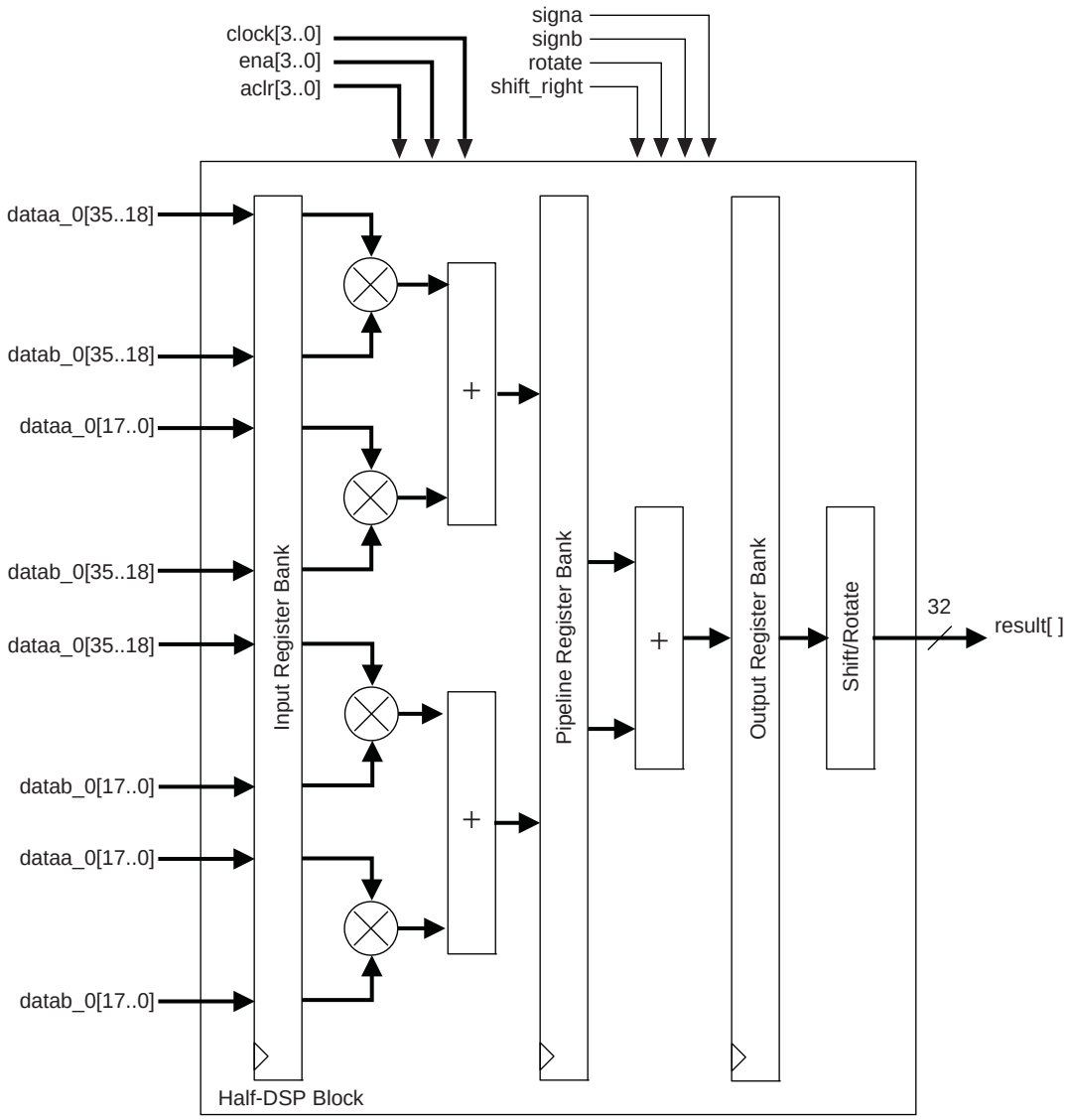


Table 5–5. Examples of Shift Operations *(Note 1)*

Example	Signa	Signb	Shift_right	Rotate	A-input	B-input	Result
Logical Shift Left `LSL[N]`	Unsigned	Unsigned	0	0	0xAABBCCDD	0x00000100	0xBBCCDD00
Logical Shift Right `LSR[32-N]`	Unsigned	Unsigned	1	0	0xAABBCCDD	0x00000100	0x000000AA
Arithmetic Shift Left `ASL[N]`	Signed	Unsigned	0	0	0xAABBCCDD	0x00000100	0xBBCCDD00
Arithmetic Shift Right `ASR[32-N]`	Signed	Unsigned	1	0	0xAABBCCDD	0x00000100	0xFFFFFFAA
Rotation `ROT[N]`	Unsigned	Unsigned	0	1	0xAABBCCDD	0x00000100	0xBBCCDDAA

Note to Table 5–5:

(1) The value of the shift is equal to the value in the bracket where `[N]` is the position of bit '1' on the B-Input. In the above examples, `[N]` is 8 and is calculated from the LSB to the MSB where LSB=0 and MSB=31.

Rounding and Saturation Mode

Round and saturation functions are often required in DSP arithmetic. Rounding is used to limit bit growth and its side effects and saturation is used to reduce overflow and underflow side effects.

Two rounding modes are supported in Stratix III devices:

- Round-to-nearest-integer mode
- Round-to-nearest-even mode

You must select one of the two options at compile time.

Round-to-nearest-integer mode provides the biased rounding support and is the simplest form of rounding commonly used in DSP arithmetic. The round-to-nearest-even method provides unbiased rounding support and is used where DC offsets are a concern. Table 5–6 lists how round-to-nearest-even mode works. Examples of the difference between the two modes are listed in Table 5–7. In this example, a 6-bit input is rounded to 4 bits. You can observe from Table 5–7 that the main difference between the two rounding options is when the residue bits are exactly half way between its nearest two integers and the LSB is zero (even).

Table 5–6. Example of Round-To-Nearest-Even Mode

6- to 4-bits Rounding	Odd/Even (Integer)	Fractional	Add to Integer	Result
010111	x	> 0.5 (11)	1	0110
001101	x	< 0.5 (01)	0	0011
001010	Even (0010)	= 0.5 (10)	0	0010
001110	Odd (0011)	= 0.5 (10)	1	0100
110111	x	> 0.5 (11)	1	1110
101101	x	< 0.5 (01)	0	1011
110110	Odd (1101)	= 0.5 (10)	1	1110
110010	Even (1100)	= 0.5 (10)	0	1100

Table 5–7. Comparison of Round-to-Nearest-Integer and Round-to-Nearest-Even

Round-To-Nearest-Integer	Round-To-Nearest-Even
010111 ⇒0110	010111 ⇒0110
001101 ⇒0011	001101 ⇒0011
001010 ⇒0011	001010 ⇒0010
001110 ⇒0100	001110 ⇒0100
110111 ⇒1110	110111 ⇒1110
101101 ⇒1011	101101 ⇒1011
110110 ⇒1110	110110 ⇒1110
110010 ⇒1101	110010 ⇒1100

Two saturation modes are supported in Stratix III:

- Asymmetric saturation mode
- Symmetric saturation mode

You must select one of the two options at compile time.

In 2's complement format, the maximum negative number that can be represented is $-2^{(n-1)}$ while the maximum positive number is $2^{(n-1)}-1$. Symmetrical saturation will limit the maximum negative number to $-2^{(n-1)} + 1$. For example, for 32 bits:

- Asymmetric 32-bit saturation: Max = 0x7FFFFFFF, Min = 0x80000000
- Symmetric 32-bit saturation: Max = 0x7FFFFFFF, Min = 0x80000001

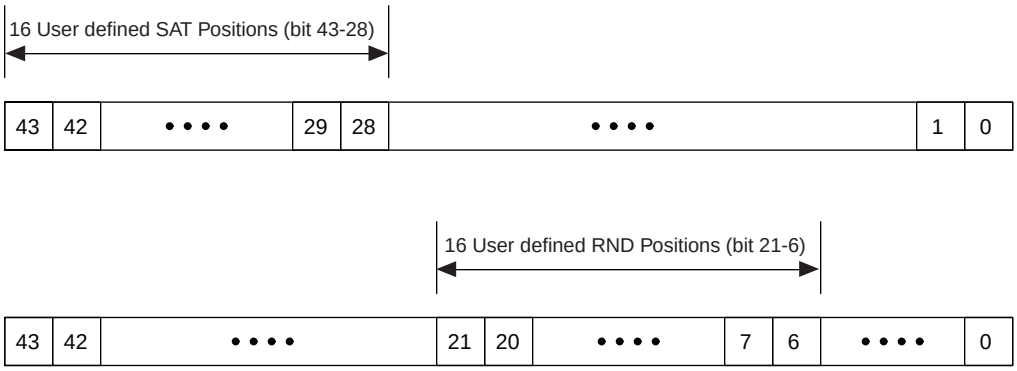
Table 5–8 lists how the saturation works. In this example, a 44-bit input is saturated to 36-bits.

Table 5–8. Examples of Saturation

44 to 36 Bits Saturation	Symmetric SAT Result	Asymmetric SAT Result
5926AC01342h	7FFFFFFFFh	7FFFFFFFFh
ADA38D2210h	800000001h	800000000h

Stratix III devices have up to 16 configurable bit positions out of the 44-bit bus (`[43:0]`) for the round and saturate logic unit providing higher flexibility. You must select the 16 configurable bit positions at compile time. These 16-bit positions are located at bits `[21:6]` for rounding and `[43:28]` for saturation, as shown in Figure 5–21.

Figure 5–21. Round and Saturation Locations



For symmetric saturation, the `RND` bit position is also used to determine where the LSP for the saturated data is located.

You can use the rounding and saturation function described above in regular supported multiplication operations as specified in Table 5–2. However, for accumulation type operations, the following convention is used.

The functionality of the round logic unit is in the format of:

`Result = RND[S(A × B)]`, when used for an accumulation type of operation.

Likewise, the functionality of the saturation logic unit is in the format of:

`Result = SAT[S(A × B)]`, when used for an accumulation type of operation.

If both the round and saturation logic units are used for an accumulation type of operation, the format is:

`Result = SAT[RND[S(A × B)]]`

DSP Block Control Signals

The Stratix III DSP block is configured using a set of static and dynamic signals. The DSP block dynamic signals are user configurable and can be set to toggled or not at run time. Table 5–10 lists the dynamic signals for the DSP block.

Table 5–10. DSP Block Dynamic Signals (Part 1 of 2)

Signal Name	Function	Count
■ signa ■ signb	Signed/unsigned control for all multipliers and adders. signa for "multiplicand" input bus to dataa[17:0] each multiplier. signb for "multiplier" input bus datab[17:0] to each multiplier. signa = 1, signb = 1 for signed-signed multiplication signa = 1, signb = 0 for signed-unsigned multiplication signa = 0, signb = 1 for unsigned-signed multiplication signa = 0, signb = 0 for unsigned-unsigned multiplication	2
output_round	Round control for first stage round/saturation block. output_round = 1 for rounding on multiply output output_round = 0 for normal multiply output	1
chainout_round	Round control for second stage round/saturation block. chainout_round = 1 for rounding on multiply output chainout_round = 0 for normal multiply output	1
output_saturate	Saturation control for first stage round/saturation block for Q-format multiply. If both rounding and saturation is enabled, saturation is done on the rounded result. output_saturate = 1 for saturation support output_saturate = 0 for no saturation support	1
chainout_saturate	Saturation control for second stage round/saturation block for Q-format multiply. If both rounding and saturation are enabled, saturation is done on the rounded result. chainout_saturate = 1 for saturation support chainout_saturate = 0 for no saturation support	1
accum_sload	Dynamically specifies whether the accumulator value is zero. accum_sload = 0, accumulation input is from the output registers accum_sload = 1, accumulation input is set to be zero	1
zero_chainout	Dynamically specifies whether the chainout value is zero.	1
zero_loopback	Dynamically specifies whether the loopback value is zero.	1
rotate	rotation = 1, rotation feature is enabled	1
shift_right	shift_right = 1, shift right feature is enabled	1
—	Total Signals per Half-block	11
clock0 clock1 clock2 clock3	DSP-block-wide clock signals	4

Table 5–10. DSP Block Dynamic Signals (Part 2 of 2)

Signal Name	Function	Count

`ena0`
`ena1`
`ena2`
`ena3`
	Input and Pipeline Register enable signals	4

`aclr0`
`aclr1`
`aclr2`
`aclr3`
	DSP block-wide asynchronous clear signals (active low).	4

—	Total Count per Full Block	34

Application Examples

FIR Example

A finite impulse response filter is a common function used in many systems to perform spectral manipulations. The basic form is shown in Equation 5–6.

Equation 5–6. Finite Impulse Response Filter Equation

$$y(n) = \sum_{k=0}^{N-1} x(n-k) \times c(k)$$

In this equation, x(n) is the input samples to the filter, c(k) are the filter coefficients, and y(n) are the filtered output samples. Typically, the coefficients do not change in time in most applications such as Digital Down Converters (DDC). FIR filters can be implemented in many forms, the most simple being the tap-delay line approach.

Stratix III DSP block can implement various types of FIR filters very efficiently. To form the tap-delay line, the input register stage of the DSP block has the ability to cascade the input in a chained fashion in 18-bit wide format. Unlike the Stratix II DSP block, which has two built-in parallel input register scan paths, Stratix III supports only one built-in 18-bit parallel input register scan path for 288 data input.

For a pair of 18-bit input buses, the A input for the first 18-bit bus is fed back to be registered again at the input of the second (lower) pair of inputs. Refer to Figure 5–22 for details.

The B input of the multiplier feeds from the general routing. You can scan in the data in 18-bit parallel form and multiply it by the 18-bit input bus from general routing in each cycle.

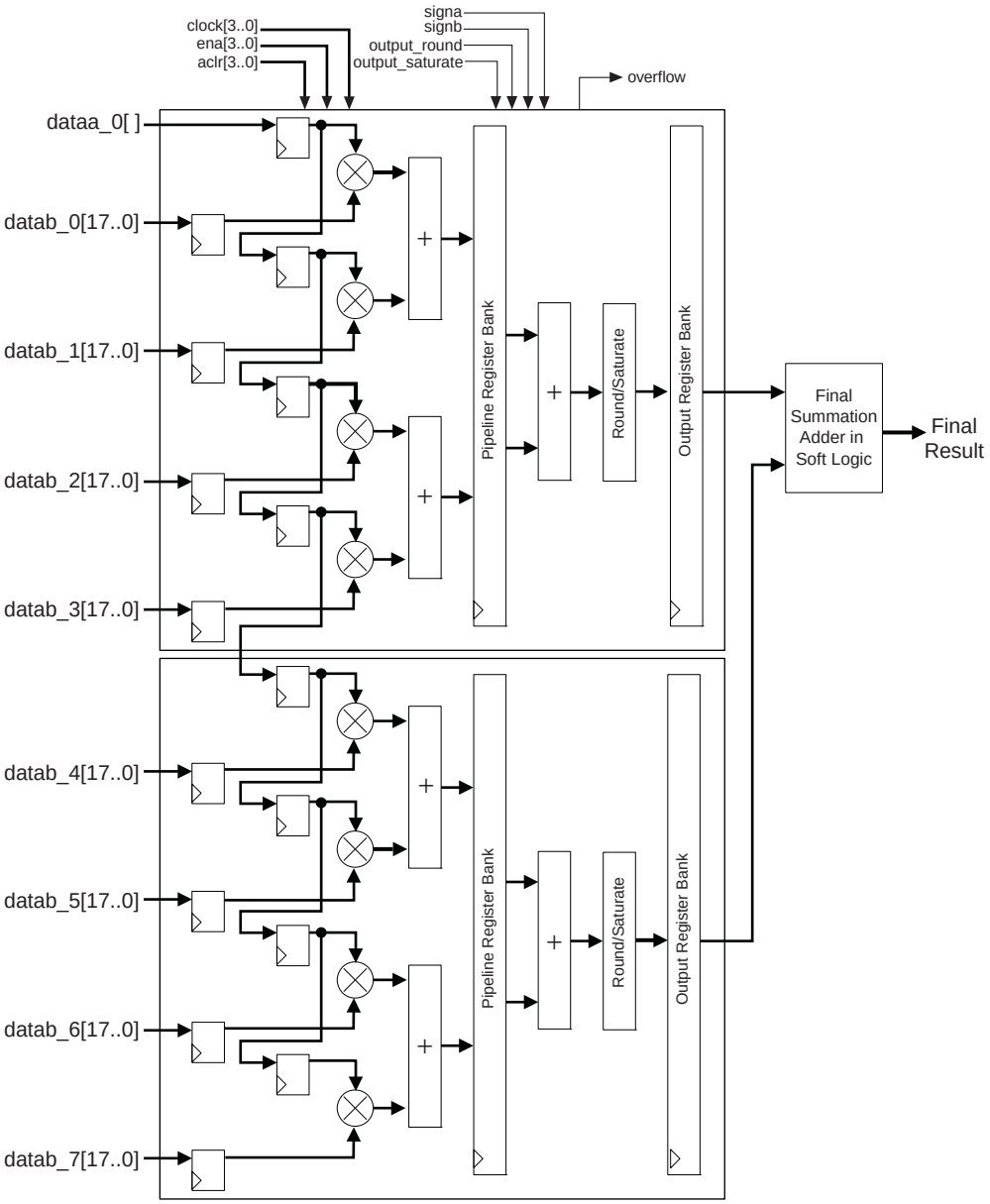
Normally in a FIR filter, the fixed data input (from general routing and not from cascade) is the constant that needs to be multiplied by the cascaded input. In 18-bit mode, the DSP block has enough input registers to register the general routing signals and the cascaded signal buses before multiplying them. This makes having eight taps for an 18-bit cascade mode possible. Each tap can be considered a single multiplier. If all eight multiplier inputs for the full DSP block are cascaded in a parallel scan chain, an eight-tap FIR filter is created, as shown in Figure 5–22.

The DSP block can be concatenated to have more than eight taps by enabling the option to output the parallel scan chain to the next (lower) DSP block. Likewise, the output of previous (above) cascade chain is used as an input to the current block. The first (top) multiplier in each half block will have the option to select the 18-bit cascade chain input from the regular routing or from the previous (above) cascade chain. Also, the last cascaded chain in each half DSP block can exit the DSP block by routing the cascade chain after the last (fourth from top) input register to the output routing channel, bypassing both the pipeline and output registers. This concatenation allows the user to easily construct their desired filter length.

You can use the Four-Multiplier Adder mode with one of the inputs to each multiplier being in a form of chained cascaded input from the previous (above) register. This is very similar to the regular Four-Multiplier Adder with the difference being that not all the inputs are from general routing.

For a complete FIR, the results per individual Four-Multiplier Adder can be combined in either a tree or chained cascade manner. Using external logic and adders, you can very easily implement a tree summation, as shown in Figure 5–22.

Figure 5–22. FIR Filter Using Tap-Delay Line Input and Tree Summation of Final Result

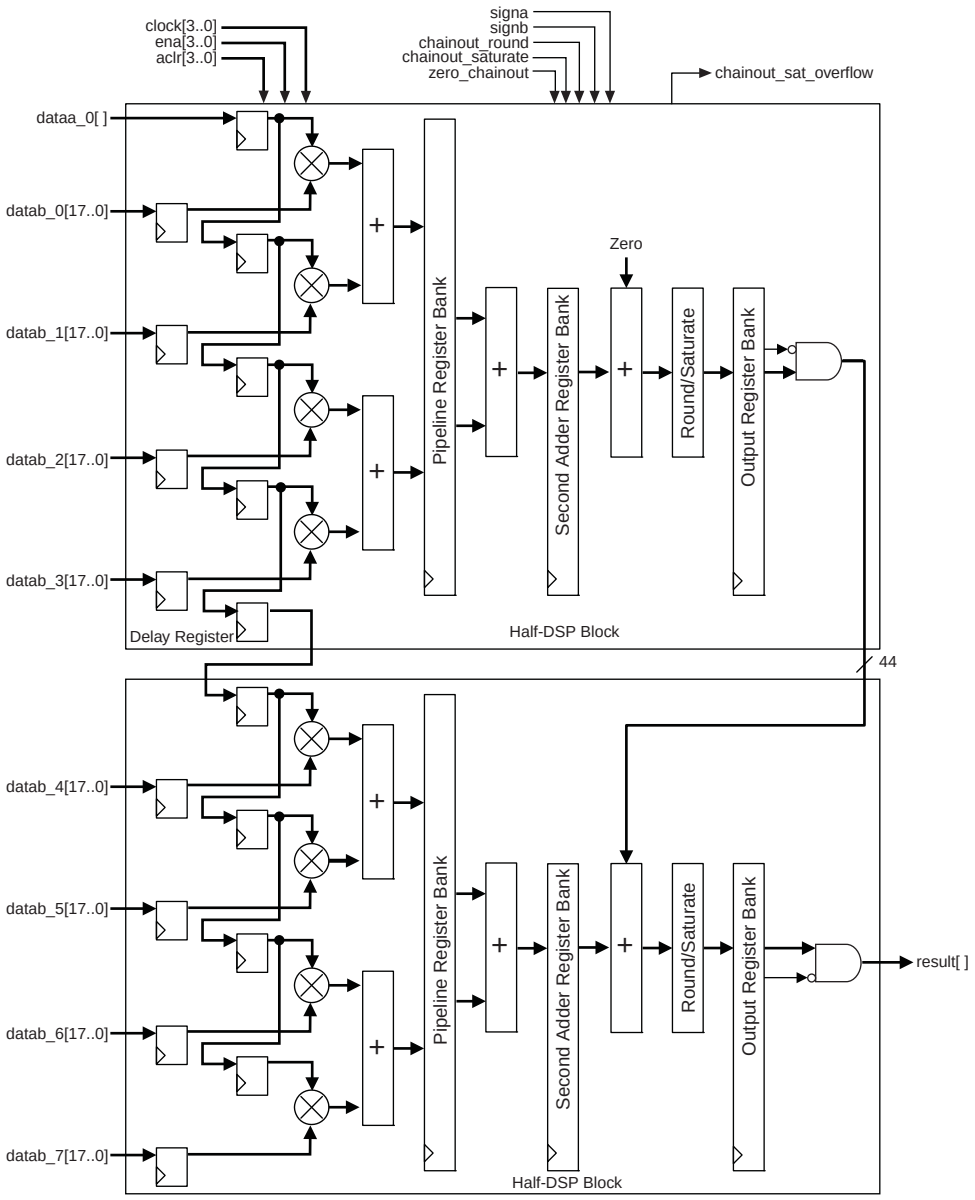


For faster and more efficient chained cascade summation, the DSP block can implement the chainout function in the cascade mode. This mode uses the second-stage 44-bit adder to add the current Four-Multiplier Adder of the half DSP block to the adjacent half DSP block of the Four-Multiplier Adder as shown in Figure 5–23.

This scheme is possible because each half DSP block has two second-stage adders. One of the two second-stage adders is used to add the current Four-Multiplier Adder. The second second-stage adder takes the output of the first second-stage adder and adds it to the adjacent half DSP block of the Four-Multiplier Adder result.

In Figure 5–23, the adder that adds the adjacent half DSP block to the current Four-Multiplier Adder is shown as the chainout adder for clarity. This scheme is used to chain and add multiple DSP blocks together. The output of the chainout adder can be registered. The registered chainout output can feed the lower adjacent DSP block for a chainout summation or it can feed general FPGA routing. The chainout result can be zeroed out by applying `logic 1` on the dynamic `zerochainout` signal. The `zerochainout` signal can also be registered.

Figure 5–23. FIR Filter using Tap-Delay Line Input and Chained Cascade Summation of Final Result

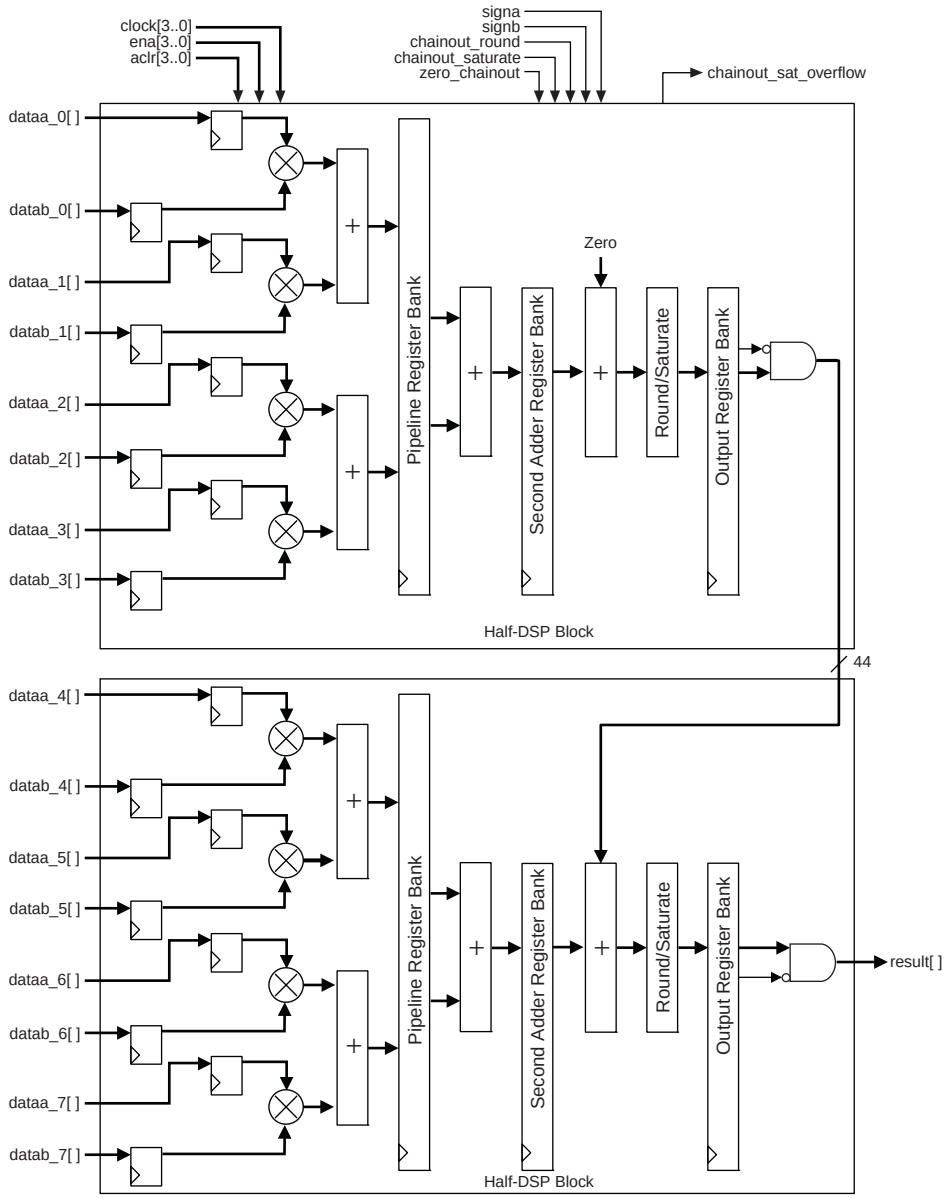


When you use **both** the input cascade and chainout features, the DSP block uses an 18-bit delay register in the boundary of each half-DSP block or from block-to-block to synchronize the input scan chain data with the chainout data. The top half computes the sum of product and chains the output to the next block after the output register. The output register uses the delay register to delay the cascade input by one clock cycle to compensate the latency for the bottom half.

For applications in which the system clock is slower than the speed of the DSP block, the multipliers can be time-multiplexed to improve efficiency. This makes multi-channel and semi-parallel FIR structures possible. The structure to achieve this is similar to Figure 5–22 and Figure 5–23. The main difference is that the input cascade chain is no longer used and each half-DSP block is used in Four-Multiplier Mode with independent inputs. Figure 5–24 shows an example for chained cascaded summation.

In most cases, only the final stage FIR tap with the rounding and saturation unit is deployed.

Figure 5–24. Semi-Parallel FIR Structure Using Chained Cascaded Summation

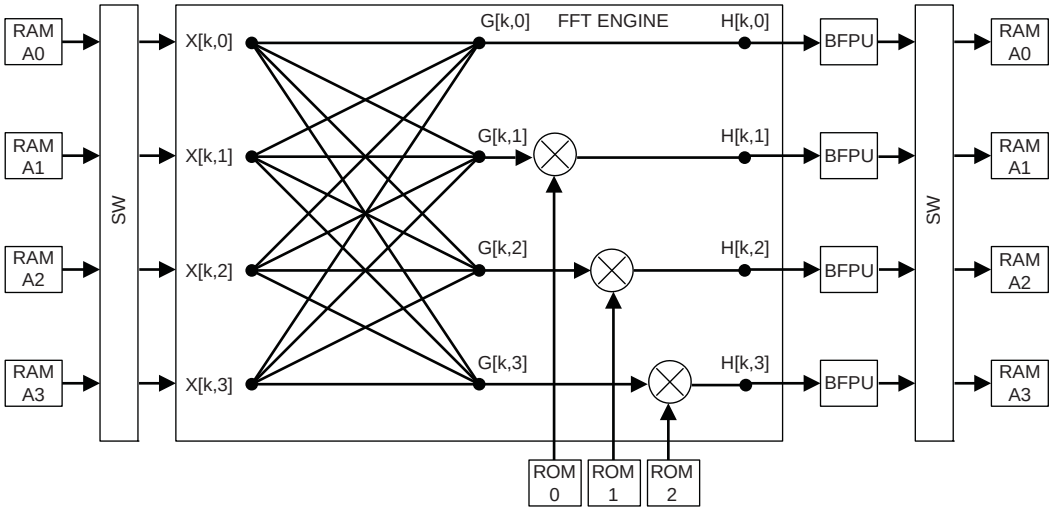


FFT Example

The Fast Fourier Transform (FFT) is a very common DSP function used to convert samples in the time domain to and from the frequency domain. A fundamental building block of the FFT is the FFT butterfly. FFTs are most efficient when operating on complex samples. You can use the Stratix III DSP block to form the core of a complex FFT butterfly very efficiently.

In Figure 5–25, a radix-4 butterfly is shown. Each butterfly requires three complex multipliers. This can be implemented in Stratix III using three half-DSP blocks assuming that the data and twiddle wordlengths are 18 bits or fewer.

Figure 5–25. Radix-4 Butterfly



Software Support

Altera provides two distinct methods for implementing various modes of the DSP block in a design: instantiation and inference. Both methods use the following Quartus II megafunctions:

- LPM_MULT
- ALTMULT_ADD
- ALTMULT_ACCUM
- ALTFP_MULT

You can instantiate the megafunctions in the Quartus II software to use the DSP block. Alternatively, with inference, you can create an HDL design and synthesize it using a third-party synthesis tool (such as LeonardoSpectrum, Synplify, or Quartus II Native Synthesis) that infers the appropriate megafunction by recognizing multipliers, multiplier adders, multiplier accumulators, and shift functions. Using either method, the Quartus II software maps the functionality to the DSP blocks during compilation.

For instructions about using the megafunctions and the *MegaWizard Plug-In Manager*, refer to the *Quartus II Software Help*.

For more information, refer to the *Synthesis* section in volume 1 of the *Quartus II Development Software Handbook*.

Chapter Revision History

Table 5–10 lists the revision history for this chapter.

Table 5–10. Document Revision History

Date	Version	Changes Made
March 2010	1.7	Updated for the Quartus II software version 9.1 SP2 release: ■ Updated Figure 5–6. ■ Removed “Conclusion” section.
May 2009	1.6	Updated Table 5–5.
February 2009	1.5	■ Updated Figure 5–6. ■ Removed “Referenced Documents” section.
October 2008	1.4	■ Updated Table 5–2. ■ Updated Figure 5–18. ■ Updated New Document Format.
May 2008	1.3	■ Updated Figure 5–12. ■ Updated Table 5–1. ■ Added “High Precision Multiplier Adder” section.
October 2007	1.2	■ Added section “Referenced Documents”. ■ Added live links for references.
May 2007	1.1	■ Updated Figures 1 to 21. ■ Added two new figures, Figure 5–10 and Figure 5–11. ■ Updated Table 5–1 and Table 5–5. ■ Deleted Table 5-10. ■ Added sections “Double Multiplier” and “Referenced Documents”. ■ Clarification added for “Shift Modes” on page 5–28.
November 2006	1.0	Initial Release.

6. Clock Networks and PLLs in Stratix III Devices

SIII51006-2.0

This chapter describes the hierarchical clock networks and multiple phase-locked loops (PLLs) with advanced features in Stratix® III devices. The large number of clocking resources, in combination with the clock synthesis precision provided by the PLLs, provide a complete clock management solution. The Altera® Quartus® II software compiler automatically turns off clock networks not used in the design, thereby reducing the overall power consumption of the device.

Stratix III devices deliver abundant PLL resources with up to 12 PLLs per device and up to 10 outputs per PLL. You can independently program every output, creating a unique, customizable clock frequency. Inherent jitter filtration and fine granularity control over multiply, divide ratios, and dynamic phase shift reconfiguration provide the high performance precision required in today's high-speed applications. Stratix III device PLLs are feature-rich, supporting advanced capabilities such as clock switchover, dynamic phase shifting, PLL reconfiguration, and reconfigurable bandwidth. Stratix III PLLs also support external feedback mode, spread-spectrum tracking, and post-scale counter cascading features.

The Quartus II software enables the PLLs and their features without requiring any external devices. The following sections describe the Stratix III clock networks and PLLs in detail.

Clock Networks in Stratix III Devices

The global clock networks (GCLKs), regional clock networks (RCLKs), and periphery clock networks (PCLKs) available in Stratix III devices are organized into hierarchical clock structures that provide up to 220 unique clock domains (16 GCLKs + 88 RCLKs + 116 PCLKs) within the Stratix III device and allow up to 67 unique GCLK, RCLK, and PCLK clock sources (16 GCLKs + 22 RCLKs + 29 PCLKs) per device quadrant.

Table 6–1 lists the clock resources available in Stratix III devices.

Table 6–1. Clock Resources in Stratix III Devices (Part 1 of 2)

Clock Resource	# of Resources Available	Source of Clock Resource
Clock input pins	32 Single-ended (16 Differential)	`CLK[0..15]p` and `CLK[0..15]n` pins
Global clock networks	16	`CLK[0..15]p/n` pins, PLL clock outputs, and logic array
Regional clock networks	64/88 *(1)*	`CLK[0..15]p/n` pins, PLL clock outputs, and logic array
Peripheral clock networks	116 (29 per device quadrant) *(2)*	DPA clock outputs, horizontal I/O pins, and logic array
GCLKs/RCLKs per quadrant	32/38 *(3)*	16 GCLKs + 16 RCLKs/ 16 GCLKs + 22 RCLKs

Table 6–1. Clock Resources in Stratix III Devices (Part 2 of 2)

Clock Resource	# of Resources Available	Source of Clock Resource
GCLKs/RCLKs per device	80/104 *(4)*	16 GCLKs + 64 RCLKs / 16 GCLKs + 88 RCLKs

Notes to Table 6–1:

(1) There are 64 RCLKs in EP3SL50, EP3SL70, EP3SL110, EP3SL150, EP3SE50, EP3SE80, and EP3SE110 devices. There are 88 RCLKs in EP3SL200, EP3SE260, and EP3SL340 devices.

(2) There are 56 PCLKs in EP3SL50, EP3SL70, and EP3SE50 devices. There are 88 PCLKs in EP3SL110, EP3SL150, EP3SL200, EP3SE80, and EP3SE110 devices. There are 112 PCLKs in EP3SE260 and 132 PCLKs in the EP3SL340 device.

(3) There are 32 GCLKs/RCLKs per quadrant in EP3SL50, EP3SL70, EP3SL110, EP3SL150, EP3SE50, EP3SE80, and EP3SE110 devices. There are 38 GCLKs/RCLKs per quadrant in EP3SL200, EP3SE260, and EP3SL340 devices.

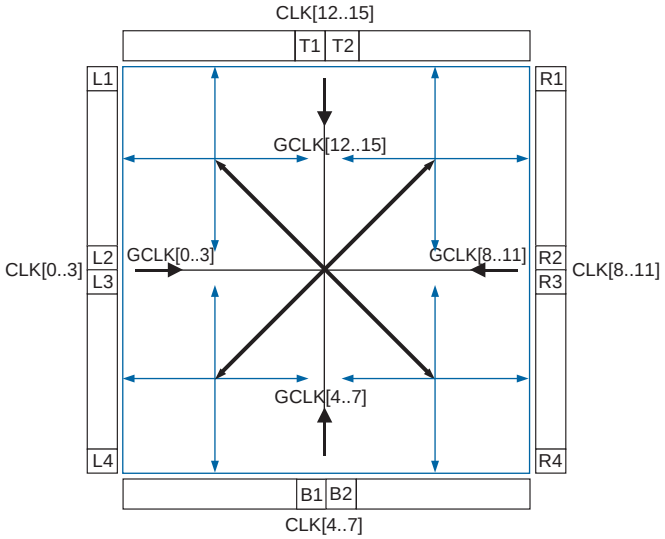
(4) There are 80 GCLKs/RCLKs per entire device in EP3SL50, EP3SL70, EP3SL110, EP3SL150, EP3SE50, EP3SE80, and EP3SE110 devices. There are 104 GCLKs/RCLKS per entire device in EP3SL200, EP3SE260, and EP3SL340 devices.

Stratix III devices have up to 32 dedicated single-ended clock pins or 16 dedicated differential clock pins (`CLK[0:15]p` and `CLK[0:15]n`) that can drive either the GCLK or RCLK networks. These clock pins are arranged on the four sides of the Stratix III device, as shown in Figure 6–1 to Figure 6–4.

Global Clock Networks

Stratix III devices provide up to 16 GCLKs that can drive throughout the entire device, serving as low-skew clock sources for functional blocks such as adaptive logic modules (ALMs), digital signal processing (DSP) blocks, TriMatrix memory blocks, and PLLs. Stratix III device I/O elements (IOEs) and internal logic can also drive GCLKs to create internally generated global clocks and other high fan-out control signals; for example, synchronous or asynchronous clears and clock enables. Figure 6–1 shows CLK pins and PLLs that can drive GCLK networks in Stratix III devices.

Figure 6–1. Global Clock Networks



Regional Clock Networks

The regional clock (RCLK) networks only pertain to the quadrant they drive into. The RCLK networks provide the lowest clock delay and skew for logic contained within a single device quadrant. Stratix III device I/O elements and internal logic within a given quadrant can also drive RCLKs to create internally generated regional clocks and other high fan-out control signals; for example, synchronous or asynchronous clears and clock enables. Figure 6–2 to Figure 6–4 show CLK pins and PLLs that can drive RCLK networks in Stratix III devices. The EP3SL50, EP3SL70, EP3SL110, EP3SL150, EP3SE50, EP3SE80, and EP3SE110 devices contain 64 RCLKs; the EP3SL200, EP3SE260, and EP3SL340 devices contain 88 RCLKs.

Figure 6–2. Regional Clock Networks (EP3SL50, EP3SL70, and EP3SE50 Devices)

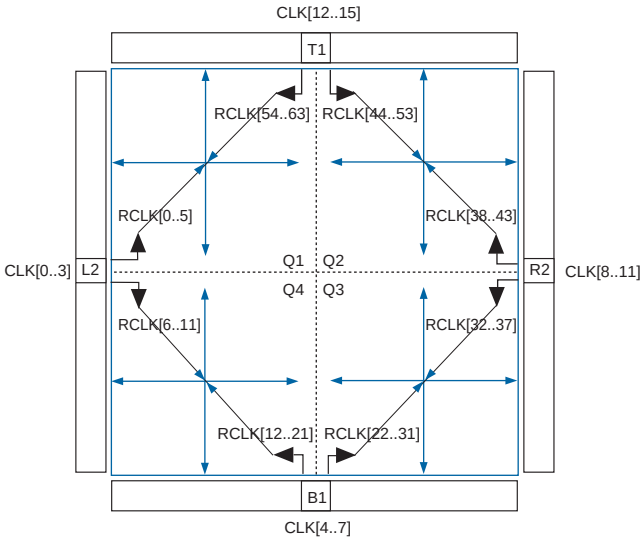


Figure 6–3. Regional Clock Networks (EP3SL110, EP3SL150, EP3SE80, and EP3SE110 Devices)

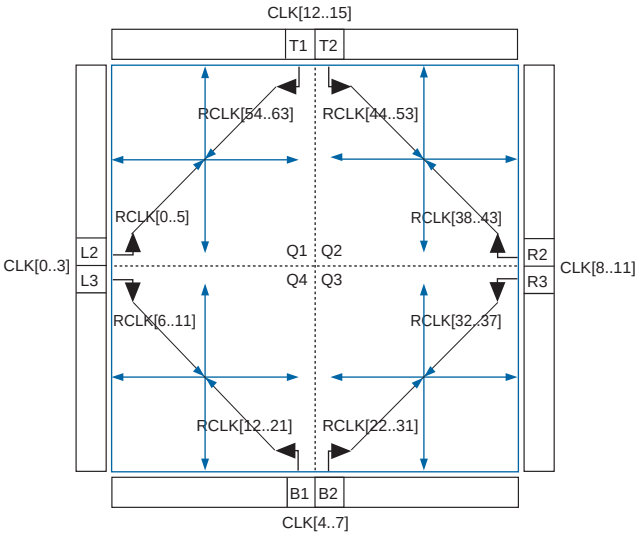
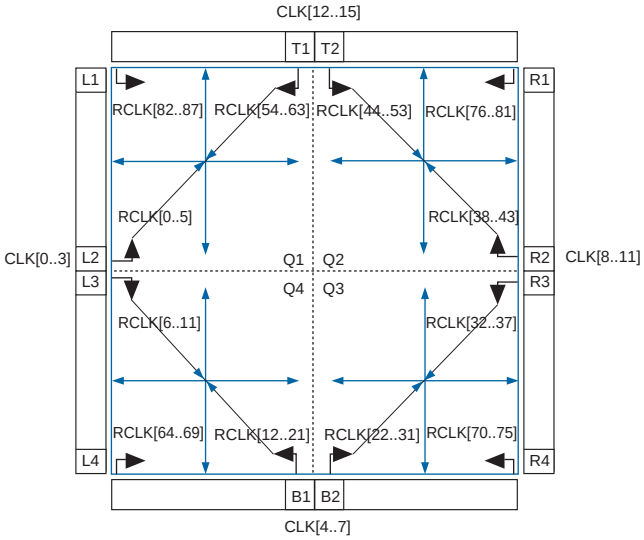


Figure 6–4. Regional Clock Networks (EP3SL200, EP3SE260, and EP3SL340 Devices) *(Note 1)*



Note to Figure 6–4:

(1) The corner RCLKs [64..87] can only be fed by their respective corner PLL outputs. Refer to Table 6–9 on page 6–13 for connectivity.

Periphery Clock Networks

Periphery clock (PCLK) networks shown in Figure 6–5 to Figure 6–9 are a collection of individual clock networks driven from the periphery of the Stratix III device. Clock outputs from the DPA block, horizontal I/O pins, and internal logic can drive the PCLK networks. The EP3SL50, EP3SL70, and EP3SE50 devices contain 56 PCLKs; the EP3SL110, EP3SL150, EP3SL200, EP3SE80, and EP3SE110 devices contain 88 PCLKs; the EP3SE260 device contains 112 PCLKs, and the EP3SL340 device contains 132 PCLKs. These PCLKs have higher skew compared to GCLK and RCLK networks and can be used instead of general purpose routing to drive signals into and out of the Stratix III device.

Figure 6–5. Periphery Clock Networks (EP3SL50, EP3SL70, and EP3SE50 Devices)

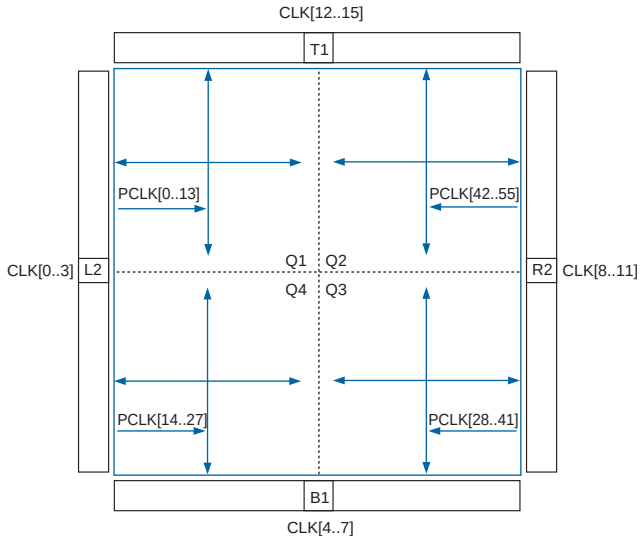


Figure 6–6. Periphery Clock Networks (EP3SL110, EP3SL150, EP3SE80, and EP3SE110 Devices)

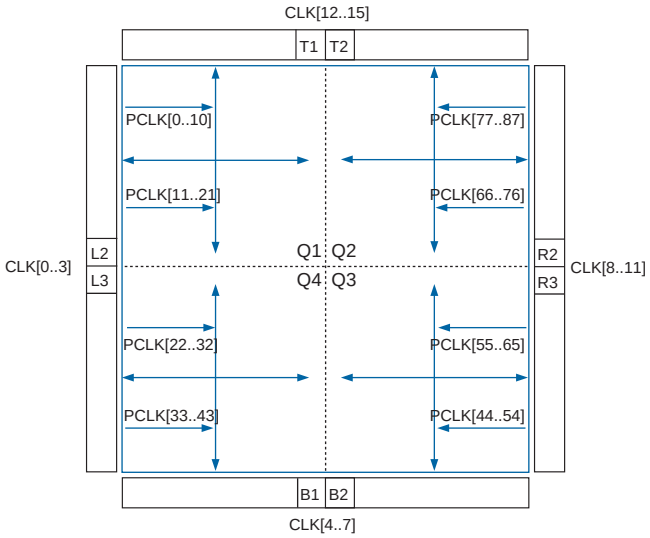


Figure 6–7. Periphery Clock Networks (EP3SL200 Devices)

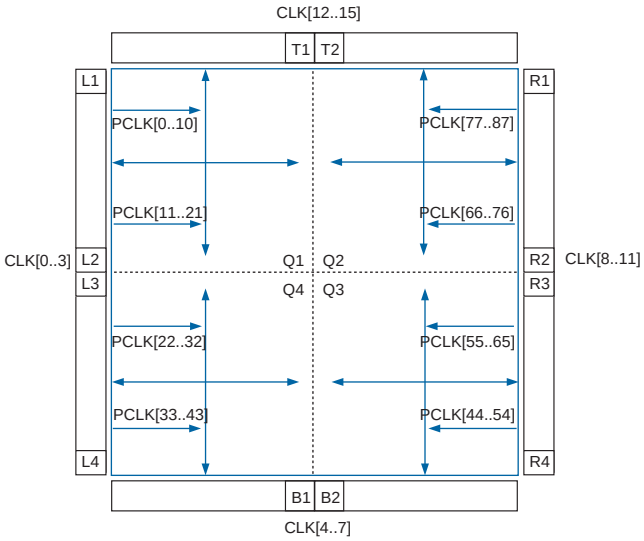


Figure 6–8. Periphery Clock Networks (EP3SE260 Devices)

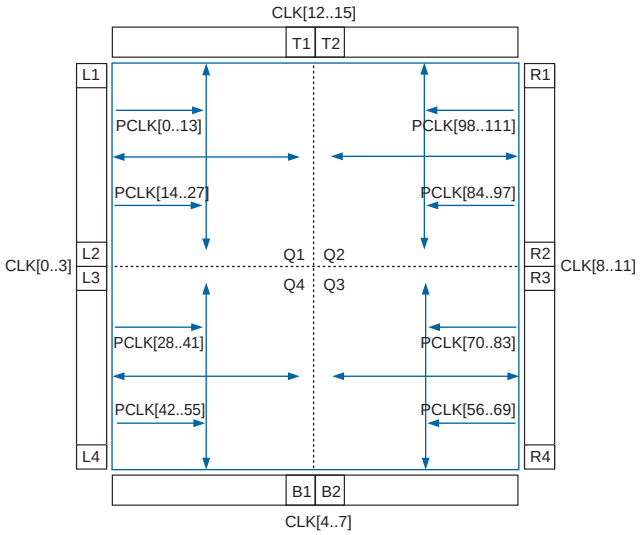
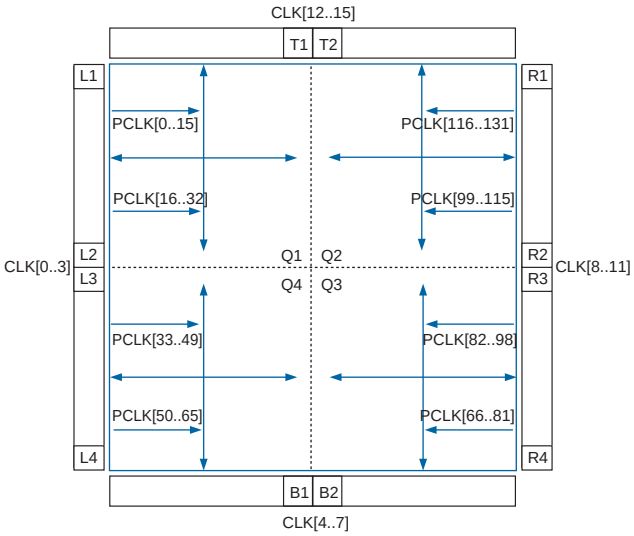


Figure 6–9. Periphery Clock Networks (EP3SL340 Devices)



Clocking Regions

Stratix III devices provide up to 104 distinct clock domains (16 GCLKs + 88 RCLKs) in the entire device. You can utilize these clock resources to form the following three different types of clock regions:

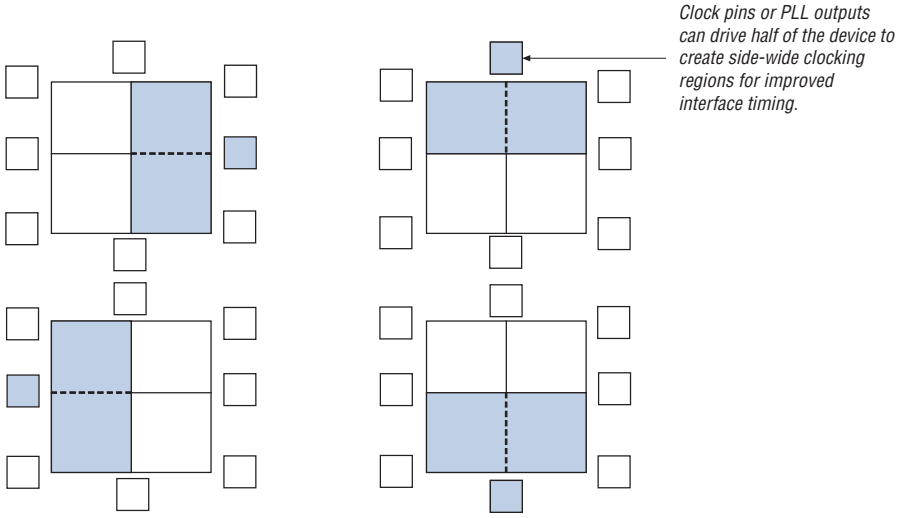
- Entire device clock region
- Regional clock region
- Dual-regional clock region

In order to form the entire device clock region, a source (not necessarily a clock signal) drives a global clock network that can be routed through the entire device. This clock region has the maximum delay compared to other clock regions but allows the signal to reach every destination within the device. This is a good option for routing global reset/clear signals or routing clocks throughout the device.

In order to form a regional clock region, a source drives a single-quadrant of the device. This clock region provides the lowest skew within a quadrant and is a good option if all destinations are within a single device quadrant.

To form a dual-regional clock region, a single source (a clock pin or PLL output) generates a dual-regional clock by driving two regional clock networks (one from each quadrant). This technique allows destinations across two device quadrants to use the same low-skew clock. The routing of this signal on an entire side has approximately the same delay as in a regional clock region. Internal logic can also drive a dual-regional clock network. Corner PLL outputs only span one quadrant and hence cannot generate a dual-regional clock network. Figure 6–10 shows the dual-regional clock region.

Figure 6–10. Stratix III Dual-Regional Clock Region



Clock Network Sources

In Stratix III devices, clock input pins, PLL outputs, and internal logic can drive the global and regional clock networks. Refer to Table 6–2 to Table 6–6 for the connectivity between dedicated `CLK[0..15]` pins and the global and regional clock networks.

Dedicated Clock Inputs Pins

The `CLK` pins can either be differential clocks or single-ended clocks. Stratix III devices support 16 differential clock inputs or 32 single-ended clock inputs. You can also use the dedicated clock input pins `CLK[15..0]` for high fan-out control signals such as asynchronous clears, presets, and clock enables for protocol signals such as `TRDY` and `IRDY` for PCI through global or regional clock networks.

Logic Array Blocks (LABs)

You can also drive each global and regional clock network via LAB-routing to enable internal logic to drive a high fan-out, low-skew signal.

☞ Stratix III device PLLs cannot be driven by internally generated GCLKs or RCLKs. The input clock to the PLL must come from dedicated clock input pins/PLL-fed GCLKs or RCLKs only.

☞ A spine clock is essentially another layer of routing below global/regional and periphery clocks before each clock is connected to the clock routing for each LAB row. The settings for a spine clock are transparent to all users. The Quartus II software takes care of the spine clock routing based on the global/regional and periphery clocks.

PLL Clock Outputs

Stratix III PLLs can drive both GCLK and RCLK networks, as detailed in Table 6–8 on page 6–12 and Table 6–9 on page 6–13.

Table 6–2 lists the connection between the dedicated clock input pins and GCLKs.

Table 6–2. Clock Input Pin Connectivity to Global Clock Networks

Clock Resources	CLK (p/n Pins)															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
GCLK0	✓	✓	✓	✓	—	—	—	—	—	—	—	—	—	—	—	—
GCLK1	✓	✓	✓	✓	—	—	—	—	—	—	—	—	—	—	—	—
GCLK2	✓	✓	✓	✓	—	—	—	—	—	—	—	—	—	—	—	—
GCLK3	✓	✓	✓	✓	—	—	—	—	—	—	—	—	—	—	—	—
GCLK4	—	—	—	—	✓	✓	✓	✓	—	—	—	—	—	—	—	—
GCLK5	—	—	—	—	✓	✓	✓	✓	—	—	—	—	—	—	—	—
GCLK6	—	—	—	—	✓	✓	✓	✓	—	—	—	—	—	—	—	—
GCLK7	—	—	—	—	✓	✓	✓	✓	—	—	—	—	—	—	—	—
GCLK8	—	—	—	—	—	—	—	—	✓	✓	✓	✓	—	—	—	—
GCLK9	—	—	—	—	—	—	—	—	✓	✓	✓	✓	—	—	—	—
GCLK10	—	—	—	—	—	—	—	—	✓	✓	✓	✓	—	—	—	—
GCLK11	—	—	—	—	—	—	—	—	✓	✓	✓	✓	—	—	—	—
GCLK12	—	—	—	—	—	—	—	—	—	—	—	—	✓	✓	✓	✓
GCLK13	—	—	—	—	—	—	—	—	—	—	—	—	✓	✓	✓	✓
GCLK14	—	—	—	—	—	—	—	—	—	—	—	—	✓	✓	✓	✓
GCLK15	—	—	—	—	—	—	—	—	—	—	—	—	✓	✓	✓	✓

Table 6–3 lists the connectivity between the dedicated clock input pins and RCLKs in device Quadrant 1. A given clock input pin can drive two adjacent regional clock networks to create a dual-regional clock network.

Table 6–3. Clock Input Pin Connectivity to Regional Clock Networks (Quadrant 1)

Clock Resource	CLK (p/n Pins)															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RCLK0	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK1	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK2	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK3	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—
RCLK4	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK5	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK54	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓
RCLK55	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—
RCLK56	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—
RCLK57	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—
RCLK58	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓
RCLK59	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—
RCLK60	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—
RCLK61	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—
RCLK62	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓
RCLK63	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—

Table 6–4 lists the connectivity between the dedicated clock input pins and RCLKs in device Quadrant 2. A given clock input pin can drive two adjacent regional clock networks to create a dual-regional clock network.

Table 6–4. Clock Input Pin Connectivity to Regional Clock Networks (Quadrant 2) (Part 1 of 2)

Clock Resource	CLK (p/n Pins)															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RCLK38	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—
RCLK39	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—
RCLK40	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—
RCLK41	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—
RCLK42	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—
RCLK43	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—
RCLK44	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓
RCLK45	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—
RCLK46	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—
RCLK47	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—
RCLK48	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓
RCLK49	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—

Table 6–4. Clock Input Pin Connectivity to Regional Clock Networks (Quadrant 2) (Part 2 of 2)

Clock Resource	CLK (p/n Pins)															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RCLK50	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—
RCLK51	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—
RCLK52	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓
RCLK53	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—

Table 6–5 lists the connectivity between the dedicated clock input pins and RCLKs in device Quadrant 3. A given clock input pin can drive two adjacent regional clock networks to create a dual-regional clock network.

Table 6–5. Clock Input Pin Connectivity to Regional Clock Networks (Quadrant 3)

Clock Resource	CLK (p/n Pins)															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RCLK22	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
RCLK23	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—
RCLK24	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—
RCLK25	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—
RCLK26	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
RCLK27	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—
RCLK28	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—
RCLK29	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—
RCLK30	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
RCLK31	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—
RCLK32	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—
RCLK33	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—
RCLK34	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—
RCLK35	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—
RCLK36	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—
RCLK37	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—

Table 6–6 lists the connectivity between the dedicated clock input pins and RCLKs in device Quadrant 4. A given clock input pin can drive two adjacent regional clock networks to create a dual-regional clock network.

Table 6–6. Clock Input Pin Connectivity to Regional Clock Networks (Quadrant 4)

Clock Resource	CLK (p/n Pins)															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RCLK6	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK7	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK8	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK9	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—
RCLK10	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK11	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK12	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
RCLK13	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—
RCLK14	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—
RCLK15	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—
RCLK16	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
RCLK17	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—
RCLK18	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—
RCLK19	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—
RCLK20	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
RCLK21	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—

Table 6–7 lists the dedicated clock input pin connectivity to Stratix III device PLLs.

Table 6–7. Stratix III Device PLLs and PLL Clock Pin Drivers (Part 1 of 2) *(Note 1)*

Dedicated Clock Input Pin (CLKp/n pins)	PLL Number											
	L1	L2	L3	L4	B1	B2	R1	R2	R3	R4	T1	T2
CLK0	✓	✓	✓	✓	—	—	—	—	—	—	—	—
CLK1	✓	✓	✓	✓	—	—	—	—	—	—	—	—
CLK2	✓	✓	✓	✓	—	—	—	—	—	—	—	—
CLK3	✓	✓	✓	✓	—	—	—	—	—	—	—	—
CLK4	—	—	—	—	✓	✓	—	—	—	—	—	—
CLK5	—	—	—	—	✓	✓	—	—	—	—	—	—
CLK6	—	—	—	—	✓	✓	—	—	—	—	—	—
CLK7	—	—	—	—	✓	✓	—	—	—	—	—	—
CLK8	—	—	—	—	—	—	✓	✓	✓	✓	—	—
CLK9	—	—	—	—	—	—	✓	✓	✓	✓	—	—
CLK10	—	—	—	—	—	—	✓	✓	✓	✓	—	—
CLK11	—	—	—	—	—	—	✓	✓	✓	✓	—	—
CLK12	—	—	—	—	—	—	—	—	—	—	✓	✓

Table 6–7. Stratix III Device PLLs and PLL Clock Pin Drivers (Part 2 of 2) *(Note 1)*

Dedicated Clock Input Pin (CLKp/n pins)	PLL Number											
	L1	L2	L3	L4	B1	B2	R1	R2	R3	R4	T1	T2
CLK13	—	—	—	—	—	—	—	—	—	—	✓	✓
CLK14	—	—	—	—	—	—	—	—	—	—	✓	✓
CLK15	—	—	—	—	—	—	—	—	—	—	✓	✓
PLL_L1_CLKp *(2)*	✓	—	—	—	—	—	—	—	—	—	—	—
PLL_L1_CLKn *(2),(3)*	✓	—	—	—	—	—	—	—	—	—	—	—
PLL_L4_CLKp *(2)*	—	—	—	✓	—	—	—	—	—	—	—	—
PLL_L4_CLKn *(2),(3)*	—	—	—	✓	—	—	—	—	—	—	—	—
PLL_R1_CLKp *(2)*	—	—	—	—	—	—	✓	—	—	—	—	—
PLL_R1_CLKn *(2),(3)*	—	—	—	—	—	—	✓	—	—	—	—	—
PLL_R4_CLKp *(2)*	—	—	—	—	—	—	—	—	—	✓	—	—
PLL_R4_CLKn *(2),(3)*	—	—	—	—	—	—	—	—	—	✓	—	—

Notes to Table 6–7:

(1) For compensated PLLs input, only the dedicated CLK pins in the same I/O bank as the PLL used are compensated inputs.

(2) If both PLL_<L1/L4/R1/R4>_CLKp and PLL_<L1/L4/R1/R4>_CLKn pins are not used as a pair of differential clock pins, they can be used independently as single-ended clock input pins.

(3) For single-ended clock input, CLKn pins use the global network to drive the PLLs.

Clock Output Connections

PLLs in Stratix III devices can drive up to 20 regional clock networks and four global clock networks. Refer to Table 6–8 for Stratix III PLL connectivity to GCLK networks. The Quartus II software automatically assigns PLL clock outputs to regional or global clock networks.

Table 6–8 lists how the PLL clock outputs connect to GCLK networks.

Table 6–8. PLL Connectivity to GCLKs on Stratix III Devices (Part 1 of 2) *(Note 1)*

Clock Network	PLL Number											
	L1	L2	L3	L4	B1	B2	R1	R2	R3	R4	T1	T2
GCLK0	✓	✓	✓	✓	—	—	—	—	—	—	—	—
GCLK1	✓	✓	✓	✓	—	—	—	—	—	—	—	—
GCLK2	✓	✓	✓	✓	—	—	—	—	—	—	—	—
GCLK3	✓	✓	✓	✓	—	—	—	—	—	—	—	—
GCLK4	—	—	—	—	✓	✓	—	—	—	—	—	—
GCLK5	—	—	—	—	✓	✓	—	—	—	—	—	—
GCLK6	—	—	—	—	✓	✓	—	—	—	—	—	—
GCLK7	—	—	—	—	✓	✓	—	—	—	—	—	—
GCLK8	—	—	—	—	—	—	✓	✓	✓	✓	—	—
GCLK9	—	—	—	—	—	—	✓	✓	✓	✓	—	—
GCLK10	—	—	—	—	—	—	✓	✓	✓	✓	—	—
GCLK11	—	—	—	—	—	—	✓	✓	✓	✓	—	—

Table 6–8. PLL Connectivity to GCLKs on Stratix III Devices (Part 2 of 2) *(Note 1)*

Clock Network	PLL Number											
	L1	L2	L3	L4	B1	B2	R1	R2	R3	R4	T1	T2
GCLK12	—	—	—	—	—	—	—	—	—	—	✓	✓
GCLK13	—	—	—	—	—	—	—	—	—	—	✓	✓
GCLK14	—	—	—	—	—	—	—	—	—	—	✓	✓
GCLK15	—	—	—	—	—	—	—	—	—	—	✓	✓

Note to Table 6–8:

(1) Only PLL counter outputs C0 - C3 can drive GCLK networks.

Table 6–9 lists how the PLL clock outputs connect to RCLK networks.

Table 6–9. Regional Clock Outputs From PLLs on Stratix III Devices *(Note 1)*

Clock Resource	PLL Number											
	L1	L2	L3	L4	B1	B2	R1	R2	R3	R4	T1	T2
RCLK[0..11]	—	✓	✓	—	—	—	—	—	—	—	—	—
RCLK[12..31]	—	—	—	—	✓	✓	—	—	—	—	—	—
RCLK[32..43]	—	—	—	—	—	—	—	✓	✓	—	—	—
RCLK[44..63]	—	—	—	—	—	—	—	—	—	—	✓	✓
RCLK[64..69]	—	—	—	✓	—	—	—	—	—	—	—	—
RCLK[70..75]	—	—	—	—	—	—	—	—	—	✓	—	—
RCLK[76..81]	—	—	—	—	—	—	✓	—	—	—	—	—
RCLK[82..87]	✓	—	—	—	—	—	—	—	—	—	—	—

Note to Table 6–9:

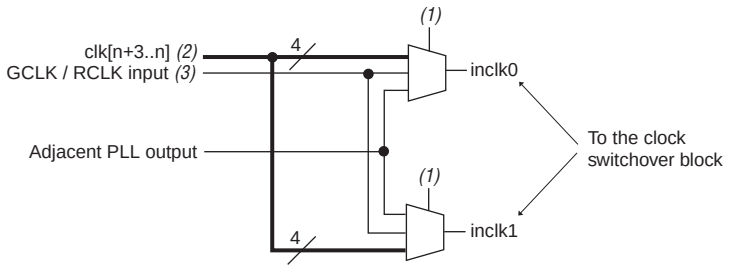
(1) All PLL counter outputs can drive RCLK networks.

Clock Source Control for PLLs

The clock input to Stratix III PLLs comes from clock input multiplexers. The clock multiplexer inputs come from dedicated clock input pins, PLLs through the GCLK and RCLK networks, or from dedicated connections between adjacent Top/Bottom and Left/Right PLLs. The clock input sources to Top/Bottom and Left/Right PLLs (L2, L3, T1, T2, B1, B2, R2, and R3) are shown in Figure 6–11; the corresponding clock input sources to Left/Right PLLs (L1, L4, R1, and R4) are shown in Figure 6–12.

The multiplexer select lines are set in the configuration file (SRAM object file [**.sof**] or programmer object file [**.pof**]) only. Once programmed, this block cannot be changed without loading a new configuration file (**.sof** or **.pof**). The Quartus II software automatically sets the multiplexer select signals depending on the clock sources selected in the design.

Figure 6–11. Clock Input Multiplexer Logic for L2, L3, T1, T2, B1, B2, R2, and R3 PLLs



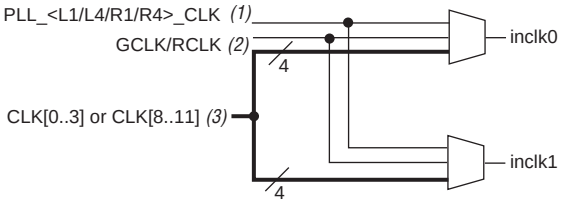
Notes to Figure 6–11:

(1) The input clock multiplexing is controlled through a configuration file (**.sof** or **.pof**) only and cannot be dynamically controlled in user mode operation.

(2) n=0 for L2 and L3 PLLs; n=4 for B1 and B2 PLLs; n=8 for R2 and R3 PLLs, and n=12 for T1 and T2 PLLs.

(3) The global (GCLK) or regional (RCLK) clock input can be driven by an output from another PLL, a pin-driven global or regional clock, or through a clock control block provided the clock control block is fed by an output from another PLL or a pin driven dedicated global or regional clock. An internally generated global signal or general purpose I/O pin cannot drive the PLL.

Figure 6–12. Clock Input Multiplexer Logic for L1, L4, R1, and R4 PLLs



Notes to Figure 6–12:

(1) Dedicated clock input pins to PLLs - L1, L4, R1 and R4, respectively. For example, `PLL_L1_CLK` is the dedicated clock input for `PLL_L1`.

(2) The global (GCLK) or regional (RCLK) clock input can be driven by an output from another PLL, a pin-driven global or regional clock, or through a clock control block provided the clock control block is fed by an output from another PLL or a pin driven dedicated global or regional clock. An internally generated global signal or general purpose I/O pin cannot drive the PLL.

(3) The center clock pins can feed the corner PLLs on the same side directly, through a dedicated path. However, these paths may not be fully compensated.

Clock Control Block

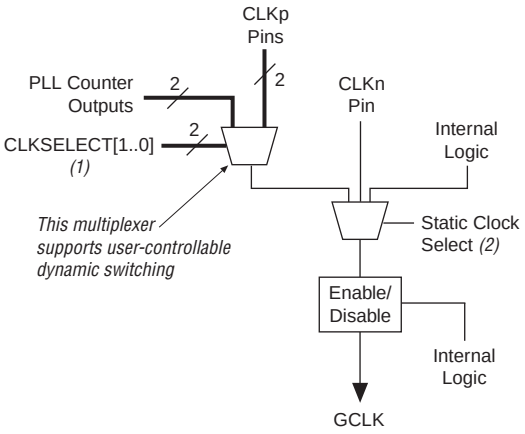
Every global and regional clock network has its own clock control block. The control block provides the following features:

- Clock source selection (dynamic selection for global clocks)
- Global clock multiplexing
- Clock power down (static or dynamic clock enable or disable)

You can select the clock source for the global clock select block either statically or dynamically. You can either statically select the clock source using a setting in the Quartus II software, or you can dynamically select the clock source using internal logic to drive the multiplexer select inputs. When selecting the clock source dynamically, you can either select two PLL outputs (such as `CLK0` or `CLK1`), or a combination of clock pins or PLL outputs.

Figure 6–13 and Figure 6–14 show the global clock and regional clock select blocks, respectively.

Figure 6–13. Stratix III Global Clock Control Block

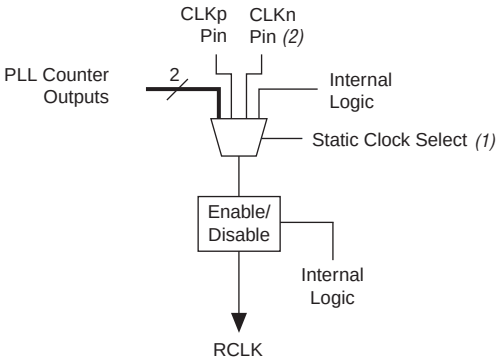


Notes to Figure 6–13:

(1) These clock select signals can only be dynamically controlled through internal logic when the device is operating in user mode.

(2) These clock select signals can only be set through a configuration file (**.sof** or **.pof**) and cannot be dynamically controlled during user mode operation.

Figure 6–14. Regional Clock Control Block



Notes to Figure 6–14:

(1) This clock select signal can only be statically controlled through a configuration file (**.sof** or **.pof**) and cannot be dynamically controlled during user mode operation.

(2) CLKn pin is not a dedicated clock input when it is use as a single-ended PLL clock input and it is not fully compensated.

The clock source selection for the regional clock select block can only be controlled statically using configuration bit settings in the configuration file (**.sof** or **.pof**) generated by the Quartus II software.

The Stratix III clock networks can be powered down by both static and dynamic approaches. When a clock net is powered down, all the logic fed by the clock net is in an off-state, thereby reducing the overall power consumption of the device. The unused global and regional clock networks are automatically powered down through configuration bit settings in the configuration file (**.sof** or **.pof**) generated by the

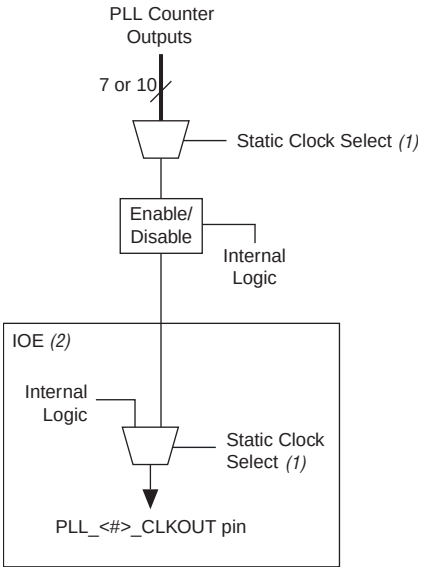
Quartus II software. The dynamic clock enable or disable feature allows the internal logic to control power-up or power-down synchronously on GCLK and RCLK networks, including dual-regional clock regions. This function is independent of the PLL and is applied directly on the clock network, as shown in Figure 6–13 and Figure 6–14.

You can set the input clock sources and the `clkena` signals for the global and regional clock network multiplexers through the Quartus II software using the ALTCLKCTRL megafunction. You can also enable or disable the dedicated external clock output pins using the ALTCLKCTRL megafunction.

When using the ALTCLKCTRL megafunction to implement clock source selection (dynamic), the inputs from the clock pins feed the `inclock[0..1]` ports of the multiplexer, while the PLL outputs feed the `inclock[2..3]` ports. You can choose from among these inputs using the `CLKSELECT[1..0]` signal.

Figure 6–15 shows the external PLL output clock control block.

Figure 6–15. Stratix III External PLL Output Clock Control Block



Notes to Figure 6–15:

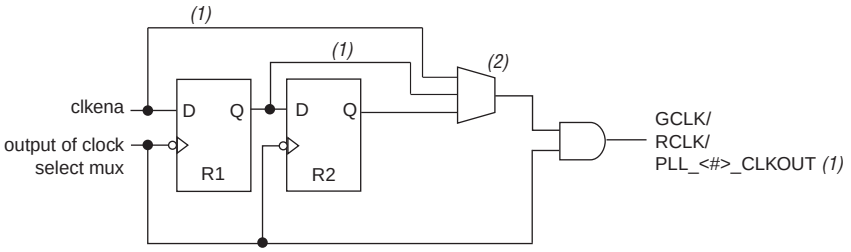
(1) This clock select signal can only be set through a configuration file (**.sof** or **.pof**) and cannot be dynamically controlled during user mode operation.

(2) The clock control block feeds to a multiplexer within the `PLL_<#>_CLKOUT` pin's IOE. The `PLL_<#>_CLKOUT` pin is a dual-purpose pin. Therefore, this multiplexer selects either an internal signal or the output of the clock control block.

Clock Enable Signals

Figure 6–16 shows how the clock enable/disable circuit of the clock control block is implemented in Stratix III devices.

Figure 6–16. clkena Implementation



Notes to Figure 6–16:

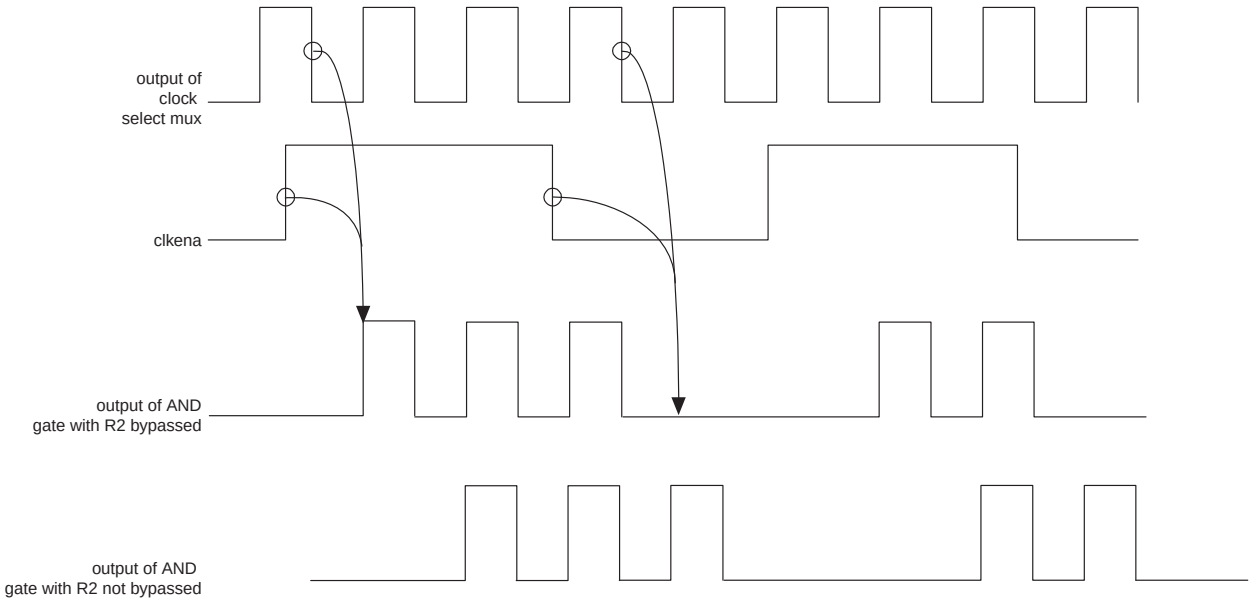
(1) The R1 and R2 bypass paths are not available for PLL external clock outputs.

(2) The select line is statically controlled by a bit setting in the configuration file (**.sof** or **.pof**).

In Stratix III devices, the `clkena` signals are supported at the clock network level instead of at the PLL output counter level. This allows you to gate off the clock even when a PLL is not being used. You can also use the `clkena` signals to control the dedicated external clocks from the PLLs. Figure 6–17 shows the waveform example for a clock output enable. `clkena` is synchronous to the falling edge of the clock output.

Stratix III devices also have an additional metastability register that aids in asynchronous enable/disable of the GCLK and RCLK networks. This register can be optionally bypassed in the Quartus II software.

Figure 6–17. clkena Signals



Note to Figure 6–17:

(1) You can use the `clkena` signals to enable or disable the global and regional networks or the `PLL_<#>_CLKOUT` pins.

The PLL can remain locked independently of the `clkena` signals because the loop-related counters are not affected. This feature is useful for applications that require a low power or sleep mode. The `clkena` signal can also disable clock outputs if the system is not tolerant of frequency overshoot during resynchronization.

PLLs in Stratix III Devices

Stratix III devices offer up to 12 PLLs that provide robust clock management and synthesis for device clock management, external system clock management, and high-speed I/O interfaces. The nomenclature for the PLLs follows their geographical location in the device floor plan. The PLLs that reside on the top and bottom sides of the device are named `PLL_T1`, `PLL_T2`, `PLL_B1` and `PLL_B2`; the PLLs that reside on the left and right sides of the device are named `PLL_L1`, `PLL_L2`, `PLL_L3`, `PLL_L4`, `PLL_R1`, `PLL_R2`, `PLL_R3`, and `PLL_R4`, respectively.

Table 6–10 lists the PLLs available in the Stratix III device family.

Table 6–10. Stratix III Device PLL Availability

Device	L1	L2	L3	L4	T1	T2	B1	B2	R1	R2	R3	R4
EP3SL50	—	✓	—	—	✓	—	✓	—	—	✓	—	—
EP3SL70	—	✓	—	—	✓	—	✓	—	—	✓	—	—
EP3SL110 *(1)*	—	✓	✓	—	✓	✓	✓	✓	—	✓	✓	—
EP3SL150 *(1)*	—	✓	✓	—	✓	✓	✓	✓	—	✓	✓	—
EP3SL200 *(1)*, *(2)*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EP3SL340 *(2)*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EP3SE50	—	✓	—	—	✓	—	✓	—	—	✓	—	—
EP3SE80 *(1)*	—	✓	✓	—	✓	✓	✓	✓	—	✓	✓	—
EP3SE110 *(1)*	—	✓	✓	—	✓	✓	✓	✓	—	✓	✓	—
EP3SE260 *(2)*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Notes to Table 6–10:

(1) PLLs T2, B2, L3, and R3 are not available in the F780 package.

(2) PLLs L1, L4, R1, and R4 are not available in the H780, F1152, and H1152 packages.

All Stratix III PLLs have the same core analog structure with only minor differences in the features that are supported. Table 6–11 lists the features of the Top/Bottom and Left/Right PLLs in Stratix III devices.

Table 6–11. Stratix III PLL Features

Feature	Stratix III Top/Bottom PLLs	Stratix III Left/Right PLLs
C (output) counters	10	7
M, N, C counter sizes	1 to 512	1 to 512
Dedicated clock outputs	6 single-ended or 4 single-ended and 1 differential pair	2 single-ended or 1 differential pair
Clock input pins	4 single-ended or 2 differential pin pairs	4 single-ended or 2 differential pin pairs
External feedback input pin	Single-ended or differential	Single-ended only
Spread-spectrum input clock tracking	Yes *(1)*	Yes *(1)*
PLL cascading	Through GCLK and RCLK and dedicated path between adjacent PLLs	Through GCLK and RCLK and dedicated path between adjacent PLLs *(2)*
Compensation modes	All except LVDS clock network compensation	All except external feedback mode when using differential I/Os
PLL drives LVDSCLK and LOADEN	No	Yes
VCO output drives DPA clock	No	Yes
Phase shift resolution	Down to 96.125 ps *(3)*	Down to 96.125 ps *(3)*
Programmable duty cycle	Yes	Yes
Output counter cascading	Yes	Yes
Input clock switchover	Yes	Yes

Notes to Table 6–11:

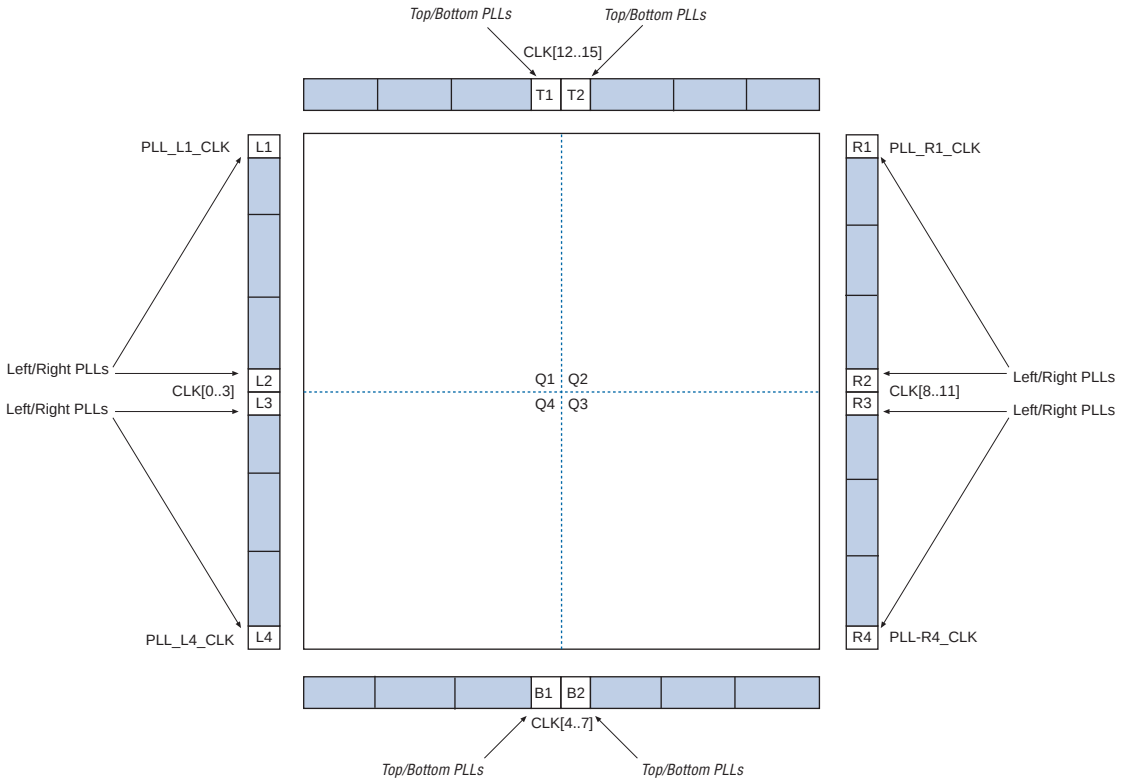
(1) Provided input clock jitter is within input jitter tolerance specifications.

(2) The dedicated path between adjacent PLLs is not available on L1, L4, R1, and R4 PLLs.

(3) The smallest phase shift is determined by the voltage-controlled oscillator (VCO) period divided by eight. For degree increments, the Stratix III device can shift all output frequencies in increments of at least 45 degrees. Smaller degree increments are possible depending on the frequency and divide parameters.

Figure 6–18 shows the location of the PLLs in Stratix III devices.

Figure 6–18. Stratix III PLL Locations



Stratix III PLL Hardware Overview

Stratix III devices contain up to 12 PLLs with advanced clock management features. The main goal of a PLL is to synchronize the phase and frequency of an internal or external clock to an input reference clock. There are a number of components that comprise a PLL to achieve this phase alignment.

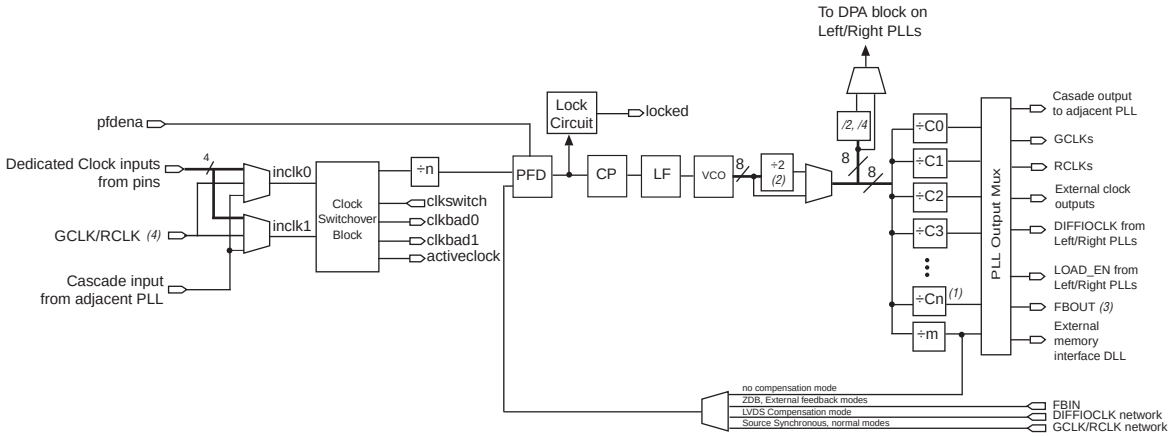
Stratix III PLLs align the rising edge of the input reference clock to a feedback clock using the phase-frequency detector (PFD). The falling edges are determined by the duty-cycle specifications. The PFD produces an up or down signal that determines whether the voltage-controlled oscillator (VCO) needs to operate at a higher or lower frequency. The output of the PFD feeds the charge pump and loop filter, which produces a control voltage for setting the VCO frequency. If the PFD produces an up signal, then the VCO frequency increases. A down signal decreases the VCO frequency. The PFD outputs these up and down signals to a charge pump. If the charge pump receives an up signal, current is driven into the loop filter. Conversely, if the charge pump receives a down signal, current is drawn from the loop filter.

The loop filter converts these up and down signals to a voltage that is used to bias the VCO. The loop filter also removes glitches from the charge pump and prevents voltage overshoot, which filters the jitter on the VCO. The voltage from the loop filter determines how fast the VCO operates. A divide counter (m) is inserted in the feedback loop to increase the VCO frequency above the input reference frequency. VCO frequency (f_{VCO}) is equal to (m) times the input reference clock (f_{REF}). The input reference clock (f_{REF}) to the PFD is equal to the input clock (f_{IN}) divided by the pre-scale counter (N). Therefore, the feedback clock (f_{FB}) applied to one input of the PFD is locked to the f_{REF} that is applied to the other input of the PFD.

The VCO output from Left/Right PLLs can feed seven post-scale counters (C[0..6]), while the corresponding VCO output from Top/Bottom PLLs can feed ten post-scale counters (C[0..9]). These post-scale counters allow a number of harmonically related frequencies to be produced by the PLL.

Figure 6–19 shows a simplified block diagram of the major components of the Stratix III PLL.

Figure 6–19. Stratix III PLL Block Diagram



Notes to Figure 6–19:

(1) The number of post-scale counters is 7 for Left/Right PLLs and 10 for Top/Bottom PLLs.

(2) This is the VCO post-scale counter K. If the design enables this ÷2 counter, the device can use a VCO frequency range of 300 to 650 MHz. The VCO frequency reported by the Quartus II software is divided by the post-scale counter K.

(3) The FBOUT port is fed by the M counter in Stratix III PLLs.

(4) The global (GCLK) or regional (RCLK) clock input can be driven by an output from another PLL, a clock pin-driven global or regional clock, or through a clock control block provided the clock control block is fed by an output from another PLL or a pin driven dedicated global or regional clock. An internally generated global signal or general purpose I/O pin cannot drive the PLL.

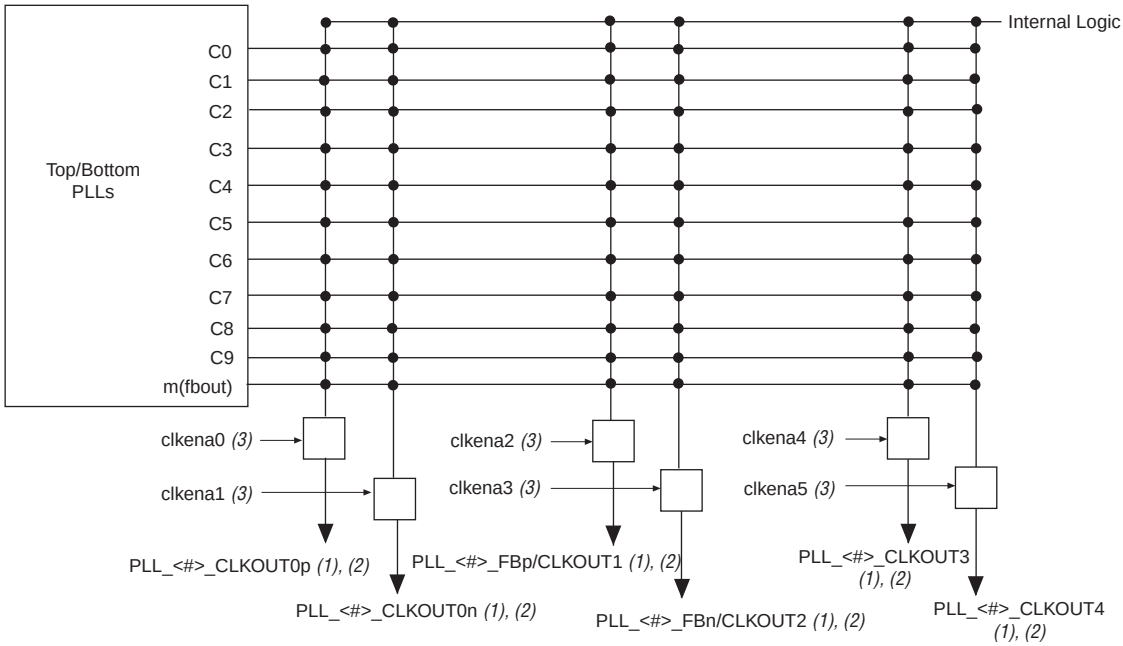
PLL Clock I/O Pins

Each Top/Bottom PLL supports six clock I/O pins, organized as three pairs of pins:

- 1st pair: 2 single-ended I/O or 1 differential I/O
- 2nd pair: 2 single-ended I/O, 1 differential external feedback input (FBp/FBn), or 1 single-ended external feedback input (FBp)
- 3rd pair: 2 single-ended I/O or 1 differential input

Figure 6–20 shows the clock I/O pins associated with Top/Bottom PLLs.

Figure 6–20. External Clock Outputs for Top/Bottom PLLs



Notes to Figure 6–20:

(1) These clock output pins can be fed by any one of the `C[9..0]`, m counters.

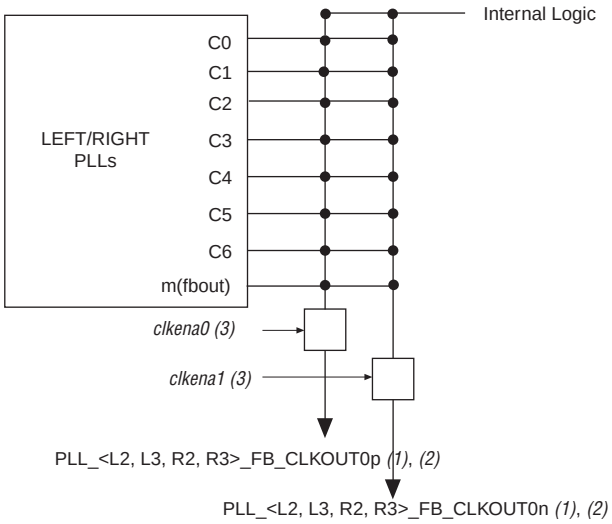
(2) The `CLKOUT0p` and `CLKOUT0n` pins can be either single-ended or differential clock outputs. `CLKOUT1` and `CLKOUT2` pins are dual-purpose I/O pins that can be used as two single-ended outputs, one differential external feedback input pin pair or one single-ended external feedback input pin (`CLKOUT1` only). `CLKOUT3` and `CLKOUT4` pins are two single-ended output pins.

(3) These external clock enable signals are available only when using the ALTCLKCTRL megafunction.

Any of the output counters (`C[9..0]` on Top/Bottom PLLs and `C[6..0]` on Left/Right PLLs) or the `M` counter can feed the dedicated external clock outputs, as shown in Figure 6–20 and Figure 6–21. Therefore, one counter or frequency can drive all output pins available from a given PLL.

Each Left/Right PLL supports two clock I/O pins, configured as either two single-ended I/Os or one differential I/O pair. When using both pins as single-ended I/Os, one of them can be the clock output while the other pin is the external feedback input (FB) pin. Hence, Left/Right PLLs only support external feedback mode for single-ended I/O standards.

Figure 6–21. External Clock Outputs for Left/Right PLLs



Notes to Figure 6–21:

(1) These clock output pins can be fed by any one of the `C[6..0]`, m counters.

(2) The `CLKOUT0p` and `CLKOUT0n` pins are dual-purpose I/O pins that can be used as two single-ended outputs or one single-ended output and one external feedback input pin.

(3) These external clock enable signals are available only when using the ALTCLKCTRL megafunction.

Each pin of a single-ended output pair can either be in-phase or 180-degrees out-of-phase. The Quartus II software places the NOT gate in the design into the IOE to implement 180-degrees phase with respect to the other pin in the pair. The clock output pin pairs support the same I/O standards as standard output pins (in the top and bottom banks) as well as LVDS, LVPECL, differential HSTL, and differential SSTL.

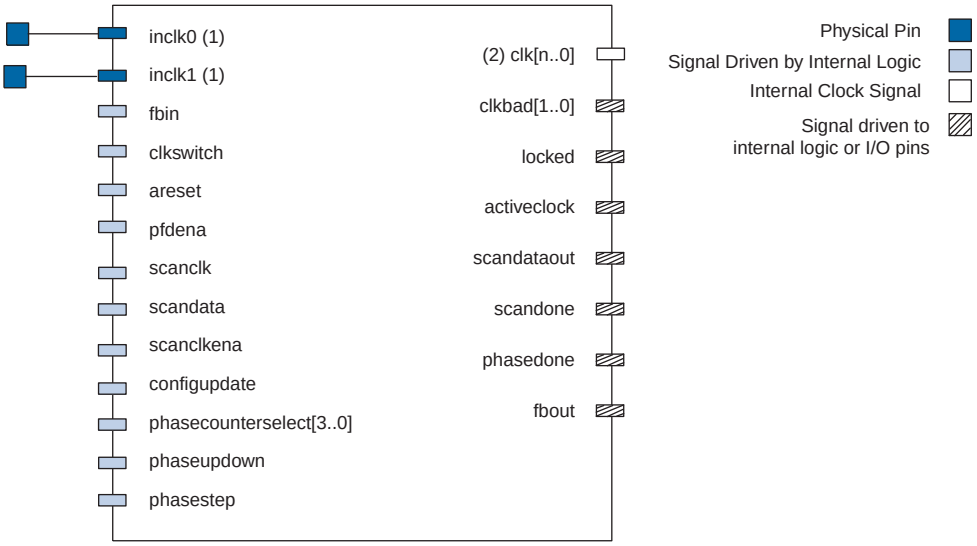
To determine which I/O standards are supported by the PLL clock input and output pins, refer to the *Stratix III Device I/O Features* chapter.

Stratix III PLLs can also drive out to any regular I/O pin through the global or regional clock network. You can use the external clock output pins as user I/O pins if external PLL clocking is not needed.

Stratix III PLL Software Overview

Stratix III PLLs are enabled in the Quartus II software by using the ALTPLL megafunction. Figure 6–22 shows the Stratix III PLL ports as they are named in the ALTPLL megafunction of the Quartus II software.

Figure 6–22. Stratix III PLL Ports



Notes to Figure 6–22:

(1) You can feed the `inclk0` or `inclk1` clock input from any one of four dedicated clock pins located on the same side of the device as the PLL.

(2) You can drive to global or regional clock networks or dedicated external clock output pins. n = 6 for Left/Right PLLs and n = 9 for Top/Bottom PLLs.

Table 6–12 lists the PLL input signals for Stratix III devices.

Table 6–12. PLL Input Signals (Part 1 of 2)

Port	Description	Source	Destination
`inclk0`	Input clock to the PLL	Dedicated pin, adjacent PLL, GCLK, or RCLK network	N counter
`inclk1`	Input clock to the PLL	Dedicated pin, adjacent PLL, GCLK, or RCLK network	N counter
`fbin`	Compensation feedback input to the PLL. Share the same clock spines used by GCLK/RCLKs.	Pin LVSDCLK	PFD
`clkswitch`	Switchover signal used to initiate clock switchover asynchronously. When used in manual switchover, `clkswitch` is used as a select signal between `inclk0` and `inclk1`. If `clkswitch` = `0`, `inclk0` is selected. If `clkswitch` = `1`, `inclk1` is selected. Both `inclk0` and `inclk1` must be switched in order for manual switchover to function.	Logic array or I/O pin	Clock switchover circuit

Table 6–12. PLL Input Signals (Part 2 of 2)

Port	Description	Source	Destination
`areset`	Signal used to reset the PLL which resynchronizes all the counter outputs. Active high	Logic array	General PLL control signal
`pfdena`	Enables the outputs from the phase frequency detector. Active high	Logic array	PFD
`scanclk`	Serial clock signal for the real-time PLL reconfiguration feature.	Logic array	Reconfiguration circuit
`scandata`	Serial input data stream for the real-time PLL reconfiguration feature.	Logic array	Reconfiguration circuit
`scanclkena`	Enables `scanclk` and allows the `scandata` to be loaded in the scan chain. Active high	Logic array or I/O pin	PLL reconfiguration circuit
`configupdate`	Writes the data in the scan chain to the PLL. Active high	Logic array or I/O pins	PLL reconfiguration circuit
`phasecounterselect[3:0]`	Selects corresponding PLL counter for dynamic phase shift	Logic array or I/O pins	PLL reconfiguration circuit
`phaseupdown`	Selects dynamic phase shift direction; 1 = UP; 0 = DOWN	Logic array or I/O pin	PLL reconfiguration circuit
`phasestep`	Logic high enables dynamic phase shifting	Logic array or I/O pin	PLL reconfiguration circuit

Table 6–13 lists the PLL output signals for Stratix III devices.

Table 6–13. PLL Output Signals (Part 1 of 2)

Port	Description	Source	Destination
`clk[9..0]` for Top/Bottom PLLs `clk[6..0]` for Left/Right PLLs	PLL output counters driving regional, global, or external clocks.	PLL counter	Internal or external clock
`clkbad[1..0]`	Signals indicating which reference clock is no longer toggling. `clkbad1` indicates `inclk1` status, `clkbad0` indicates `inclk0` status. 0 = good; 1 = bad	PLL switchover circuit	Logic array
`locked`	Lock output from lock detect circuit. Active high	PLL lock detect	Logic array
`activeclock`	Signal to indicate which clock (`0 = inclk0` or `1 = inclk1`) is driving the PLL. If this signal is low, `inclk0` drives the PLL. If this signal is high, `inclk1` drives the PLL.	PLL clock multiplexer	Logic array
`scandataout`	Output of the last shift register in the scan chain.	PLL scan chain	Logic array

Table 6–13. PLL Output Signals (Part 2 of 2)

Port	Description	Source	Destination
scandone	Signal indicating when the PLL has completed reconfiguration. One-to-0 transition indicates that the PLL has been reconfigured.	PLL scan chain	Logic array
phasedone	When asserted it indicates that the phase reconfiguration is complete and the PLL is ready to act on a possible second reconfiguration. Asserts based on internal PLL timing. De-asserts on rising edge of SCANCLK.	PLL scan chain	Logic array
fbout	Output of m counter. Used for clock delay compensation.	M counter	Logic array

Clock Feedback Modes

Stratix III PLLs support up to six different clock feedback modes. Each mode allows clock multiplication and division, phase shifting, and programmable duty cycle. Table 6–14 lists the clock feedback modes supported by Stratix III PLLs.

Table 6–14. Clock Feedback Mode Availability

Clock Feedback Mode	Availability	
	Top/Bottom PLLs	Left/Right PLLs
Source-synchronous mode	Yes	Yes
No-compensation mode	Yes	Yes
Normal mode	Yes	Yes
Zero-delay buffer (ZDB) mode	Yes	Yes
External feedback mode *(2)*	Yes *(3)*	Yes *(1)*
LVDS compensation	No	Yes

Notes to Table 6–14:

(1) External feedback mode supported for single-ended inputs and outputs only on Left/Right PLLs.

(2) High-bandwidth PLL settings are not supported in external feedback mode. Select a "low" or "medium" PLL bandwidth in the ALTPLL MegaWizard™ Plug-in Manager when using PLLs in external feedback mode.

(3) Differential HSTL and SSTL I/O standards are not supported in Top/Bottom PLLs for external feedback mode.

☞ The input and output delays are fully compensated by a PLL only when they are using the dedicated clock input pins associated with a given PLL as the clock source. Input and output delays are not compensated when cascading two adjacent top or bottom PLLs even if they are using dedicated routing for cascading. For example, when using PLL_T1 in normal mode, the clock delays from the input pin to the PLL clock output-to-destination register are fully compensated provided the clock input pin is one of the following four pins: CLK12, CLK13, CLK14, or CLK15. When an RCLK or GCLK network drives the PLL, the input and output delays may not be fully compensated in the Quartus II software.

Source Synchronous Mode

If the data and clock signals arrive at the same time on the input pins, the same phase relationship is maintained at the clock and data ports of any IOE input register. Figure 6–23 shows an example waveform of the clock and data in this mode. This mode is recommended for source-synchronous data transfers. Data and clock signals at the IOE experience similar buffer delays when you use the same I/O standard.

Figure 6–23. Phase Relationship Between Clock and Data in Source-Synchronous Mode

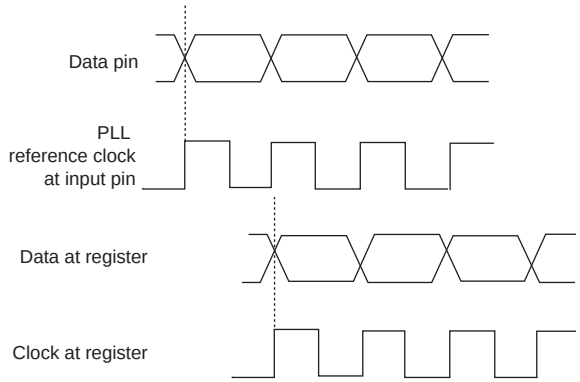
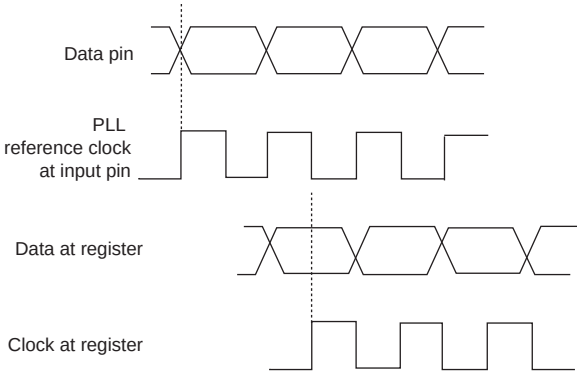


Figure 6–24 shows an example waveform of the clock and data in the LVDS mode.

Figure 6–24. Phase Relationship Between Clock and Data LVDS Modes



The source-synchronous mode compensates for the delay of the clock network used plus any difference in the delay between these two paths:

- Data pin to IOE register input
- Clock input pin to the PLL PFD input

Set the input pin to register delay chain within the IOE to zero in the Quartus II software for all data pins clocked by a source-synchronous mode PLL. Also, all data pins must use the **PLL COMPENSATED** logic option in the Quartus II software.

Source-Synchronous Mode for LVDS Compensation

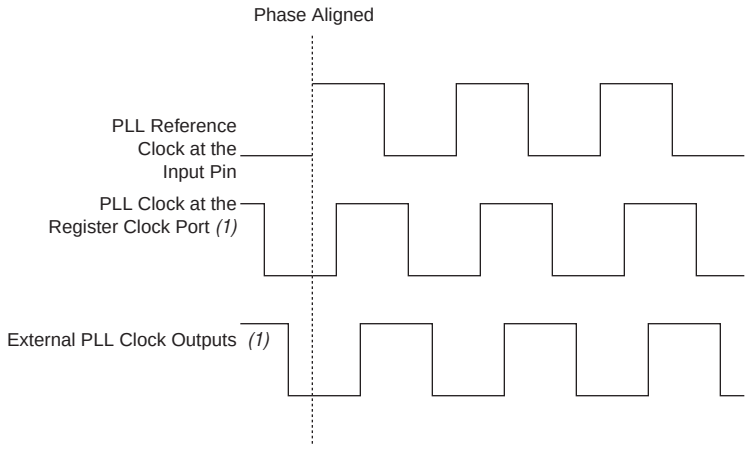
The goal of this mode is to maintain the same data and clock timing relationship seen at the pins at the internal SERDES capture register, except that the clock is inverted (180-degree phase shift). Thus, this mode ideally compensates for the delay of the LVDS clock network plus any difference in delay between these two paths:

- Data pin-to-SERDES capture register
- Clock input pin-to-SERDES capture register. In addition, the output counter needs to provide the 180-degree phase shift.

No-Compensation Mode

In the no-compensation mode, the PLL does not compensate for any clock networks. This mode provides better jitter performance because the clock feedback into the PFD passes through less circuitry. Both the PLL internal- and external-clock outputs are phase-shifted with respect to the PLL clock input. Figure 6–25 shows an example waveform of the PLL clocks' phase relationship in this mode.

Figure 6–25. Phase Relationship Between PLL Clocks in No Compensation Mode



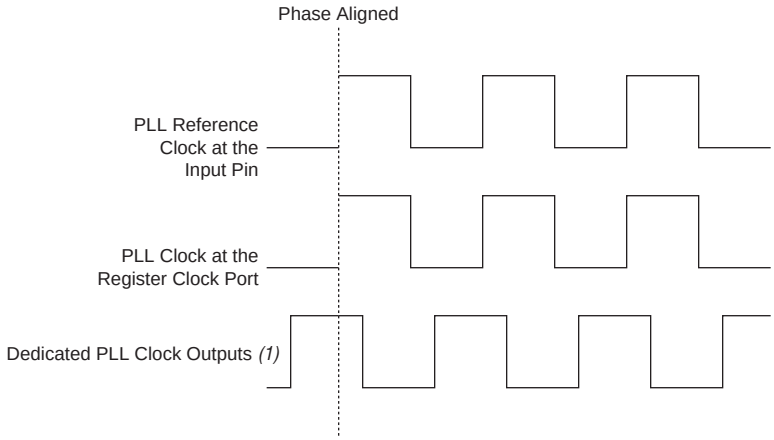
Note to Figure 6–25:

(1) The PLL clock outputs will lag the PLL input clocks, depending on routing delays.

Normal Mode

An internal clock in normal mode is phase-aligned to the input clock pin. The external clock-output pin has a phase delay relative to the clock input pin if connected in this mode. The Quartus II software timing analyzer reports any phase difference between the two. In normal mode, the delay introduced by the GCLK or RCLK network is fully compensated. Figure 6–26 shows an example waveform of the PLL clocks' phase relationship in this mode.

Figure 6–26. Phase Relationship Between PLL Clocks in Normal Mode



Note to Figure 6–26:

(1) The external clock output can lead or lag the PLL internal clock signals.

Zero-Delay Buffer Mode

In zero-delay buffer (ZDB) mode, the external clock output pin is phase-aligned with the clock input pin for zero delay through the device. When using this mode, you must use the same I/O standard on the input clocks and output clocks in order to guarantee clock alignment at the input and output pins. This mode is supported on all Stratix III PLLs.

When using Stratix III PLLs in ZDB mode, along with single-ended I/O standards, to ensure phase alignment between the clock input pin (`CLK`) and the external clock output (`CLKOUT`) pin, you must instantiate a bi-directional I/O pin in the design to serve as the feedback path connecting the `FBOUT` and `FBIN` ports of the PLL. The PLL uses this bi-directional I/O pin to mimic, and hence compensate for, the output delay from the clock output port of the PLL to the external clock output pin. Figure 6–27 shows ZDB mode implementation in Stratix III PLLs. You cannot use differential I/O standards on the PLL clock input or output pins when using ZDB mode.

To avoid reflection, do not place a board trace on the bi-directional I/O pins.

The bi-directional I/O pin that you instantiate in your design should always be assigned a single-ended I/O standard.

Figure 6–27. Zero-Delay Buffer Mode in Stratix III PLLs

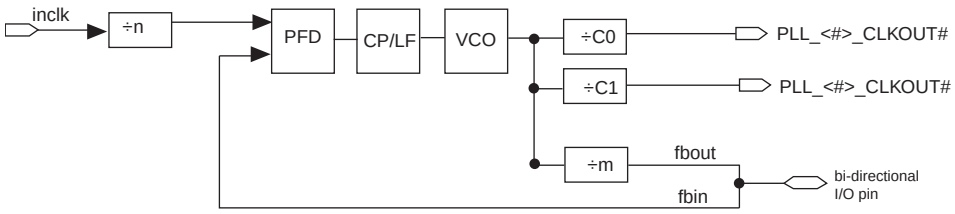
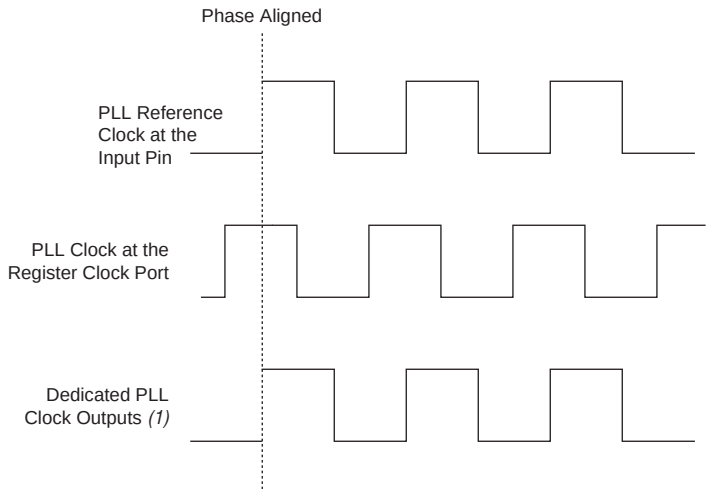


Figure 6–28 shows an example waveform of the PLL clocks' phase relationship in ZDB mode.

Figure 6–28. Phase Relationship Between PLL Clocks in Zero Delay Buffer Mode



Note to Figure 6–28:

(1) The internal PLL clock output can lead or lag the external PLL clock outputs.

External Feedback Mode

In external feedback (EFB) mode, the external feedback input pin (`fbin`) is phase-aligned with the clock input pin, as shown in Figure 6–30. Aligning these clocks allows you to remove clock delay and skew between devices. This mode is supported on all Stratix III PLLs.

In this mode, the output of the `M` counter (`FBOUT`) feeds back to the PLL `fbin` input (using a trace on the board) becoming part of the feedback loop. Also, you can use one of the dual-purpose external clock outputs as the `fbin` input pin in EFB mode.

When using this mode, you must use the same I/O standard on the input clock, feedback input, and output clocks. Left/Right PLLs support EFB mode when using single-ended I/O standards only. Figure 6–29 shows the EFB mode implementation in Stratix III devices.

High-bandwidth PLL settings are not supported in external feedback mode. Select a "low" or "medium" PLL bandwidth in the ALTPLL MegaWizard Plug-In Manager when using PLLs in external feedback mode.

Figure 6–29. External Feedback Mode in Stratix III Devices

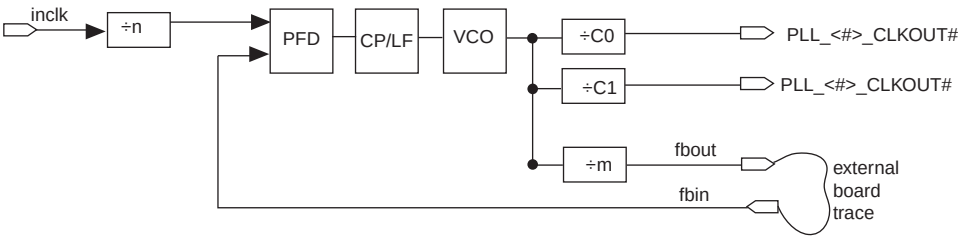
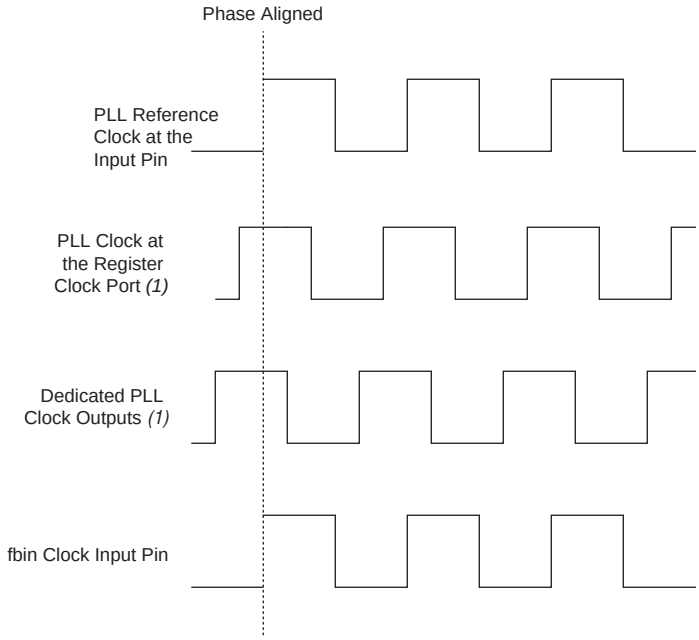


Figure 6–30 shows an example waveform of the phase relationship between PLL clocks in EFB mode.

Figure 6–30. Phase Relationship Between PLL Clocks in External-Feedback Mode



Note to Figure 6–30:

(1) The PLL clock outputs can lead or lag the fbin clock input.

Clock Multiplication and Division

Each Stratix III PLL provides clock synthesis for PLL output ports using $m/(n*$ post-scale counter) scaling factors. The input clock is divided by a pre-scale factor, n, and is then multiplied by the m feedback factor. The control loop drives the VCO to match f_{in} (m/n). Each output port has a unique post-scale counter that divides down the high-frequency VCO. For multiple PLL outputs with different frequencies, the VCO is set to the least common multiple of the output frequencies that meets its frequency specifications. For example, if output frequencies required from one PLL are 33 and 66 MHz, then the Quartus II software sets the VCO to 660 MHz (the least common multiple of 33 and 66 MHz within the VCO range). Then the post-scale counters scale down the VCO frequency for each output port.

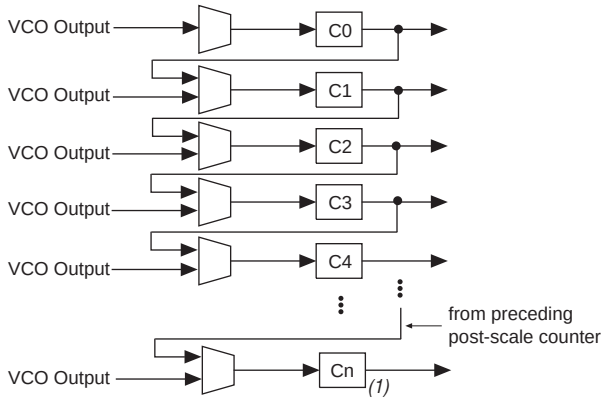
Each PLL has one pre-scale counter, `n`, and one multiply counter, `m`, with a range of 1 to 512 for both `m` and `n`. The `n` counter does not use duty-cycle control because the only purpose of this counter is to calculate frequency division. There are seven generic post-scale counters per Left/Right PLL and ten post-scale counters per Top/Bottom PLL that can feed GCLKs, RCLKs, or external clock outputs. These post-scale counters range from 1 to 512 with a 50% duty cycle setting. The high- and low-count values for each counter range from 1 to 256. The sum of the high- and low-count values chosen for a design selects the divide value for a given counter.

The Quartus II software automatically chooses the appropriate scaling factors according to the input frequency, multiplication, and division values entered into the ALTPLL megafunction.

Post-Scale Counter Cascading

The Stratix III PLLs support post-scale counter cascading to create counters larger than 512. This is automatically implemented in the Quartus II software by feeding the output of one `C` counter into the input of the next `C` counter as shown in Figure 6–31.

Figure 6–31. Counter Cascading



Note to Figure 6–31:

(1) n = 6 or n = 9

When cascading post-scale counters to implement a larger division of the high-frequency VCO clock, the cascaded counters behave as one counter with the product of the individual counter settings. For example, if C0 = 40 and C1 = 20, then the cascaded value is C0*C1 = 800.

Post-scale counter cascading is set in the configuration file. It cannot be done using PLL reconfiguration.

Programmable Duty Cycle

The programmable duty cycle allows PLLs to generate clock outputs with a variable duty cycle. This feature is supported on the PLL post-scale counters. The duty-cycle setting is achieved by a low and high time-count setting for the post-scale counters. The Quartus II software uses the frequency input and the required multiply or divide rate to determine the duty cycle choices. The post-scale counter value determines the precision of the duty cycle. The precision is defined by 50% divided by the post-scale counter value. For example, if the `C0` counter is 10, then steps of 5% are possible for duty-cycle choices between 5% to 90%.

If the PLL is in external feedback mode, you must set the duty cycle for the counter driving the `fbin` pin to 50%. Combining the programmable duty cycle with programmable phase shift allows the generation of precise non-overlapping clocks.

PLL Control Signals

You can use the following three signals to observe and control the PLL operation and resynchronization.

pfdena

Use the `pfdena` signal to maintain the most recent locked frequency so your system has time to store its current settings before shutting down. The `pfdena` signal controls the PFD output with a programmable gate. If you disable the PFD, the VCO is free running and the PLL output drifts. The PLL output jitter may not meet the datasheet specifications. The lock signal cannot be used as an indicator when the PFD is disabled.

areset

The `areset` signal is the reset or resynchronization input for each PLL. The device input pins or internal logic can drive these input signals. When `areset` is driven high, the PLL counters reset, clearing the PLL output and placing the PLL out-of-lock. The VCO is then set back to its nominal setting. When `areset` is driven low again, the PLL will resynchronize to its input as it re-locks.

You should assert the `areset` signal every time the PLL loses lock to guarantee the correct phase relationship between the PLL input clock and output clocks. You can set up the PLL to automatically reset (self reset) upon a loss-of-lock condition using the Quartus II MegaWizard Plug-In Manager. You should include the `areset` signal in designs if the following condition is true:

PLL reconfiguration or clock switchover is enabled in the design.

☞ If the input clock to the PLL is not toggling or is unstable upon power up, assert the `areset` signal after the input clock is stable and within specifications.

locked

The lock signal is an asynchronous output of the PLL. The locked output of the PLL indicates that the PLL has locked onto the reference clock and the PLL clock outputs are operating at the desired phase and frequency set in the Quartus II MegaWizard Plug-In Manager. The lock detection circuit provides a signal to the core logic that gives an indication if the feedback clock has locked onto the reference clock both in phase and frequency.

☞ Altera recommends that you use the `areset` and `locked` signals in your designs to control and observe the status of your PLL.

Clock Switchover

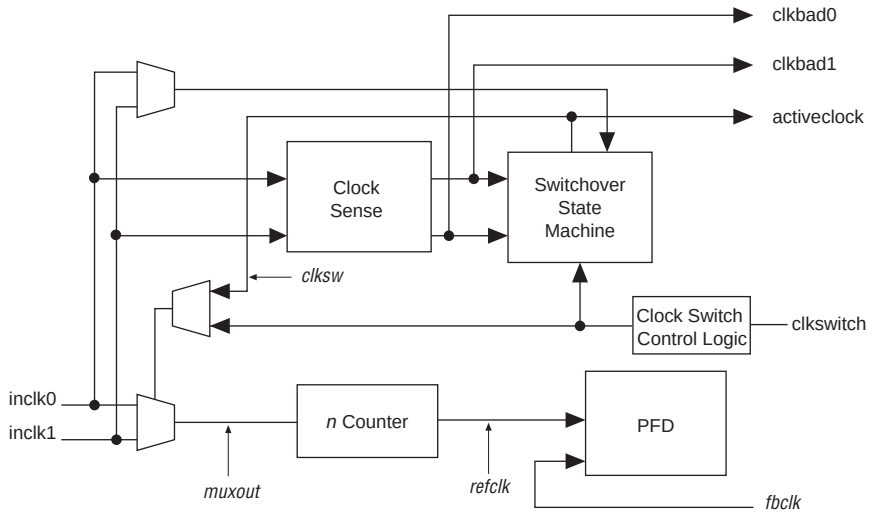
The clock switchover feature allows the PLL to switch between two reference input clocks. Use this feature for clock redundancy or for a dual-clock domain application such as in a system that turns on the redundant clock if the previous clock stops running. The design can perform clock switchover automatically, when the clock is no longer toggling or based on a user control signal, `clkswitch`.

The following clock switchover modes are supported in Stratix III PLLs:

- Automatic switchover—The clock sense circuit monitors the current reference clock and if it stops toggling, automatically switches to the other clock `inclk0` or `inclk1`.
- Manual clock switchover—Clock switchover is controlled via the `clkswitch` signal in this mode. When the `clkswitch` signal goes from logic low to logic high, and stays high for at least three clock cycles, the reference clock to the PLL is switched from `inclk0` to `inclk1`, or vice-versa.
- Automatic switchover with manual override—This mode combines Modes 1 and 2. When the `clkswitch` signal goes high, it overrides automatic clock switchover mode.

Stratix III device PLLs support a fully configurable clock switchover capability. Figure 6–32 shows the block diagram of the switchover circuit built into the PLL. When the current reference clock is not present, the clock sense block automatically switches to the backup clock for PLL reference. The clock switchover circuit also sends out three status signals—`clkbad[0]`, `clkbad[1]`, and `activeclock`—from the PLL to implement a custom switchover circuit in the logic array. You can select a clock source as the backup clock by connecting it to the `inclk1` port of the PLL in your design.

Figure 6–32. Automatic Clock Switchover Circuit Block Diagram



Automatic Clock Switchover

Use the switchover circuitry to automatically switch between `inclk0` and `inclk1` when the current reference clock to the PLL stops toggling. For example, in applications that require a redundant clock with the same frequency as the reference clock, the switchover state machine generates a signal (`clksw`) that controls the multiplexer select input as shown in Figure 6–32. In this case, `inclk1` becomes the reference clock for the PLL. When using the automatic switchover mode, you can switch back and forth between `inclk0` and `inclk1` clocks any number of times, when one of the two clocks fails and the other clock is available.

When using the automatic clock switchover mode, the following requirements must be satisfied:

- Both clock inputs must be running.
- The period of the two clock inputs can differ by no more than 100% (2×).

If the current clock input stops toggling while the other clock is also not toggling, switchover will not be initiated and the `clkbad[0..1]` signals will not be valid. Also, if both clock inputs are not the same frequency, but their period difference is within 100%, the clock sense block will detect when a clock stops toggling, but the PLL may lose lock after the switchover is completed and need time to re-lock.

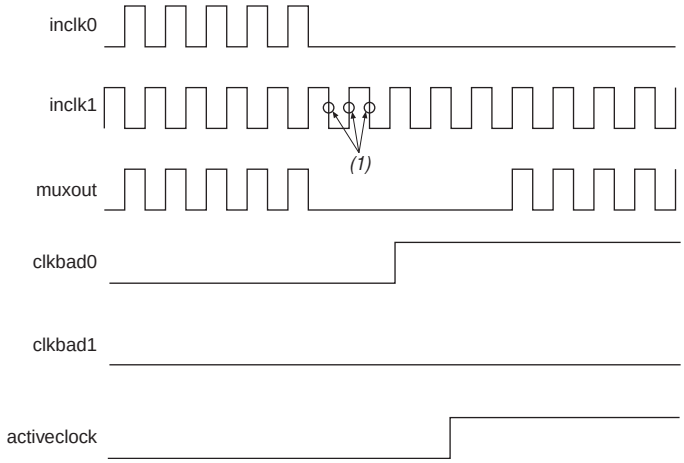
Altera recommends resetting the PLL using the `areset` signal to maintain the phase relationships between the PLL input and output clocks when using clock switchover.

When using automatic switchover mode, the `clkbad[0]` and `clkbad[1]` signals indicate the status of the two clock inputs. When they are asserted, the clock sense block has detected that the corresponding clock input has stopped toggling. These two signals are not valid if the frequency difference between `inclk0` and `inclk1` is greater than 20%.

The `activeclock` signal indicates which of the two clock inputs (`inclk0` or `inclk1`) is being selected as the reference clock to the PLL. When the frequency difference between the two clock inputs is more than 20%, the `activeclock` signal is the only valid status signal.

Figure 6–33 shows an example waveform of the switchover feature when using the automatic switchover mode. In this example, the `inclk0` signal remains low. After the `inclk0` signal remains low for approximately two clock cycles, the clock sense circuitry drives the `clkbad[0]` signal high. Also, because the reference clock signal is not toggling, the switchover state machine controls the multiplexer through the `clksw` signal to switch to the backup clock, `inclk1`.

Figure 6–33. Automatic Switchover Upon Loss of Clock Detection



Note to Figure 6–33:

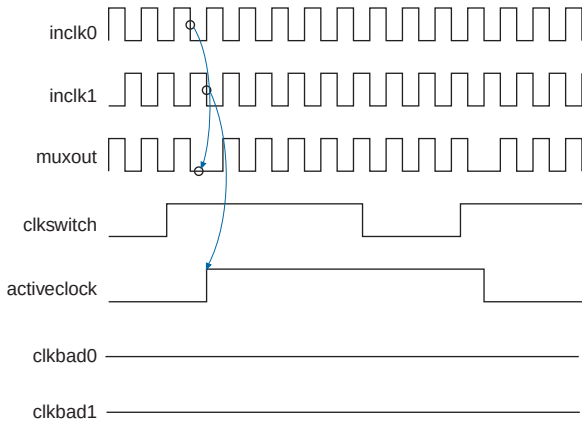
(1) Switchover is enabled on the falling edge of `inclk0` or `inclk1`, depending on which clock is available. In this figure, switchover is enabled on the falling edge of `inclk1`.

Manual Override

In the automatic switchover with manual override mode, you can use the `clkswitch` input for user- or system-controlled switch conditions. You can use this mode for same-frequency switchover or to switch between inputs of different frequencies. For example, if `inclk0` is 66 MHz and `inclk1` is 200 MHz, you must control the switchover using `clkswitch` because the automatic clock-sense circuitry cannot monitor clock input (`inclk0`, `inclk1`) frequencies with a frequency difference of more than 100% (2×). This feature is useful when the clock sources originate from multiple cards on the backplane, requiring a system-controlled switchover between the frequencies of operation. You should choose the backup clock frequency and set the `m`, `n`, `c`, and `k` counters accordingly so the VCO operates within the recommended operating frequency range of 600 to 1,300 MHz. The ALTPLL MegaWizard Plug-in Manager notifies users if a given combination of `inclk0` and `inclk1` frequencies cannot meet this requirement. In the Quartus II software, the VCO value reported is divided by the post scale counter (`K`).

Figure 6–34 shows an example of a waveform illustrating the switchover feature when controlled by `clkswitch`. In this case, both clock sources are functional and `inclk0` is selected as the reference clock. `clkswitch` goes high, which starts the switchover sequence. On the falling edge of `inclk0`, the counter's reference clock, `muxout`, is gated off to prevent any clock glitching. On the falling edge of `inclk1`, the reference clock multiplexer switches from `inclk0` to `inclk1` as the PLL reference, and the `activeclock` signal changes to indicate which clock is currently feeding the PLL.

Figure 6–34. Clock Switchover Using the clkswitch (Manual) Control *(Note 1)*



Note to Figure 6–34:

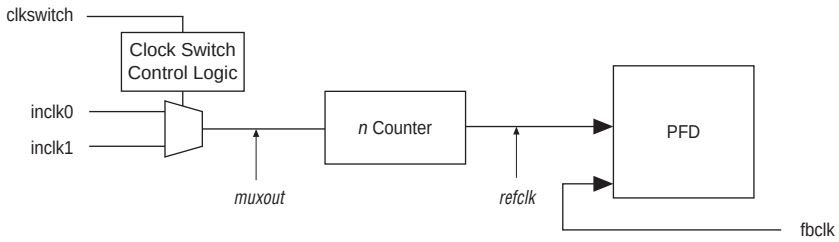
(1) Both `inclk0` and `inclk1` must be running when the `clkswitch` signal goes high to initiate a manual clock switchover event.

In this mode, the `activeclock` signal mirrors the `clkswitch` signal. As both clocks are still functional during the manual switch, neither `clkbad` signal goes high. Since the switchover circuit is positive-edge sensitive, the falling edge of the `clkswitch` signal does not cause the circuit to switch back from `inclk1` to `inclk0`. When the `clkswitch` signal goes high again, the process repeats. `clkswitch` and automatic switch only work if the clock being switched to is available. If the clock is not available, the state machine waits until the clock is available.

Manual Clock Switchover

In manual clock switchover mode, the `clkswitch` signal controls whether `inclk0` or `inclk1` is selected as the input clock to the PLL. By default, `inclk0` is selected. A low-to-high transition on `clkswitch` and `clkswitch` being held high for at least three `inclk` cycles initiates a clock switchover event. You must bring `clkswitch` back low again in order to perform another switchover event in the future. If you do not require another switchover event in the future, you can leave `clkswitch` in a logic high state after the initial switch. Pulsing `clkswitch` high for at least three `inclk` cycles performs another switchover event. If `inclk0` and `inclk1` are different frequencies and are always running, the `clkswitch` minimum high time must be greater than or equal to three of the slower frequency `inclk0/inclk1` cycles. Take note that manual switchover is only applicable when both clocks are switching. Figure 6–35 shows the block diagram of the manual switchover circuit.

Figure 6–35. Manual Clock Switchover Circuitry in Stratix III PLLs



For more information about PLL software support in the Quartus II software, refer to the *ALTPLL Megafunction User Guide*.

Guidelines

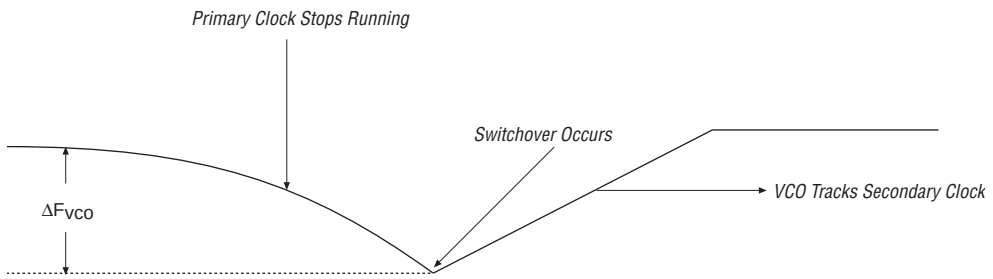
Use the following guidelines when implementing clock switchover in Stratix III PLLs.

- Automatic clock switchover requires that the `inclk0` and `inclk1` frequencies be within 100% (2×) of each other. Failing to meet this requirement causes the `clkbad[0]` and `clkbad[1]` signals to not function properly.
- When using manual clock switchover, the difference between `inclk0` and `inclk1` can be more than 100% (2×). However, differences in frequency and/or phase of the two clock sources will likely cause the PLL to lose lock. Resetting the PLL ensures that the correct phase relationships are maintained between input and output clocks.

  Both `inclk0` and `inclk1` must be running when the `clkswitch` signal goes high to instantiate the manual clock switchover event. Failing to meet this requirement causes the clock switchover to not function properly.

- Applications that require a clock switchover feature and a small frequency drift should use a low-bandwidth PLL. The low-bandwidth PLL reacts more slowly than a high-bandwidth PLL to reference input clock changes. When the switchover happens, a low-bandwidth PLL propagates the stopping of the clock to the output more slowly than a high-bandwidth PLL. However, be aware that the low-bandwidth PLL also increases lock time.
- After a switchover occurs, there may be a finite resynchronization period for the PLL to lock onto a new clock. The exact amount of time it takes for the PLL to re-lock depends on the PLL configuration.
- The phase relationship between the input clock to the PLL and the output clock from the PLL is important in your design. Assert `areset` for at least 10 ns after performing a clock switchover. Wait for the locked signal to go high and be stable before re-enabling the output clocks from the PLL.
- Figure 6–36 shows how the VCO frequency gradually decreases when the current clock is lost and then increases as the VCO locks on to the backup clock.

Figure 6–36. VCO Switchover Operating Frequency



- Disable the system during clock switchover if it is not tolerant of frequency variations during the PLL resynchronization period. You can use the `clkbad[0]` and `clkbad[1]` status signals to turn off the PFD (`PFDENA = 0`) so the VCO maintains its most recent frequency. You can also use the state machine to switch over to the secondary clock. When the PFD is re-enabled, output clock-enable signals (`clkena`) can disable clock outputs during the switchover and resynchronization period. Once the lock indication is stable, the system can re-enable the output clocks.

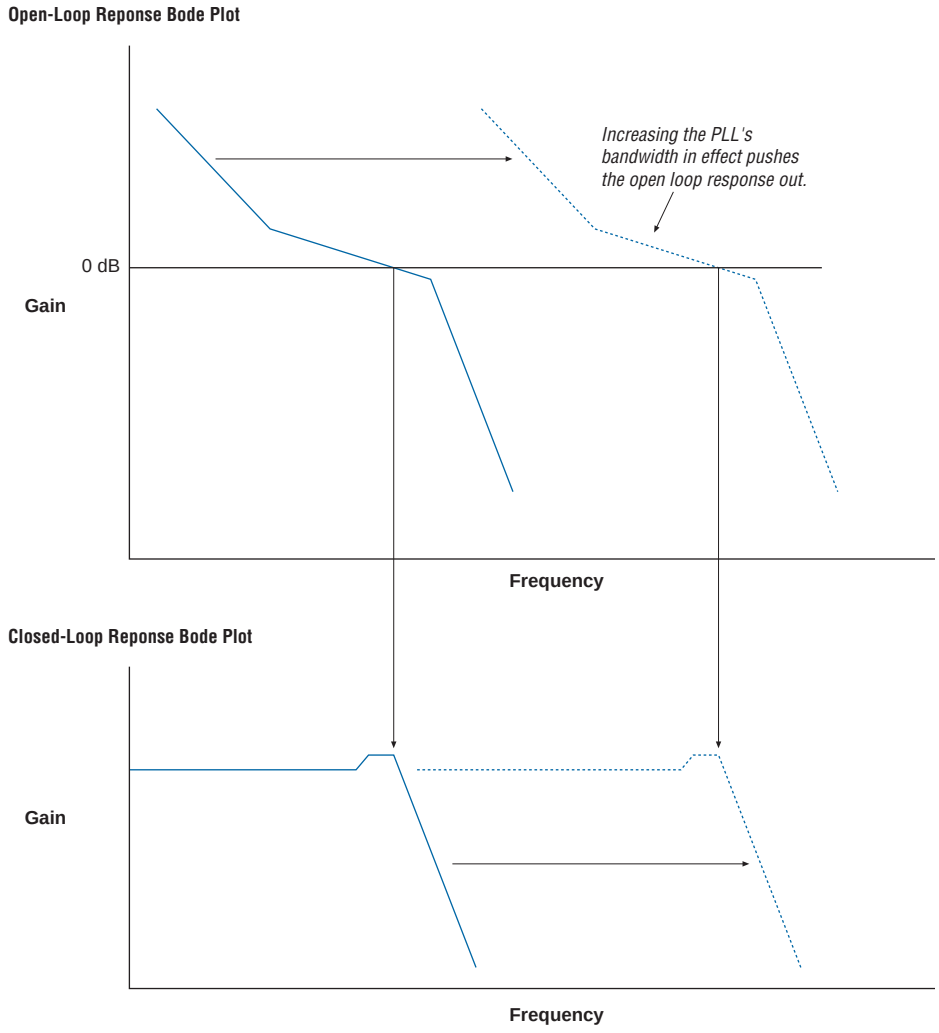
Programmable Bandwidth

Stratix III PLLs provide advanced control of the PLL bandwidth using the PLL loop's programmable characteristics, including loop filter and charge pump.

Background

PLL bandwidth is the measure of the PLL's ability to track the input clock and its associated jitter. The closed-loop gain 3-dB frequency in the PLL determines the PLL bandwidth. The bandwidth is approximately the unity gain point for open loop PLL response. As Figure 6–37 shows, these points correspond to approximately the same frequency. Stratix III PLLs provide three bandwidth settings—low, medium (default), and high.

Figure 6–37. Open- and Closed-Loop Response Bode Plots



A high-bandwidth PLL provides a fast lock time and tracks jitter on the reference clock source, passing it through to the PLL output. A low-bandwidth PLL filters out reference clock jitter but increases lock time. Stratix III PLLs allow you to control the bandwidth over a finite range to customize the PLL characteristics for a particular application. The programmable bandwidth feature in Stratix III PLLs benefits applications requiring clock switchover.

A high-bandwidth PLL can benefit a system that needs to accept a spread-spectrum clock signal. Stratix III PLLs can track a spread-spectrum clock by using a high-bandwidth setting. Using a low-bandwidth in this case could cause the PLL to filter out the jitter on the input clock.

A low-bandwidth PLL can benefit a system using clock switchover. When the clock switchover happens, the PLL input temporarily stops. A low-bandwidth PLL reacts more slowly to changes on its input clock and takes longer to drift to a lower frequency (caused by the input stopping) than a high-bandwidth PLL.

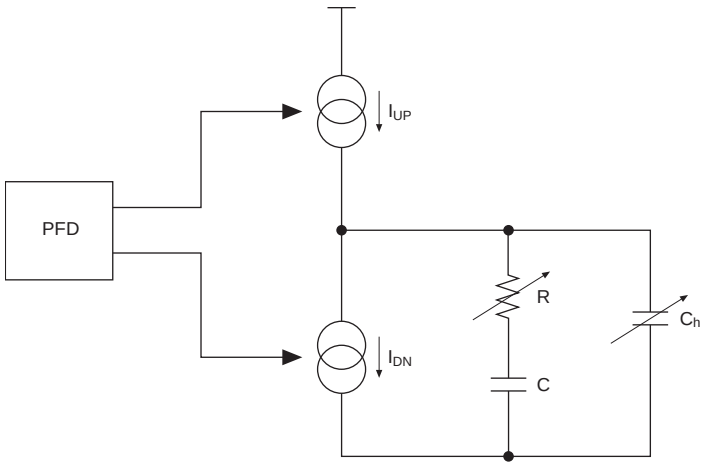
Implementation

Traditionally, external components such as the VCO or loop filter control a PLL's bandwidth. Most loop filters consist of passive components such as resistors and capacitors that take up unnecessary board space and increase cost. With Stratix III PLLs, all the components are contained within the device to increase performance and decrease cost.

When you specify the bandwidth setting (low, medium, or high) in the ALTPLL MegaWizard Plug-in Manager, the Quartus II software automatically sets the corresponding charge pump and loop filter (I_{cp}, R, C) values to achieve the desired bandwidth range.

Figure 6–38 shows the loop filter and the components that you can set using the Quartus II software. The components are the loop filter resistor, R, the high frequency capacitor, C_H, and the charge pump current, I_{UP} or I_{DN}.

Figure 6–38. Loop Filter Programmable Components



Phase-Shift Implementation

Phase shift is used to implement a robust solution for clock delays in Stratix III devices. Phase shift is implemented by using a combination of the VCO phase output and the counter starting time. The VCO phase output and counter starting time is the most accurate method of inserting delays, since it is based purely on counter settings, which are independent of process, voltage, and temperature.

You can phase-shift the output clocks from the Stratix III PLLs in either of these two resolutions:

- Fine resolution using VCO phase taps
- Coarse resolution using counter starting time

Fine-resolution phase shifts are implemented by allowing any of the output counters (`C[n..0]`) or the `m` counter to use any of the eight phases of the VCO as the reference clock. This allows you to adjust the delay time with a fine resolution. The minimum delay time that you can insert using this method is defined by Equation 6–1.

Equation 6–1.

$$\Phi_{fine} = \frac{1}{8}T_{VCO} = \frac{1}{8f_{VCO}} = \frac{N}{8Mf_{REF}}$$

where f_{REF} is the input reference clock frequency.

For example, if f_{REF} is 100 MHz, n is 1, and m is 8, then f_{VCO} is 800 MHz and Φ_{fine} equals 156.25 ps. This phase shift is defined by the PLL operating frequency, which is governed by the reference clock frequency and the counter settings.

Coarse-resolution phase shifts are implemented by delaying the start of the counters for a predetermined number of counter clocks. You can express coarse phase shift as shown in Equation 6–2.

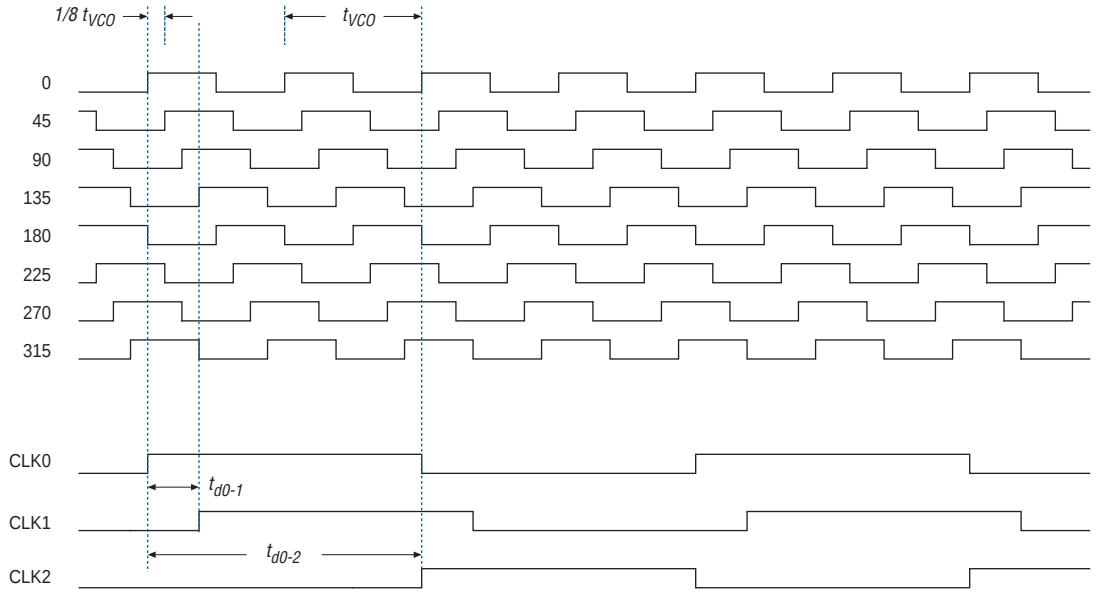
Equation 6–2.

$$\Phi_{coarse} = \frac{C-1}{f_{VCO}} = \frac{(C-1)N}{Mf_{REF}}$$

where C is the count value set for the counter delay time (this is the initial setting in the PLL usage section of the compilation report in the Quartus II software). If the initial value is 1, C – 1 = 0° phase shift.

Figure 6–39 shows an example of phase-shift insertion with the fine resolution using the VCO phase taps method. The eight phases from the VCO are shown and labeled for reference. For this example, `CLK0` is based off the `0phase` from the VCO and has the C value for the counter set to one. The `CLK0` signal is divided by four, two VCO clocks for high time and two VCO clocks for low time. `CLK1` is based off the 135° phase tap from the VCO and also has the C value for the counter set to one. The `CLK1` signal is also divided by 4. In this case, the two clocks are offset by $3\Phi_{fine}$. `CLK2` is based off the `0phase` from the VCO but has the C value for the counter set to three. This arrangement creates a delay of $2\Phi_{coarse}$ (two complete VCO periods).

Figure 6–39. Delay Insertion Using VCO Phase Output and Counter Delay Time



You can use the coarse- and fine-phase shifts to implement clock delays in Stratix III devices.

Stratix III devices support dynamic phase-shifting of VCO phase taps only. The phase shift is reconfigurable any number of times, and each phase shift takes about one `SCANCLK` cycle, allowing you to implement large phase shifts quickly.

PLL Reconfiguration

PLLs use several divide counters and different VCO phase taps to perform frequency synthesis and phase shifts. In Stratix III PLLs, you can reconfigure both the counter settings and phase-shift the PLL output clock in real time. You can also change the charge pump and loop-filter components, which dynamically affects the PLL bandwidth. You can use these PLL components to update the output-clock frequency and the PLL bandwidth and to phase-shift in real time, without reconfiguring the entire Stratix III device.

The ability to reconfigure the PLL in real time is useful in applications that operate at multiple frequencies. It is also useful in prototyping environments, allowing you to sweep PLL output frequencies and adjust the output-clock phase dynamically. For example, a system generating test patterns is required to generate and transmit patterns at 75 or 150 MHz, depending on the requirements of the device under test. Reconfiguring the PLL components in real time allows you to switch between two such output frequencies within a few microseconds. You can also use this feature to adjust clock-to-out (**t_{co}**) delays in real time by changing the PLL output clock phase shift. This approach eliminates the need to regenerate a configuration file with the new PLL settings.

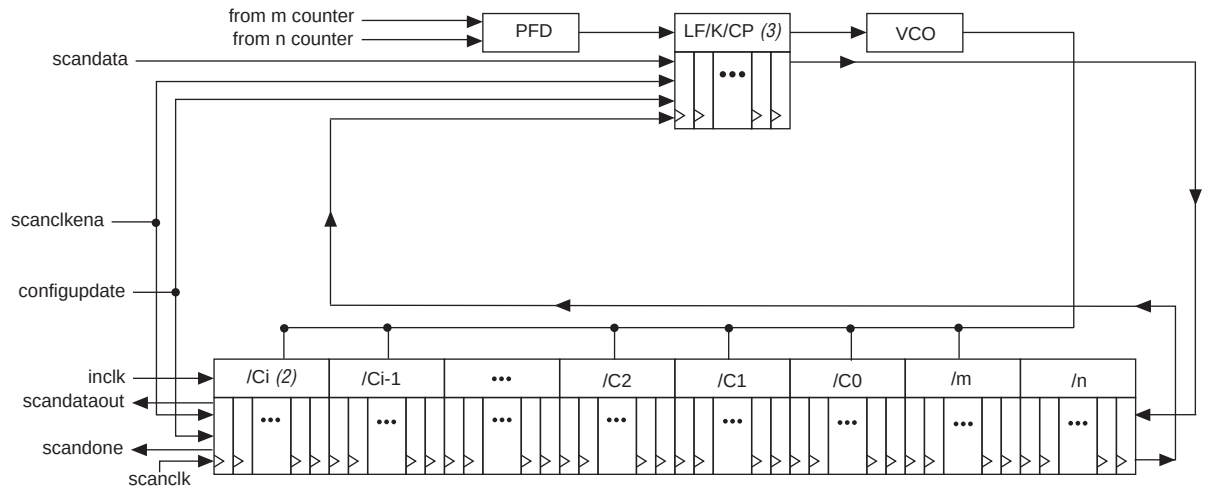
PLL Reconfiguration Hardware Implementation

The following PLL components are reconfigurable in real time:

- Pre-scale counter (`n`)
- Feedback counter (`m`)
- Post-scale output counters (`C0` – `C9`)
- Post VCO Divider (`K`)
- Dynamically adjust the charge-pump current (I_{cp}) and loop-filter components (R, C) to facilitate reconfiguration of the PLL bandwidth

Figure 6–40 shows how PLL counter settings can be dynamically adjusted by shifting their new settings into a serial shift-register chain or scan chain. Serial data is input to the scan chain via the `scandataport` and shift registers are clocked by `scanclk`. The maximum `scanclk` frequency is 100 MHz. Serial data is shifted through the scan chain as long as the `scanclkena` signal stays asserted. After the last bit of data is clocked, asserting the `configupdate` signal for at least one `scanclk` clock cycle causes the PLL configuration bits to be synchronously updated with the data in the scan registers.

Figure 6–40. PLL Reconfiguration Scan Chain



Notes to Figure 6–40:

(1) The Stratix III Left/Right PLLs support `C0 - C6` counters.

(2) i = 6 or i = 9.

(3) This figure shows the corresponding scan register for the K counter in between the scan registers for the charge pump and loop filter. The K counter is physically located after the VCO.

The counter settings are updated synchronously to the clock frequency of the individual counters. Therefore, all counters are not updated simultaneously.

Table 6–15 lists how these signals can be driven by the programmable logic device (PLD) logic array or I/O pins.

Table 6–15. Real-Time PLL Reconfiguration Ports

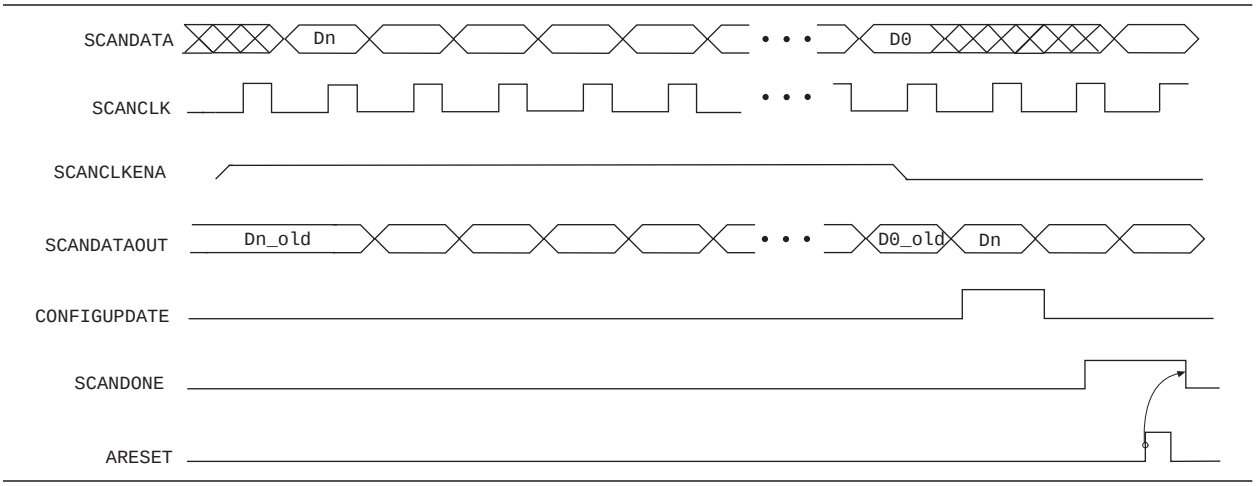
PLL Port Name	Description	Source	Destination
`scandata`	Serial input data stream to scan chain.	Logic array or I/O pin	PLL reconfiguration circuit
`scanclk`	Serial clock input signal. This clock can be free running.	GCLK/RCLK or I/O pins	PLL reconfiguration circuit
`scanclkena`	Enables `scanclk` and allows the `scandata` to be loaded in the scan chain. Active high	Logic array or I/O pin	PLL reconfiguration circuit
`configupdate`	Writes the data in the scan chain to the PLL. Active high	Logic array or I/O pin	PLL reconfiguration circuit
`scandone`	Indicates when the PLL has finished reprogramming. A rising edge indicates the PLL has begun reprogramming. A falling edge indicated the PLL has finished reprogramming.	PLL reconfiguration circuit	Logic array or I/O pins
`scandataout`	Used to output the contents of the scan chain.	PLL reconfiguration circuit	Logic array or I/O pins

Use the following procedure to reconfigure the PLL counters:

1. The `scanclkena` signal is asserted at least one `scanclk` cycle prior to shifting in the first bit of `scandata` (`Dn`).
2. Serial data (`scandata`) is shifted into the scan chain on the second rising edge of `scanclk`.
3. After all 234 bits (Top/Bottom PLLs) or 180 bits (Left/Right PLLs) have been scanned into the scan chain, the `scanclkena` signal is de-asserted to prevent inadvertent shifting of bits in the scan chain.
4. The `configupdate` signal is asserted for one `scanclk` cycle to update the PLL counters with the contents of the scan chain.
5. The `scandone` signal goes high indicating the PLL is being reconfigured. A falling edge indicates the PLL counters have been updated with new settings.
6. Reset the PLL using the `areset` signal if you make any changes to the `M`, `N`, or post-scale C counters or the I_{cp}, R, or C settings.
7. Steps 1 through 5 can be repeated to reconfigure the PLL any number of times.

Figure 6–41 shows a functional simulation of the PLL reconfiguration feature.

Figure 6–41. PLL Reconfiguration Waveform



When you reconfigure the counter clock frequency, you cannot reconfigure the corresponding counter phase shift settings using the same interface. Instead, reconfigure the phase shifts in real time using the dynamic phase shift reconfiguration interface. If you reconfigure the counter frequency, but wish to keep the same non-zero phase shift setting (for example, 90 degrees) on the clock output, you must reconfigure the phase shift immediately after reconfiguring the counter clock frequency.

Post-Scale Counters (C0 to C9)

The multiply or divide values and duty cycle of post-scale counters can be reconfigured in real time. Each counter has an 8-bit high-time setting and an 8-bit low-time setting. The duty cycle is the ratio of output high- or low-time to the total cycle time, which is the sum of the two. Additionally, these counters have two control bits, `rbypass`, for bypassing the counter, and `rselodd`, to select the output clock duty cycle.

When the `rbypass` bit is set to 1, it bypasses the counter, resulting in a divide by 1. When this bit is set to 0, the high- and low-time counters are added to compute the effective division of the VCO output frequency. For example, if the post-scale divide factor is 10, the high- and low-count values could be set to 5 and 5, respectively, to achieve a 50-50% duty cycle. The PLL implements this duty cycle by transitioning the output clock from high to low on the rising edge of the VCO output clock. However, a 4 and 6 setting for the high- and low-count values, respectively, would produce an output clock with 40-60% duty cycle.

The `rselodd` bit indicates an odd divide factor for the VCO output frequency along with a 50% duty cycle. For example, if the post-scale divide factor is 3, the high- and low-time count values could be set to 2 and 1, respectively, to achieve this division. This implies a 67%-33% duty cycle. If you need a 50%-50% duty cycle, you can set the `rselodd` control bit to 1 to achieve this duty cycle despite an odd division factor. The PLL implements this duty cycle by transitioning the output clock from high to low on a falling edge of the VCO output clock. When you set `rselodd` = 1, you subtract 0.5 cycles from the high time and you add 0.5 cycles to the low time. For example:

- High-time count = 2 cycles
- Low-time count = 1 cycle
- `rselodd` = 1 effectively equals:
  - High-time count = 1.5 cycles
  - Low-time count = 1.5 cycles
  - Duty cycle = (1.5/3) % high-time count and (1.5/3) % low-time count

Scan Chain Description

The length of the scan chain varies for different Stratix III PLLs. The Top/Bottom PLLs have 10 post-scale counters and a 234-bit scan chain, while the Left/Right PLLs have 7 post-scale counters and a 180-bit scan chain. Table 6–16 lists the number of bits for each component of a Stratix III PLL.

Table 6–16. Top/Bottom PLL Reprogramming Bits (Part 1 of 2)

Block Name	Number of Bits		Total
	Counter	Other *(1)*	
C9 *(2)*	16	2	18
C8	16	2	18
C7	16	2	18
C6 *(3)*	16	2	18
C5	16	2	18
C4	16	2	18
C3	16	2	18
C2	16	2	18
C1	16	2	18
C0	16	2	18
N	16	2	18
M	16	2	18
Charge Pump Current	0	3	3
VCO Post-Scale divider (K)	1	0	1
Loop Filter Capacitor *(4)*	0	2	2
Loop Filter Resistor	0	5	5
Unused CP/LF	0	7	7

Table 6–16. Top/Bottom PLL Reprogramming Bits (Part 2 of 2)

Block Name	Number of Bits		Total
	Counter	Other *(1)*	
Total number of bits	—	—	234

Notes to Table 6–16:

(1) Includes two control bits, `rbypass`, for bypassing the counter, and `rselodd`, to select the output clock duty cycle.

(2) LSB bit for `C9` low-count value is the first bit shifted into the scan chain for Top/Bottom PLLs.

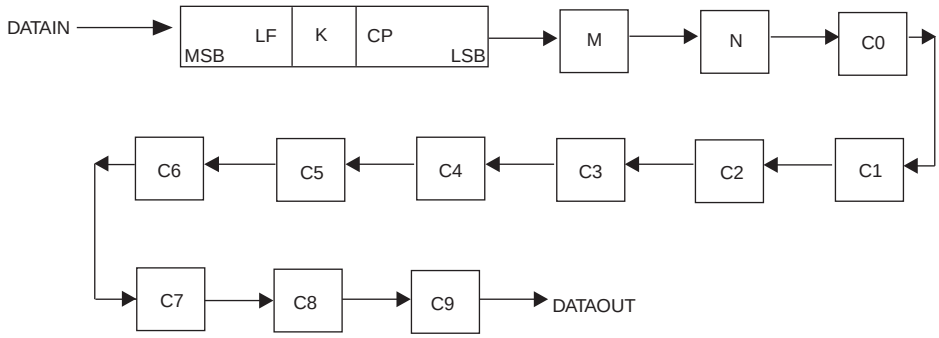
(3) LSB bit for `C6` low-count value is the first bit shifted into the scan chain for Left/Right PLLs.

(4) MSB bit for loop filter is the last bit shifted into the scan chain.

Table 6–16 lists the scan chain order of PLL components for Top/Bottom PLLs which have 10 post-scale counters. The order of bits is the same for the Left/Right PLLs, but the reconfiguration bits start with the `C6` post-scale counter.

Figure 6–42 shows the scan-chain order of PLL components for the Top/Bottom PLLs.

Figure 6–42. Scan-Chain Order of PLL Components for Top/Bottom PLLs *(Note 1)*

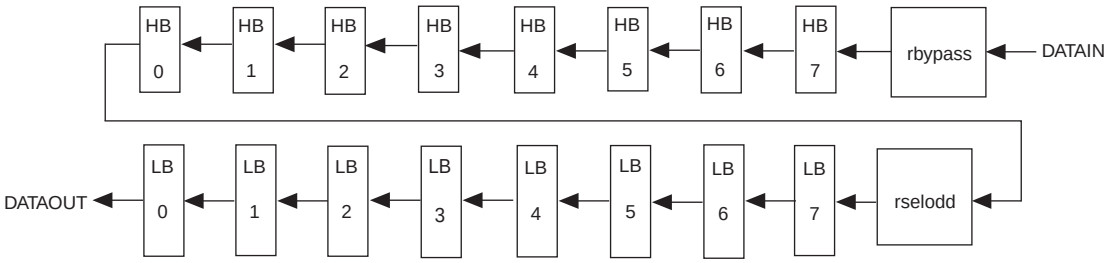


Note to Figure 6–42:

(1) Left/Right PLLs have the same scan-chain order. The post-scale counters end at `C6`.

Figure 6–43 shows the scan-chain bit-order sequence for post-scale counters in all Stratix III PLLs.

Figure 6–43. Scan-Chain Bit-Order Sequence for Post-Scale Counters in Stratix III PLLs



Charge Pump and Loop Filter

You can reconfigure the charge-pump and loop-filter settings to update the PLL bandwidth in real time. Table 6–17, Table 6–18, and Table 6–19 list the possible settings for charge pump current (I_{cp}), loop-filter resistor (R), and capacitor (C) values for Stratix III PLLs.

Table 6–17. charge_pump_current Bit Settings

CP[2]	CP[1]	CP[0]	Decimal Value for Setting
0	0	0	0
0	0	1	1
0	1	1	3
1	1	1	7

Table 6–18. loop_filter_r Bit Settings

LFR[4]	LFR[3]	LFR[2]	LFR[1]	LFR[0]	Decimal Value for Setting
0	0	0	0	0	0
0	0	0	1	1	3
0	0	1	0	0	4
0	1	0	0	0	8
1	0	0	0	0	16
1	0	0	1	1	19
1	0	1	0	0	20
1	1	0	0	0	24
1	1	0	1	1	27
1	1	1	0	0	28
1	1	1	1	0	30

Table 6–19. loop_filter_c Bit Settings

LFC[1]	LFC[0]	Decimal Value for Setting
0	0	0
0	1	1
1	1	3

Bypassing PLL

Bypassing a PLL counter results in a multiply (m counter) or a divide (n and C0 to C9 counters) factor of one.

Table 6–20 lists the settings for bypassing the counters in Stratix III PLLs.

Table 6–20. PLL Counter Settings

PLL Scan Chain Bits [0..10] Settings											
LSB *(2)*	**[1]**	**[2]**	**[3]**	**[4]**	**[5]**	**[6]**	**[7]**	**[8]**	**[9]**	**MSB** *(1)*	**Description**
0	X	X	X	X	X	X	X	X	X	1 *(3)*	PLL counter bypassed
X	X	X	X	X	X	X	X	X	X	0 *(3)*	PLL counter not bypassed because bit 10 (MSB) is set to 0

Notes to Table 6–20:

(1) Most significant bit (MSB).
(2) Least significant bit (LSB).
(3) Counter-bypass bit.

To bypass any of the PLL counters, set the bypass bit to 1. The values on the other bits are ignored. To bypass the VCO post-scale counter (K), set the corresponding bit to 1.

Dynamic Phase-Shifting

The dynamic phase-shifting feature allows the output phases of individual PLL outputs to be dynamically adjusted relative to each other and to the reference clock without the need to send serial data through the scan chain of the corresponding PLL. This feature simplifies the interface and allows you to quickly adjust clock-to-out (t_{co}) delays by changing the output clock phase-shift in real time. This adjustment is achieved by incrementing or decrementing the VCO phase-tap selection to a given C counter or to the M counter. The phase is shifted by 1/8 of the VCO frequency at a time. The output clocks are active during this phase-reconfiguration process.

Table 6–21 lists the control signals that are used for dynamic phase-shifting.

Table 6–21. Dynamic Phase-Shifting Control Signals (Part 1 of 2)

Signal Name	Description	Source	Destination
PHASECOUNTERSELECT [3:0]	Counter select. Four bits decoded to select either the M or one of the C counters for phase adjustment. One address maps to select all C counters. This signal is registered in the PLL on the rising edge of SCANCLK.	Logic array or I/O pins	PLL reconfiguration circuit
PHASEUPDOWN	Selects dynamic phase shift direction; 1= UP; 0= DOWN. Signal is registered in the PLL on the rising edge of SCANCLK.	Logic array or I/O pin	PLL reconfiguration circuit
PHASESTEP	Logic high enables dynamic phase shifting.	Logic array or I/O pin	PLL reconfiguration circuit

Table 6–21. Dynamic Phase-Shifting Control Signals (Part 2 of 2)

Signal Name	Description	Source	Destination
SCANCLK	Free running clock from core used in combination with PHASESTEP to enable/disable dynamic phase shifting. Shared with SCANCLK for dynamic reconfiguration.	GCLK/RCLK or I/O pin	PLL reconfiguration circuit
PHASEDONE	When asserted, it indicates to core-logic that the phase adjustment is complete and PLL is ready to act on a possible second adjustment pulse. Asserts based on internal PLL timing. De-asserts on rising edge of SCANCLK.	PLL reconfiguration circuit	Logic array or I/O pins

Table 6–22 lists the PLL counter selection based on the corresponding PHASECOUNTERSELECT setting.

Table 6–22. Phase Counter Select Mapping

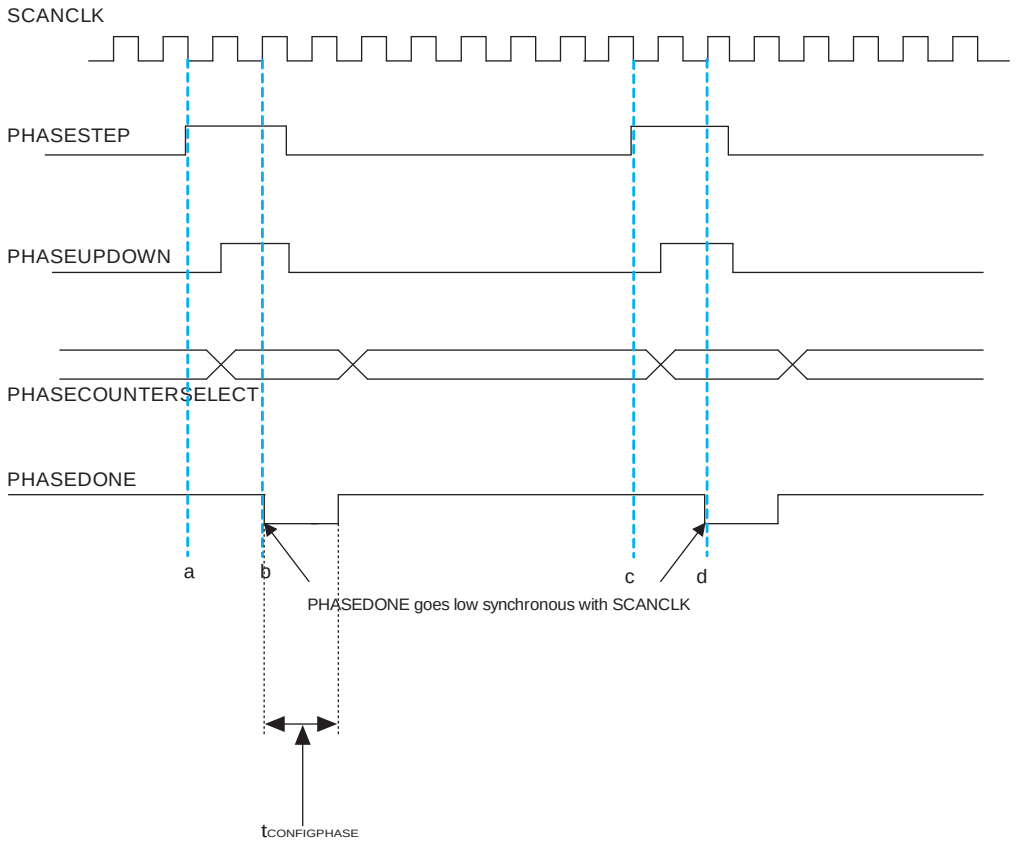
PHASECOUNTERSELECT[3]	[2]	[1]	[0]	Selects
0	0	0	0	All Output Counters
0	0	0	1	M Counter
0	0	1	0	C0 Counter
0	0	1	1	C1 Counter
0	1	0	0	C2 Counter
0	1	0	1	C3 Counter
0	1	1	0	C4 Counter
0	1	1	1	C5 Counter
1	0	0	0	C6 Counter
1	0	0	1	C7 Counter
1	0	1	0	C8 Counter
1	0	1	1	C9 Counter

The procedure to perform one dynamic phase-shift step is as follows:

1. Set phaseupdown and phasecounterselect as required.
2. Assert phasestep for at least two scanclk cycles. Each phasestep pulse enables one phase shift.
3. De-assert phasestep.
4. Wait for phasedone to go high.
5. Repeat steps 1 through 4 as many times as required to perform multiple phase-shifts.

All signals are synchronous to scanclk. They are latched on scanclk edges and must meet t_{su}/t_h requirements with respect to scanclk edges.

Figure 6–44. Dynamic Phase Shifting Waveform



Dynamic phase-shifting can be repeated indefinitely. All signals are synchronous to `scanclk` and must meet t_{su}/t_h requirements with respect to `scanclk` edges.

The `phasestep` signal is latched on the negative edge of `scanclk`. In Figure 6–44, this is shown by the second `scanclk` falling edge. `phasestep` must stay high for at least two `scanclk` cycles. On the second `scanclk` rising edge after `phasestep` is latched (the fourth `scanclk` rising edge in Figure 6–44), the values of `phaseupdown` and `phasecounterselect` are latched and the PLL starts dynamic phase-shifting for the specified counters and in the indicated direction. On the fourth `scanclk` rising edge, `phasedone` goes high to low and remains low until the PLL finishes dynamic phase-shifting. You can perform another dynamic phase-shift after the `phasedone` signal goes from low to high.

Depending on the VCO and `scanclk` frequencies, `phasedone` low time ($t_{CONFIGPHASE}$) may be greater than or less than one `scanclk` cycle.

After `phasedone` goes from low to high, you can perform another dynamic phase shift. `Phasestep` pulses must be at least one `scanclk` cycle apart.

For more information about the ALTPLL_RECONFIG MegaWizard Plug-In Manager, refer to the *ALTPLL_RECONFIG Megafunction User Guide*.

PLL Cascading and Clock Network Guidelines

When cascading PLLs in Stratix III devices, the source (upstream) PLL must have a low-bandwidth setting while the destination (downstream) PLL must have a high-bandwidth setting. There must be no overlap of the bandwidth ranges of the two PLLs.

To ensure that the memory interface's PLL is configured correctly in the external memory interface design, the following settings are required:

- The PLL used to generate the memory output clock signals and write data/clock signals must be set to **No compensation** mode to minimize output clock jitter.
- The reference input clock signal to the PLL must be driven by the dedicated clock input pin located adjacent to the PLL, or from the clock output signal from the adjacent PLL. To minimize output clock jitter, the reference input clock pin must not be routed through the core using global or regional clock networks. If reference clock is cascaded from another PLL, that upstream PLL must be configured in **No compensation** mode and **Low bandwidth** mode.

Spread-Spectrum Tracking

Stratix III devices can accept a spread-spectrum input with typical modulation frequencies. However, the device cannot automatically detect that the input is a spread-spectrum signal. Instead, the input signal looks like deterministic jitter at the input of PLL. Stratix III PLLs can track a spread-spectrum input clock as long as it is within the input-jitter tolerance specifications and the modulation frequency of the input clock is below the PLL bandwidth which is specified in the fitter report. Stratix III devices cannot internally generate spread-spectrum clocks.

PLL Specifications

For information about PLL timing specifications, refer to the *DC and Switching Characteristics of Stratix III Devices* chapter.

Chapter Revision History

Table 6–23 lists the revision history for this chapter.

Table 6–23. Chapter Revision History (Part 1 of 2)

Date	Version	Changes Made
July 2010	2.0	Updated Figure 6–44.
March 2010	1.9	Updated for the Quartus II software version 9.1 SP2 release: ■ Updated Table 6–10 and Table 6–11. ■ Updated Figure 6–42. ■ Updated the "Guidelines" and "PLL Cascading and Clock Network Guidelines" sections. ■ Removed "sub-regional clock networks" information. ■ Minor text edits.
July 2009	1.8	■ Updated "Clock Switchover" section. ■ Updated Figure 6–37.
May 2009	1.7	■ Added "PLL and Clock Network Guidelines for External Memory Interface" and "Zero-Delay Buffer Mode" sections. ■ Updated Figure 6–17.
February 2009	1.6	■ Updated Table 6–7 and Table 6–10. ■ Updated Figure 6–23. ■ Updated "PLL Clock I/O Pins", "Logic Array Blocks (LABs)", and "Clock Feedback Modes" sections. ■ Removed "Reference Documents" section.
October 2008	1.5	■ Updated Table 6–10, Table 6–13, and Table 6–14. ■ Updated "locked", "Manual Override", "Bypassing PLL", "PLL Clock I/O Pins", and "Dynamic Phase-Shifting" sections. ■ Updated Figure 6–22, Figure 6–24, and Figure 6–26. ■ Updated (Note 2) to Figure 6–22. ■ Added (Note 3) to Table 6–14. ■ Added Figure 6–27. ■ Updated New Document Format.
May 2008	1.4	■ Updated Table 6–3, Table 6–4, Table 6–5, Table 6–6, Table 6–7, and Table 6–14. ■ Added new Figure 6–5 through Figure 6–9 to "Periphery Clock Networks" section. ■ Updated "Logic Array Blocks (LABs)", "External Feedback Mode", "Phase-Shift Implementation", and "Spread-Spectrum Tracking" sections. ■ Updated notes to Figure 6–17. ■ Updated notes to Figure 6–22. ■ Updated notes to Figure 6–27. ■ Updated Figure 6–43.
November 2007	1.3	Updated "pfdena" on page 6–42.

Table 6–23. Chapter Revision History (Part 2 of 2)

Date	Version	Changes Made
October 2007	1.2	■ Updated Table 6–13 to remove a reference to gated locks. Updated Table 6–16 and added new rows to it. ■ Modified Figure 6–3 and Figure 6–40. ■ Edited notes for Figure 6–9, Figure 6–10, and Figure 6–17. ■ Replaced Figure 6–41. ■ Added section “Referenced Documents”. ■ Added live links for references.
May 2007	1.1	Changed frequency difference between inclk0 and inclk1 to more than 20% instead of 100% on page 42. Updated Table 6–16, note to Figure 6–17, and Figure 6–19.
November 2006	1.0	Initial Release.

Section II. I/O Interfaces

This section provides information on Stratix® III device I/O features, external memory interfaces, and high-speed differential interfaces with DPA. This section includes the following chapters:

- Chapter 7, Stratix III Device I/O Features
- Chapter 8, External Memory Interfaces in Stratix III Devices
- Chapter 9, High-Speed Differential I/O Interfaces and DPA in Stratix III Devices

Revision History

Refer to each chapter for its own specific revision history. For information on when each chapter was updated, refer to the Chapter Revision Dates section, which appears in the full handbook.

7. Stratix III Device I/O Features

SIII51007-1.9

Stratix® III I/Os are specifically designed for ease of use and rapid system integration while simultaneously providing the high bandwidth required to maximize internal logic capabilities and produce system-level performance. Independent modular I/O banks with a common bank structure for vertical migration lend efficiency and flexibility to the high speed I/O. Package and die enhancements with dynamic termination and output control provide best-in-class signal integrity. Numerous I/O features assist in high-speed data transfer into and out of the device, including:

- Single-ended, non-voltage-referenced, and voltage-referenced I/O standards
- Low-voltage differential signaling (LVDS), reduced swing differential signal (RSDS), mini-LVDS, high-speed transceiver logic (HSTL), and stub series terminated logic (SSTL)
- Single data rate (SDR) and half data rate (HDR—half frequency and twice data width of SDR) input and output options
- Up to 132-full duplex 1.6-Gbps true LVDS channels (132 Tx + 132 Rx) on the row I/O banks
- Hard dynamic phase alignment (DPA) block with serializer/deserializer (SERDES)
- De-skew, read and write leveling, and clock-domain crossing functionality
- Programmable output current strength
- Programmable slew rate
- Programmable delay
- Programmable bus-hold
- Programmable pull-up resistor
- Open-drain output
- Serial, parallel, and dynamic on-chip termination (OCT)
- Differential OCT
- Programmable pre-emphasis
- Programmable differential output voltage (V_{OD})

Stratix III I/O Standards Support

Stratix III devices support a wide range of industry I/O standards. Table 7–1 lists the I/O standards supported by Stratix III devices as well as typical applications. Stratix III devices support V_{CCIO} voltage levels of 3.3, 3.0, 2.5, 1.8, 1.5, and 1.2 V.

Table 7–1. I/O Standard Applications for Stratix III Devices (Part 1 of 2)

I/O Standard	Typical Application
3.3-V LVTTL/LVCMOS	General purpose
3.0-V LVTTL/LVCMOS	General purpose
2.5-V LVTTL/LVCMOS	General purpose
1.8-V LVTTL/LVCMOS	General purpose
1.5-V LVTTL/LVCMOS	General purpose
1.2-V LVTTL/LVCMOS	General purpose
3.0-V PCI	PC and embedded system
3.0-V PCI-X	PC and embedded system
SSTL-2 Class I	DDR SDRAM
SSTL-2 Class II	DDR SDRAM
SSTL-18 Class I	DDR2 SDRAM
SSTL-18 Class II	DDR2 SDRAM
SSTL-15 Class I	DDR3 SDRAM
SSTL-15 Class II	DDR3 SDRAM
HSTL-18 Class I	QDR II/RLDRAM II
HSTL-18 Class II	QDR II/RLDRAM II
HSTL-15 Class I	QDR II/QDR II+/RLDRAM II
HSTL-15 Class II	QDR II/QDR II+/RLDRAM II
HSTL-12 Class I	General purpose
HSTL-12 Class II	General purpose
Differential SSTL-2 Class I	DDR SDRAM
Differential SSTL-2 Class II	DDR SDRAM
Differential SSTL-18 Class I	DDR2 SDRAM
Differential SSTL-18 Class II	DDR2 SDRAM
Differential SSTL-15 Class I	DDR3 SDRAM
Differential SSTL-15 Class II	DDR3 SDRAM
Differential HSTL-18 Class I	Clock interfaces
Differential HSTL-18 Class II	Clock interfaces
Differential HSTL-15 Class I	Clock interfaces
Differential HSTL-15 Class II	Clock interfaces
Differential HSTL-12 Class I	Clock interfaces
Differential HSTL-12 Class II	Clock interfaces
LVDS	High-speed communications
RSDS	Flat panel display

Table 7–1. I/O Standard Applications for Stratix III Devices (Part 2 of 2)

I/O Standard	Typical Application
mini-LVDS	Flat panel display
LVPECL	Video graphics and clock distribution

I/O Standards and Voltage Levels

Stratix III devices support a wide range of industry I/O standards, including single-ended, voltage-referenced single-ended, and differential I/O standards.

Table 7–2 lists the supported I/O standards and the typical values for input and output V_{CCIO}, V_{CCPD}, V_{REF}, and board V_{TT}.

Table 7–2. I/O Standards and Voltage Levels for Stratix III Devices *(Note 1)*, *(3)* (Part 1 of 3)

I/O Standard	Standard Support	V_{CCIO} (V)				V_{CCPD} (V) (Pre-Driver Voltage)	V_{REF} (V) (Input Ref Voltage)	V_{TT} (V) (Board Termination Voltage)
		Input Operation		Output Operation				
		Column I/O Banks	Row I/O Banks	Column I/O Banks	Row I/O Banks			
3.3-V LVTTL	JESD8-B	3.3/3.0/2.5	3.3/3.0/2.5	3.3	3.3	3.3	—	—
3.3-V LVCMOS	JESD8-B	3.3/3.0/2.5	3.3/3.0/2.5	3.3	3.3	3.3	—	—
3.0-V LVTTL	JESD8-B	3.3/3.0/2.5	3.3/3.0/2.5	3.0	3.0	3.0	—	—
3.0-V LVCMOS	JESD8-B	3.3/3.0/2.5	3.3/3.0/2.5	3.0	3.0	3.0	—	—
2.5-V LVTTL/LVCMOS	JESD8-5	3.3/3.0/2.5	3.3/3.0/2.5	2.5	2.5	2.5	—	—
1.8-V LVTTL/LVCMOS	JESD8-7	1.8/1.5	1.8/1.5	1.8	1.8	2.5	—	—
1.5-V LVTTL/LVCMOS	JESD8-11	1.8/1.5	1.8/1.5	1.5	1.5	2.5	—	—
1.2-V LVTTL/LVCMOS	JESD8-12	1.2	1.2	1.2	1.2	2.5	—	—
3.0-V PCI	PCI Rev 2.2	3.0	3.0	3.0	3.0	3.0	—	—
3.0-V PCI-X	PCI-X Rev 1.0	3.0	3.0	3.0	3.0	3.0	—	—
SSTL-2 Class I	JESD8-9B	*(2)*	*(2)*	2.5	2.5	2.5	1.25	1.25
SSTL-2 Class II	JESD8-9B	*(2)*	*(2)*	2.5	2.5	2.5	1.25	1.25
SSTL-18 Class I	JESD8-15	*(2)*	*(2)*	1.8	1.8	2.5	0.90	0.90
SSTL-18 Class II	JESD8-15	*(2)*	*(2)*	1.8	1.8	2.5	0.90	0.90
SSTL-15 Class I	—	*(2)*	*(2)*	1.5	1.5	2.5	0.75	0.75
SSTL-15 Class II	—	*(2)*	*(2)*	1.5	—	2.5	0.75	0.75
HSTL-18 Class I	JESD8-6	*(2)*	*(2)*	1.8	1.8	2.5	0.90	0.90
HSTL-18 Class II	JESD8-6	*(2)*	*(2)*	1.8	1.8	2.5	0.90	0.90
HSTL-15 Class I	JESD8-6	*(2)*	*(2)*	1.5	1.5	2.5	0.75	0.75
HSTL-15 Class II	JESD8-6	*(2)*	*(2)*	1.5	—	2.5	0.75	0.75
HSTL-12 Class I	JESD8-16A	*(2)*	*(2)*	1.2	1.2	2.5	0.6	0.6
HSTL-12 Class II	JESD8-16A	*(2)*	*(2)*	1.2	—	2.5	0.6	0.6

Table 7–2. I/O Standards and Voltage Levels for Stratix III Devices *(Note 1)*, *(3)* (Part 2 of 3)

I/O Standard	Standard Support	V_{CCIO} (V)				V_{CCPD} (V) (Pre-Driver Voltage)	V_{REF} (V) (Input Ref Voltage)	V_{TT} (V) (Board Termination Voltage)
		Input Operation		Output Operation				
		Column I/O Banks	Row I/O Banks	Column I/O Banks	Row I/O Banks			
Differential SSTL-2 Class I	JESD8-9B	*(2)*	*(2)*	2.5	2.5	2.5	—	1.25
Differential SSTL-2 Class II	JESD8-9B	*(2)*	*(2)*	2.5	2.5	2.5	—	1.25
Differential SSTL-18 Class I	JESD8-15	*(2)*	*(2)*	1.8	1.8	2.5	—	0.90
Differential SSTL-18 Class II	JESD8-15	*(2)*	*(2)*	1.8	1.8	2.5	—	0.90
Differential SSTL-15 Class I	—	*(2)*	*(2)*	1.5	1.5	2.5	—	0.75
Differential SSTL-15 Class II	—	*(2)*	*(2)*	1.5	—	2.5	—	0.75
Differential HSTL-18 Class I	JESD8-6	*(2)*	*(2)*	1.8	1.8	2.5	—	0.90
Differential HSTL-18 Class II	JESD8-6	*(2)*	*(2)*	1.8	1.8	2.5	—	0.90
Differential HSTL-15 Class I	JESD8-6	*(2)*	*(2)*	1.5	1.5	2.5	—	0.75
Differential HSTL-15 Class II	JESD8-6	*(2)*	*(2)*	1.5	—	2.5	—	0.75
Differential HSTL-12 Class I	JESD8-16A	*(2)*	*(2)*	1.2	1.2	2.5	—	0.60
Differential HSTL-12 Class II	JESD8-16A	*(2)*	*(2)*	1.2	—	2.5	—	0.60
LVDS *(6)*, *(8)*	ANSI/TIA/EIA-644	*(2)*	*(2)*	2.5	2.5	2.5	—	—
RSDS *(6)*, *(7)*, *(8)*	—	*(2)*	*(2)*	2.5	2.5	2.5	—	—
mini-LVDS *(6)*, *(7)*, *(8)*	—	*(2)*	*(2)*	2.5	2.5	2.5	—	—

Table 7–2. I/O Standards and Voltage Levels for Stratix III Devices *(Note 1)*, *(3)* (Part 3 of 3)

I/O Standard	Standard Support	V_{CCIO} (V)				V_{CCPD} (V) (Pre-Driver Voltage)	V_{REF} (V) (Input Ref Voltage)	V_{TT} (V) (Board Termination Voltage)
		Input Operation		Output Operation				
		Column I/O Banks	Row I/O Banks	Column I/O Banks	Row I/O Banks			
LVPECL	—	*(4)*	*(4)*	—	—	2.5	—	—

Notes to Table 7–2:

(1) V_{CCPD} is either 2.5 V, 3.0 V, or 3.3 V. For V_{CCIO} = 3.3 V, V_{CCPD}=3.3 V. For V_{CCIO} = 3.0 V, V_{CCPD} = 3.0 V. For V_{CCIO} = 2.5 V or less, V_{CCPD} = 2.5 V.

(2) Single-ended HSTL/SSTL, differential SSTL/HSTL, and LVDS input buffers are powered by V_{CCPD}.

(3) $V_{CCCLKIN}$ powers the Column I/O bank dedicated clock input pins when configured as differential inputs. Clock input pins on the Column I/O banks use V_{CCIO} when configured as single-ended inputs.

(4) Column and row I/O banks support LVPECL I/O standards for input operation only on dedicated clock input pins. Clock inputs on column I/O are powered by $V_{CCCLKIN}$ when configured as differential clock input. Differential clock inputs in row I/O are powered by V_{CCPD}.

(5) Row I/O banks support LVDS outputs using a dedicated output buffer. Column and row I/O banks support emulated LVDS outputs using two single-ended output buffers and external one-resistor (LVDS_E_1R) and a three-resistor (LVDS_E_3R) network.

(6) Row I/O banks support RSDS and mini-LVDS I/O standards using a true LVDS output buffer without a resistor network.

(7) Column and row I/O banks support emulated-RSDS and mini-LVDS I/O standards using two single-ended output buffers with one-resistor (RSDS_E_1R and mini-LVDS_E_1R) and three-resistor (RSDS_E_3R and mini-LVDS_E_3R) networks.

(8) The emulated differential output standard that supports the tri-state feature includes: LVDS_E_1R, LVDS_E_3R, RSDS_E_1R, RSDS_E_3R, Mini_LVDS_E_1R, and Mini_LVDS_E_3R.

For detailed electrical characteristics of each I/O standard, refer to the *DC and Switching Characteristics of Stratix III Devices* chapter.

Stratix III I/O Banks

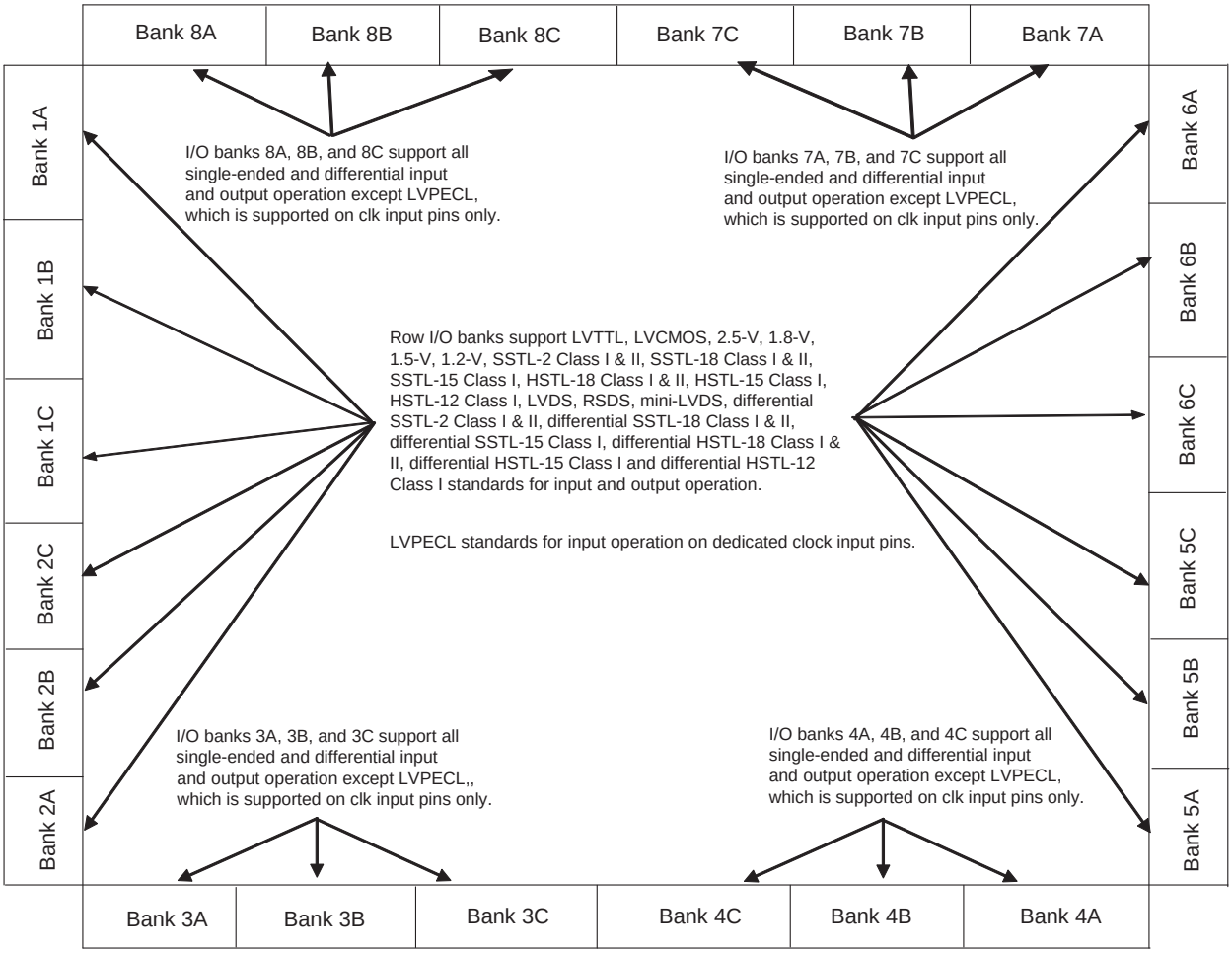
Stratix III devices contain up to 24 I/O banks, as shown in Figure 7–1. The row I/O banks contain true differential input and output buffers and dedicated circuitry to support differential standards at speeds up to 1.6 Gbps.

Every I/O bank in Stratix III devices can support high-performance external memory interfaces with dedicated circuitry. The I/O pins are organized in pairs to support differential standards. Each I/O pin pair can support both differential input and output buffers. The only exceptions are the `CLK[1,3,8,10][p,n]`, `PLL_L[1,4]_CLK[p,n]`, and `PLL_R[1,4]_CLK[p,n]` pins, which support differential or single-ended input operations, these pins do not support output operations.

Pins that do not support output operations do not support the programmable current strength, programmable slew rate, programmable pull-up, bus hold, open-drain, or on-chip series termination (OCT R_S) options.

For the number of channels available for the LVDS I/O standard, refer to the *High-Speed Differential I/O Interface with DPA in Stratix III Devices* chapter.

Figure 7–1. I/O Banks for Stratix III Devices *(Note 1), (2), (3), (4), (5), (6), (7), (8), (9)*



Notes to Figure 7–1:

(1) Differential HSTL and SSTL outputs are not true differential outputs. They use two single-ended outputs with the second output programmed as inverted.

(2) Column and row I/O differential HSTL and SSTL inputs use LVDS differential input buffers without OCT R_D support.

(3) Column and row I/O supports emulated LVDS output buffer.

(4) Column I/O supports PCI/PCI-X with on-chip clamp diode, and row I/O supports PCI/PCI-X with external clamp diode.

(5) Clock inputs on column I/O are powered by $V_{CCCLKIN}$ when configured as differential clock input. They are powered by V_{CCIO} when configured as single-ended clock input. All outputs use the corresponding bank V_{CCIO}.

(6) Row I/O supports the true LVDS output buffer.

(7) Column and row I/O banks support LVPECL standards for input operation on dedicated clock input pins.

(8) Figure 7–1 is a top view of the silicon die that corresponds to a reverse view for flip chip packages. It is a graphical representation only.

(9) 3.0-V PCI/PCI-X and 3.3-V LVTTL/LVCMOS outputs are not supported in the same I/O bank.

Modular I/O Banks

The I/O pins in Stratix III devices are arranged in groups called modular I/O banks. Depending on device densities, the number of I/O banks range from 16 to 24 banks. The size of each bank is 24, 32, 36, 40, or 48 I/O pins. Figure 7–3 to Figure 7–5 show the number of I/O pins available in each I/O bank and packaging information for different sets of available devices.

In Stratix III devices, the maximum number of I/O banks per side is four or six, depending on the device density. When migrating between devices with a different number of I/O banks per side, it is the middle or "B" bank which is removed or inserted. For example, when moving from a 24-bank device to a 16-bank device, the banks that are dropped are "B" banks, namely: 1B, 2B, 3B, 4B, 5B, 6B, 7B, and 8B. Similarly, when moving from a 16-bank device to a 24-bank device, the banks that are added are "B" banks, namely: 1B, 2B, 3B, 4B, 5B, 6B, 7B, and 8B.

During migration from a smaller device to a larger device, the bank size increases or remains the same but never decreases. For example, banks may increase from a size of 24 I/O to a bank of size 32, 36, 40, or 48 I/O, but never decrease. Table 7–3 lists the increase in bank size when migrating from a smaller device to a larger device.

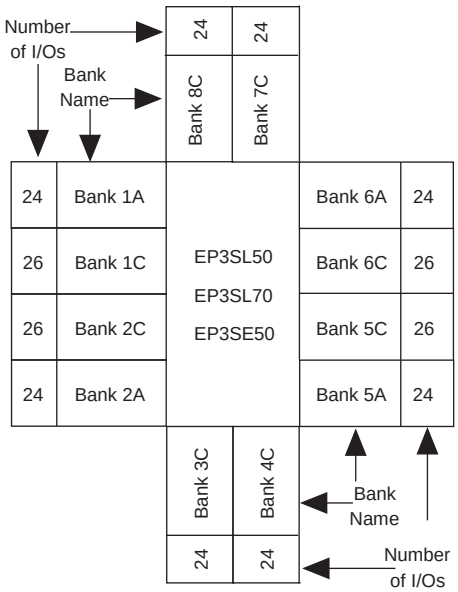
Table 7–3. Bank Migration Path with Increasing Device Size *(Note 1)*

	Banks	Increase in Bank Size (number of I/O)		
Column I/O	A	40	48	—
	B	24	48	—
	C	24	32	48
Row I/O	A	32	48	—
	B	24	36	—
	C	24	40	48

Note to Table 7–3:

(1) Number of I/O shown does not include dedicated clock input pins `CLK[1,3,8,10][p,n]`.

Figure 7–2. Number of I/Os in Each Bank in EP3SL50, EP3SL70, and EP3SE50 Devices in 484-Pin FineLine BGA Package *(Note 1)*, *(2)*,

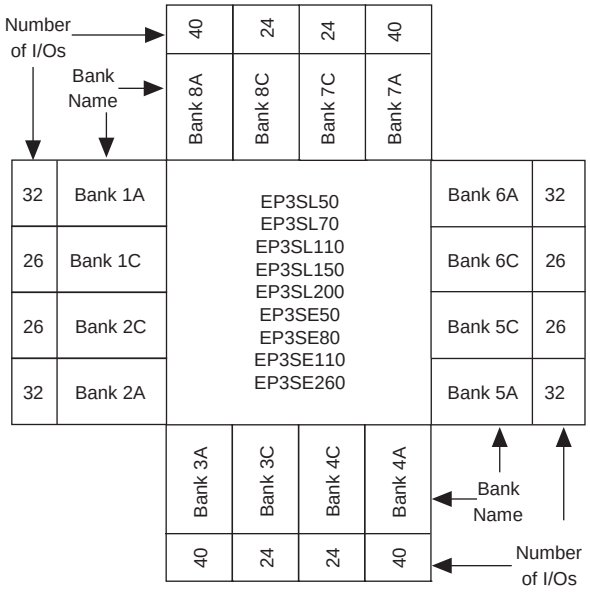


Notes to Figure 7–2:

(1) All I/O pin counts include dedicated clock inputs pins. The pin count includes all general purpose I/O, dedicated clock pins, and dual-purpose configuration pins. Transceiver pins and dedicated configuration pins are not included in the pin count.

(2) Figure 7–2 is a top view of the silicon die that corresponds to a reverse view for flip chip packages. It is a graphical representation only.

Figure 7–3. Number of I/Os in Each Bank in the 780-pin FineLine BGA Package *(Note 1)*, *(2)*, *(3)*, *(4)*



Notes to Figure 7–3:

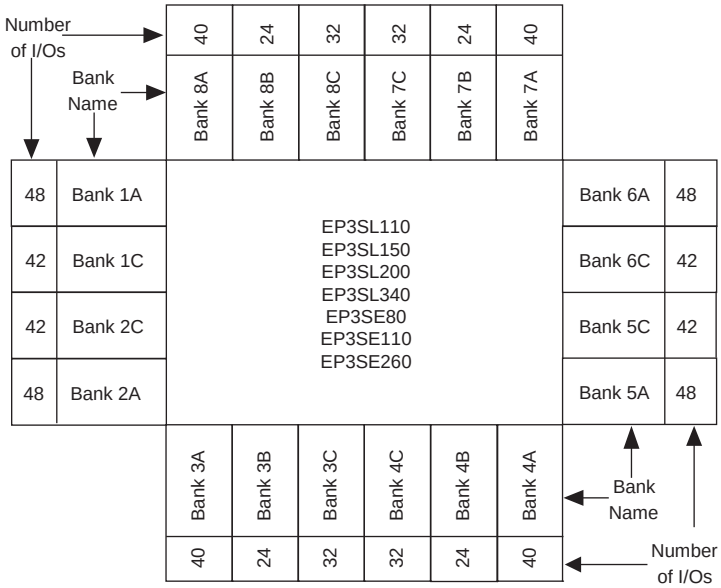
(1) All I/O pin counts include dedicated clock inputs pins. The pin count includes all general purpose I/O, dedicated clock pins, and dual-purpose configuration pins. Dedicated configuration pins are not included in the pin count.

(2) Figure 7–3 is a top view of the silicon die that corresponds to a reverse view for flip chip packages. It is a graphical representation only.

(3) Number of I/Os in each Bank in EP3SL50,EP3SL70, EP3SL110, EP3SL150, EP3SE50, EP3SE80 and EP3SE110 in the 780-pin FineLine BGA package.

(4) Number of I/Os in each Bank in EP3SL200 and EP3SE260 in the 780-pin Hybrid FineLine BGA package.

Figure 7–4. Number of I/Os in Each Bank in the 1152-pin FineLine BGA Package *(Note 1)*, *(2)*, *(3)*, *(4)*



Notes to Figure 7–4:

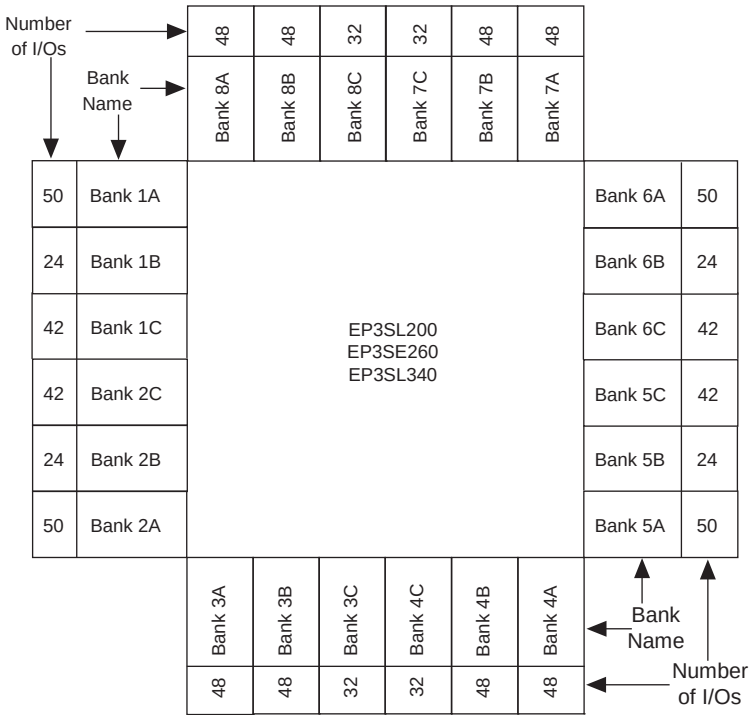
(1) All I/O pin counts include dedicated clock inputs pins. The pin count includes all general purpose I/O, dedicated clock pins, and dual-purpose configuration pins. Dedicated configuration pins are not included in the pin count.

(2) Figure 7–4 is a top view of the silicon die that corresponds to a reverse view for flip chip packages. It is a graphical representation only.

(3) Number of I/Os in Each Bank in EP3SL110, EP3SL150, EP3SL200, EP3SE80, EP3SE110, and EP3SE260 Devices in the 1152-pin FineLine BGA package.

(4) Number of I/Os in Each Bank in EP3SL340 in the 1152-pin Hybrid FineLine BGA package.

Figure 7–5. Number of I/Os in Each Bank in EP2SL200, EP3SE260, and EP3SL340 Devices in the 1517-Pin FineLine BGA Package *(Note 1)*, *(2)*

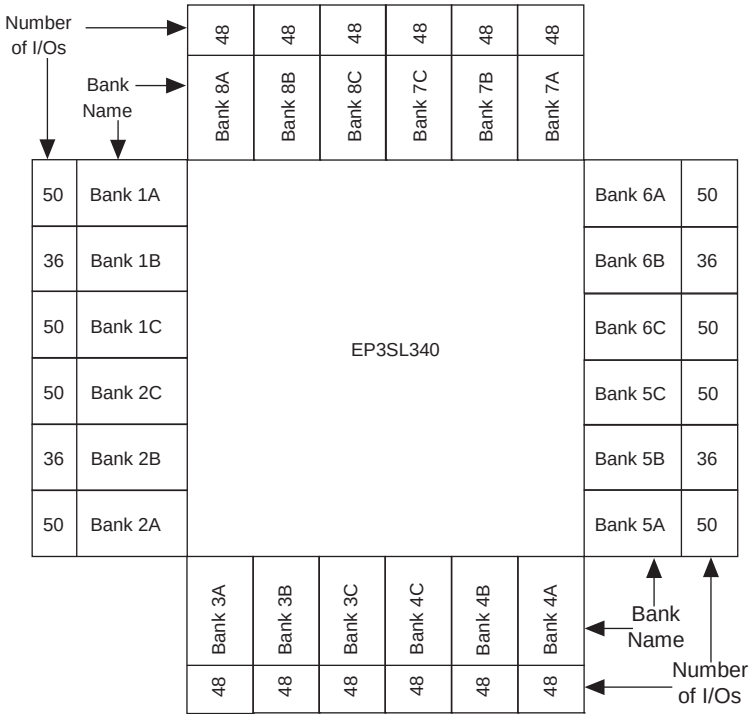


Notes to Figure 7–5:

(1) All I/O pin counts include dedicated clock inputs pins. The pin count includes all general purpose I/O, dedicated clock pins, and dual-purpose configuration pins. Dedicated configuration pins are not included in the pin count.

(2) Figure 7–5 is a top view of the silicon die that corresponds to a reverse view for flip chip packages. It is a graphical representation only.

Figure 7–6. Number of I/Os in Each Bank in EP3SL340 Devices in the 1760-pin FineLine BGA Package *(Note 1)*, *(2)*



Notes to Figure 7–6:

(1) All I/O pin counts include dedicated clock inputs pins. The pin count includes all general purpose I/O, dedicated clock pins, and dual-purpose configuration pins. Dedicated configuration pins are not included in the pin count.

(2) Figure 7–6 is a top view of the silicon die that corresponds to a reverse view for flip chip packages. It is a graphical representation only.

Stratix III I/O Structure

The I/O element (IOE) in Stratix III devices contains a bi-directional I/O buffer and I/O registers to support a complete embedded bi-directional single data rate or DDR transfer. The IOEs are located in I/O blocks around the periphery of the Stratix III device. There are up to four IOEs per row I/O block and four IOEs per column I/O block. The row IOEs drive row, column, or direct link interconnects. The column IOEs drive column interconnects.

The Stratix III bi-directional IOE also supports the following features:

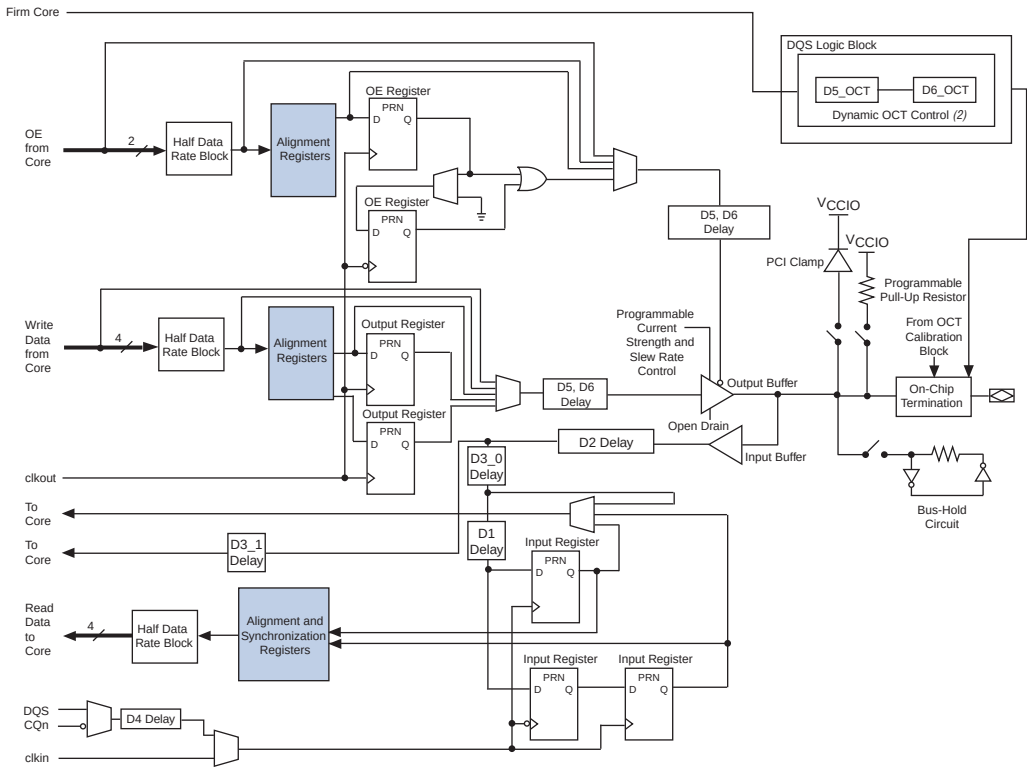
- Programmable input delay
- Programmable output-current strength
- Programmable slew rate
- Programmable output delay
- Programmable bus-hold
- Programmable pull-up resistor
- Open-drain output
- On-chip series termination with calibration

- On-chip series termination without calibration
- On-chip parallel termination with calibration (OCT R_T)
- On-chip differential termination (OCT R_D)
- PCI clamping diode

The I/O registers are composed of the input path for handling data from the pin to the core, the output path for handling data from the core to the pin, and the output-enable (OE) path for handling the OE signal for the output buffer. These registers allow faster source-synchronous register-to-register transfers and resynchronization. The input path consists of the DDR input registers, alignment and synchronization registers, and HDR. You can bypass each block of the input path.

Figure 7–7 shows the Stratix III IOE structure.

Figure 7–7. IOE Structure for Stratix III Devices *(Note 1)*, *(2)*



Notes to Figure 7–7:

(1) `D3_0` and `D3_1` delays have the same available settings in the Quartus® II software.

(2) One dynamic OCT control is available per DQ/DQS group.

The output and OE paths are divided into output or OE registers, alignment registers, and HDR blocks. You can bypass each block of the output and OE path.

For more information about I/O registers and how they are used for memory applications, refer to the *External Memory Interfaces in Stratix III Devices* chapter.

3.3-V I/O Interface

Stratix III I/O buffers are fully compatible with 3.3-V I/O standards, and you can use them as transmitters or receivers in your system. The output high voltage (V_{OH}), output low voltage (V_{OL}), input high voltage (V_{IH}), and input low voltage (V_{IL}) levels meet the 3.3-V I/O standards specifications defined by EIA/JEDEC Standard JESD8-B with margin when the Stratix III V_{CCIO} voltage is powered by 3.3 V or 3.0 V.

For device reliability and proper operation when interfacing with a 3.3 V I/O system using Stratix III devices, ensure that the absolute maximum ratings of Stratix III devices are not violated. Altera recommends performing IBIS simulation to determine that the overshoot and undershoot voltages are within the guidelines.

When using a Stratix III device as a transmitter, some techniques can limit the overshoot and undershoot at the I/O pins, such as slow slew rate and series termination, but they are not mandatory. Transmission line effects that cause large voltage deviation at the receiver are associated with impedance mismatch between the driver and transmission line. By matching the impedance of the driver to the characteristic impedance of the transmission line, overshoot voltage can be significantly reduced. You can use a series termination resistor placed physically close to the driver to match the total driver impedance to the transmission line impedance. Stratix III devices support OCT R_S for all LVTTL/LVCMOS I/O standards in all I/O banks.

When using a Stratix III device as a receiver, a clamping diode can be used to limit the overshoot (on-chip or off-chip), but it is not mandatory. Stratix III devices provide an optional on-chip PCI-clamp diode for column I/O pins. You can use this diode to protect I/O pins against overshoot voltage.

Another method for limiting overshoot is reducing the bank supply voltage (V_{CCIO}) to 3.0 V. With this method, the clamp diode (on-chip or off-chip), though not mandatory, can sufficiently clamp overshoot voltage to within the DC and AC input voltage specification. The clamped voltage can be expressed as the sum of the supply voltage (V_{CCIO}) and the diode forward voltage. By lowering V_{CCIO} to 3.0 V you can reduce overshoot and undershoot for all I/O standards, including 3.3-V LVTTL/LVCMOS, 3.0-V LVTTL/LVCMOS, and 3.0-V PCI/PCI-X. Additionally, lowering V_{CCIO} to 3.0 V reduces power consumption.

For more information about absolute maximum rating and maximum allowed overshoot during transitions, refer to the *DC and Switching Characteristics of Stratix III Devices* chapter.

External Memory Interfaces

In addition to the I/O registers in each IOE, Stratix III devices also have dedicated registers and phase-shift circuitry on all I/O banks for interfacing with external memory interfaces. Table 7–4 lists the memory interfaces and the corresponding I/O standards supported by Stratix III devices.

Table 7–4. Memory Interface Standards Supported (Part 1 of 2)

Memory Interface Standard	I/O Standard
DDR SDRAM	SSTL-2
DDR2 SDRAM	SSTL-18

Table 7–4. Memory Interface Standards Supported (Part 2 of 2)

Memory Interface Standard	I/O Standard
DDR3 SDRAM	SSTL-15
RLDRAM II	HSTL-18
QDR II SRAM	HSTL-18
QDR II+ SRAM	HSTL-15

For more information about external memory interfaces, refer to the *External Memory Interfaces in Stratix III Devices* chapter.

High-Speed Differential I/O with DPA Support

Stratix III devices contain dedicated circuitry for supporting differential standards at speeds up to 1.6 Gbps. The high-speed differential I/O circuitry supports the following high speed I/O interconnect standards and applications: Utopia IV, SPI-4.2, SFI-4, 10 Gigabit Ethernet XSBI, RapidIOTM, and NPSI. Stratix III devices support ×2, ×4, ×6, ×7, ×8, and ×10 SERDES modes for high-speed differential I/O interfaces and ×4, ×6, ×7, ×8, and ×10 SERDES modes with dedicated DPA circuitry. DPA minimizes bit errors, simplifies PCB layout and timing management for high-speed data transfer, and eliminates channel-to-channel and channel-to-clock skew in high-speed data transmission systems.

×2 mode is supported by the DDR registers and is not included in SERDES. For Stratix III devices, SERDES can be bypassed in the Quartus II MegaWizard™ Plug-In Manager for the ALTLVDS megafunction to support DDR (×2) operation.

Stratix III devices have the following dedicated circuitry for high-speed differential I/O support:

- Differential I/O buffer
- Transmitter serializer
- Receiver deserializer
- Data realignment
- DPA
- Synchronizer (FIFO buffer)
- Phase-locked loops (PLLs)

For more information about DPA support, refer to the *High-Speed Differential I/O Interfaces with DPA in Stratix III Devices* chapter.

Programmable Current Strength

The output buffer for each Stratix III device I/O pin has a programmable current-strength control for certain I/O standards. You can use programmable current strength to mitigate the effects of high signal attenuation due to a long transmission line or a legacy backplane. The LVTTL, LVCMOS, SSTL, and HSTL standards have several levels of current strength that you can control. Table 7–5 lists information about programmable current strength.

Table 7–5. Programmable Current Strength *(Note 1)*

I/O Standard	I_{OH} / I_{OL} Current Strength Setting (mA) for Column I/O Pins	I_{OH} / I_{OL} Current Strength Setting (mA) for Row I/O Pins
3.3-V LVTTL	16, 12, 8, 4	12, 8, 4
3.3-V LVCMOS	16, 12, 8, 4	8, 4
3.0-V LVTTL	16, 12, 8, 4	12, 8, 4
3.0-V LVCMOS	16, 12, 8, 4	8, 4
2.5-V LVTTL/LVCMOS	16, 12, 8, 4	12, 8, 4
1.8-V LVTTL/LVCMOS	12, 10, 8, 6, 4, 2	8, 6, 4, 2
1.5-V LVTTL/LVCMOS	12, 10, 8, 6, 4, 2	8, 6, 4, 2
1.2-V LVTTL/LVCMOS	8, 6, 4, 2	4, 2
SSTL-2 Class I	12, 10, 8	12, 8
SSTL-2 Class II	16	16
SSTL-18 Class I	12, 10, 8, 6, 4	12, 10, 8, 6, 4
SSTL-18 Class II	16, 8	16, 8
SSTL-15 Class I	12, 10, 8, 6, 4	8, 6, 4
SSTL-15 Class II	16, 8	—
HSTL-18 Class I	12, 10, 8, 6, 4	12, 10, 8, 6, 4
HSTL-18 Class II	16	16
HSTL-15 Class I	12, 10, 8, 6, 4	8, 6, 4
HSTL-15 Class II	16	—
HSTL-12 Class I	12, 10, 8, 6, 4	8, 6, 4
HSTL-12 Class II	16	—

Note to Table 7–5:

(1) The default setting in the Quartus II software is 50-Ω OCT R_S without calibration for all non-voltage reference and HSTL/SSTL class I I/O standards. The default setting is 25-Ω OCT R_S without calibration for HSTL/SSTL class II I/O standards.

Altera recommends performing IBIS or SPICE simulations to determine the right current strength setting for your specific application.

Programmable Slew Rate Control

The output buffer for each Stratix III device regular- and dual-function I/O pin has a programmable output slew-rate control that you can configure for low-noise or high-speed performance. A faster slew rate provides high-speed transitions for high-performance systems. A slow slew rate can help reduce system noise, but adds a nominal delay to rising and falling edges. Each I/O pin has an individual slew-rate control, allowing you to specify the slew rate on a pin-by-pin basis.

You cannot use the programmable slew rate feature when using OCT R_S.

The Quartus II software allows four settings for programmable slew rate control—0, 1, 2, and 3—where 0 is slow slew rate and 3 is fast slew rate. Table 7–6 lists the default setting for the I/O standards supported in the Quartus II software.

Table 7–6. Default Programmable Slew Rate

I/O Standard	Default Slew Rate Setting
1.2-V, 1.5-V, 1.8-V, 2.5-V, 3.0-V, and 3.3-V LVTTL / LVCMOS	3
3.0-V PCI / PCI-X	3
SSTL-2, -18, -15 Class I and Class II	3
HSTL-18, -15, -12 Class I and II	3
Differential SSTL-2, -18, -15 Class I and Class II	3
Differential HSTL-18, -15, -12 Class I and Class II	3
LVDS_E_1R, mini-LVDS_E_1R, RSDS_E_1R	3
LVDS_E_3R, mini-LVDS_E_3R, RSDS_E_3R	3

You can use faster slew rates to improve the available timing margin in memory-interface applications or when the output pin has high-capacitive loading. Altera recommends performing IBIS or SPICE simulations to determine the right slew rate setting for your specific application.

Programmable Delay

The Stratix III device IOE includes programmable delays (refer to Figure 7–7) that you can activate to ensure zero hold times, minimize setup times, or increase clock-to-output times. Each pin can have a different input delay from pin to input register or a delay from the output register to the output pin values to ensure that the bus has the same delay going into or out of the device. This feature helps read and time margins as it minimizes the uncertainties between signals in the bus.

For the programmable IOE delay specifications, refer to the *DC and Switching Characteristics of Stratix III Devices* chapter.

Programmable Output Buffer Delay

Stratix III devices support delay chains built inside the single-ended output buffer, as shown in Figure 7–7 on page 7–13. The delay chains can independently control the rising and falling edge delays of the output buffer, providing the ability to adjust the output-buffer duty cycle, compensate channel-to-channel skew, reduce simultaneous switching output (SSO) noise by deliberately introducing channel-to-channel skew, and improve high-speed memory-interface timing margins. Stratix III devices support four levels of output buffer delay settings. The default setting is **No Delay**.

For the programmable output buffer delay specifications, refer to the *DC and Switching Characteristics of Stratix III Devices* chapter.

Open-Drain Output

Stratix III devices provide an optional open-drain output (equivalent to an open-collector output) for each I/O pin. When configured as open-drain, the logic value of the output is either `high-Z` or `0`. Typically, an external pull-up resistor is required to provide logic high.

Bus Hold

Each Stratix III device I/O pin provides an optional bus-hold feature. The bus-hold circuitry can weakly hold the signal on an I/O pin at its last-driven state. Because the bus-hold feature holds the last-driven state of the pin until the next input signal is present, you do not need an external pull-up or pull-down resistor to hold a signal level when the bus is tri-stated.

The bus-hold circuitry also pulls non-driven pins away from the input threshold voltage where noise can cause unintended high-frequency switching. You can select this feature individually for each I/O pin. The bus-hold output drives no higher than V_{CCIO} to prevent over-driving signals. If the bus-hold feature is enabled, the programmable pull-up option cannot be used. Disable the bus-hold feature if the I/O pin is configured for differential signals.

The bus-hold circuitry uses a resistor with a nominal resistance (R_{BH}) of approximately 7 kΩ to weakly pull the signal level to the last-driven state.

For the specific sustaining current driven through this resistor and the overdrive current used to identify the next-driven input level, refer to the *DC and Switching Characteristics of Stratix III Devices* chapter. This information is provided for each V_{CCIO} voltage level.

The bus-hold circuitry is active only after configuration. When going into user mode, the bus-hold circuit captures the value on the pin present at the end of configuration.

Programmable Pull-Up Resistor

Each Stratix III device I/O pin provides an optional programmable pull-up resistor during user mode. If you enable this feature for an I/O pin, the pull-up resistor (typically 25 kΩ) weakly holds the I/O to the V_{CCIO} level.

Programmable pull-up resistors are only supported on user I/O pins and are not supported on dedicated configuration pins, JTAG pins, or dedicated clock pins. If the programmable pull-up option is enabled, you cannot use the bus-hold feature.

When the optional `DEV_OE` signal drives low, all I/O pins remain tri-stated even with programmable pull-up option enabled.

Programmable Pre-Emphasis

Stratix III LVDS transmitters support programmable pre-emphasis to compensate for the frequency dependent attenuation of the transmission line. The Quartus II software allows four settings for programmable pre-emphasis—zero, low, medium, and high. The default setting is low.

For more information about programmable pre-emphasis, refer to the *High-Speed Differential I/O Interfaces with DPA* in the *Stratix III Devices* chapter.

Programmable Differential Output Voltage

Stratix III LVDS transmitters support programmable V_{OD}. The programmable V_{OD} settings enable you to adjust output eye height to optimize for trace length and power consumption. A higher V_{OD} swing improves voltage margins at the receiver end while a smaller V_{OD} swing reduces power consumption. The Quartus II software allows four settings for programmable V_{OD}—low, medium low, medium high, and high. The default setting is medium low.

For more information about programmable V_{OD}, refer to the *High Speed Differential I/O Interfaces with DPA in the Stratix III Devices* chapter.

MultiVolt I/O Interface

The Stratix III architecture supports the MultiVolt™ I/O interface feature that allows Stratix III devices in all packages to interface with systems of different supply voltages.

You can connect the V_{CCIO} pins to a 1.2-, 1.5-, 1.8-, 2.5-, 3.0 or 3.3-V power supply, depending on the output requirements. The output levels are compatible with systems of the same voltage as the power supply. (For example, when V_{CCIO} pins are connected to a 1.5-V power supply, the output levels are compatible with 1.5-V systems.)

You must connect the Stratix III `VCCPD` power pins to a 2.5-, 3.0 or 3.3-V power supply. Using these power pins to supply the pre-driver power to the output buffers increases the performance of the output pins. Table 7–7 summarizes Stratix III MultiVolt I/O support.

For V_{CCIO} = 3.3 V, V_{CCPD}=3.3 V. For V_{CCIO} = 3.0 V, V_{CCPD} = 3.0 V. For V_{CCIO} = 2.5 V or less, V_{CCPD} = 2.5 V.

Table 7–7. MultiVolt I/O Support for Stratix III Devices *(Note 1)*, *(2)*

	Input Signal (V)						Output Signal (V)					
V_{CCIO} (V)	1.2	1.5	1.8	2.5	3.0	3.3	1.2	1.5	1.8	2.5	3.0	3.3
1.2	✓	—	—	—	—	—	✓	—	—	—	—	—
1.5	—	✓	✓ *(1)*	—	—	—	—	✓	—	—	—	—
1.8	—	✓	✓	—	—	—	—	—	✓	—	—	—
2.5	—	—	—	✓	✓ *(2)*	✓ *(2)*	—	—	—	✓	—	—
3.0	—	—	—	✓	✓	✓	—	—	—	—	✓	—
3.3	—	—	—	✓	✓	✓	—	—	—	—	—	✓

Notes to Table 7–7:

(1) The pin current may be slightly higher than the default value. You must verify that the driving device's V_{OL} maximum and V_{OH} minimum voltages do not violate the applicable Stratix III V_{IL} maximum and V_{IH} minimum voltage specifications.

(2) Use on-chip PCI clamp diode for column I/Os or external PCI clamp diode for row I/Os to protect the input pins against overshoot voltage.

(3) Each I/O bank of a Stratix III device has its own `VCCIO` pins and supports only one V_{ccio}, either 1.2, 1.5, 1.8, or 3.0 V. The LVDS I/O standard requires that a V_{CCIO} of 2.5 V cannot be assigned in a same bank with a 3.0-V or 3.3-V output signal.

OCT Support

Stratix III devices feature dynamic series and parallel on-chip termination to provide I/O impedance matching and termination capabilities. OCT improves signal quality over external termination by reducing parasitic, saving board space, and reducing external component costs.

Stratix III devices support OCT R_S with or without calibration, OCT R_T with calibration, and dynamic series and parallel termination for single-ended I/O standards as well as OCT R_D for differential LVDS I/O standards. Stratix III devices support OCT in all I/O banks by selecting one of the OCT I/O standards.

Stratix III devices support OCT R_S and R_T in the same I/O bank for different I/O standards if they use the same V_{CCIO} supply voltage. Each I/O in an I/O bank can be independently configured to support OCT R_S, programmable current strength, or OCT R_T.

You cannot configure both OCT R_S and programmable current strength for the same I/O buffer.

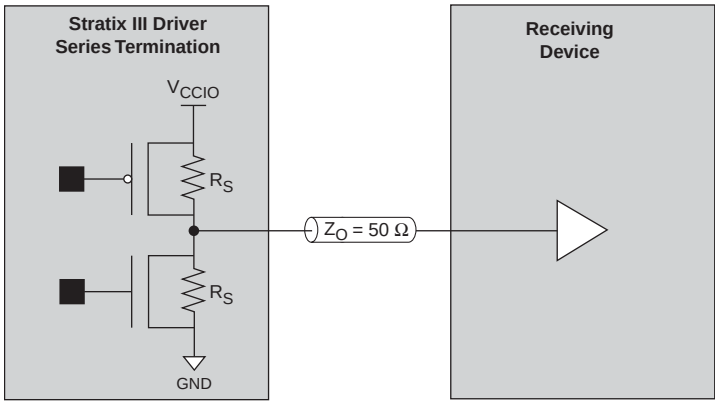
A pair of RUP and RDN pins are available in a given I/O bank, and are shared for series- and parallel-calibrated termination. The RUP and RDN pins share the same V_{CCIO} and GND, respectively, with the I/O bank where they are located. The RUP and RDN pins are dual-purpose I/Os, and function as regular I/Os if you do not use the calibration circuit. When used for calibration, the RUP pin is connected to V_{CCIO} through an external 25-Ω ±1% or 50-Ω ±1% resistor for an OCT R_S value of 25 Ω or 50 Ω, respectively; the RDN pin is connected to GND through an external 25-Ω ±1% or 50-Ω ±1% resistor for an OCT R_S value of 25 Ω or 50 Ω, respectively. For OCT R_T, the RUP pin is connected to V_{CCIO} through an external 50-Ω ±1% resistor; the RDN pin is connected to GND through an external 50-Ω ±1% resistor.

On-Chip Series Termination without Calibration

Stratix III devices support driver-impedance matching to provide the I/O driver with controlled output impedance that closely matches the impedance of the transmission line. As a result, you can significantly reduce reflections. Stratix III devices support OCT R_S for single-ended I/O standards (see Figure 7–8).

The R_S shown in Figure 7–8 is the intrinsic impedance of the output transistors. The typical R_S values are 25 Ω and 50 Ω. When matching impedance is selected, current strength is no longer selectable.

Figure 7–8. On-Chip Series Termination without Calibration for Stratix III Devices



To use OCT for the SSTL Class I standard, you should select the **50-Ω on-chip series termination** setting, eliminating the external 25-Ω R_S (to match the 50-Ω transmission line). For the SSTL Class II standard, you should select the **25-Ω on-chip series termination** setting (to match the 50-Ω transmission line and the near-end external 50-Ω pull-up to V_{TT}).

On-Chip Series Termination with Calibration

Stratix III devices support OCT R_S with calibration in all banks. The OCT R_S calibration circuit compares the total impedance of the I/O buffer to the external 25-Ω ±1% or 50-Ω ±1% resistors connected to the RUP and RDN pins, and dynamically enables or disables the transistors until they match. The R_S shown in Figure 7–9 is the intrinsic impedance of transistors. Calibration occurs at the end of device configuration. When the calibration circuit finds the correct impedance, it powers down and stops changing the characteristics of the drivers. When calibration is not taking place, the RUP and RDN pins go to a tri-state condition.

Figure 7–9. On-Chip Series Termination with Calibration for Stratix III Devices

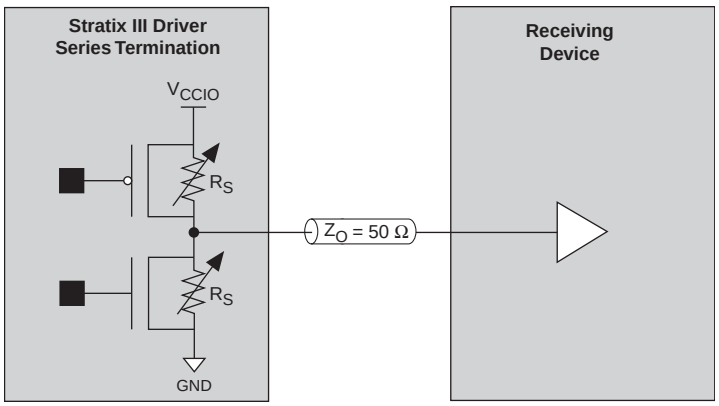


Table 7–8 lists I/O standards that support OCT R_S with calibration.

Table 7–8. Selectable I/O Standards with On-Chip Series Termination With or Without Calibration

I/O Standard	On-Chip Series Termination Setting		
	Row I/O	Column I/O	Unit
3.3-V LVTTL/LVCMOS	50	50	Ω
	25	25	Ω
3.0-V LVTTL/LVCMOS	50	50	Ω
	25	25	Ω
2.5-V LVTTL/LVCMOS	50	50	Ω
	25	25	Ω
1.8-V LVTTL/LVCMOS	50	50	Ω
	25	25	Ω
1.5-V LVTTL/LVCMOS	50	50	Ω
		25	Ω
1.2-V LVTTL/LVCMOS	50	50	Ω
		25	Ω
SSTL-2 Class I	50	50	Ω
SSTL-2 Class II	25	25	Ω
SSTL-18 Class I	50	50	Ω
SSTL-18 Class II	25	25	Ω
SSTL-15 Class I	50	50	Ω
SSTL-15 Class II	—	25	Ω
HSTL-18 Class I	50	50	Ω
HSTL-18 Class II	25	25	Ω
HSTL-15 Class I	50	50	Ω
HSTL-15 Class II	—	25	Ω
HSTL-12 Class I	50	50	Ω
HSTL-12 Class II	—	25	Ω

Expanded On-Chip Series Termination with Calibration

OCT calibration circuits always adjust OCT R_S to match the external resistors connected to the RUP and RDN pins, it is possible to achieve different OCT R_S values besides the 25- and 50-Ω resistors. Theoretically you can always change the resistance connected to the RUP and RDN pins accordingly if you require a different OCT R_S value. Practically, the OCT R_S range, which Stratix III devices can support, is limited due to the output buffer size and granularity limitations. Table 7–9 shows expanded OCT R_S with calibration supported in Stratix III devices.The Quartus II software only allows discrete OCT R_S calibration settings of 25 Ω, 40 Ω, 50 Ω, and 60 Ω You can select the closest discrete value of OCT R_S with calibration settings in the Quartus II software to your system to get the closest timing and IBIS model information. For example, if you use 20-Ω OCT R_S with calibration in your system, you can select 25-Ω OCT R_S with calibration setting in the Quartus II software to get the closest timing and IBIS model information.

Table 7–9. Selectable I/O Standards with Expanded On-Chip Series Termination with Calibration Range

I/O Standard	Expanded OCT R_s range		
	Row I/O	Column I/O	Unit
3.3-V LVTTL/LVCMOS	20–60	20–60	Ω
3.0-V LVTTL/LVCMOS	20–60	20–60	Ω
2.5-V LVTTL/LVCMOS	20–60	20–60	Ω
1.8-V LVTTL/LVCMOS	20–60	20–60	Ω
1.5-V LVTTL/LVCMOS	40–60	20–60	Ω
1.2-V LVTTL/LVCMOS	40–60	20–60	Ω
SSTL-2	20–60	20–60	Ω
SSTL-18	20–60	20–60	Ω
SSTL-15	40–60	20–60	Ω
HSTL-18	20–60	20–60	Ω
HSTL-15	40–60	20–60	Ω
HSTL-12	40–60	20–60	Ω

Note to Table 7–9:

(1) The expanded On-Chip Series Termination with calibration of SSTL and HSTL is for impedance matching to improve signal integrity and not for meeting JEDEC standard.

Left Shift Series Termination Control

Stratix III devices support left shift series termination control. You can use the left shift series termination control to get the calibrated OCT R_S with half of the impedance value of the external reference resistors connected to RUP and RDN pins. This feature is useful in applications which require both 25-Ω and 50-Ω calibrated OCT R_S at the same V_{CCIO}. For example, if your applications require 25-Ω and 50-Ω calibrated OCT R_S for SSTL–2 Class I and Class II I/O standards, you would only require one OCT calibration block with 50-Ω external reference resistors. You can enable this feature in the ALTIOBUF megafunction in the Quartus II software. The Quartus II software only allows the left shift series termination control for 25-Ω calibrated OCT R_S with 50-Ω external reference resistors connected to RUP and RDN pins. You can only use left shift series termination control for I/O standards that support 25 Ω-calibrated OCT R_S.

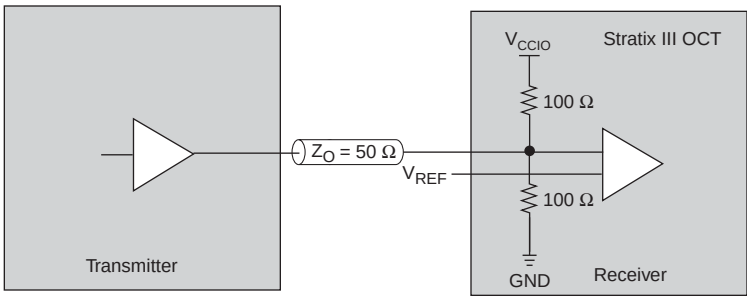
Left shift series termination control is automatically enabled if you use a bidirectional I/O with 25- Ω calibrated OCT R_S and 50- Ω parallel OCT.

For more information about how to enable left shift series termination in the ALTIOBUF megafunction, refer to the *ALTIOBUF Megafunction User Guide*.

On-Chip Parallel Termination with Calibration

Stratix III devices support OCT R_T with calibration in all banks. OCT R_T with calibration is only supported for input or bi-directional pin configurations. For input pins, you can enable OCT R_T continuously. However, for bi-directional I/O, OCT R_T is enabled or disabled depending on whether or not the bi-directional I/O acts as a transmitter or receiver. Output pin configurations do not support OCT R_T with calibration. Figure 7–10 shows OCT R_T with calibration. When OCT R_T is used, the V_{CCIO} of the bank has to match the I/O standard of the pin where the parallel OCT is enabled.

Figure 7–10. On-Chip Parallel Termination with Calibration for Stratix III Devices



The OCT R_T calibration circuit compares the total impedance of the I/O buffer to the external 50-Ω ±1% resistors connected to the RUP and RDN pins and dynamically enables or disables the transistors until they match. Calibration occurs at the end of device configuration. When the calibration circuit finds the correct impedance, it powers down and stops changing the characteristics of the drivers. Table 7–10 lists the I/O standards that support OCT R_T with calibration.

Table 7–10. Selectable I/O Standards that Support On-Chip Parallel Termination with Calibration

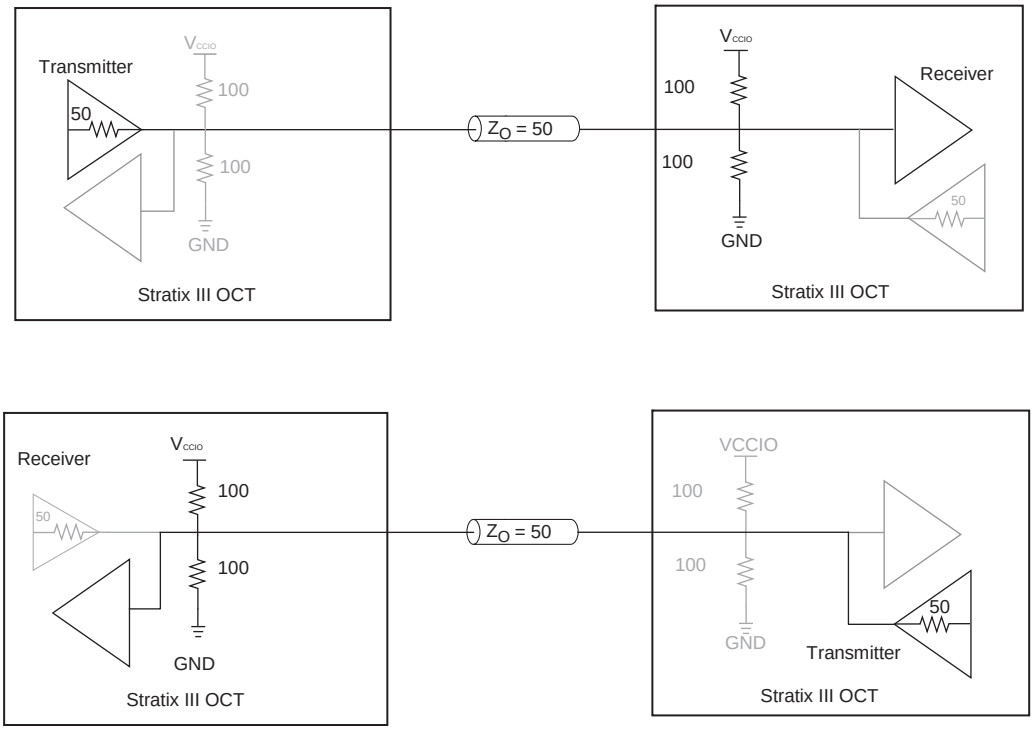
I/O Standard	On-Chip Parallel Termination Setting (Column I/O)	On-Chip Parallel Termination Setting (Row I/O)	Unit
SSTL-2 Class I, II	50	50	Ω
SSTL-18 Class I, II	50	50	Ω
SSTL-15 Class I, II	50	50	Ω
HSTL-18 Class I, II	50	50	Ω
HSTL-15 Class I, II	50	50	Ω
HSTL-12 Class I, II	50	50	Ω
Differential SSTL-2 Class I, II	50	50	Ω
Differential SSTL-18 Class I, II	50	50	Ω
Differential SSTL-15 Class I, II	50	50	Ω
Differential HSTL-18 Class I, II	50	50	Ω
Differential HSTL-15 Class I, II	50	50	Ω
Differential HSTL-12 Class I, II	50	50	Ω

Dynamic OCT

Stratix III devices support on-off dynamic series and parallel termination for a bi-directional I/O in all I/O banks. Figure 7–11 shows the termination schemes supported in the Stratix III device. Dynamic parallel termination is enabled only when the bi-directional I/O acts as a receiver and is disabled when it acts as a driver. Similarly, dynamic series termination is enabled only when the bi-directional I/O acts as a driver and is disabled when it acts as a receiver. This feature is useful for terminating any high-performance bi-directional path because the signal integrity is optimized depending on the direction of the data.

You should connect a bi-directional pin that uses both 25-Ω or 50-Ω series termination and 50-Ω input termination to a calibration block that has a 50-Ω external resistor connected to its RUP and RDN pins. The 25-Ω series termination on the bi-directional pin is achieved through internal divide by two circuits.

Figure 7–11. Dynamic Parallel OCT in Stratix III Devices

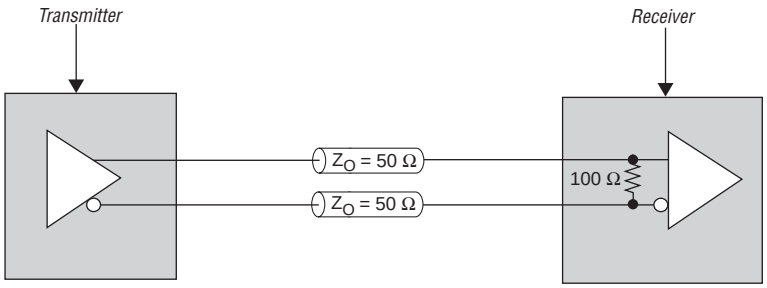


For more information about tolerance specifications for OCT with calibration, refer to the *DC and Switching Characteristics of Stratix III Devices* chapter.

LVDS Input On-Chip Termination (R_D)

Stratix III devices support OCT for differential LVDS input buffers with a nominal resistance value of 10 Ω, as shown in Figure 7–12. You can enable OCT R_D in row I/O banks when V_{CCIO} and V_{CCPD} are set to 2.5 V. The column I/O banks do not support OCT R_D. The dedicated clock input pairs `CLK[1,3,8,10][p,n]`, `PLL_L[1,4]_CLK[p,n]`, and `PLL_R[1,4]_CLK[p,n]` on the row I/O banks of the Stratix III devices do not support OCT R_D. Dedicated clock input pairs `CLK[0,2,9,11][p,n]` on row I/O banks support OCT R_D. Dedicated clock input pairs `CLK[4,5,6,7][p,n]` and `CLK[12,13,14,15][p,n]` on column I/O banks do not support OCT R_D.

Figure 7–12. Differential Input On-Chip Termination



For more information about OCT R_D, refer to the *High Speed Differential I/O Interfaces with DPA in Stratix III Devices* chapter.

Table 7–11 lists the assignment name and its value for OCT R_D in the Quartus II software Assignment Editor.

You must set the V_{CCIO} to 2.5 V when OCT R_D is used for the LVDS input buffer, even if the LVDS input buffer is powered by V_{CCPD}.

Table 7–11. On-Chip Differential Termination in Quartus II Software Assignment Editor

Assignment Name	Allowed Values	Applies To
Input Termination (Accepts wildcards/groups)	Parallel 50 Ω with calibration	Input buffers for single-ended and differential-HSTL/SSTL standards
	Differential	Input buffers for LVDS receivers on row I/O banks.
Output Termination	Series 25 Ω without calibration	Output buffers for single-ended LVTTL/LVCMOS and HSTL/SSTL standards as well as differential HSTL/SSTL standards.
	Series 50 Ω without calibration	
	Series 25 Ω with calibration	
	Series 40 Ω with calibration	
	Series 50 Ω with calibration	
	Series 60 Ω with calibration	

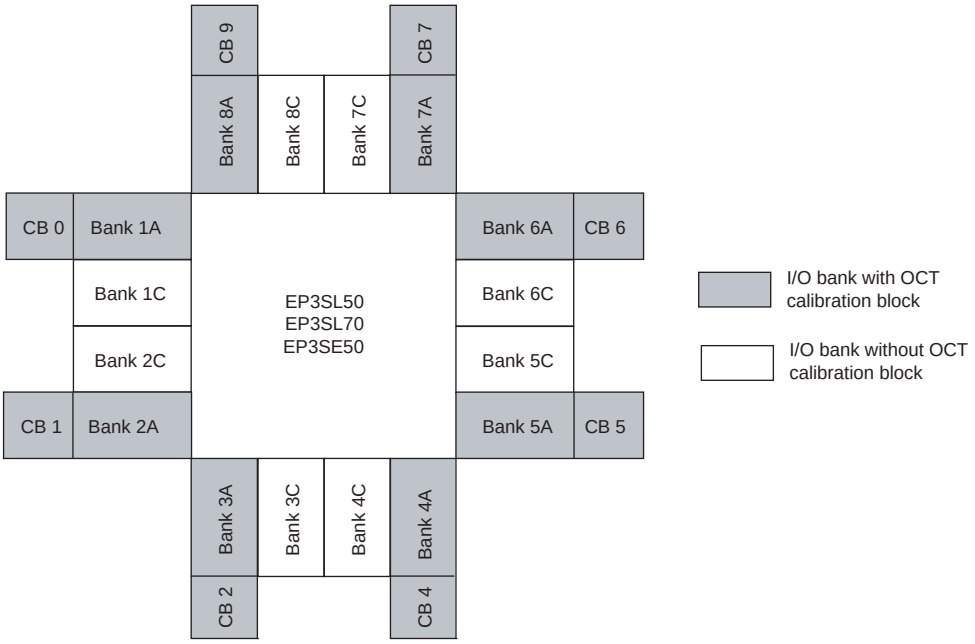
OCT Calibration

Stratix III devices support calibrated OCT R_S and calibrated OCT R_T on all I/O pins. You can calibrate the Stratix III I/O bank with any of eight OCT calibration blocks in EP3SL50, EP3SL70, EP3SL110, EP3SL150, EP3SE50, EP3SE80, and EP3SE110 devices and ten OCT calibration blocks in EP3SL200, EP3SE260, and EP3SL340 devices.

OCT Calibration Block Location

Figure 7–13, Figure 7–14, and Figure 7–15 show the location of OCT calibration blocks in Stratix III devices.

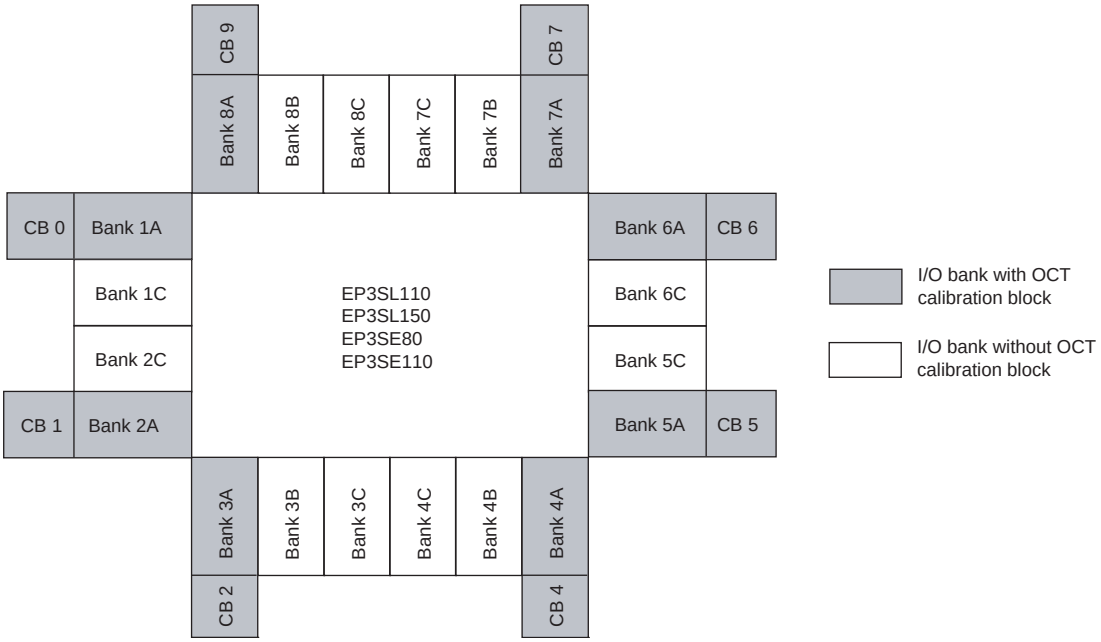
Figure 7–13. OCT Calibration Block (CB) Location in EP3SL50, EP3SL70, and EP3SE50 Devices *(Note 1)*



Note to Figure 7–13:

(1) Figure 7–13 is a top view of the silicon die that corresponds to a reverse view for flip chip packages. It is a graphical representation only.

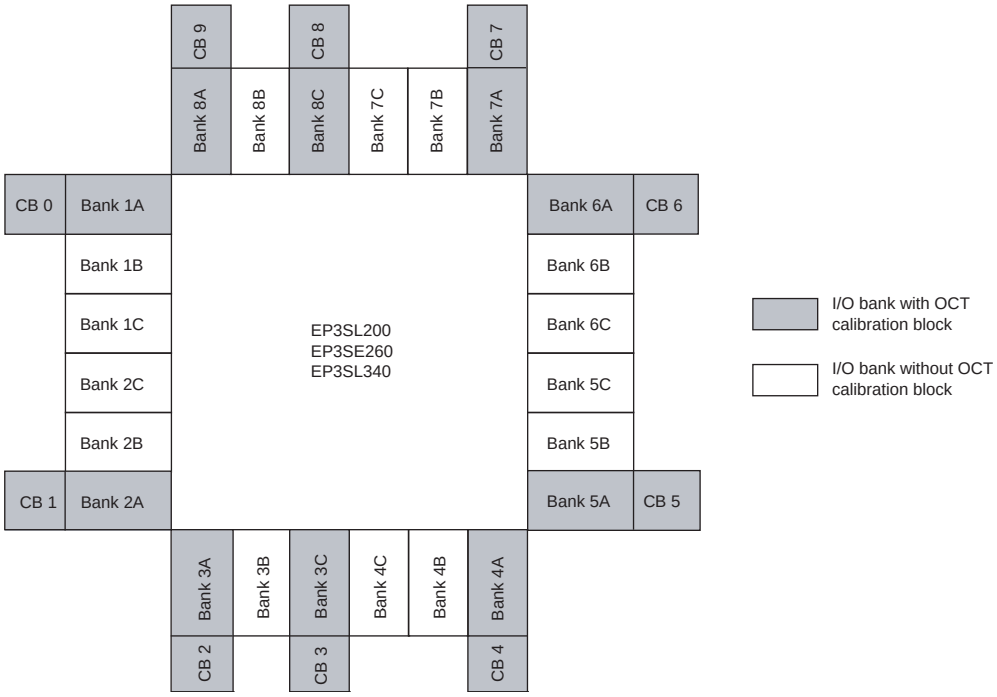
Figure 7–14. OCT Calibration Block (CB) Location in EP3SL110, EP3SL150, EP3SE80, and EP3SE110 Devices *(Note 1)*



Note to Figure 7–14:

(1) Figure 7–14 is a top view of the silicon die that corresponds to a reverse view for flip chip packages. It is a graphical representation only.

Figure 7–15. OCT Calibration Block (CB) Location in EP3SL200, EP3SE260 and EP3SL340 *(Note 1)*



Note to Figure 7–15:

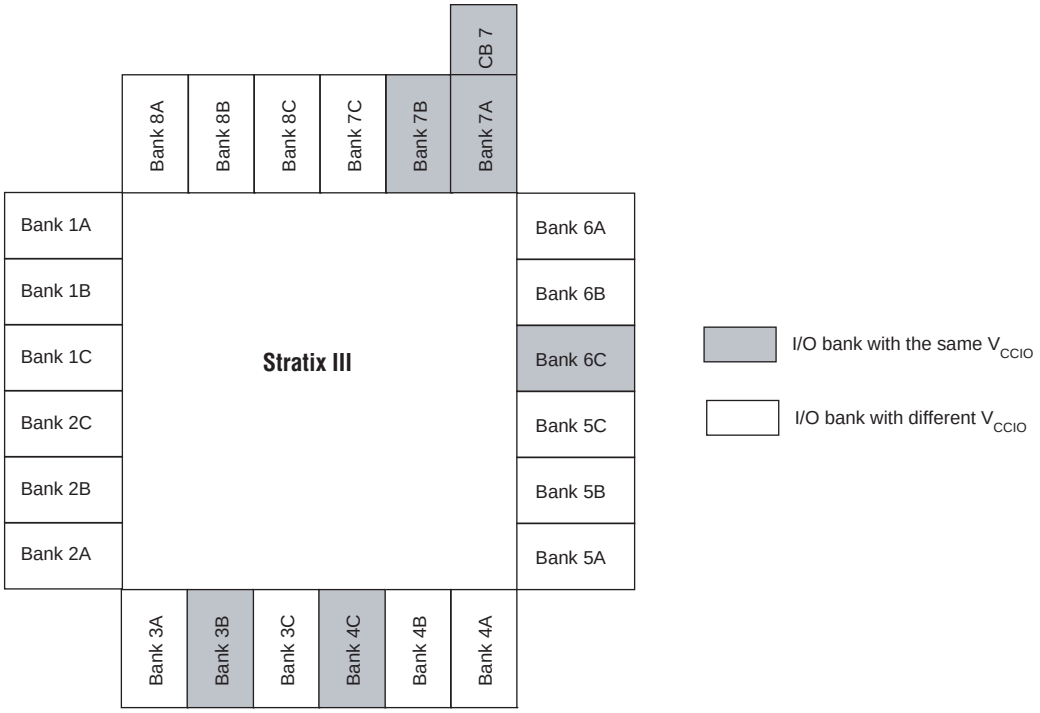
(1) Figure 7–15 is a top view of the silicon die that corresponds to a reverse view for flip chip packages. It is a graphical representation only.

Sharing an OCT Calibration Block in Multiple I/O Banks

An OCT calibration block has the same V_{CCIO} as the I/O bank that contains the block. OCT R_S calibration is supported on all I/O banks with different V_{CCIO} voltage standards, up to the number of available OCT calibration blocks. You can configure I/O banks to receive calibrated codes from any OCT calibration block with the same V_{CCIO}. All I/O banks with the same V_{CCIO} can share one OCT calibration block, even if that particular I/O bank has an OCT calibration block.

For example, Figure 7–16 shows a group of I/O banks that have the same V_{CCIO} voltage. If a group of I/O banks have the same V_{CCIO} voltage, you can use one OCT calibration block to calibrate the group of I/O banks placed around the periphery. Because 3B, 4C, 6C, and 7B have the same V_{CCIO} as bank 7A, you can calibrate all four I/O banks (3B, 4C, 6C, and 7B) with the OCT calibration block located in bank 7A. You can enable this by serially shifting out OCT R_S calibration codes from the OCT calibration block located in bank 7A to the I/O banks located around the periphery.

Figure 7–16. Example of Sharing Multiple I/O Banks with One OCT Calibration Block *(Note 1)*



Note to Figure 7–16:

(1) Figure 7–16 is a top view of the silicon die that corresponds to a reverse view for flip chip packages. It is a graphical representation only.

OCT Calibration Block Modes of Operation

Stratix III devices support calibration OCT R_S and OCT R_T in all I/O banks. The calibration can occur in either power-up mode or user mode.

Power-Up Mode

In power-up mode, OCT calibration is automatically performed at power up and calibrated codes are shifted to selected I/O buffers before transitioning to user mode.

User Mode

During user mode, OCTUSRCLK, ENAOCT, nCLRUSR, and ENASER[9..0] signals are used to calibrate and serially transfer calibrated codes from each OCT calibration block to any I/O. Table 7–12 lists the user controlled calibration block signal names and their descriptions.

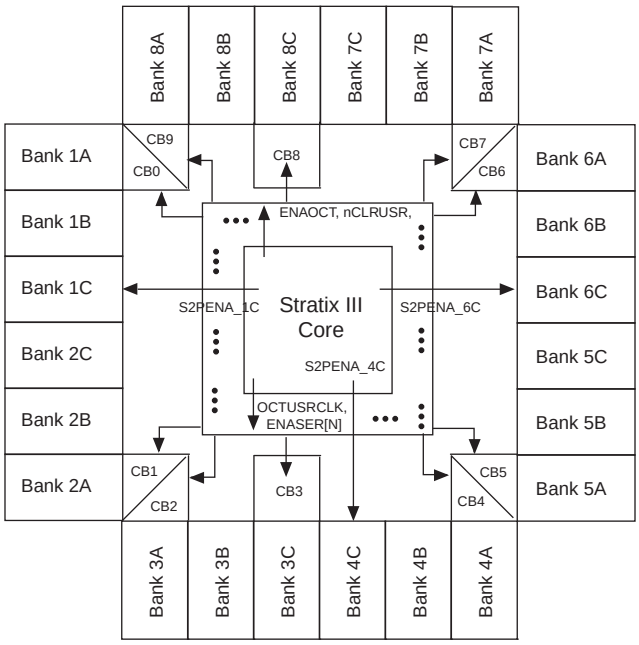
Table 7–12. OCT Calibration Block Ports for User Control and Description

Signal Name	Description
OCTUSRCLK	Clock for OCT block.
ENAOCT	Enable OCT Termination (generated by user IP).
ENASER[9..0]	When ENAOCT = 0, each signal enables the OCT serializer for the corresponding OCT calibration block. When ENAOCT = 1, each signal enables OCT calibration for the corresponding OCT calibration block.
S2PENA_<*bank#*>	Serial-to-parallel load enable per I/O bank.
nCLRUSR	Clear user.

Figure 7–17 shows the flow of the user signal. When ENAOCT is 1, all OCT calibration blocks are in calibration mode, and when ENAOCT is 0, all OCT calibration blocks are in serial data transfer mode. The OCTUSRCLK clock frequency must be 20 MHz or less.

You must generate all user signals on the rising edge of OCTUSRCLK.

Figure 7–17. Signals Used for User Mode Calibration *(Note 1)*



Note to Figure 7–17:

(1) Figure 7–17 is a top view of the silicon die that corresponds to a reverse view for flip chip packages. It is a graphical representation only.

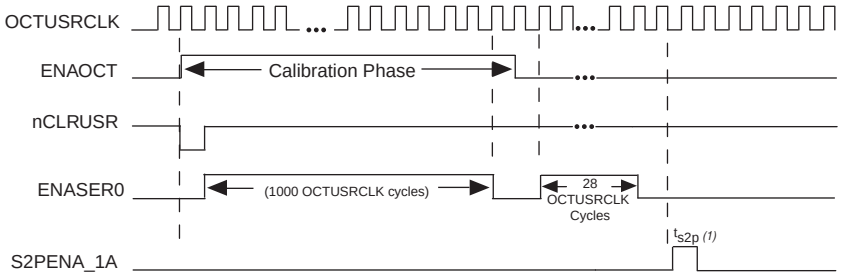
OCT Calibration

Figure 7–18 shows the user-mode signal-timing waveforms. To calibrate OCT block[N] (where N is a calibration block number), you must assert ENAOCT one cycle before asserting ENASER[N]. Also, nCLRUSR must be set to low for one OCTUSRCLK cycle before ENASER[N] signal is asserted. An asserted ENASER[N] signals for 1000 OCTUSRCLK cycles to perform $OCTR_S$ and $OCTR_T$ calibration. ENAOCT can be deasserted one clock cycle after the last ENASER is deasserted.

Serial Data Transfer

When calibration is complete, you must serially shift out the 28-bit OCT calibration code (14-bit OCT RS code and 14-bit OCT RT) from each OCT calibration block to the corresponding I/O buffers. Only one OCT calibration block can send out the codes at any given time by asserting only one ENASER[N] signal at a time. After ENAOCT is deasserted, you must wait at least 1 OCTUSRCLK cycle to enable any ENASER[N] signal to begin serial transfer. To shift 28-bit code from OCT calibration block[N], ENASER[N] must be asserted for exactly 28 OCTUSRCLK cycles. There must be at least one OCTUSRCLK cycle gap between two consecutive asserted ENASER signals. For these requirements, refer to Figure 7–18.

Figure 7–18. OCT User-Mode Signal Timing Waveform for One OCT Block



Note to Figure 7–18:

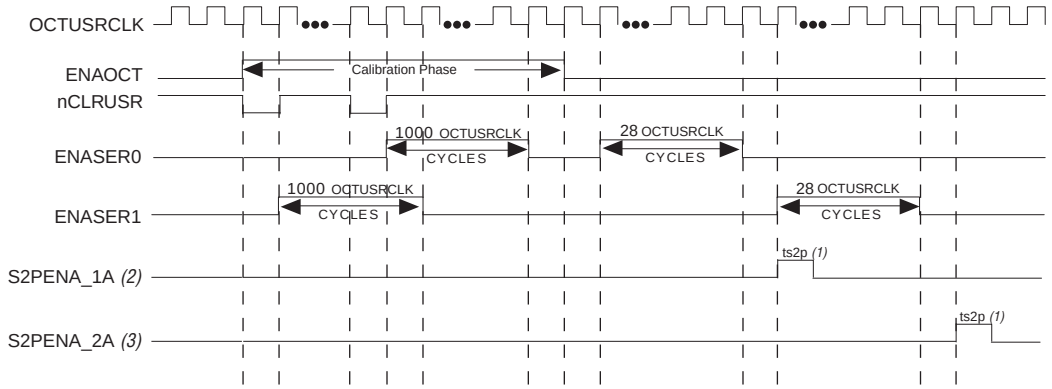
(1) ts2p ≥ 25 ns

After calibrated codes are shifted serially to the corresponding I/O buffers, they must be converted from serial format to parallel format before being used in the I/O buffers. Figure 7–18 shows S2PENA signals that can be asserted at any time to update the calibration codes in each I/O bank. All I/O banks that received the codes from the same OCT calibration block can have S2PENA asserted at the same time, or at a different time, even while another OCT calibration block is calibrating and serially shifting codes. The S2PENA signal is asserted one OCTUSRCLK cycle after ENASER is deasserted for at least 25 ns. You cannot use I/Os for transmitting or receiving data when their S2PENA is asserted for parallel codes transfer.

Example of Using Multiple OCT Calibration Blocks

Figure 7–19 shows a signal timing waveform for two OCT calibration blocks doing R_S and R_T calibration. Calibration blocks can start calibrating at different times by asserting ENASER signals at different times. ENAOCT must stay asserted while any calibration is ongoing. nCLRUSR must be set to low for one OCTUSRCLK cycle before each ENASER[N] signal is asserted. In Figure 7–19, when nCLRUSR is set to 0 for the second time to initialize OCT calibration block 0, this does not affect OCT calibration block 1, whose calibration is already in progress.

Figure 7–19. OCT User-Mode Signal Timing Waveform for Two OCT Blocks



Notes to Figure 7–19:

(1) ts2p ≥ 25 ns

(2) S2PENA_1A is asserted in Bank 1A for calibration block 0.

(3) S2PENA_2A is asserted in Bank 2A for calibration block 1.

RS Calibration

If only RS calibration is used for an OCT calibration block, its corresponding ENASER signal must be asserted for 240 OCTUSRCLK cycles for calibration.

You still have to assert the ENASER signal for 28 OCTUSRCLK cycles for serial transfer.

For more information, refer to the *ALT_OCT Megafunction User Guide* and *AN 465: Implementing OCT Calibration in Stratix III Devices*.

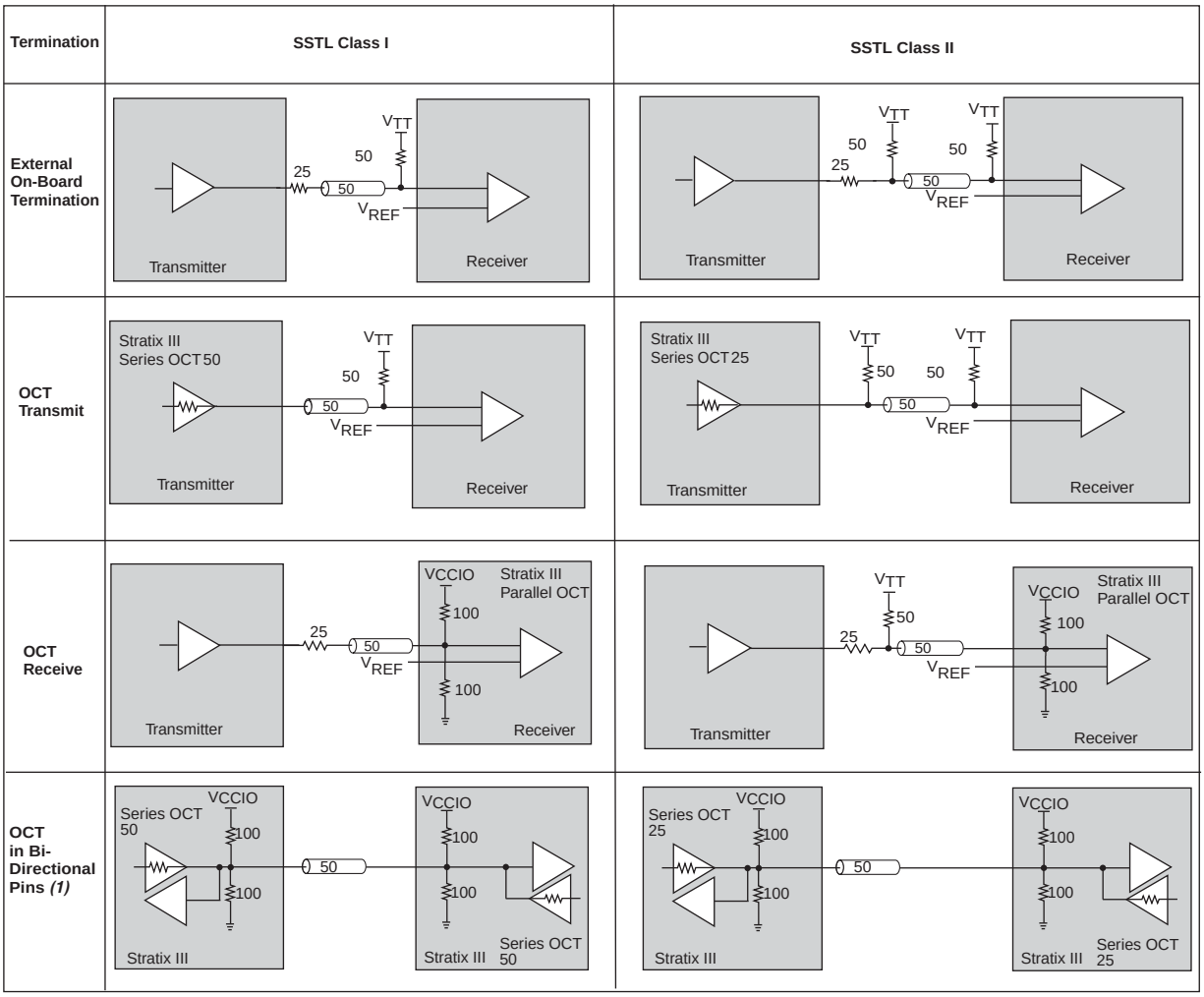
Termination Schemes for I/O Standards

The following section describes the different termination schemes for the I/O standards used in Stratix III devices.

Single-Ended I/O Standards Termination

Voltage-referenced I/O standards require both an input reference voltage, V_{REF}, and a termination voltage (V_{TT}). The reference voltage of the receiving device tracks the termination voltage of the transmitting device. Figure 7–20 and Figure 7–21 show the details of SSTL and HSTL I/O termination on Stratix III devices.

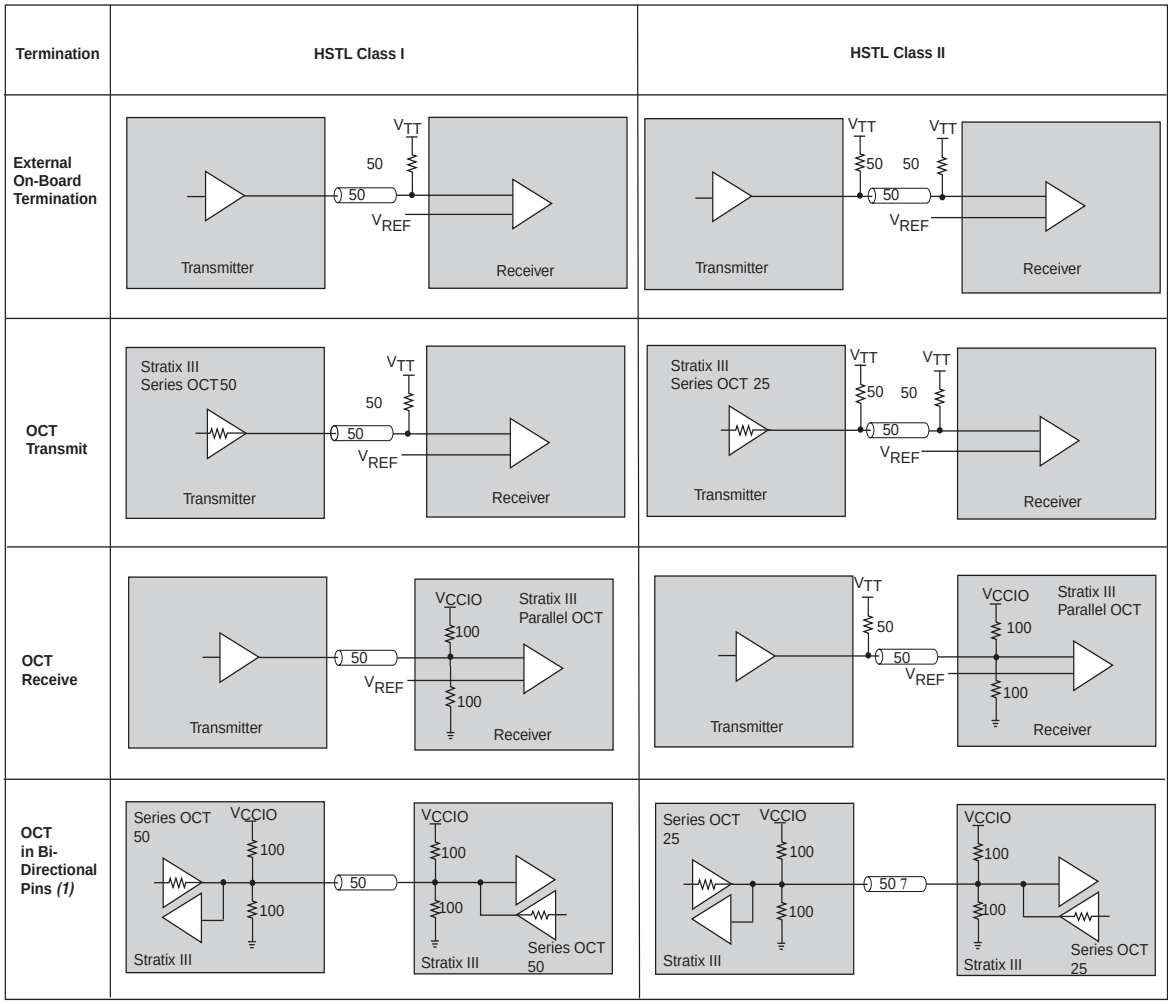
Figure 7–20. SSTL I/O Standard Termination for Stratix III Devices



Note to Figure 7–20:

(1) In Stratix III devices, series and parallel OCT cannot be used simultaneously. For more information, refer to "Dynamic OCT" on page 7–25.

Figure 7–21. HSTL I/O Standard Termination for Stratix III Devices



Note to Figure 7–21:

(1) In Stratix III devices, you cannot use simultaneously series and parallel OCT. For more information, refer to "Dynamic OCT" on page 7–25.

Differential I/O Standards Termination

Stratix III devices support differential SSTL-2 and SSTL-18, differential HSTL-18, HSTL-15, HSTL-12, LVDS, LVPECL, RSDS, and mini-LVDS. Figure 7–22 through Figure 7–28 show the details of various differential I/O termination on Stratix III devices.

☞ Differential HSTL and SSTL outputs are not true differential outputs. They use two single-ended outputs with the second output programmed as inverted.

Figure 7–22. Differential SSTL I/O Standard Termination for Stratix III Devices

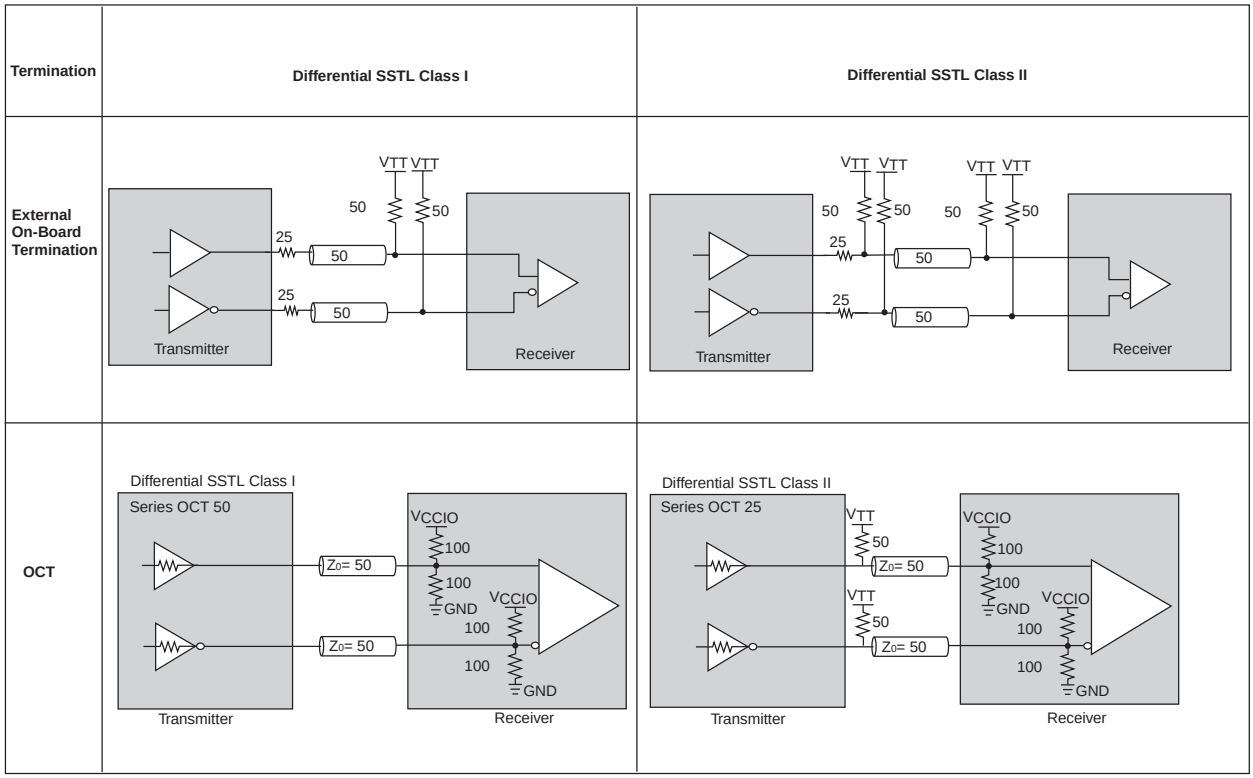
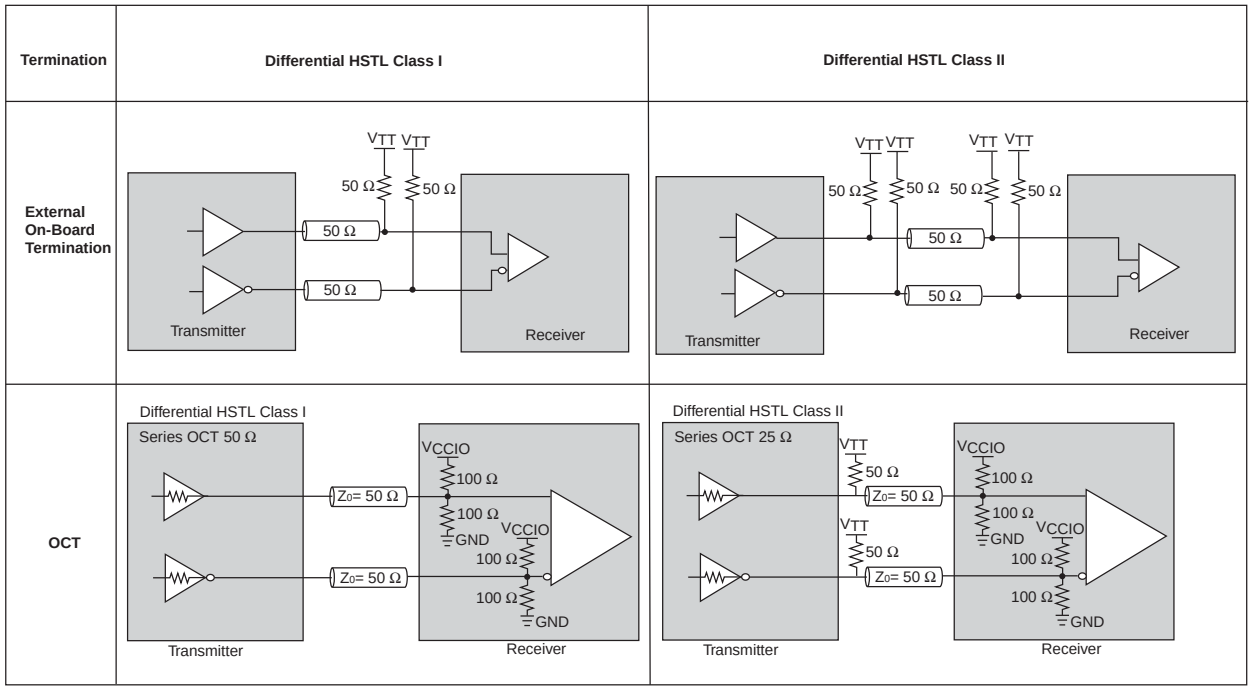


Figure 7–23. Differential HSTL I/O Standard Termination for Stratix III Devices

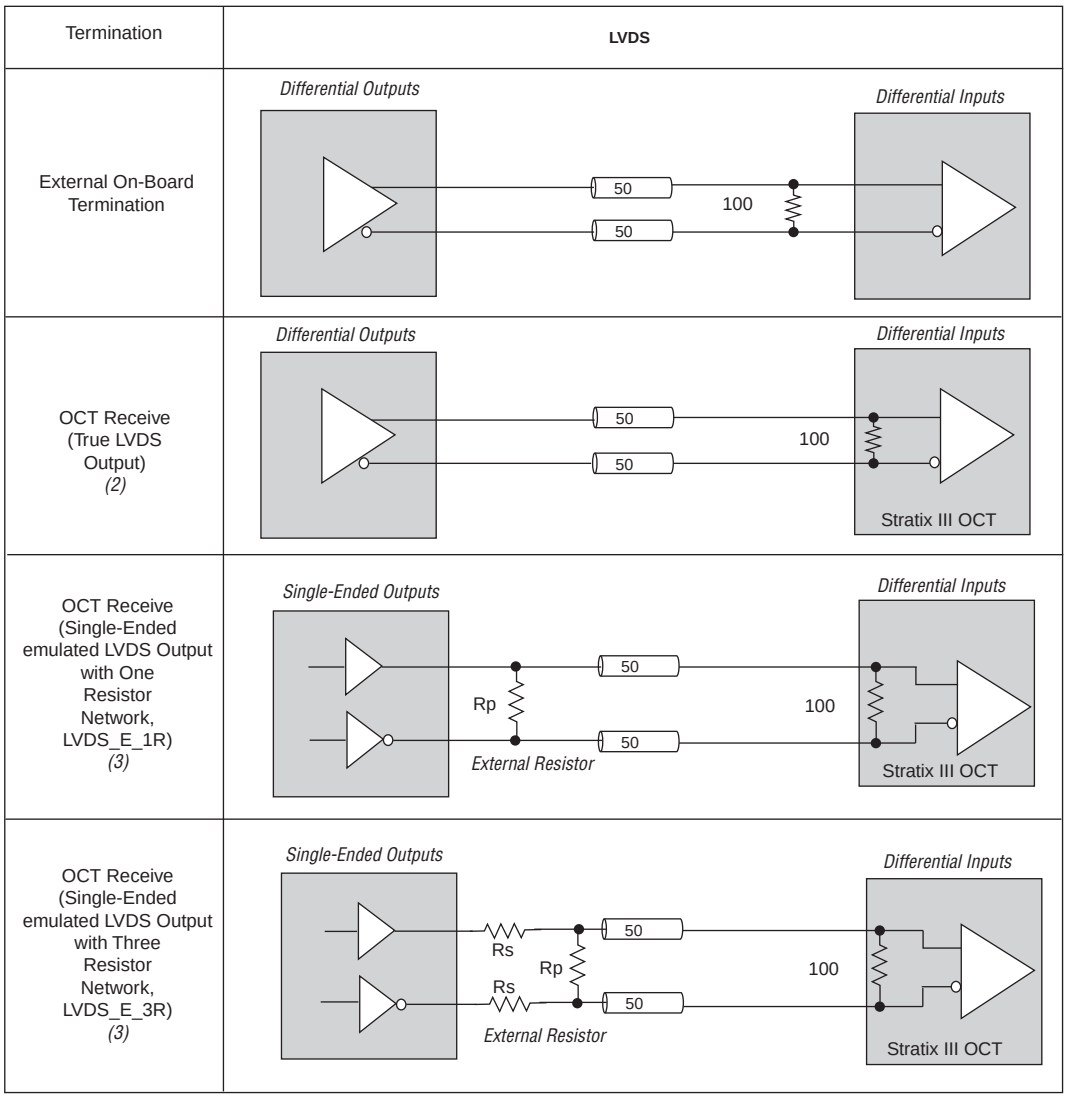


LVDS

The LVDS I/O standard is a differential high-speed, low-voltage swing, low-power, general-purpose I/O interface standard. In Stratix III devices, the LVDS I/O standard requires a 2.5-V V_{CCIO} level. The LVDS input buffer requires 2.5-V V_{CCPD}. Use this standard in applications requiring high-bandwidth data transfer, backplane drivers, and clock distribution. LVDS requires a 100-Ω termination resistor between the two signals at the input buffer. Stratix III devices provide an optional 100-Ω differential termination resistor in the device using on-chip differential termination.

Figure 7–24 shows the details of LVDS termination. The OCT R_D is only available in row I/O banks.

Figure 7–24. LVDS I/O Standard Termination for Stratix III Devices *(Note 1)*



Notes to Figure 7–24:

(1) R_P=120 Ω for LVDS_E_1R, R_P=170 Ω and R_S=120 Ω for LVDS_E_3R.

(2) Row I/O banks support true LVDS output buffers.

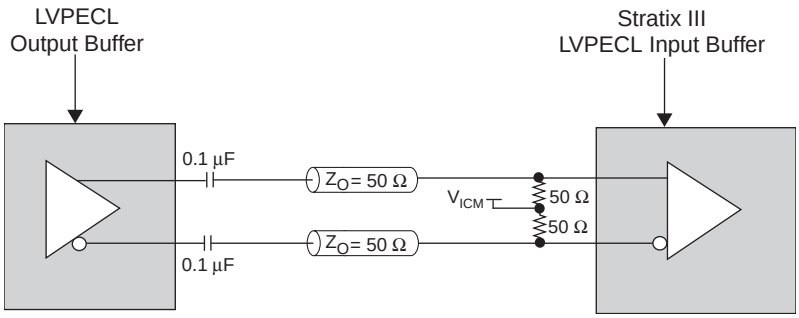
(3) Column and row I/O banks support LVDS_E_1R and LVDS_E_3R I/O standards using two single-ended output buffers.

Differential LVPECL

In Stratix III devices, the LVPECL I/O standard is supported on input clock pins on column and row I/O banks. LVPECL output operation is not supported by Stratix III devices. LVDS input buffers are used to support LVPECL input operation. AC coupling is required when LVPECL common mode voltage of the output buffer is higher than Stratix III LVPECL input common mode voltage. Figure 7–25 shows the AC coupled termination scheme. The 50-Ω resistors used at the receiver end are external to the device.

DC-coupled LVPECL is supported if the driving device's LVPECL output common mode voltage is within the Stratix III LVPECL input buffer specification (see Figure 7–26).

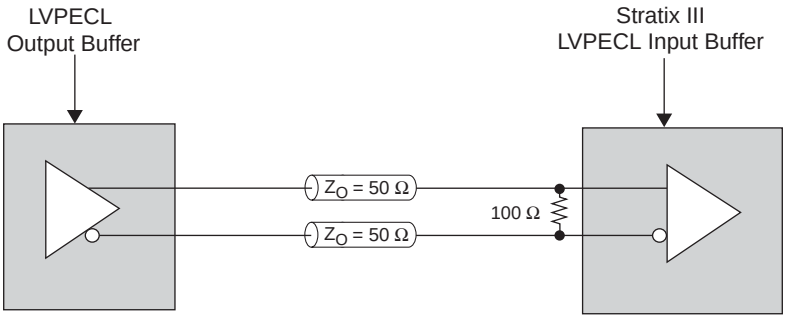
Figure 7–25. LVPECL AC Coupled Termination *(Note 1)*



Note to Figure 7–25:

(1) The LVPECL AC-coupled termination is applicable only when an Altera FPGA LVPECL transmitter is used.

Figure 7–26. LVPECL DC Coupled Termination *(Note 1)*



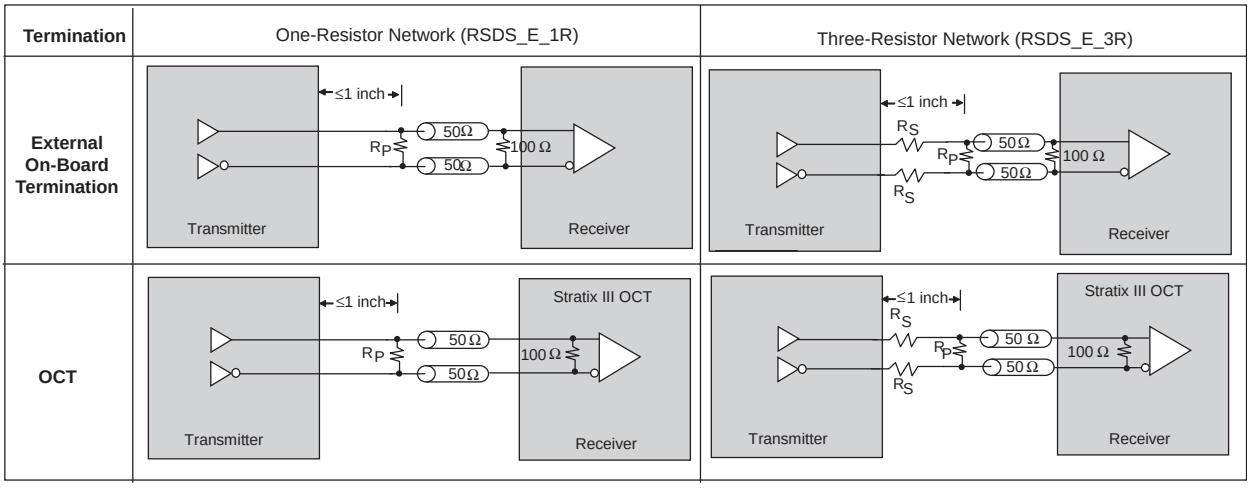
Note to Figure 7–26:

(1) The LVPECL DC-coupled termination is applicable only when an Altera FPGA LVPECL transmitter is used.

RSDS

The row I/O banks support RSDS output using true LVDS output buffers without an external resistor network. The column I/O banks support RSDS output using two single-ended output buffers with the external one- or three-resistor networks, as shown in Figure 7–27.

Figure 7–27. RSDS I/O Standard Termination for Stratix III Devices *(Note 1), (2)*



Notes to Figure 7–27:

(1) R_P=120 Ω for RSDS_E_1R, R_P=170 Ω and R_S=120 Ω for RSDS_E_3R.

(2) Column and row I/O banks support RSDS_E_1R and RSDS_E_3R I/O standards using two single-ended output buffers.

A resistor network is required to attenuate the LVDS output-voltage swing to meet the RSDS specifications. You can modify the three-resistor network values to reduce power or improve the noise margin. The resistor values chosen should satisfy Equation 7–1:

Equation 7–1.

$$\frac{Rs \times \frac{Rp}{2}}{Rs + \frac{Rp}{2}} = 50\Omega$$

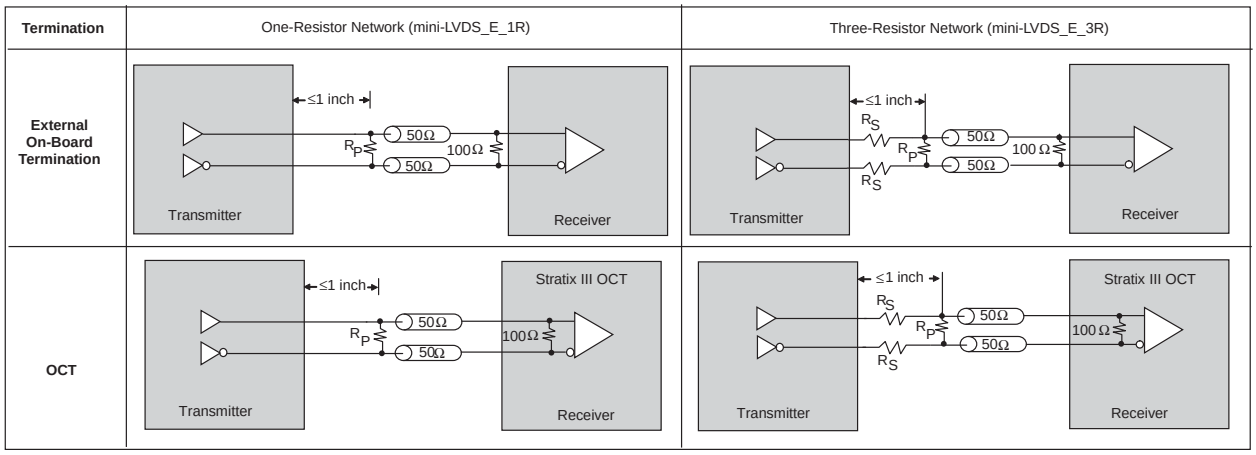
Altera recommends that you perform additional simulations using IBIS models to validate that custom resistor values meet the RSDS requirements.

For more information about the RSDS I/O standard, refer to the *RSDS Specification* from the National Semiconductor website.

Mini-LVDS

The row I/O banks support mini-LVDS output using true LVDS output buffers without an external resistor network. The column I/O banks support mini-LVDS output using two single-ended output buffers with the external one- or three-resistor network, as shown in Figure 7–28.

Figure 7–28. Mini-LVDS I/O Standard Termination for Stratix III Devices *(Note 1)*, *(2)*



Notes to Figure 7–28:

(1) R_P=120 Ω for mini-LVDS_E_1R, R_P=170 Ω, and R_S=120 Ω for mini-LVDS_E_3R.

(2) Column and row I/O banks support mini-LVDS_E_1R and mini-LVDS_E_3R I/O standards using two single-ended output buffers.

A resistor network is required to attenuate the LVDS output voltage swing to meet the mini-LVDS specifications. You can modify the three-resistor network values to reduce power or improve the noise margin. The resistor values chosen should satisfy Equation 7–2:

Equation 7–2.

$$\frac{Rs \times \frac{Rp}{2}}{Rs + \frac{Rp}{2}} = 50\Omega$$

Altera recommends that you perform additional simulations using IBIS models to validate that custom resistor values meet the RSDS requirements.

For more information about the mini-LVDS I/O standard, refer to the *mini-LVDS Specification* from the Texas Instruments website.

Design Considerations

While Stratix III devices feature various I/O capabilities for high-performance and high-speed system designs, there are several other considerations that require attention to ensure the success of those designs.

I/O Termination

I/O termination requirements for single-ended and differential I/O standards are discussed in this section.

Single-Ended I/O Standards

Although single-ended, non-voltage-referenced I/O standards do not require termination, impedance matching may be necessary to reduce reflections and improve signal integrity.

Voltage-referenced I/O standards require both an input reference voltage, $V_{REF,}$ and a termination voltage, V_{TT}. The reference voltage of the receiving device tracks the termination voltage of the transmitting device. Each voltage-referenced I/O standard requires a unique termination setup. For example, a proper resistive signal termination scheme is critical in SSTL2 standards to produce a reliable DDR memory system with superior noise margin.

Stratix III OCT R_S and OCT R_T provide the convenience of no external components. Alternatively, you can use external pull-up resistors to terminate the voltage-referenced I/O standards, such as SSTL and HSTL.

Differential I/O Standards

Differential I/O standards typically require a termination resistor between the two signals at the receiver. The termination resistor must match the differential load impedance of the signal line. Stratix III devices provide an optional differential on-chip resistor when using LVDS.

For PCB layout guidelines, refer to *AN 224: High-Speed Board Layout Guidelines* and *AN 315: Guidelines for Designing High-Speed FPGA PCBs*.

I/O Banks Restrictions

Each I/O bank can simultaneously support multiple I/O standards. The following sections provide guidelines for mixing non-voltage-referenced and voltage-referenced I/O standards in Stratix III devices.

Non-Voltage-Referenced Standards

Each Stratix III device I/O bank has its own V_{CCIO} pins and supports only one V_{CCIO}, either 1.2, 1.5, 1.8, 2.5, 3.0, or 3.3 V. An I/O bank can simultaneously support any number of input signals with different I/O standard assignments, as listed in Table 7–2.

For output signals, a single I/O bank supports non-voltage-referenced output signals that are driving at the same voltage as V_{CCIO}. Since an I/O bank can only have one V_{CCIO} value, it can only drive out that one value for non-voltage-referenced signals. For example, an I/O bank with a 2.5-V V_{CCIO} setting can support 2.5-V standard inputs and outputs and 3-V LVCMOS inputs (not output or bi-directional pins).

Voltage-Referenced Standards

To accommodate voltage-referenced I/O standards, each Stratix III device I/O bank has one VREF pin feeding a common V_{REF} bus. If it is not used as a VREF pin, it cannot be used as a generic I/O pin and should be tied to V_{CCIO} or GND. Each bank can only have a single V_{CCIO} voltage level and a single V_{REF} voltage level at a given time.

An I/O bank featuring single-ended or differential standards can support voltage-referenced standards as long as all voltage-referenced standards use the same V_{REF} setting.

For performance reasons, voltage-referenced input standards use their own V_{CCPD} level as the power source. This feature allows you to place voltage-referenced input signals in an I/O bank with a V_{CCIO} of 2.5 or below. For example, you can place HSTL-15 input pins in an I/O bank with a 2.5-V V_{CCIO}. However, voltage-referenced input with parallel OCT enabled requires the V_{CCIO} of the I/O bank to match the voltage of the input standard.

Voltage-referenced bi-directional and output signals must be the same as the I/O bank's V_{CCIO} voltage. For example, you can only place SSTL-2 output pins in an I/O bank with a 2.5-V V_{CCIO}.

Mixing Voltage-Referenced and Non-Voltage-Referenced Standards

An I/O bank can support both non-voltage-referenced and voltage-referenced pins by applying each of the rule sets individually. For example, an I/O bank can support SSTL-18 inputs and 1.8-V inputs and outputs with a 1.8-V V_{CCIO} and a 0.9-V V_{REF}. Similarly, an I/O bank can support 1.5-V standards, 1.8-V inputs (but not outputs), and HSTL and HSTL-15 I/O standards with a 1.5-V V_{CCIO} and 0.75-V V_{REF}.

For pin connection guidelines, refer to the *Stratix III Device Family Pin Connection Guidelines*.

Chapter Revision History

Table 7–13 lists the revision history for this chapter.

Table 7–13. Chapter Revision History (Part 1 of 2)

Date and Revision	Version	Changes Made
July 2010	1.9	■ Updated Figure 7–25, Figure 7–26, and Figure 7–28. ■ Updated Equation 7–1 and Equation 7–2.
March 2010	1.8	Updated for the Quartus II software version 9.1 SP2 release: ■ Updated "Programmable Pull-Up Resistor" section. ■ Updated Figure 7–2, Figure 7–3, Figure 7–4, Figure 7–5, Figure 7–6. ■ Updated Table 7–2, Table 7–3, and Table 7–7. ■ Added reference before Table 7–11. ■ Removed "Conclusion" section. ■ Minor text edit.
May 2009	1.7	■ Updated "Expanded On-Chip Series Termination with Calibration" and "Mixing Voltage-Referenced and Non-Voltage-Referenced Standards" sections. ■ Added "Left Shift Series Termination Control" section. ■ Updated Table 7–8 and Table 7–9. ■ Updated Figure 7–24.
February 2009	1.6	■ Updated Table 7–3, Table 7–7, Table 7–8, and Table 7–11. ■ Updated Figure 7–2, Figure 7–3, Figure 7–4, Figure 7–5, and Figure 7–6. ■ Updated "LVDS Input On-Chip Termination (R_D)" section. ■ Removed "Referenced Documents" section.
October 2008	1.5	Text, Table, and Figure updates: ■ Updated Table 7–2, Table 7–4, Table 7–7, and Table 7–10. ■ Updated notes for Table 7–2. ■ Updated notes for Figure 7–3, Figure 7–4, Figure 7–5, Figure 7–6, and Figure 7–7. ■ Updated "Stratix III I/O Banks", "Modular I/O Banks", "High-Speed Differential I/O with DPA Support", "Dynamic On-Chip Termination", "LVDS Input On-Chip Termination (RD)", "Serial Data Transfer", "LVDS", "RSDS", "mini-LVDS", "Voltage-Referenced Standards", "Stratix III I/O Banks", "MultiVolt I/O Interface", and "On-Chip Parallel Termination with Calibration" sections. ■ Updated Figure 7–1. ■ Added Table 7–3. ■ Updated New Document Format.

Table 7–13. Chapter Revision History (Part 2 of 2)

Date and Revision	Version	Changes Made
May 2008	1.4	Text, Table, and Figure updates: ■ Updated Table 7–2 headers and notes. ■ Updated Figure 7–1. ■ Updated "Programmable Slew Rate Control", "Programmable Pre-Emphasis", "LVDS Input On-Chip Termination (RD)", and "Programmable Differential Output Voltage". ■ Added Note (1) for Figure 7–17. ■ Updated notes for Figure 7–24. ■ Added Note (2) for Figure 7–27. ■ Added Note (2) for Figure 7–28.
November 2007	1.3	Figure updates: ■ Updated Figure 7–4, Figure 7–5, Figure 7–6, Figure 7–15, and Figure 7–16. ■ Updated Note (1) of Figure 7–25.
October 2007	1.2	Text changes, figure updates, removal of a section: ■ Minor text edits to second to last paragraph on pg 7-47. ■ Updated Table 7–2, Table 7–4, Table 7–5, Table 7–8. ■ Updated "Introduction", "OCT Calibration Block Modes of Operation", "Power Up Mode", "User Mode". ■ Changed 3.0-V LVTTL and 3.0-V LVCMOS to be 3.3/3.0-V LVTTL and 3.3/3.0-V LVCMOS throughout the document. ■ Added a note to Figure 7–1, Figure 7–3, Figure 7–4, Figure 7–5, Figure 7–6, Figure 7–7, Figure 7–14, Figure 7–15, Figure 7–16, and Figure 7–17. ■ Updated Figure 7–8, Figure 7–18, Figure 7–22, Figure 7–23, Figure 7–25, Figure 7–28, and Figure 7–29. ■ Added Figure 7–18 and Figure 7–20. ■ Expanded "3.3-V I/O Interface" on page 7–15 to include new information. ■ Removed section "OCT Calibration Block Architecture", "OCT Calibration Block Ports", and "OCT Calibration Block Code Data Transfer". ■ Added section "OCT Calibration", "Serial Data Transfer", "Example of Using Multiple OCT Calibration Blocks", "RS Calibration", and "Referenced Documents." ■ Added live links for references.
May 2007	1.1	■ Added the feature programmable input delay to "Stratix III I/O Structure" on page 7–13. ■ Updated Table 7–4 and Table 7–7. ■ Updated "LVDS Input On-Chip Termination (RD)" on page 7–29. ■ Updated Figure 7–3 through Figure 7–7. ■ Updated Figure 7–23, Figure 7–24. ■ Minor text edits to page 14.
November 2006	1.0	Initial Release.

8. External Memory Interfaces in Stratix III Devices

SIII51008-1.9

The Stratix® III I/O structure has been completely redesigned to provide flexible, high-performance support for existing and emerging external memory standards. These include high-performance double data rate (DDR) memory standards such as DDR3, DDR2, DDR SDRAM, QDR II+, QDR II SRAM, and RLDRAM II.

Packed with features such as dynamic on-chip termination (OCT), trace mismatch compensation, read and write leveling, half data rate (HDR) blocks, and 4- to 36- bit programmable DQ group widths, Stratix III I/O elements provide easy-to-use built-in functionality required for a rapid and robust implementation.

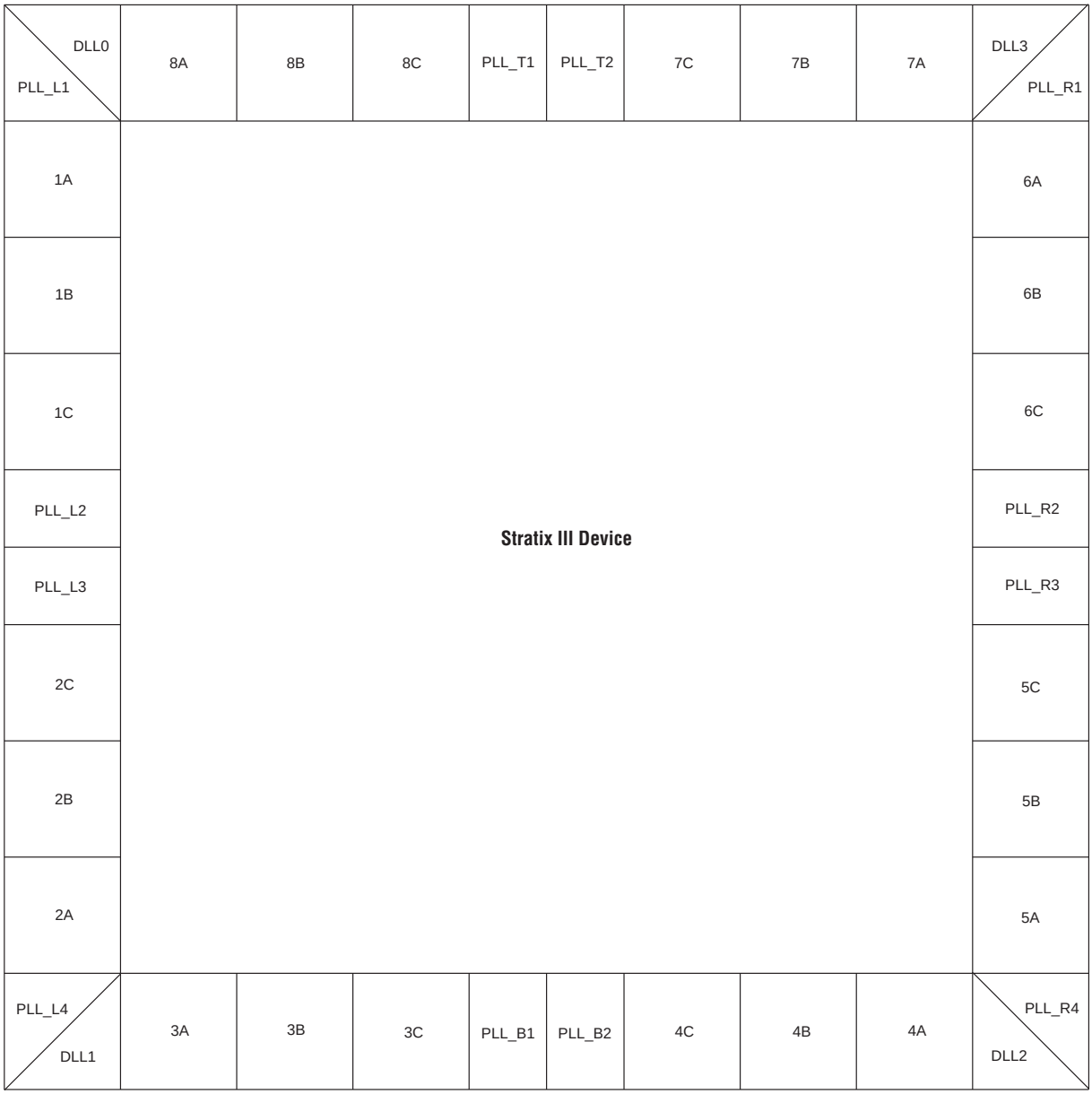
DDR external memory support is found on all sides of the Stratix III FPGA. Stratix III devices provide an efficient architecture to quickly and easily fit wide external–memory interfaces with the new small modular I/O bank structure.

A self-calibrating megafunction (ALTMEMPHY) is optimized to take advantage of the Stratix III I/O structure, along with the Quartus® II software's TimeQuest Timing Analyzer, which provides the total solution for the highest reliable frequency of operation across process, voltage, and temperature (PVT) variations.

While this chapter describe the silicon capability of Stratix III devices, for more information about the external memory system specifications, implementation, board guidelines, timing analysis, simulation, and design debugging, refer to the Literature: External Memory Interfaces section of the Altera website.

Figure 8–1 shows a package bottom view for Stratix III external memory support, showing the phase-locked loop (PLL), delay-locked loop (DLL), and I/O banks. The number of available I/O banks and PLLs depend on the device density.

Figure 8–1. Package Bottom View for Stratix III Devices *(Note 1), (2)*



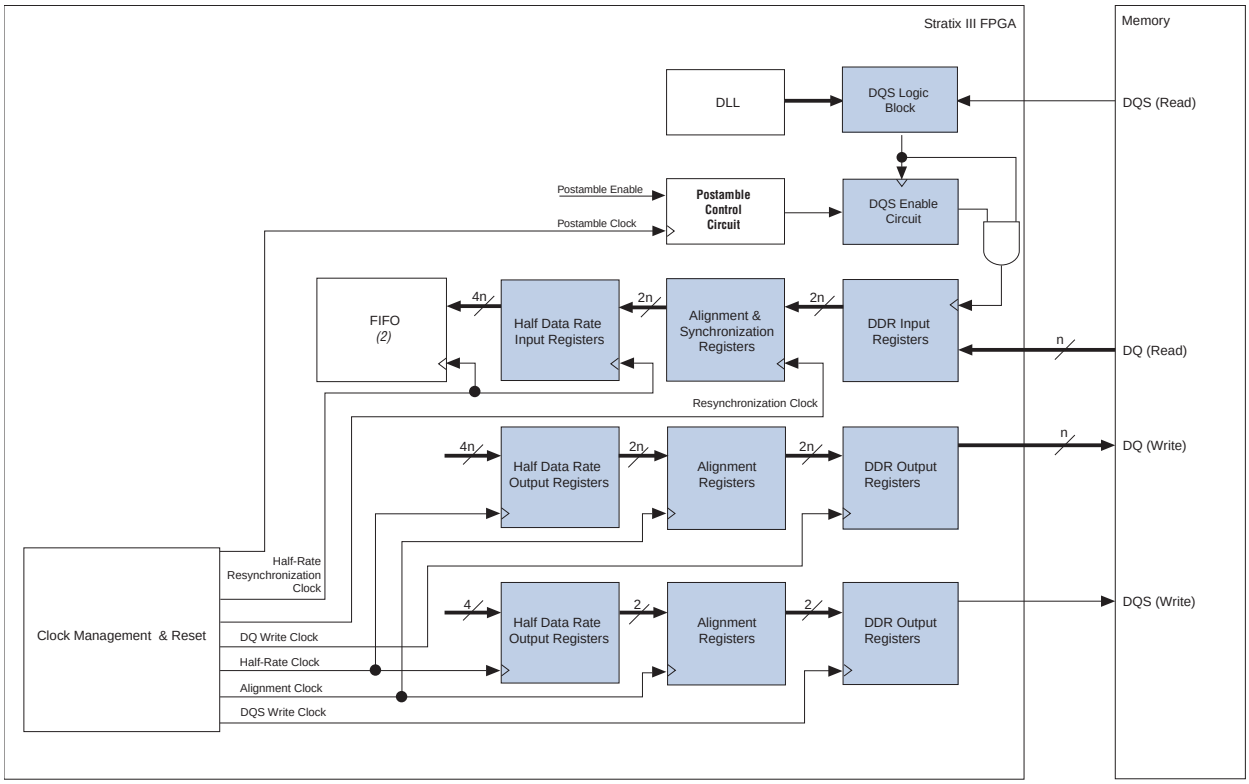
Notes to Figure 8–1:

(1) The number of I/O banks and PLLs available depends on the device density.

(2) There is only one PLL in the center of each side of the device in EP3SL50, EP3SL70, and EP3SE50 devices.

Figure 8–2 shows an overview of the memory interface data path that uses all the Stratix III I/O Element (IOE) features.

Figure 8–2. External Memory Interface Data Path Overview *(Note 1)*, *(2)*, *(3)*



Notes to Figure 8–2:

(1) Each register block can be bypassed.

(2) The blocks for each memory interface may differ slightly.

(3) These signals may be bi-directional or uni-directional, depending on the memory standard. When bi-directional, the signal is active during both read and write operations.

This chapter describes the hardware features in Stratix III devices that facilitate high-speed memory interfacing for each DDR memory standard. Stratix III devices feature DLLs, PLLs, dynamic OCT, read and write leveling, and deskew ciruitry.

Memory Interfaces Pin Support

A typical memory interface requires data (D, Q, or DQ), data strobe (DQS/CQ and DQSn/CQn), address, command, and clock pins. Some memory interfaces use data mask (DM) pins to enable write masking and QVLD pins to indicate that the read data is ready to be captured. This section describes how Stratix III devices support all these different pins.

For more information on memory interfaces, refer to the *Stratix III Pin Connection Guidelines*.

Data and Data-Strobe/Clock Pins

Read data-strobes or clocks are called DQS pins. Depending on the memory specifications, DQS pins can be bi-directional single-ended signals (in DDR2 and DDR SDRAM), uni-directional differential signals (in RLDRAM II), bi-directional differential signals (DDR3 and DDR2 SDRAM), or uni-directional complementary signals (QDR II+ and QDR II SRAM). Connect the uni-directional read and write data-strobes or clocks to Stratix III DQS pins.

Stratix III devices offer differential input buffers for differential read data-strobe/clock operations and provide an independent DQS logic block for each CQn pin for complementary read data-strobe/clock operations. The differential DQS pin-pairs are denoted as DQS and DQSn pins, while the complementary DQS signals are denoted as CQ and CQn pins. DQSn and CQn pins are marked separately in the pin table. Each CQn pin connects to a DQS logic block and the shifted CQn signals go to the negative-edge input registers in the IOE registers.

☞ Use differential DQS signaling for DDR2 SDRAM interfaces running higher than 333 MHz.

☞ For DDR3 and DDR2 SDRAM application, pseudo-differential DQS signaling is used for write operation.

Stratix III DDR memory interface data pins are called DQ pins. DQ pins can be bi-directional signals (in DDR3, DDR2, and DDR SDRAM, and RLDRAM II common I/O (CIO) interfaces), or uni-directional signals (in QDR II+, QDR II SRAM, and RLDRAM II separate I/O (SIO) devices). Connect the uni-directional read data signals to Stratix III DQ pins and the uni-directional write data signals to a different DQS/DQ group other than the read DQS/DQ group. You must assign the write clocks to the DQS/DQSn pins associated to this write DQS/DQ group. Do not use the CQ/CQn pin-pair for write clocks.

☞ Using a DQS/DQ group for write data signals minimizes output skew, allows access to the write leveling circuitry (for DDR3 SDRAM interfaces), and allows for vertical migration. These pins also have access to deskewing circuitry that can compensate for delay mismatch between signals on the bus.

For more information about pin planning, refer to *Section I. Device and Pin Planning* chapter in volume 2 of the *External Memory Interface Handbook*.

The DQS and DQ pin locations are fixed in the pin table. Memory interface circuitry is available in every Stratix III I/O bank. All memory interface pins support the I/O standards required to support DDR3, DDR2, DDR SDRAM, QDR II+, QDR II SRAM, and RLDRAM II devices.

The Stratix III device supports DQS and DQ signals with DQ bus modes of ×4, ×8/×9, ×16/×18, or ×32/×36, although not all devices support DQS bus mode ×32/×36. When any of these pins are not used for memory interfacing, you can use them as user I/Os. In addition, you can use any DQSn or CQn pins not used for clocking as DQ (data) pins. Table 8–1 lists pin support per DQS/DQ bus mode, including the DQS/CQ and DQSn/CQn pin pair.

Table 8–1. DQS and DQ Bus Mode Pins for Stratix III Devices

Mode	DQSn Support	CQn Support	Parity or DM (Optional)	QVLD (Optional) *(1)*	Typical Number of Data Pins per Group	Maximum Number of Data Pins per Group *(2)*
×4	Yes	No	No *(3)*	No	4	5
×8/×9 *(4)*	Yes	Yes	Yes	Yes	8 or 9	11
×16/×18 *(5)*	Yes	Yes	Yes	Yes	16 or 18	23
×32/×36 *(6)*	Yes	Yes	Yes	Yes	32 or 36	47

Notes to Table 8–1:

(1) The QVLD pin is not used in the ALTMEMPHY megafunction.

(2) This represents the maximum number of DQ pins (including parity, data mask, and QVLD pins) connected to the DQS bus network with single-ended DQS signaling. When you use differential or complementary DQS signaling, the maximum number of data per group decreases by one. This number may vary per DQS/DQ group in a particular device. Check with the pin table for the accurate number per group. For DDR3, DDR2, and DDR interfaces, the number of pins is further reduced for interfaces larger than ×8 mode because a DQS pin for each ×8/×9 group that is used to form the ×16/×18 and ×32/×36 groups is required.

(3) The DM pin can be supported if differential DQS is not used and the group does not have additional signals.

(4) Two ×4 DQS/DQ groups are stitched to make a ×8/×9 group, so there are a total of 12 pins in this group.

(5) Four ×4 DQS/DQ groups are stitched to make a ×16/×18 group.

(6) Eight ×4 DQS/DQ groups are stitched to make a ×32/×36 group.

Using R_{UP}/R_{DN} Pins in a DQS/DQ Group Used for Memory Interfaces

You can also use DQS/DQSn pins in some of the ×4 groups as R_{UP}/R_{DN} pins (listed in Table 8–1). You cannot use a ×4 DQS/DQ group for memory interfaces if any of its pin members are being used as RUP and RDN pins for OCT calibration. You may be able to use the ×8/×9 group that includes this ×4 DQS/DQ group, if either of the following applies:

- You are not using DM pins with your differential DQS pins
- You are not using complementary or differential DQS pins

This is because a DQS/DQ ×8/×9 group is comprised of 12 pins, as the groups are formed by stitching two DQS/DQ groups in ×4 mode with six total pins each (refer to Table 8–1). A typical ×8 memory interface consists of one DQS, one DM, and eight DQ pins which add up to 10 pins. If you choose your pin assignment carefully, you can use the two extra pins for RUP and RDN. In a DDR3 SDRAM interface, you have to use differential DQS, which means that you only have one extra pin. In this case, pick different pin locations for the RUP and RDN pins (for example, in the bank that contains the address and control/command pins).

You cannot use RUP and RDN pins shared with DQS/DQ group pins when using ×9 QDR II+/QDR II SRAM devices, as the RUP and RDN pins may have dual purpose with the CQn pins. In this case, pick different pin locations for RUP and RDN pins to avoid conflicts with the memory interface pin placement. In this case, you have the choice of placing the RUP and RDN pins in the data-write group or in the same bank as the address and control/command pins. There is no restriction when using ×16/×18 or ×32/×36 DQS/DQ groups that include the ×4 groups whose pin members are being used as RUP and RDN pins, because there are enough extra pins that you can use as DQS pins.

You must pick your DQS and DQ pins manually for the ×8, ×16/×18, or ×32/×36 DQS/DQ group whose members are being used for RUP and RDN because the Quartus II software might not be able to place this correctly when there are no specific pin assignments and might give you a "no-fit" instead.

Table 8–2 lists the maximum number of DQS/DQ groups per side of the Stratix III device. For a more detailed listing of the number of DQS/DQ groups available per bank in each Stratix III device, refer to Figure 8–3 through Figure 8–7. These figures represent the package bottom view of the Stratix III device.

Table 8–2. Number of DQS/DQ Groups in Stratix III Devices per Side (Part 1 of 2)

Device	Package	Side	×4 *(1)*	×8/×9	×16/×18	×32/×36 *(2)*
EP3SE50 EP3SL50 EP3SL70	484-pin FineLine BGA	Left/ Right	12	4	0	0
		Top/ Bottom	5	2	0	0
	780-pin FineLine BGA	Left/ Right	14	6	2	0
		Top/ Bottom	17	8	2	0
EP3SE80 EP3SE110 EP3SL110 EP3SL150	780-pin FineLine BGA	Left/ Right	14	6	2	0
		Top/ Bottom	17	8	2	0
	1152-pin FineLine BGA	Left/ Right	26	12	4	0
		Top/ Bottom	26	12	4	0
EP3SL200	780-pin Hybrid FineLine BGA	Left/ Right	14	6	2	0
		Top/ Bottom	17	8	2	0
	1152-pin FineLine BGA	Left/ Right	26	12	4	0
		Top/ Bottom	26	12	4	0
	1517-pin FineLine BGA	Left/ Right	34	16	6	0
		Top/ Bottom	38	18	8	4
EP3SE260	780-pin Hybrid FineLine BGA	Left/ Right	14	6	2	0
		Top/ Bottom	17	8	2	0
	1152-pin FineLine BGA	Left/ Right	26	12	4	0
		Top/ Bottom	26	12	4	0
	1517-pin FineLine BGA	Left/ Right	34	16	6	0
		Top/ Bottom	38	18	8	4

Table 8–2. Number of DQS/DQ Groups in Stratix III Devices per Side (Part 2 of 2)

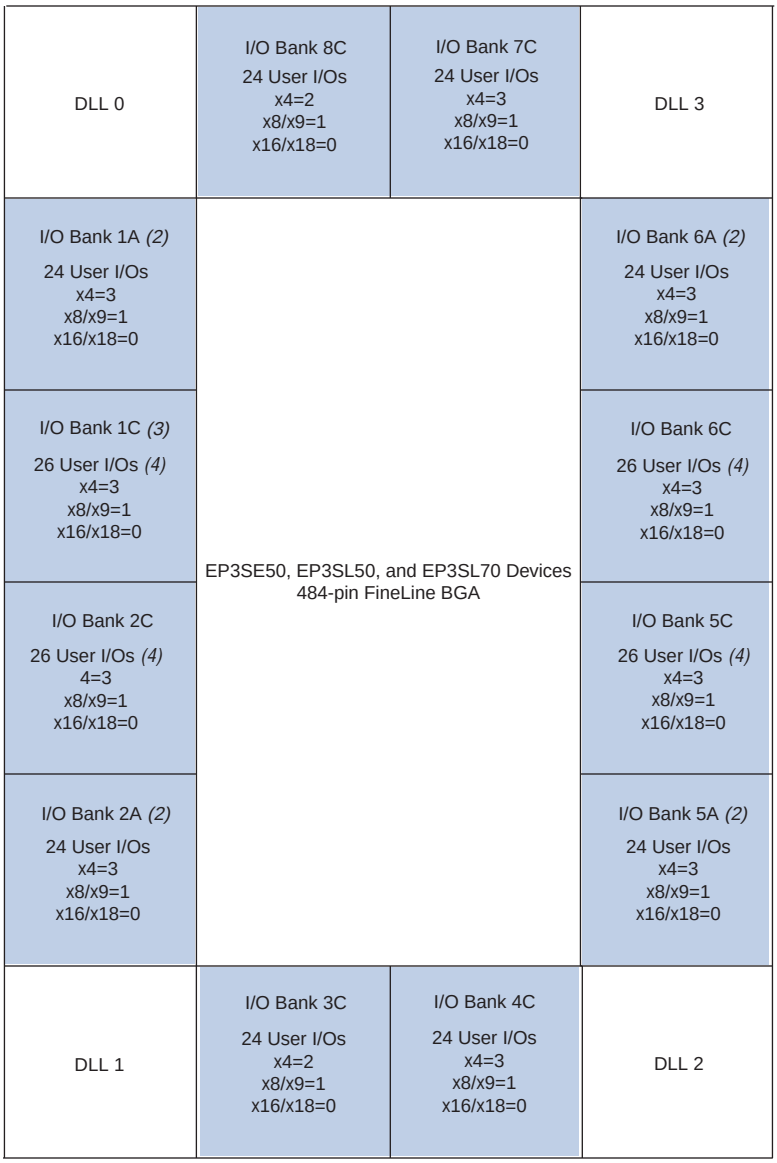
Device	Package	Side	×4 *(1)*	×8/×9	×16/×18	×32/×36 *(2)*
EP3SL340	1152-pin Hybrid FineLine BGA	Left/ Right	26	12	4	0
		Top/ Bottom	26	12	4	0
	1517-pin FineLine BGA	Left/ Right	34	16	6	0
		Top/ Bottom	38	18	8	4
	1760-pin FineLine BGA	Left/ Right	40	18	6	0
		Top/ Bottom	44	22	10	4

Notes to Table 8–2:

(1) Some of the ×4 groups may use configuration or RUP/RDN pins. You cannot use these ×4 groups if the pins are used for configuration or as RUP and RDN pins for OCT calibration.

(2) To interface with a ×36 QDR II+/QDR II SRAM device in a Stratix III FPGA that does not support the ×32/×36 DQS/DQ group, refer to the *Device, Pin, and Board Layout Guidelines* in volume 2 of the *External Memory Interface Handbook*.

Figure 8–3. Number of DQS/DQ Groups per Bank in EP3SE50, EP3SL50, and EP3SL70 Devices in the 484-pin FineLine BGA Package *(Note 1)*



Notes to Figure 8–3:

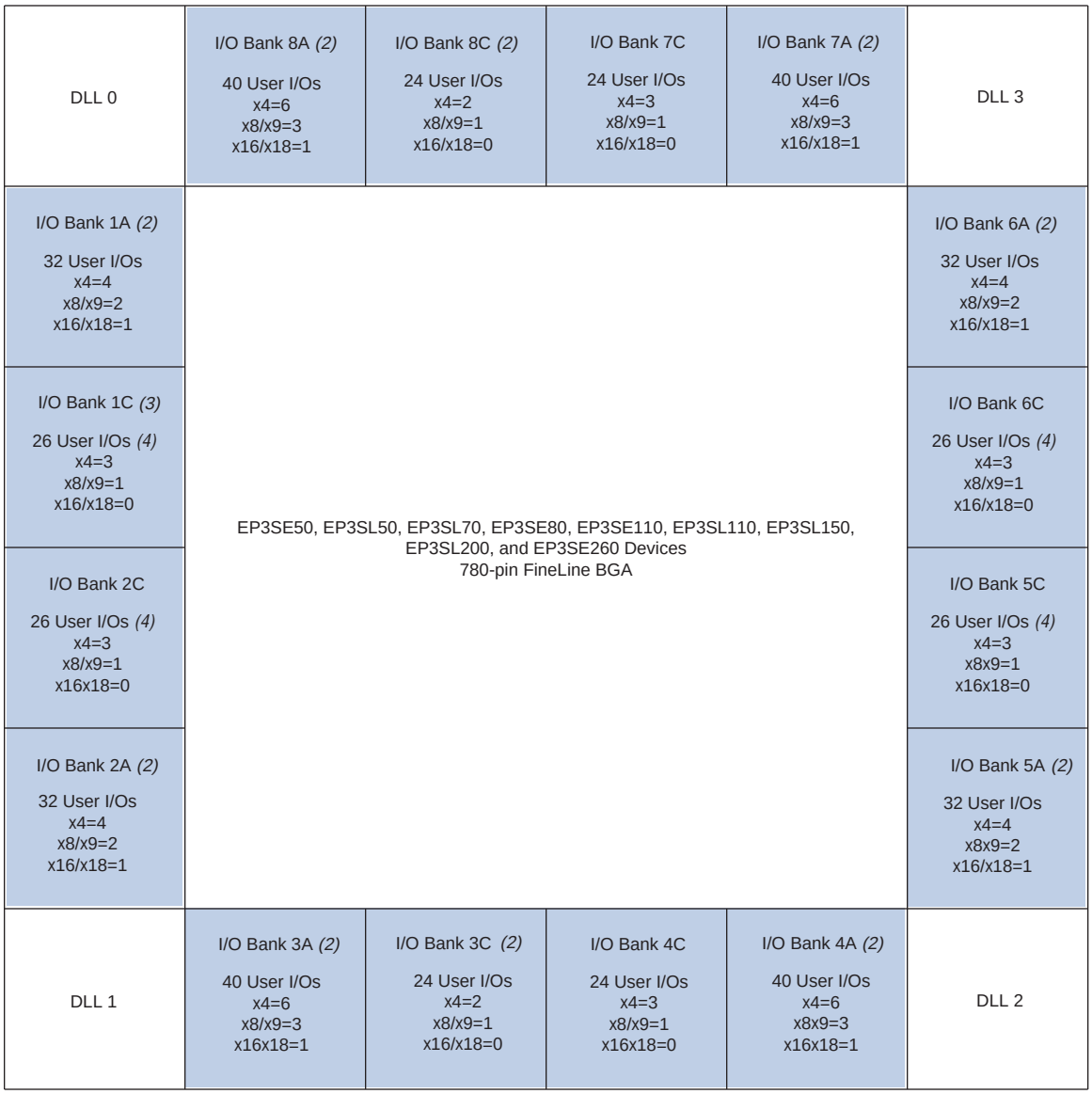
(1) This device does not support ×32/×36 mode.

(2) You can also use DQS/DQSn pins in some of the ×4 groups as RUP/RDN pins. You cannot use a ×4 group for memory interfaces if two pins of the group are being used as RUP and RDN pins for OCT calibration. You can still use the ×16/×18 or ×32/×36 groups that includes these ×4 groups. However, there are restrictions on using ×8/×9 groups that include these ×4 groups as described on page 8–5.

(3) Some of the DQS/DQ pins in this bank can also be used as configuration pins. Choose the DQS/DQ pins that are not going to be used by your configuration scheme.

(4) All I/O pin counts include eight dedicated clock inputs (`CLK1p`, `CLK1n`, `CLK3p`, `CLK3n`, `CLK8p`, `CLK8n`, `CLK10p`, and `CLK10n`)

Figure 8–4. Number of DQS/DQ Groups per Bank in EP3SE50, EP3SL50, EP3SL70, EP3SE80, EP3SE110, EP3SL110, EP3SL150, EP3SL200, and EP3SE260 Devices in the 780-pin FineLine BGA Package *(Note 1)*



Notes to Figure 8–4:

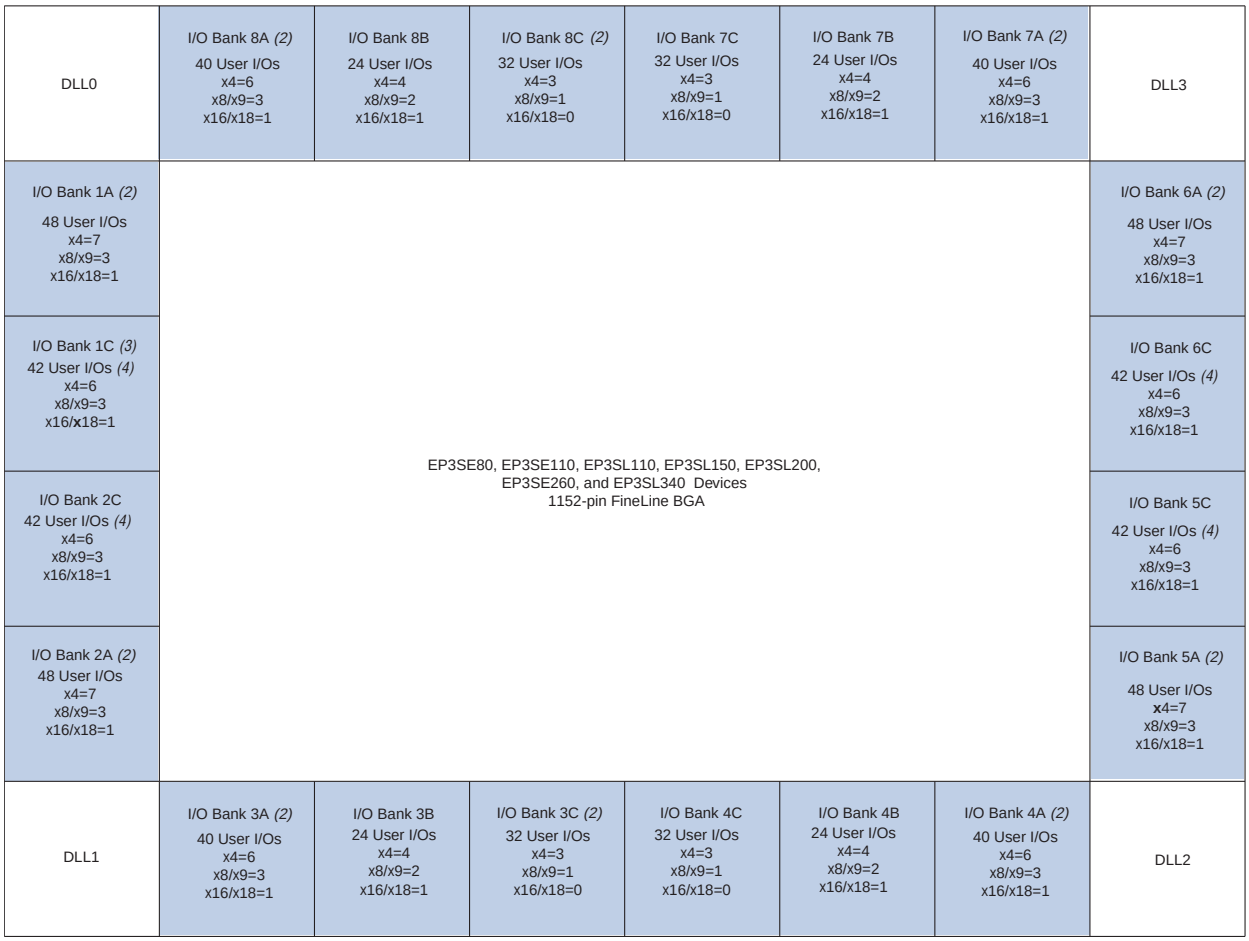
(1) This device does not support ×32/×36 mode.

(2) You can also use DQS/DQSn pins in some of the ×4 groups as RUP/RDN pins. You cannot use a ×4 group for memory interfaces if two pins of the group are being used as RUP and RDN pins for OCT calibration. You can still use the ×16/×18 or ×32/×36 groups that includes these ×4 groups. However, there are restrictions on using ×8/×9 groups that include these ×4 groups as described on page 8–5.

(3) Some of the DQS/DQ pins in this bank can also be used as configuration pins. Choose the DQS/DQ pins that are not going to be used by your configuration scheme.

(4) All I/O pin counts include eight dedicated clock inputs (CLK1p, CLK1n, CLK3p, CLK3n, CLK8p, CLK8n, CLK10p, and CLK10n).

Figure 8–5. Number of DQS/DQ Groups in EP3SE80, EP3SE110, EP3SL110, EP3SL150, EP3SL200, EP3SE260, and EP3SL340 Devices in the 1152-pin FineLine BGA Package *(Note 1)*



Notes to Figure 8–5:

(1) This device does not support ×32/×36 mode.

(2) You can also use DQS/DQSn pins in some of the ×4 groups as RUP/RDN pins. You cannot use a ×4 group for memory interfaces if two pins of the group are being used as RUP and RDN pins for OCT calibration. You can still use the ×16/×18 or ×32/×36 groups that includes these ×4 groups. However, there are restrictions on using ×8/×9 groups that include these ×4 groups as described on page 8–5.

(3) Some of the DQS/DQ pins in this bank can also be used as configuration pins. Choose the DQS/DQ pins that are not going to be used by your configuration scheme.

(4) All I/O pin counts include eight dedicated clock inputs (CLK1p, CLK1n, CLK3p, CLK3n, CLK8p, CLK8n, CLK10p, and CLK10n).

Figure 8–6. Number of DQS/DQ Groups per Bank in EP3SL200, EP3SE260 and EP3SL340 Devices in the 1517-pin FineLine BGA Package

DLL0	I/O Bank 8A *(1)* 48 User I/Os x4=8 x8/x9=4 x16/x18=2 x32/x36=1	I/O Bank 8B 48 User I/Os x4=8 x8/x9=4 x16/x18=2 x32/x36=1	I/O Bank 8C *(1)* 32 User I/Os x4=3 x8/x9=1 x16/x18=0 x32/x36=0	I/O Bank 7C 32 User I/Os x4=3 x8/x9=1 x16/x18=0 x32/x36=0	I/O Bank 7B 48 User I/Os x4=8 x8/x9=4 x16/x18=2 x32/x36=1	I/O Bank 7A *(1)* 48 User I/Os x4=8 x8/x9=4 x16/x18=2 x32/x36=1	DLL3
I/O Bank 1A *(1)* 50 User I/Os *(3)* x4=7 x8/x9=3 x16/x18=1 x32/x36=0							I/O Bank 6A *(1)* 50 User I/Os *(3)* x4=7 x8/x9=3 x16/x18=1 x32/x36=0
I/O Bank 1B 24 User I/Os x4=4 x8/x9=2 x16/x18=1 x32/x36=0							I/O Bank 6B 24 User I/Os x4=4 x8/x9=2 x16/x18=1 x32/x36=0
I/O Bank 1C *(2)* 42 User I/Os *(3)* x4=6 x8/x9=3 x16/x18=1 x32/x36=0							I/O Bank 6C 42 User I/Os *(3)* x4=6 x8/x9=3 x16/x18=1 x32/x36=0
I/O Bank 2C 42 User I/Os *(3)* x4=6 x8/x9=3 x16/x18=1 x32/x36=0			EP3SL200, EP3SE260, and EP3SL340 Devices 1517-Pin FineLine BGA				I/O Bank 5C 42 User I/Os *(3)* x4=6 x8/x9=3 x16/x18=1 x32/x36=0
I/O Bank 2B 24 User I/Os x4=4 x8/**x**9=2 x16/x18=1 x32/x36=0							I/O Bank 5B 24 User I/Os x4=4 x8/x9=2 x16/x18=1 x32/x36=0
I/O Bank 2A *(1)* 50 User I/Os *(3)* x4=7 x8/x9=3 x16/x18=1 x32/x36=0							I/O Bank 5A *(1)* 50 User I/Os *(3)* x4=7 x8/x9=3 x16/x18=1 x32/x36=0
DLL1	I/O Bank 3A *(1)* 48 User I/Os x4=8 x8/x9=4 x16/x18=2 x32/x36=1	I/O Bank 3B 48 User I/Os x4=8 x8/x9=4 x16/x18=2 x32/x36=1	I/O Bank 3C *(1)* 32 User I/Os x4=3 x8/x9=1 x16/x18=0 x32/x36=0	I/O Bank 4C 32 User I/Os x4=3 x8/x9=1 x16/x18=0 x32/x36=0	I/O Bank 4B 48 User I/Os x4=8 x8/x9=4 x16/x18=2 x32/x36=1	I/O Bank 4A *(1)* 48 User I/Os x4=8 x8/x9=4 x16/x18=2 x32/x36=1	DLL2

Notes to Figure 8–6:

(1) You can also use DQS/DQSn pins in some of the ×4 groups as RUP/RDN pins. You cannot use a ×4 group for memory interfaces if two pins of the group are being used as RUP and RDN pins for OCT calibration. You can still use the ×16/×18 or ×32/×36 groups that includes these ×4 groups. However, there are restrictions on using ×8/×9 groups that include these ×4 groups as described on page 8–5.

(2) Some of the DQS/DQ pins in this bank can also be used as configuration pins. Choose the DQS/DQ pins that are not going to be used by your configuration scheme.

(3) All I/O pin counts include eight dedicated clock inputs (CLK1p, CLK1n, CLK3p, CLK3n, CLK8p, CLK8n, CLK10p, and CLK10n) and eight dedicated corner PLL clock inputs (PLL_L1_CLKp, PLL_L1_CLKn, PLL_L4_CLKp, PLL_L4_CLKn, PLL_R4_CLKp, PLL_R4_CLKn, PLL_R1_CLKp, and PLL_R1_CLKn) that can be used for data inputs.

Figure 8–7. DQS/DQ Bus Mode Support per Bank in EP3SL340 Devices in the 1760-pin FineLine BGA Package

DLL0	I/O Bank 8A *(1)* 48 User I/Os x4=8 x8/x9=4 x16/x18=2 x32/x36=1	I/O Bank 8B 48 User I/Os x4=8 x8/x9=4 x16/x18=2 x32/x36=1	I/O Bank 8C *(1)* 48 User I/Os x4=6 x8/x9=3 x16/x18=1 x32/x36=0	I/O Bank 7C 48 User I/Os x4=6 x8/x9=3 x16/x18=1 x32/x36=0	I/O Bank 7B 48 User I/Os x4=8 x8/x9=4 x16/x18=2 x32/x36=1	I/O Bank 7A *(1)* 48 User I/Os x4=8 x8/x9=4 x16/x18=2 x32/x36=1	DLL3
I/O Bank 1A *(1)* 50 User I/Os *(2)* x4=7 x8/x9=3 x16/x18=1 x32/x36=0							I/O Bank 6A *(1)* 50 User I/Os *(2)* x4=7 x8/x9=3 x16/x18=1 x32/x36=0
I/O Bank 1B 36 User I/Os x4=6 x8/x9=3 x16/x18=1 x32/x36=0							I/O Bank 6B 36 User I/Os x4=6 x8/x9=3 x16/x18=1 x32/x36=0
I/O Bank 1C *(3)* 50 User I/Os *(2)* x4=7 x8/x9=3 x16/x18=1 x32/x36=0							I/O Bank 6C 50 User I/Os *(2)* x4=7 x8/x9=3 x16/x18=1 x32/x36=0
I/O Bank 2C 50 User I/Os *(2)* x4=7 x8/x9=3 x16/x18=1 x32/x36=0			EP3SL340 Devices 1760-pin FineLine BGA				I/O Bank 5C 50 User I/Os *(2)* x4=7 x8/x9=3 x16/x18=1 x32/x36=0
I/O Bank 2B 36 User I/Os x4=6 x8/x9=3 x16/x18=1 x32/x36=0							I/O Bank 5B 36 User I/Os *(2)* x4=6 x8/x9=3 x16/x18=1 x32/x36=0
I/O Bank 2A *(1)* 50 User I/Os *(2)* x4=7 x8/x9=3 x16/x18=1 x32/x36=0							I/O Bank 5A *(1)* 50 User I/Os *(2)* x4=7 x8/x9=3 x16/x18=1 x32/x36=0
DLL1	I/O Bank 3A *(1)* 48 User I/Os x4=8 x8/x9=4 x16/x18=2 x32/x36=1	I/O Bank 3B 48 User I/Os x4=8 x81/x9=4 x16/x18=2 x32/x36=1	I/O Bank 3C *(1)* 48 User I/Os x4=6 x8/x9=3 x16/x18=1 x32/x36=0	I/O Bank 4C 48 User I/Os x4=6 x8/x9=3 x16/x18=1 x32/x36=0	I/O Bank 4B 48 User I/Os x4=8 x8/x9=4 x16/x18=2 x32/x36=1	I/O Bank 4A *(1)* 48 User I/Os x4=8 x8/x9=4 x16/x18=2 x32/x36=1	DLL2

Notes to Figure 8–7:

(1) You can also use DQS/DQSn pins in some of the ×4 groups as RUP/RDN pins. You cannot use a ×4 group for memory interfaces if two pins of the group are being used as RUP and RDN pins for OCT calibration. You can still use the ×16/×18 or ×32/×36 groups that includes these ×4 groups. However, there are restrictions on using ×8/×9 groups that include these ×4 groups as described on page 8–5.

(2) All I/O pin counts include eight dedicated clock inputs (CLK1p, CLK1n, CLK3p, CLK3n, CLK8p, CLK8n, CLK10p, and CLK10n) and eight dedicated corner PLL clock inputs (PLL_L1_CLKp, PLL_L1_CLKn, PLL_L4_CLKp, PLL_L4_CLKn, PLL_R4_CLKp, PLL_R4_CLKn, PLL_R1_CLKp, and PLL_R1_CLKn) that can be used for data inputs.

(3) Some of the DQS/DQ pins in this bank can also be used as configuration pins. Choose the DQS/DQ pins that are not going to be used by your configuration scheme.

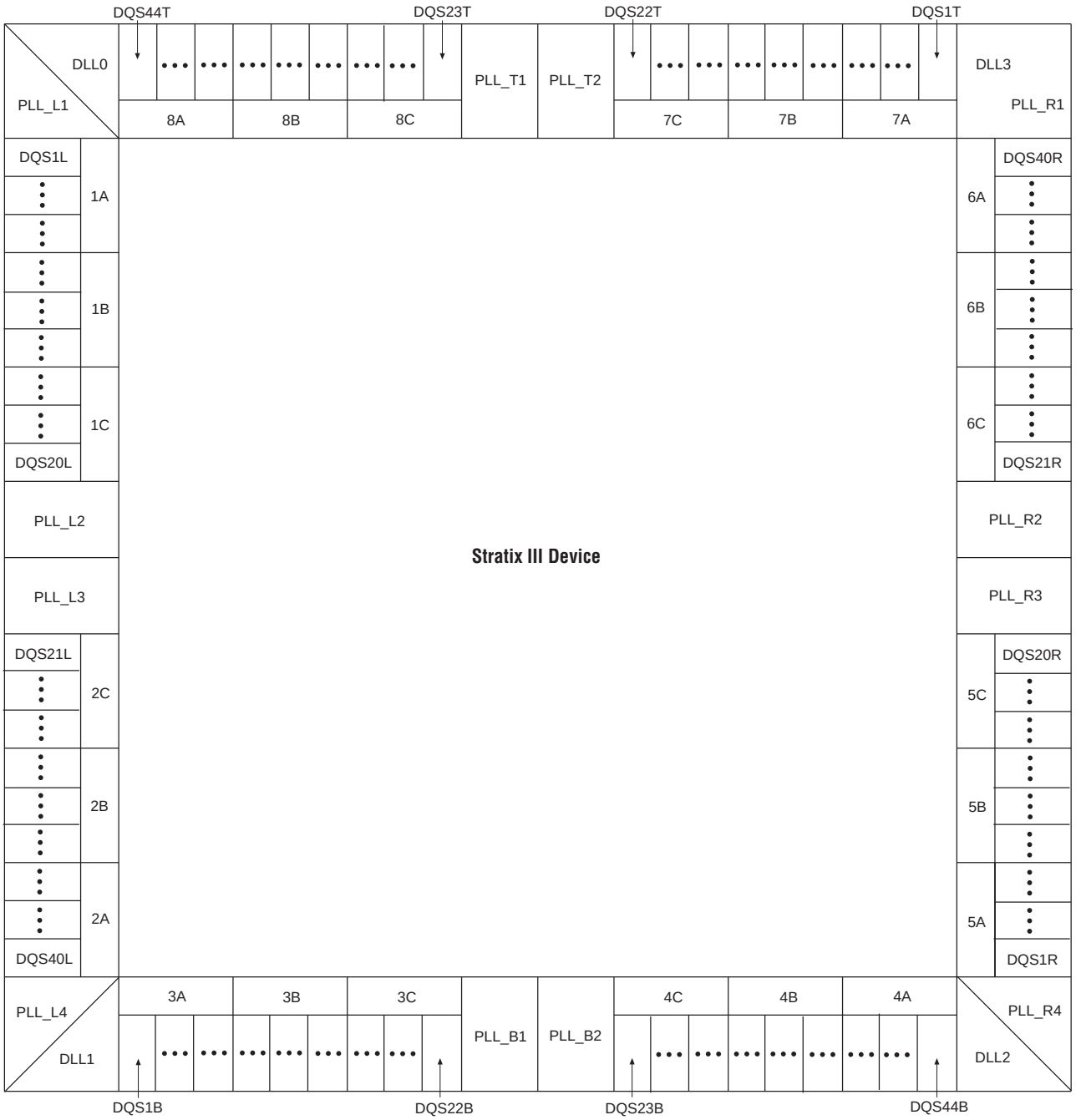
The DQS and DQSn pins are listed in the Stratix III pin tables as DQSXY and DQSnXY, respectively, where X denotes the DQS/DQ grouping number, and Y denotes whether the group is located on the top (*T*), bottom (*B*), left (*L*), or right (*R*) side of the device.

The corresponding DQ pins are marked as DQXY, where X indicates which DQS group the pins belong to and Y indicates whether the group is located on the top (*T*), bottom (*B*), left (*L*), or right (*R*) side of the device. For example, `DQS1L` indicates a DQS pin, located on the left side of the device. Refer to Figure 8–8 for an illustration. The DQ pins belonging to that group are shown as `DQ1L` in the pin table.

The numbering scheme starts from the top-left side of the device going counter-clockwise. Figure 8–8 shows how the DQS/DQ groups are numbered in a package bottom view of the device. The top and bottom sides of the device can contain up to 44 ×4 DQS/DQ groups. The left and right sides of the device can contain up to 40 ×4 DQS/DQ groups.

The parity, DM, BWSn, NWSn, ECC, and QVLD pins are shown as DQ pins in the pin table. When not used as memory interface pins, these pins are available as regular I/O pins.

Figure 8–8. DQS Pins in Stratix III I/O Banks



DQ pin numbering is based on ×4 mode. In ×4 mode, there are up to eight DQS/DQ groups per I/O bank. Each ×4 mode DQS/DQ group consists of a DQS pin, a DQSn pin, and four DQ pins. In ×8/×9 mode, the I/O bank combines two adjacent ×4 DQS/DQ groups; one pair of DQS and DQSn/CQn pins can drive all the DQ and parity pins in the new combined group that consists of up to 10 DQ pins (including parity or DM and QVLD pins) and a pair of DQS and DQSn/CQn pins.

Similarly, in ×16/×18 mode, the I/O bank combines four adjacent ×4 DQS/DQ groups to create a group with a maximum of 19 DQ pins (including parity or DM and QVLD pins) and a pair of DQS/CQ and DQSn/CQn pins. In ×32/×36 mode, the I/O bank combines eight adjacent ×4 DQS DQ groups together to create a group with a maximum of 37 DQ pins (including parity or DM and QVLD pins) and a pair of DQS/CQ and DQSn/CQn pins.

Stratix III modular I/O banks allow easy formation of the DQS/DQ groups. If all the pins in the I/O banks are user I/O pins and are not used for programming, RUP/RDN used for OCT calibration, or PLL clock output pins, you can divide the number of I/O pins in the bank by six to get the maximum possible number of ×4 groups. You can then divide that number by two, four, or eight to get the maximum possible number of ×8/×9, ×16/×18, or ×32/×36, respectively (refer to Table 8–3). However, some of the pins in the I/O bank may be used for other functions.

Table 8–3. DQ/DQS Group in Stratix III Modular I/O Banks

Modular I/O Bank Size	Maximum Possible Number of ×4 Groups *(1)*	Maximum Possible Number of ×8/×9 Groups	Maximum Possible Number of ×16/×18 Groups	Maximum Possible Number of ×32/×36 Groups
24 pins	4	2	1	0
32 pins	5	2	1	0
40 pins	6	3	1	0
48 pins	8	4	2	1

Note to Table 8–3:

(1) Some of the ×4 groups may use R_{UP}/R_{DN} pins. You cannot use these groups if you use the Stratix III calibrated OCT feature, as described in Table 8–1 on page 8–5.

Combining ×16/×18 DQS/DQ groups for ×36 QDR II+/QDR II SRAM Interface

This implementation combines two ×16/×18 DQS/DQ groups to interface with a ×36 QDR II+/QDR II SRAM device. The ×36 read data bus uses two ×16/×18 groups, while the ×36 write data uses another two ×16/×18 groups or four ×8/×9 groups. The CQ/CQn signal traces are split on the board trace to connect two pairs of DQS/CQn pins in the FPGA. This is the only connection on the board that you need to change for this implementation. Other QDR II+/QDR II SRAM interface rules for Stratix III devices also apply for this implementation.

Altera's ALTMEMPHY megafunction does not use the QVLD signal, so you can leave the QVLD signal unconnected as in any QDR II+/QDR II SRAM interfaces in the Stratix III devices.

For more information about the ALTMEMPHY megafunction, refer to the *ALTMEMPHY Megafunction User Guide*.

Rules to Combine Groups

In 780- and 1152-pin package devices, there is at most one ×16/×18 group per I/O sub-bank. You can combine ×16/×18 groups from a single side of the device for a ×36 interface. For devices that do not have four ×16/×18 groups in a single side of the device to form two ×36 groups for read and write data, you can form one ×36 group on one side of the device, and another ×36 group on the other side of the device. For

vertical migration with the ×36 emulation implementation, check if migration is possible by enabling device migration in the Quartus II project. Table 8–4 shows the possible I/O sub-bank combinations to form two ×36 groups. On Stratix III devices that do not have ×36 groups. Other Stratix III devices in the 1517 - and 1760 - pin packages support this implementation as well.

☞ Splitting the read or write data bus over more than one device edge is not recommended.

Table 8–4. I/O Sub-Bank Combinations for Stratix III Devices that do not have ×36 Groups to form two ×36 Groups.

Package	Device	I/O Sub-Bank Combinations
780-pin FineLine BGA	EP3SL50, EP3SL70, EP3SE80, EP3SE110, EP3SL110, EP3SL150, EP3SL200, and EP3SE260	■ 1A and 2A ■ 5A and 6A ■ 3A and 4A ■ 7A and 8A
1152-pin FineLine BGA	EP3SE80, EP3SE110, EP3SL110, EP3SL150, EP3SL200, EP3SE260, and EP3SL340	■ 1A and 1C ■ 2A and 2C ■ 3A and 3B ■ 4A and 4B ■ 5A and 5C ■ 6A and 6C ■ 7A and 7B ■ 8A and 8B
1517-pin FineLine BGA	EP3SL200, EP3SE260, and EP3SL340	■ 1A and 1B ■ 2A and 2B or 1B and 1C ■ 2B and 2C *(2)* ■ 5A and 5B ■ 6A and 6B or 5B and 5C ■ 6B and 6C *(2)*
1760-pin FineLine BGA *(1)*	EP3SL340	■ 1A and 1B ■ 2A and 2B or 1B and 1C ■ 2B and 2C *(2)* ■ 5A and 5B ■ 6A and 6B or 5B and 5C ■ 6B and 6C *(2)*

Notes to Table 8–4:

(1) This device supports ×36 DQ/DQS groups on the top and bottom I/O banks natively.

(2) You can combine the ×16/×18 DQ/DQS groups from I/O banks 1A and 1C, 2A and 2C, 5A and 5C, 6A and 6C. However, this process is discouraged because of the size of the package. Similarly, crossing a bank number (for example combining groups from I/O banks 6C and 5C) is not supported in this package.

Optional Parity, DM, BWSn, NWSn, ECC and QVLD Pins

In Stratix III devices, you can use any of the DQ pins from the same DQS/DQ group for data as parity pins. The Stratix III device family supports parity in ×8/×9, ×16/×18, and ×32/×36 modes. There is one parity bit available per eight bits of data pins. Use any of the DQ (or D) pins in the same DQS/DQ group as data for parity as they are treated, configured, and generated like a DQ pin.

DM pins are only required when writing to DDR3, DDR2, DDR SDRAM, and RLDRAM II devices. QDR II+ and QDR II SRAM devices use the BWSn (or NWSn in the ×8 QDR II SRAM devices) signal to select which byte to write into the memory. Each group of DQS and DQ signals in DDR3, DDR2, and DDR SDRAM devices require a DM pin. There is one DM pin per RLDRAM II device and one BWSn pin per 9 bits of data in ×9, ×18, and ×36 QDR II+/QDR II SRAM. The ×8 QDR II SRAM device has two BWSn pins per 8 data bits, which are referred to as NWSn pins.

A low signal on DM, NWSn, or BWSn indicates that the write is valid. If the DM/BWSn/NWSn signal is high, the memory masks the DQ signals. If the system does not require write data masking, connect the memory DM pins low to indicate every write data is valid. You can use any of the DQ pins in the same DQS/DQ group as write data for the DM/BWSn/NWSn signals. Generate the DM or BWSn signals using DQ pins and configure the signals similar to the DQ (or D) output signals. Stratix III devices do not support the DM signal in ×4 DDR3 SDRAM or in ×4 DDR2 SDRAM interfaces with differential DQS signaling.

Some DDR3, DDR2, and DDR SDRAM devices or modules support error correction coding (ECC), which is a method of detecting and automatically correcting errors in data transmission. In a 72-bit DDR3, DDR2, or DDR SDRAM interface, typically eight ECC pins are used in addition to the 64 data pins. Connect the DDR3, DDR2, and DDR SDRAM ECC pins to a Stratix III device DQS/DQ group. These signals are also generated similar to DQ pins. The memory controller requires encoding and decoding logic for ECC data. You can also use the extra byte of data for other error checking methods.

QVLD pins are used in RLDRAM II and QDR II+ SRAM interfaces to indicate read data availability. There is one QVLD pin per memory device. A high on QVLD indicates that the memory is outputting the data requested. Similar to DQ inputs, this signal is edge-aligned with the read clock signals (CQ/CQn in QDR II+/QDR II SRAM and QK/QK# in RLDRAM II) and is sent half a clock cycle before data starts coming out of the memory. The QVLD pin is not used in the ALTMEMPHY megafunction solution for QDR II+ SRAM.

For more information about the parity, ECC, and QVLD pins as these pins are treated as DQ pins refer to "Data and Data-Strobe/Clock Pins" on page 8–4.

Address and Control/Command Pins

Address and control/command signals are typically sent at a single data rate. The only exception is in QDR II SRAM burst-of-two devices, where the read address must be captured on the rising edge of the clock while the write address must be captured on the falling edge of the clock by the memory. There is no special circuitry required for the address and control/command pins. You can use any of the user I/O pins in the same I/O bank as the data pins.

Memory Clock Pins

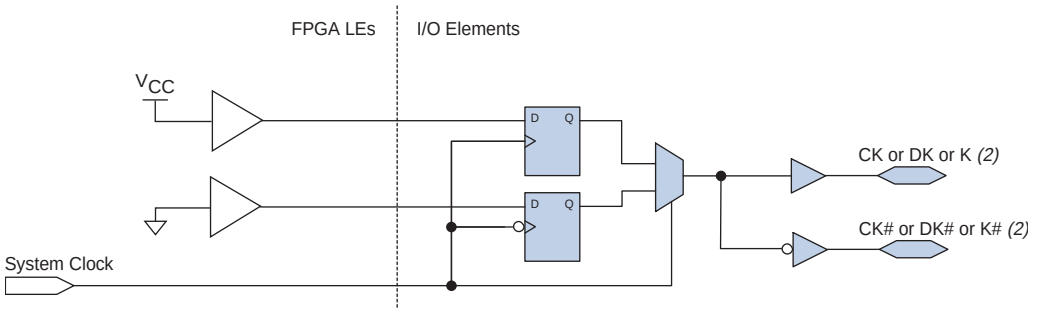
In addition to DQS (and CQn) signals to capture data, DDR3, DDR2, DDR SDRAM, and RLDRAM II use an extra pair of clocks, called CK and CK# signals, to capture the address and control/command signals. The CK/CK# signals must be generated to mimic the write data-strobe using Stratix III DDR I/O registers (DDIOs) to ensure that timing relationships between the CK/CK# and DQS signals (t_{DQSS} in DDR3, DDR2, and DDR SDRAM or t_{CKDK} in RLDRAM II) are met. QDR II+ and QDR II SRAM devices use the same clock (K/K#) to capture data, address, and control/command signals.

Memory clock pins in Stratix III devices are generated with a DDIO register going to differential output pins, marked in the pin table with `DIFFOUT`, `DIFFIO_TX`, and `DIFFIO_RX` prefixes.

For more information about which pins to use for memory clock pins, refer to the *Section I. Device and Pin Planning* chapter in volume 2 of the *External Memory Interface Handbook*.

Figure 8–9 shows the memory clock generation block diagram for Stratix III devices.

Figure 8–9. Memory Clock Generation Block Diagram *(Note 1)*



Notes to Figure 8–9:

(1) For more information about pin location requirements for these pins, refer *Section I. Device and Pin Planning* chapter in volume 2 of the *External Memory Interface Handbook*.

(2) The `mem_clk[0]` and `mem_clk_n[0]` pins for DDR3, DDR2, and DDR SDRAM interfaces use the I/O input buffer for feedback; therefore, bi-directional I/O buffers are used for these pins. For memory interfaces using a differential DQS input, the input feedback buffer is configured as differential input; for memory interfaces using a single-ended DQS input, the input buffer is configured as a single-ended input. Using a single-ended input feedback buffer requires that I/O standard's V_{REF} voltage is provided to that I/O bank's VREF pins.

Stratix III External Memory Interface Features

Stratix III devices are rich with features that allow robust high-performance external memory interfacing. The ALTMEMPHY megafunction allows you to set these external memory interface features and helps set up the physical interface (PHY) best suited for your system. This section describes each Stratix III device feature that is used in external memory interfaces from the DQS phase-shift circuitry, DQS logic block, leveling multiplexers, dynamic OCT control block, IOE registers, IOE features, and PLLs.

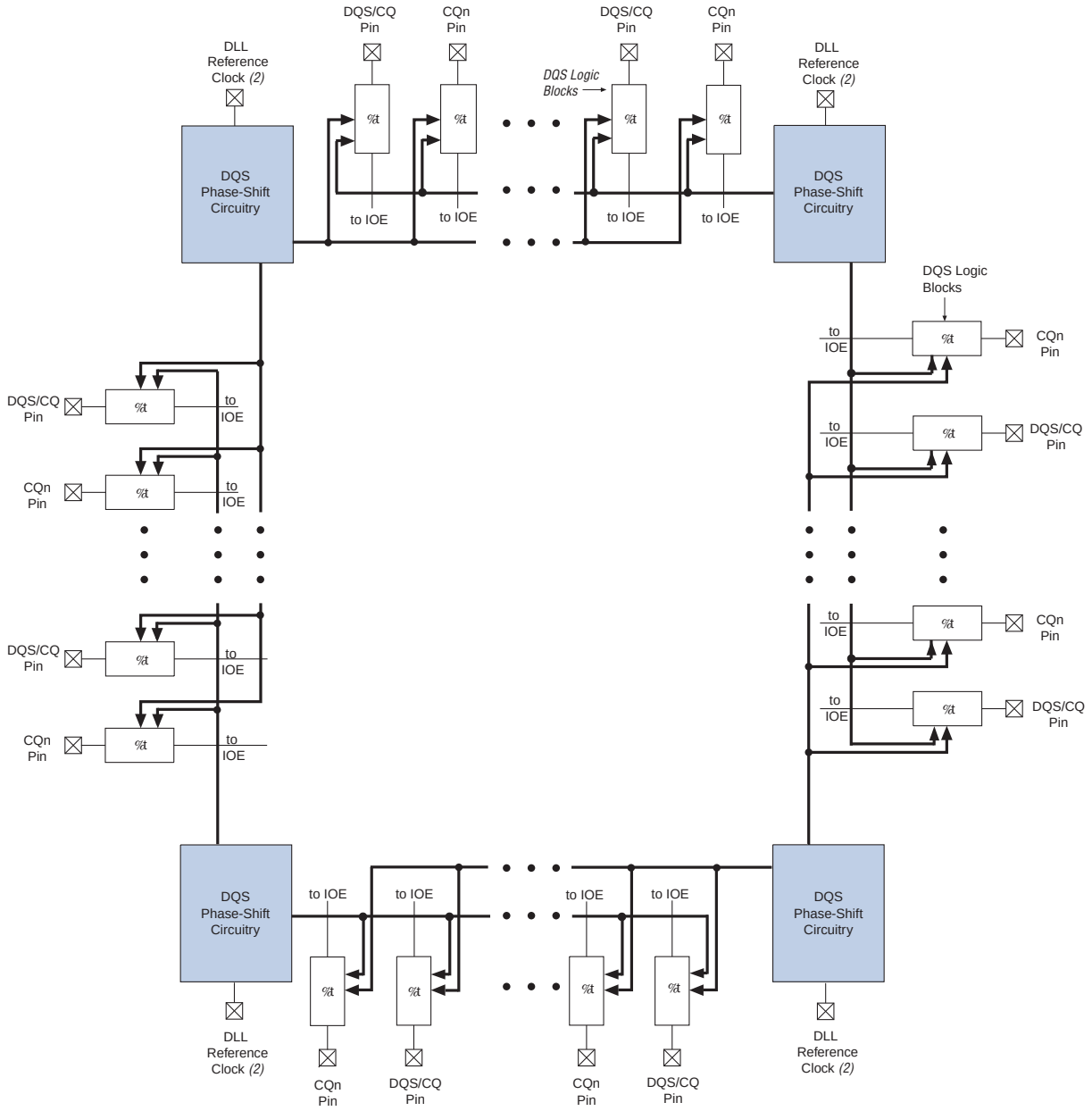
When using the Altera memory controller MegaCore® functions, the PHY is instantiated for you.

The ALTMEMPHY megafunction and the Altera memory controller MegaCore functions can run at half the frequency of the I/O interface of the memory devices to allow better timing management in high-speed memory interfaces. Stratix III devices have built-in registers to convert data from full-rate (I/O frequency) to half-rate (controller frequency) and vice versa. You can bypass these registers if your memory controller is not running at half the rate of the I/O frequency.

DQS Phase-Shift Circuitry

Stratix III phase-shift circuitry provides phase shift to the DQS and CQn pins on read transactions, when the DQS/CQ and CQn pins are acting as input clocks or strobes to the FPGA. DQS phase-shift circuitry consists of DLLs that are shared between multiple DQS pins and the phase-offset module to further fine-tune the DQS phase shift for different sides of the device. Figure 8–10 shows how the DQS phase-shift circuitry is connected to the DQS/CQ and CQn pins in the device.

Figure 8–10. DQS and CQn Pins and DQS Phase-Shift Circuitry *(Note 1)*



Notes to Figure 8–10:

(1) For possible reference input clock pins for each DLL, refer to"DLL" on page 8–21.

(2) You can configure each DQS/CQn signal pair with a phase shift based on one of two possible DLL output settings.

DQS phase-shift circuitry is connected to the DQS logic blocks that control each DQS/CQ or CQn pin. The DQS logic blocks allow the DQS delay settings to be updated concurrently at every DQS/CQ or CQn pin.

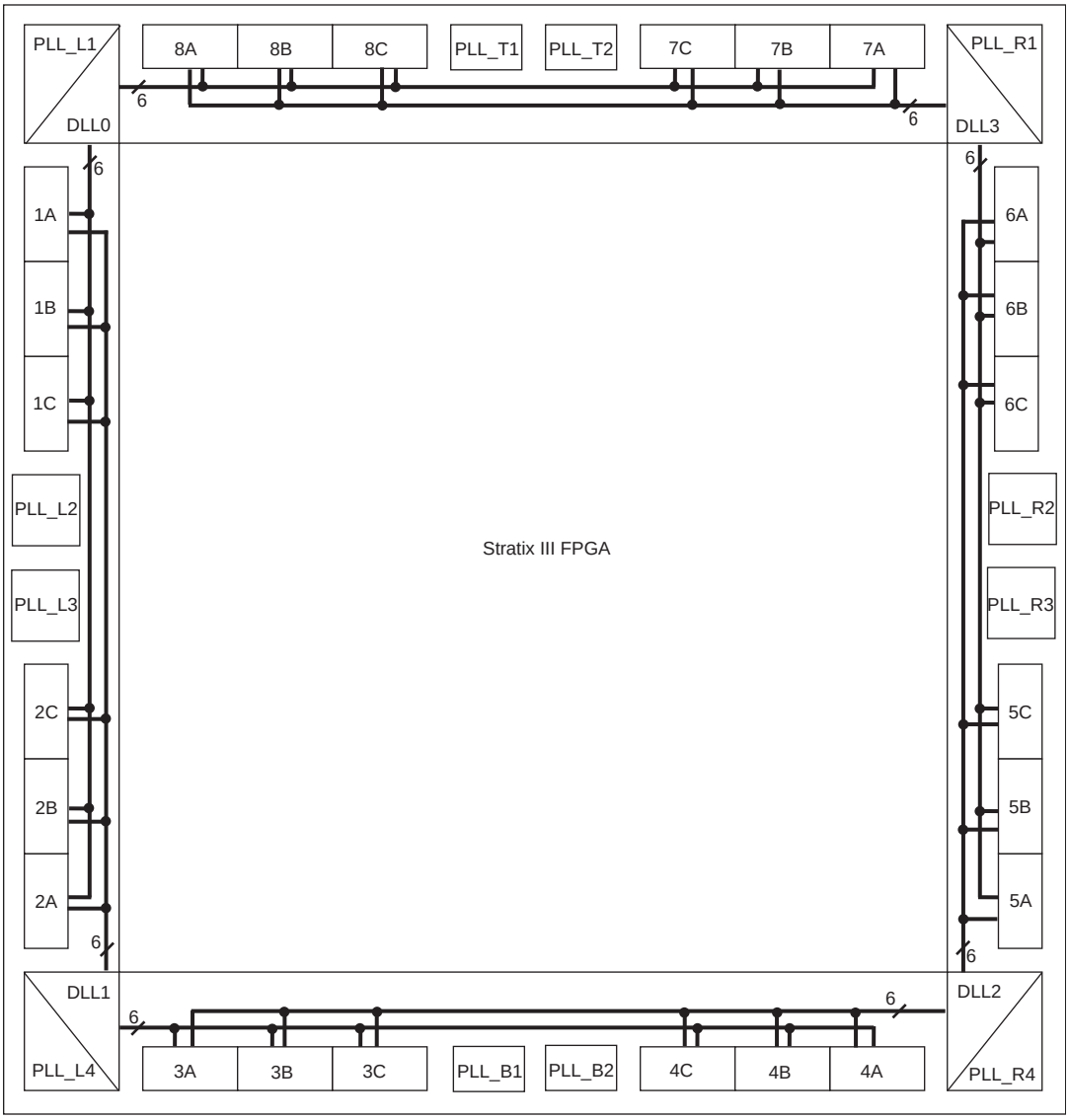
DLL

DQS phase-shift circuitry uses a DLL to dynamically measure the clock period required by the DQS/CQ and CQn pin. The DLL, in turn, uses a frequency reference to dynamically generate control signals for the delay chains in each of the DQS/CQ and CQn pins, allowing it to compensate for PVT variations. The DQS delay settings are Gray-coded to reduce jitter when the DLL updates the settings. The phase-shift circuitry requires a maximum of 1280 clock cycles to lock and calculate the correct input clock period when the DLL is in low jitter mode. Otherwise, only 256 clock cycles are required. Do not send data during these clock cycles because there is no guarantee it can be properly captured. As the settings from the DLL may not be stable until this lock period has elapsed, you should be aware that anything using these settings (including the leveling delay system) may be unstable during this period.

☞ Use the DQS phase-shift circuitry for any memory interfaces that are less than 100 MHz. The DQS signal is shifted by 2.5 ns. Even if the DQS signal is not shifted exactly to the middle of the DQ valid window, the I/O element should still be able to capture the data in low frequency applications where a large amount of timing margin is available.

There are four DLLs in a Stratix III device, located in each corner of the device. These DLLs support a maximum of four unique frequencies, with each DLL running at one frequency. Each DLL can have two outputs with different phase offsets, which allow one Stratix III device to have eight different DLL phase shift settings. Figure 8–11 shows the DLL and I/O bank locations in Stratix III devices from a package bottom view.

Figure 8–11. Stratix III DLL and I/O Bank Locations (Package Bottom View)



The DLL can access the two adjacent sides from its location within the device. For example, DLL0 on the top left of the device can access the top side (I/O banks 7A, 7B, 7C, 8A, 8B, and 8C) and the left side of the device (I/O banks 1A, 1B, 1C, 2A, 2B, and 2C). This means that each I/O bank is accessible by two DLLs, giving more flexibility to create multiple frequencies and multiple-type interfaces. For example, you can design an interface spanning one side of the device or within two sides adjacent to the DLL. The DLL outputs the same DQS delay settings for both sides of the device adjacent to the DLL.

Each bank can use settings from either or both DLLs that the bank is adjacent to. For example, `DQS1L` can get its phase-shift settings from DLL0, while `DQS2L` can get its phase-shift settings from DLL1. Table 8–5 lists the DLL location and supported I/O banks for Stratix III devices.

You can only have one memory interface in each I/O sub-bank (such as I/O sub-banks 1A, 1B, and 1C) when you use leveling delay chains. This is because there is only one leveling delay chain per I/O sub-bank.

Table 8–5. DLL Location and Supported I/O Banks

DLL	Location	Accessible I/O Banks
DLL0	Top left corner	1A, 1B, 1C, 2A, 2B, 2C, 7A, 7B, 7C, 8A, 8B, 8C
DLL1	Bottom left corner	1A, 1B, 1C, 2A, 2B, 2C, 3A, 3B, 3C, 4A, 4B, 4C
DLL2	Bottom right corner	3A, 3B, 3C, 4A, 4B, 4C, 5A, 5B, 5C, 6A, 6B, 6C
DLL3	Top right corner	5A, 5B, 5C, 6A, 6B, 6C, 7A, 7B, 7C, 8A, 8B, 8C

The reference clock for each DLL may come from PLL output clocks or any of the two dedicated clock input pins located in either side of the DLL. Table 8–6 through Table 8–9 lists the available DLL reference clock input resources for the Stratix III device family.

When you have a dedicated PLL that only generates the DLL input reference clock, set the PLL mode to **No Compensation**, or the Quartus II software changes it automatically. As the PLL does not use any other outputs, it does not require to compensate for any clock paths.

Table 8–6. DLL Reference Clock Input for EP3SE50, EP3SL50, and EP3SL70 Devices

DLL	CLKIN (Top/Bottom)	CLKIN (Left/Right)	PLL (Top/Bottom)	PLL (Left/Right)
DLL0	CLK12P CLK13P CLK14P CLK15P	CLK0P CLK1P CLK2P CLK3P	PLL_T1	PLL_L2
DLL1	CLK4P CLK5P CLK6P CLK7P	CLK0P CLK1P CLK2P CLK3P	PLL_B1	PLL_L2
DLL2	CLK4P CLK5P CLK6P CLK7P	CLK8P CLK9P CLK10P CLK11P	PLL_B1	PLL_R2
DLL3	CLK12P CLK13P CLK14P CLK15P	CLK8P CLK9P CLK10P CLK11P	PLL_T1	PLL_R2

Table 8–7. DLL Reference Clock Input for EP3SE80, EP3SE110, and EP3SL150 Devices in the 780-pin Package

DLL	CLKIN (Top/Bottom)	CLKIN (Left/Right)	PLL (Top/Bottom)	PLL (Left/Right)
DLL0	CLK12P CLK13P CLK14P CLK15P	CLK0P CLK1P CLK2P CLK3P	PLL_T1	PLL_L2
DLL1	CLK4P CLK5P CLK6P CLK7P	CLK0P CLK1P CLK2P CLK3P	PLL_B1	—
DLL2	CLK4P CLK5P CLK6P CLK7P	CLK8P CLK9P CLK10P CLK11P	—	—
DLL3	CLK12P CLK13P CLK14P CLK15P	CLK8P CLK9P CLK10P CLK11P	—	PLL_R1

Table 8–8. DLL Reference Clock Input for EP3SE80, EP3SE110, EP3SL110, and EP3SL150 Devices in the 1152-pin Package

DLL	CLKIN (Top/Bottom)	CLKIN (Left/Right)	PLL (Top/Bottom)	PLL (Left/Right)
DLL0	CLK12P CLK13P CLK14P CLK15P	CLK0P CLK1P CLK2P CLK3P	PLL_T1	PLL_L2
DLL1	CLK4P CLK5P CLK6P CLK7P	CLK0P CLK1P CLK2P CLK3P	PLL_B1	PLL_L3
DLL2	CLK4P CLK5P CLK6P CLK7P	CLK8P CLK9P CLK10P CLK11P	PLL_B2	PLL_R3
DLL3	CLK12P CLK13P CLK14P CLK15P	CLK8P CLK9P CLK10P CLK11P	PLL_T2	PLL_R2

Table 8–9. DLL Reference Clock Input for EP3SL200, EP3SE260 and EP3SL340 Devices *(Note 1)*, *(2)*

DLL	CLKIN (Top/Bottom)	CLKIN (Left/Right)	PLL (Top/Bottom)	PLL (Left/Right)
DLL0	CLK12P CLK13P CLK14P CLK15P	CLK0P CLK1P CLK2P CLK3P	PLL_T1	PLL_L1 PLL_L2
DLL1	CLK4P CLK5P CLK6P CLK7P	CLK0P CLK1P CLK2P CLK3P	PLL_B1	PLL_L3 PLL_L4
DLL2	CLK4P CLK5P CLK6P CLK7P	CLK8P CLK9P CLK10P CLK11P	PLL_B2	PLL_R3 PLL_R4
DLL3	CLK12P CLK13P CLK14P CLK15P	CLK8P CLK9P CLK10P CLK11P	PLL_T2	PLL_R1 PLL_R2

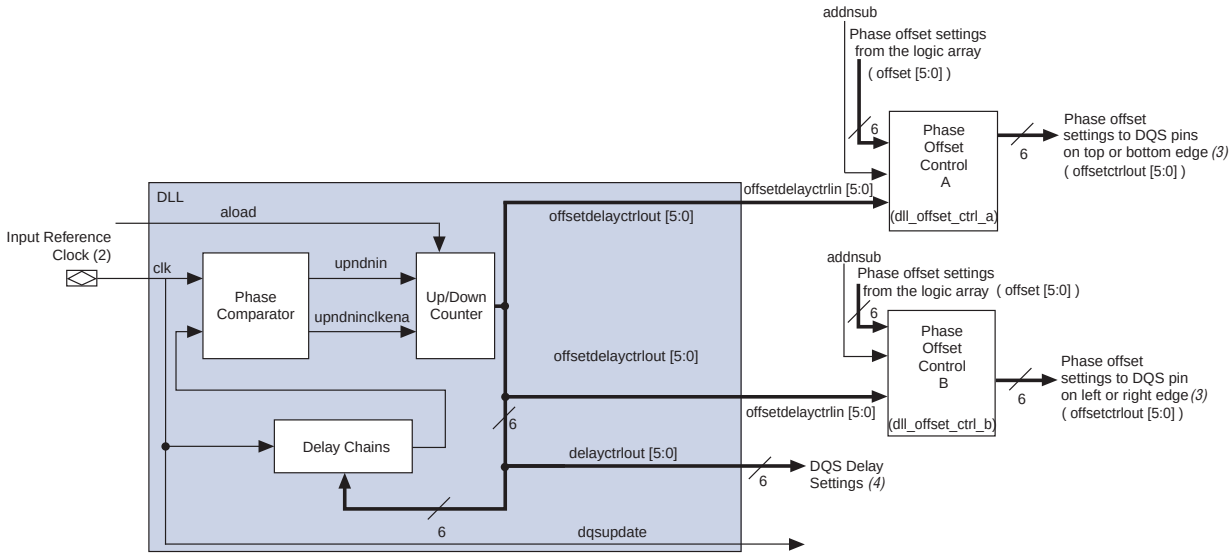
Notes to Table 8–9:

(1) PLLs L1, L3, L4, B2, R1, R3, R4, and T2 are not available for the EP3SL200 H780 package.

(2) PLLs L1, L4, R1 and R4 are not available for the EP3SL200 F1152 package.

Figure 8–12 shows a simple block diagram of the DLL. The input reference clock goes into the DLL to a chain of up to 16 delay elements. The phase comparator compares the signal coming out of the end of the delay chain block to the input reference clock. The phase comparator then issues the upndn signal to the Gray-code counter. This signal increments or decrements a 6-bit delay setting (DQS delay settings) that increases or decreases the delay through the delay element chain to bring the input reference clock and the signals coming out of the delay element chain in phase.

Figure 8–12. Simplified Diagram of the DQS Phase Shift Circuitry *(Note 1)*



Notes to Figure 8–12:

(1) All features of the DQS phase-shift circuitry are accessible from the ALTMEMPHY megafunction in the Quartus II software.

(2) The input reference clock for the DQS phase-shift circuitry can come from a PLL output clock or an input clock pin. For the exact PLL and input clock pin location, refer to Table 8–6 through Table 8–9.

(3) Phase offset settings can only go to the DQS logic blocks.

(4) DQS delay settings can go to the logic array, the DQS logic block, and the leveling circuitry.

The phase offset control block 'A' is designated as `DLLOFFSETCTRL_<coordinate x>_<coordinate y>_N1` and phase offset control block 'B' is designated as `DLLOFFSETCTRL_<coordinate x>_<coordinate y>_N2` in the Quartus II assignment.

You can reset the DLL from either the logic array or a user I/O pin. Each time the DLL is reset, you must wait for 1280 clock cycles before you can capture the data properly.

Depending on the DLL frequency mode, the DLL can shift the incoming DQS signals by 0°, 22.5°, 30°, 36°, 45°, 60°, 67.5°, 72°, 90°, 108°, 120°, 135°, 144°, or 180°. The shifted DQS signal is then used as the clock for the DQ IOE input registers.

All DQS and CQn pins referenced to the same DLL can have their input signal phase shifted by a different degree amount but all must be referenced at one particular frequency. For example, you can have a 90° phase shift on `DQS1T` and a 60° phase shift on `DQS2T` referenced from a 200-MHz clock. Not all phase-shift combinations are supported, however. The phase shifts on the DQS pins referenced by the same DLL must all be a multiple of 22.5° (up to 90°), a multiple of 30° (up to 120°), a multiple of 36° (up to 144°), or a multiple of 45° (up to 180°).

There are seven different frequency modes for the Stratix III DLL, as listed in Table 8–10. Each frequency mode provides different phase shift selections. In frequency modes 0, 1, 2, and 3, the 6-bit DQS delay settings vary with PVT to implement the phase-shift delay. In frequency modes 4, 5, 6, and 7, only 5 bits of the DQS delay settings vary with PVT to implement a phase-shift delay; the most significant bit of the DQS delay setting is set to 0.

For the frequency range of each mode, refer to the *DC and Switching Characteristics of Stratix III Devices* chapter.

Table 8–10. Stratix III DLL Frequency Modes

Frequency Mode	Available Phase Shift	Number of Delay Chains
0	22.5°, 45°, 67.5°, 90°	16
1	30°, 60°, 90°, 120°	12
2	36°, 72°, 108°, 144°	10
3	45°, 90°, 135°, 180°	8
4	30°, 60°, 90°, 120°	12
5	36°, 72°, 108°, 144°	10
6	45°, 90°, 135°, 180°	8
7	60°, 120°, 180°, 240°	6

For 0° shift, the DQS signal bypasses both the DLL and DQS logic blocks. The Quartus II software automatically sets DQ input delay chains so that the skew between the DQ and DQS pin at the DQ IOE registers is negligible when the 0° shift is implemented. You can feed the DQS delay settings to the DQS logic block and logic array.

The shifted DQS signal goes to the DQS bus to clock the IOE input registers of the DQ pins. The signal can also go into the logic array for resynchronization if you are not using the IOE resynchronization registers. The shifted CQn signal can only go to the negative-edge input register in the DQ IOE and is only used for QDR II+ and QDR II SRAM interfaces.

Phase Offset Control

Each DLL has two phase-offset modules and can provide two separate DQS delay settings with independent offset, one for the top and bottom I/O bank and one for the left and right I/O bank, so you can fine-tune the DQS phase shift settings between two different sides of the device. Even though you have independent phase offset control, the frequency of the interface using the same DLL has to be the same. Use the phase offset control module for making small shifts to the input signal; Use the DQS phase-shift circuitry for larger signal shifts. For example, if the DLL only offers a multiple of 30° phase shift, but your interface requires a 67.5° phase shift on the DQS signal, you can use two delay chains in the DQS logic blocks to give you 60° phase shift and use the phase offset control feature to implement the extra 7.5° phase shift.

You can either use a static phase offset or a dynamic phase offset to implement the additional phase shift. The available additional phase shift is implemented in 2's-complement in Gray-code between settings –64 to +63 for frequency modes 0, 1, 2, and 3, and between settings –32 to +31 for frequency modes 4, 5, and 6. An additional bit indicates whether the setting has a positive or negative value. The DQS phase shift is the sum of the DLL delay settings and the user selected phase offset settings. The maximum is setting 64 for frequency modes 0, 1, 2, and 3, and setting 32 for frequency modes 4, 5, 6, and 7 so the actual physical offset setting range is 64 or 32 subtracted by the DQS delay settings from the DLL.

When using this feature, you must to monitor the DQS delay settings to know how many offsets you can add and subtract in the system. The DQS delay settings output by the DLL are also Gray-coded.

For example, if the DLL determines that DQS delay settings of 28 is required to achieve a 30° phase shift in DLL frequency mode 1, you can subtract up to 28 phase offset settings and you can add up to 35 phase offset settings to achieve the optimal delay that you require. However, if the same DQS delay settings of 28 is required to achieve 30° phase shift in DLL frequency mode 4, you can still subtract up to 28 phase offset settings, but you can only add up to 3 phase offset settings before the DQS delay settings reach their maximum settings because DLL frequency mode 4 only uses 5-bit DLL delay settings.

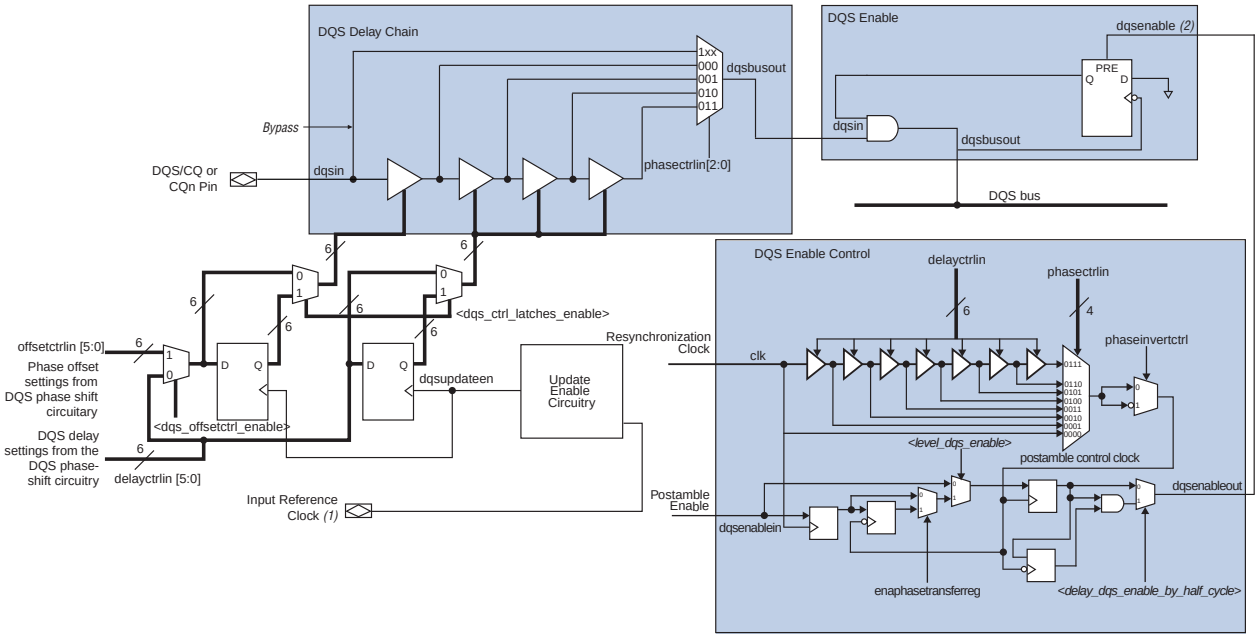
For more information about the value for each step, refer to the *DC and Switching Characteristics of Stratix III Devices* chapter.

When using static phase offset, you can specify the phase offset amount in the ALTMEMPHY megafunction as a positive number for addition or a negative number for subtraction. You can also have a dynamic phase offset that is always added to, subtracted from, or both added to and subtracted from the DLL phase shift. When you always add or subtract, you can dynamically input the phase offset amount into the `dll_offset[5..0]` port. When you want to both add and subtract dynamically, you control the `addnsub` signal in addition to the `dll_offset[5..0]` signals.

DQS Logic Block

Each DQS and CQn pin is connected to a separate DQS logic block, which consists of the DQS delay chains, update enable circuitry, and DQS postamble circuitry, as shown in Figure 8–13.

Figure 8–13. Stratix III DQS Logic Block



Notes to Figure 8–13:

(1) The input reference clock for the DQS phase-shift circuitry can come from a PLL output clock or an input clock pin. For the exact PLL and input clock pin location, refer to Table 8–6 through Table 8–9.

(2) The `dqsenable` signal can also come from the Stratix III FPGA fabric.

DQS Delay Chain

The DQS delay chains consist of a set of variable delay elements to allow the input DQS/CQ and CQn signals to be shifted by the amount specified by the DQS phase-shift circuitry or the logic array. There are four delay elements in the DQS delay chain; the first delay chain closest to the DQS/CQ pin can either be shifted by the DQS delay settings or by the sum of the DQS delay setting and the phase-offset setting. The number of delay chains required is transparent because the ALTMEMPHY megafunction automatically sets it when you choose the operating frequency. The DQS delay settings can come from the DQS phase-shift circuitry on either end of the I/O banks or from the logic array.

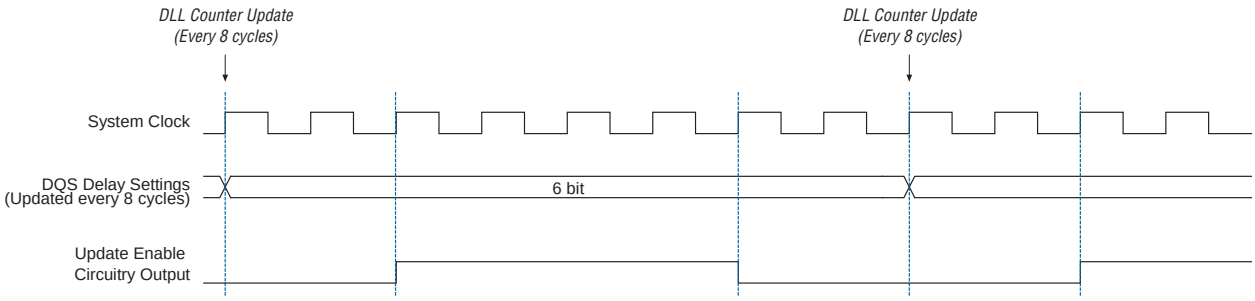
Delay elements in the DQS logic block have the same characteristics as the delay elements in the DLL. When the DLL is not used to control the DQS delay chains, you can input your own Gray-coded 6-bit or 5-bit settings using the `dqs_delayctrlin[5..0]`signals available in the ALTMEMPHY megafunction. These settings control 1, 2, 3, or all 4 delay elements in the DQS delay chains. The ALTMEMPHY megafunction can also dynamically choose the number of DQS delay chains required for the system. The amount of delay is equal to the sum of the delay element's intrinsic delay and the product of the number of delay steps and the value of the delay steps.

You can also bypass the DQS delay chain to achieve 0° phase shift.

Update Enable Circuitry

Both the DQS delay settings and phase-offset settings pass through a register before going into the DQS delay chains. The registers are controlled by the update enable circuitry to allow enough time for any changes in the DQS delay setting bits to arrive at all the delay elements. This allows them to be adjusted at the same time. The update enable circuitry enables the registers to allow enough time for the DQS delay settings to travel from the DQS phase-shift circuitry or core logic to all the DQS logic blocks before the next change. It uses the input reference clock or a user clock from the core to generate the update enable output. The ALTMEMPHY megafunction uses this circuit by default. See Figure 8–14 for an example waveform of the update enable circuitry output.

Figure 8–14. Example of a DQS Update Enable Waveform



DQS Postamble Circuitry

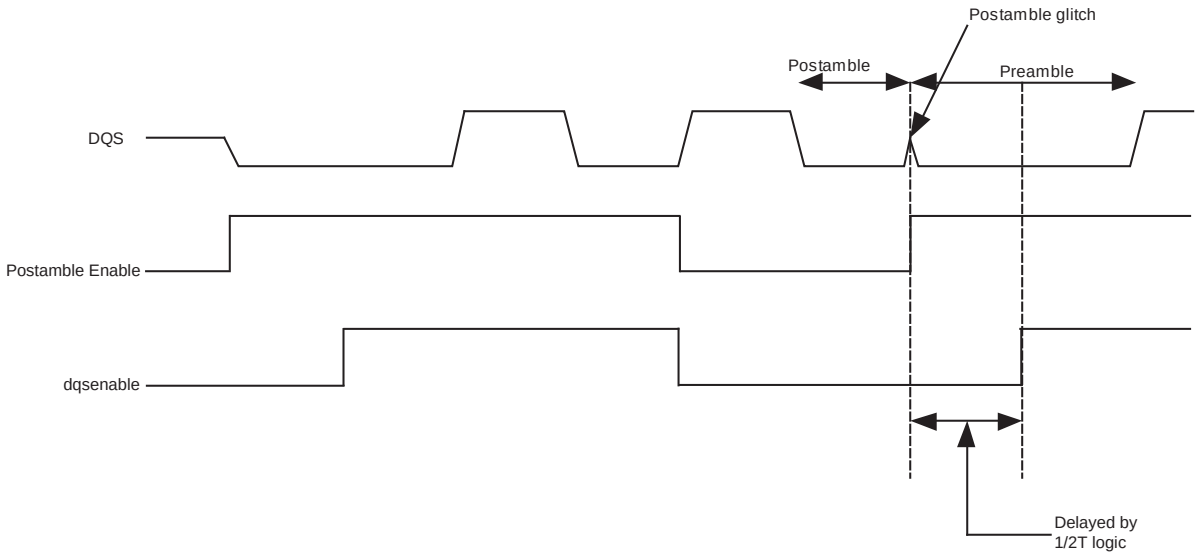
For external memory interfaces that use a bi-directional read strobe like DDR3, DDR2, and DDR SDRAM, the DQS signal is low before going to or coming from a high-impedance state. The state where DQS is low, just after a high-impedance state, is called the preamble. The state where DQS is low, just before it returns to a high-impedance state, is called the postamble. There are preamble and postamble specifications for both read and write operations in DDR3, DDR2, and DDR SDRAM. The DQS postamble circuitry ensures that the data is not lost if there is noise on the DQS line during the end of a read operation that occurs while the DQS is in a postamble state.

Stratix III devices have a dedicated postamble register that you can control to ground the shifted DQS signal used to clock the DQ input registers at the end of a read operation. This ensures that any glitches on the DQS input signals at the end of the read postamble time do not affect the DQ IOE registers.

In addition to the dedicated postamble register, Stratix III devices also have an HDR block inside the postamble enable circuitry. These registers are used if the controller is running at half the frequency of the I/Os.

Using the HDR block as the first stage capture register in the postamble enable circuitry block is optional. The HDR block is clocked by the half-rate resynchronization clock, which is the output of the I/O clock divider circuit (shown in Figure 8–20 on page 8–35). There is an AND gate after the postamble register outputs that is used to avoid postamble glitches from a previous read burst on a non-consecutive read burst. This scheme allows a half-a-clock cycle latency for `dqsenable` assertion and zero latency for `dqsenable` deassertion, as shown in Figure 8–15.

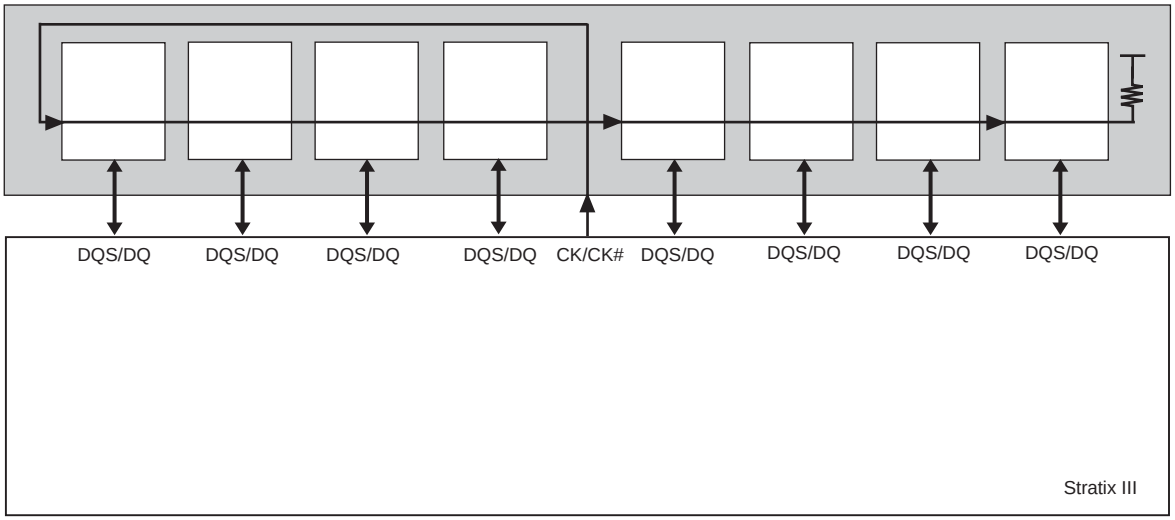
Figure 8–15. Avoiding a Glitch on a Non-Consecutive Read Burst Waveform



Leveling Circuitry

DDR3 SDRAM unbuffered modules use a fly-by clock distribution topology for better signal integrity. This means that the CK/CK# signals arrive at each DDR3 SDRAM device in the module at different times. The difference in arrival time between the first DDR3 SDRAM device and the last device on the module can be as long as 1.6 ns. Figure 8–16 shows the clock topology in DDR3 SDRAM unbuffered modules.

Figure 8–16. DDR3 SDRAM Unbuffered Module Clock Topology



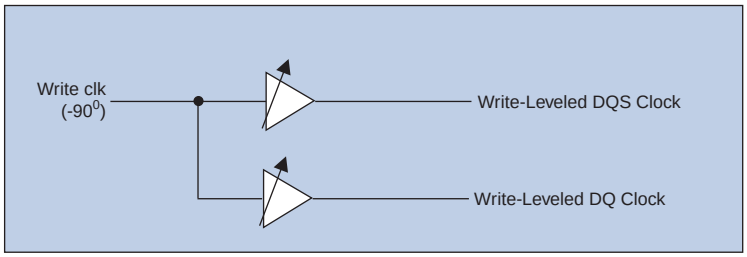
Because the data and read strobe signals are still point-to-point, special consideration must be taken to ensure that the timing relationship between CK/CK# and DQS signals (t_{DQSS}) during a write is met at every device on the modules. Furthermore, read data coming back into the FPGA from the memory is also staggered in a similar way.

Stratix III FPGAs have leveling circuitry to take care of these two requirements. There is one group of leveling circuitry per I/O bank, with the same I/O number (for example, there is one leveling circuitry shared between I/O bank 1A, 1B, and 1C) located in the middle of the I/O bank. These delay chains are PVT-compensated by the same DQS delay settings as the DLL and DQS delay chains. For frequencies equal to and above 400 MHz, the DLL uses eight delay chains such that each delay chain generates a 45° delay.

The generated clock phases are distributed to every DQS logic block that is available in the I/O bank. The delay chain taps, then feeds a multiplexer controlled by the ALTMEMPHY megafunction to select which clock phases are to be used for that ×4 or ×8 DQS group. Each group can use a different tap output from the read-leveling and write-leveling delay chains to compensate for the different CK/CK# delay going into each device on the module.

Figure 8–17 illustrates the Stratix III write leveling circuitry.

Figure 8–17. Stratix III Write Leveling Delay Chains and Multiplexers *(Note 1)*

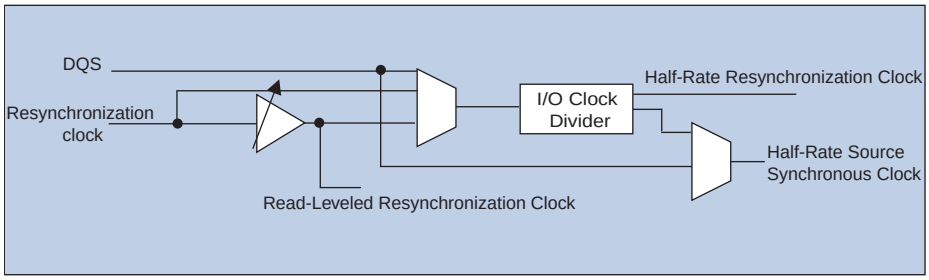


Note to Figure 8–17:

(1) There is only one leveling delay chain per I/O bank with the same I/O number (for example, I/O banks 1A, 1B, and 1C). You can only have one memory controller in these I/O banks when you use leveling delay chains.

Figure 8–18 illustrates the Stratix III read leveling circuitry.

Figure 8–18. Stratix III Read Leveling Delay Chains and Multiplexers *(Note 1)*



Note to Figure 8–18:

(1) There is only one leveling delay chain per I/O bank with the same I/O number (for example, I/O banks 1A, 1B, and 1C). You can only have one memory controller in these I/O banks when you use leveling delay chains.

The –90° write clock of the ALTMEMPHY megafunction feeds the write-leveling circuitry to produce the clock that generates the DQS and DQ signals. During initialization, the ALTMEMPHY megafunction picks the correct write-leveled clock for the DQS and DQ clocks for each DQS/DQ group after sweeping all the available clocks in the write calibration process. The DQ clock output is –90° phase-shifted compared to the DQS clock output.

Similarly, the resynchronization clock feeds the read-leveling circuitry to produce the optimal resynchronization and postamble clock for each DQS/DQ group in the calibration process. Resynchronization and the postamble clocks can use different clock outputs from the leveling circuitry. Output from the read-leveling circuitry can also generate the half-rate resynchronization clock that goes to the FPGA fabric.

The ALTMEMPHY megafunction calibrates the alignment for read and write leveling dynamically during the initialization process.

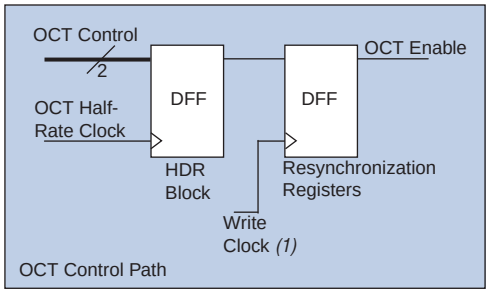
For more information about the ALTMEMPHY megafunction, refer to the *Volume 3: Implementing Altera Memory Interface IP*.

Dynamic OCT Control

Figure 8–19 shows the dynamic OCT control block. The block includes all the registers required to dynamically turn on OCT R_T during a read and turn OCT R_T off during a write.

For more information about dynamic OCT control, refer to the *Stratix III Device I/O Features* chapter.

Figure 8–19. Stratix III Dynamic OCT Control Block



Note to Figure 8–19:

(1) The write clock comes from either the PLL or the write leveling delay chain.

IOE Registers

The IOE registers have been expanded to allow source-synchronous systems to have faster register-to-register transfers and resynchronization. Both top/bottom and left/right IOEs have the same capability with left/right IOEs having extra features to support LVDS data transfer.

Figure 8–20 shows the registers available in the Stratix III input path. The input path consists of the DDR input registers, resynchronization registers, and HDR block. You can bypass each block of the input path.

Figure 8–20. Stratix III IOE Input Registers *(Note 1)*

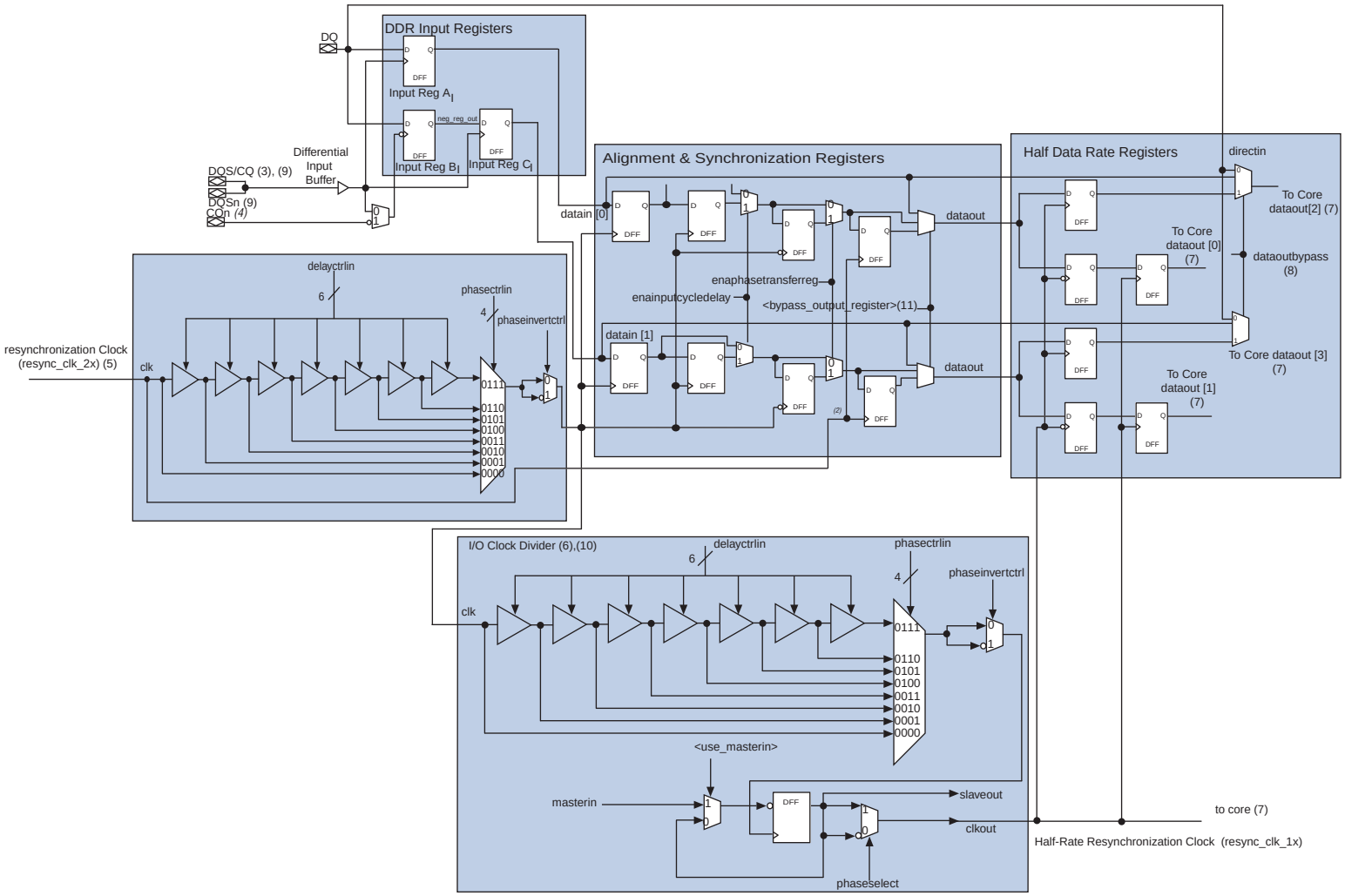


Figure 8–20. Stratix III IOE Input Registers *(Note 1)*

Notes to Figure 8–20:

(1) You can bypass each register block in this path.

(2) This is the 0-phase resynchronization clock (from the read-leveling delay chain).

(3) The input clock can be from the DQS logic block (whether the postamble circuitry is bypassed or not) or from a global clock line.

(4) This input clock comes from the CQn logic block.

(5) This resynchronization clock can come either from the PLL or from the read-leveling delay chain.

(6) The I/O clock divider resides adjacent to the DQS logic block. In addition to the PLL and read-leveled resync clock, the I/O clock divider can also be fed by the DQS bus or CQn bus.

(7) The half-rate data and clock signals feed into a dual-port RAM in the FPGA core.

(8) You can dynamically change the `dataoutbypass` signal after configuration to select either the `directin` input or the output from the half data rate register to feed `dataout`.

(9) You must invert the strobe signal needs for DDR, DDR2, and DDR3 interfaces, except for QDR II or QDR II+ SRAM interfaces. This inversion is automatically done if you use the Altera external memory interface IPs.

(10) Each divider feeds up to six pins (from a $\times 4$ DQS group) in the device. To feed wider DQS groups, you must chain multiple clock dividers together by feeding the `slaveout` output of one divider to the `masterin` input of the neighboring pins' divider.

(11) The **bypass_output_register** option allows you to select either the output from the second mux or the output of the fourth alignment/ synchronization register to feed `dataout`.

© March 2010 Altera Corporation

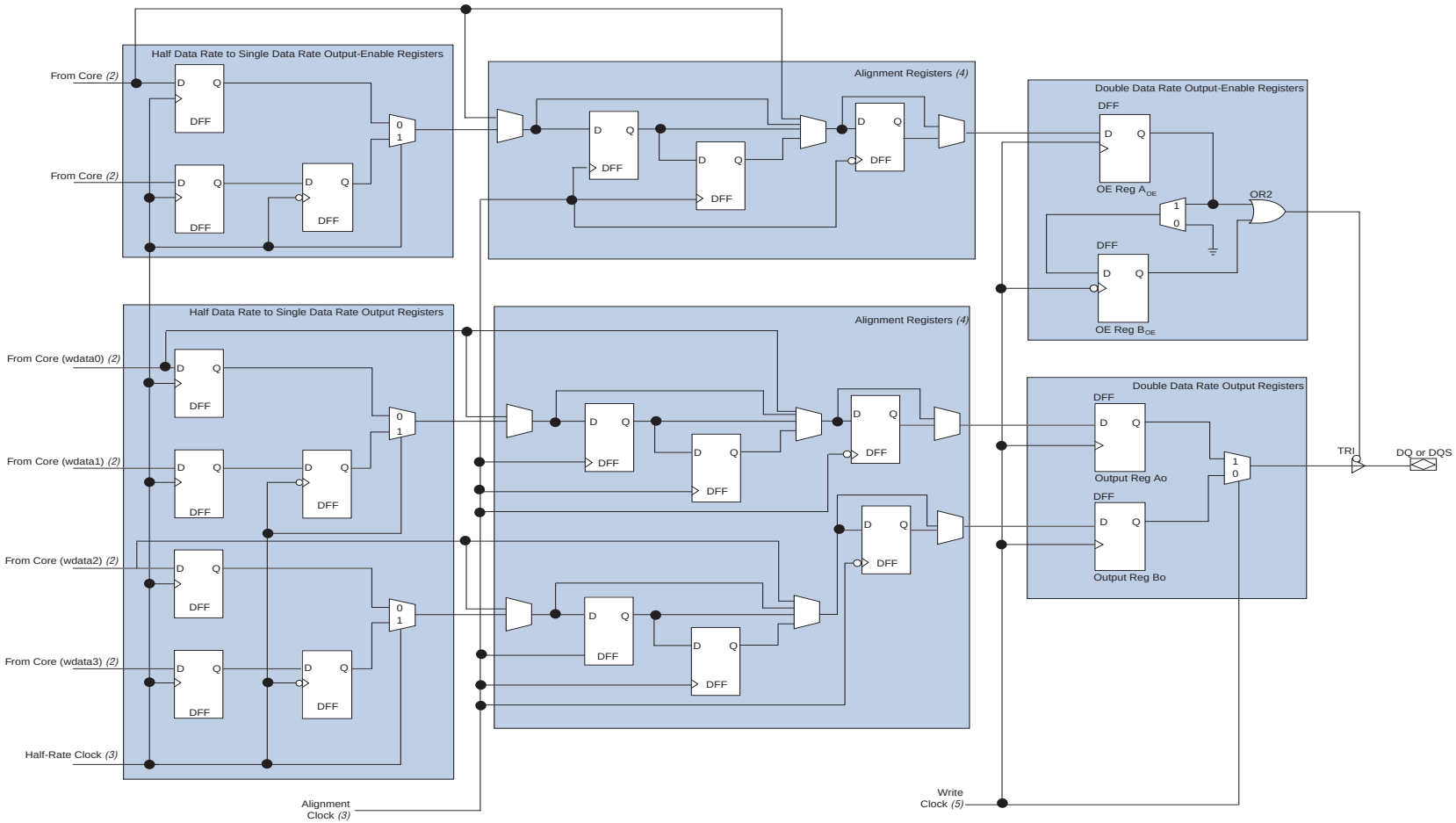
There are three registers in the DDR input registers block. Two registers capture data on the positive and negative edges of the clock, while the third register aligns the captured data. You can choose to have the same clock for the positive edge and negative edge registers, or two different clocks (DQS for positive edge register, and CQn for negative edge register). The third register that aligns the captured data uses the same clock as the positive edge registers.

Resynchronization registers consist of up to three levels of registers to resynchronize the data to the system clock domain. These registers are clocked by the resynchronization clock that is either generated by the PLL or the read-leveling delay chain. The outputs of the resynchronization registers can go straight to the core or to the HDR blocks, which are clocked by the divided-down resynchronization clock.

For more information about the read-leveling delay chain, refer to "Leveling Circuitry" on page 8–31.

Figure 8–21 shows the registers available in the Stratix III output and output-enable paths. The path is divided into the HDR block, resynchronization registers, and output/output-enable registers. The device can bypass each block of the output and output-enable path.

Figure 8–21. Stratix III IOE Output and Output-Enable Path Registers *(Note 1)*



Notes to Figure 8–21:

(1) You can bypass each register block of the output and output-enable paths.

(2) Data coming from the FPGA core are at half the frequency of the memory interface.

(3) Half-rate and alignment clocks come from the PLL.

(4) These registers are only used in DDR3 SDRAM interfaces for write-leveling purposes.

(5) The write clock can come from either the PLL or from the write-leveling delay chain. The DQ write clock and DQS write clock have a 90° offset between them.

© March 2010 Altera Corporation

The output path is designed to route combinatorial or registered single data rate (SDR) outputs and full-rate or half-rate DDR outputs from the FPGA core. Half-rate data is converted to full-rate using the HDR block and is clocked by the half-rate clock from the PLL. Resynchronization registers are also clocked by the same 0° system clock, except in the DDR3 SDRAM interface where the leveling registers are clocked by the write-leveling clock.

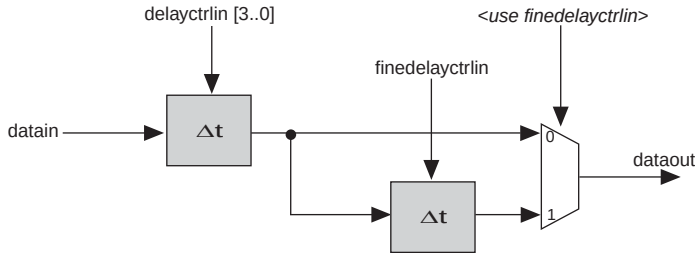
For more information about the write leveling delay chain, refer to "Leveling Circuitry" on page 8–31.

The output-enable path has structure similar to the output path. You can have a combinatorial or registered output in SDR applications and you can use half-rate or full-rate operation in DDR applications. You also have the resynchronization registers similar to the output path registers structure, ensuring that the output-enable path goes through the same delay and latency as the output path.

Delay Chain

Stratix III devices have run-time adjustable delay chains in the I/O blocks and the DQS logic blocks. You can control the delay chain setting through the I/O or the DQS configuration block output. Figure 8–22 shows the delay chain ports.

Figure 8–22. Delay Chain

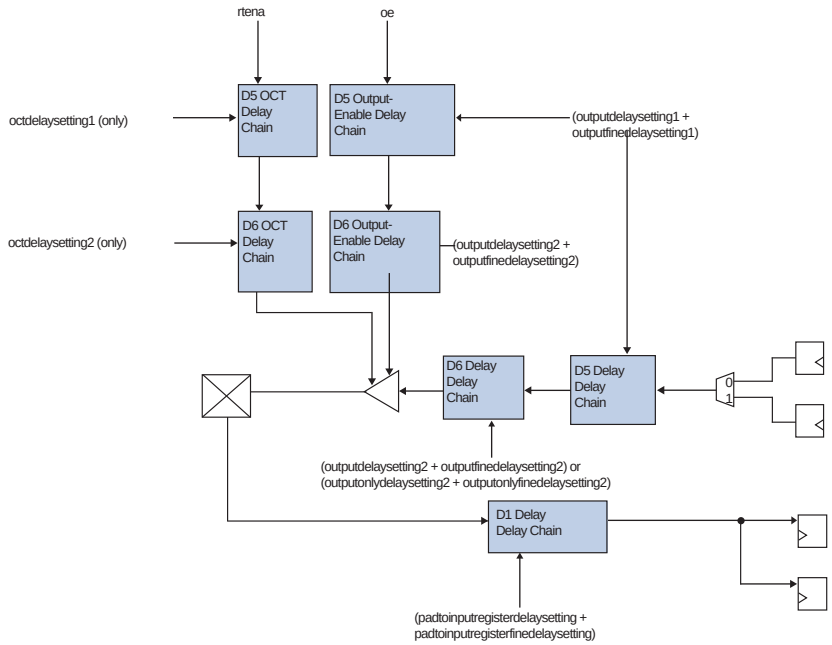


Every I/O block contains the following:

- Two delay chains in series between the output registers and output buffer
- One delay chain between the input buffer and input register
- Two delay chains between the output enable and output buffer
- Two delay chains between the OCT R_T enable control register and output buffer

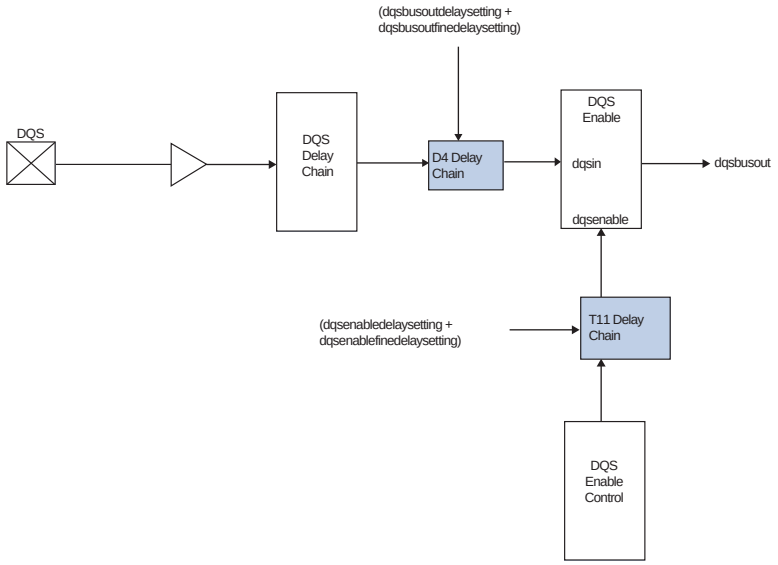
Figure 8–23 shows the delay chains in an I/O block.

Figure 8–23. Delay Chains in an I/O Block



Each DQS logic block contains a delay chain after the `dqsbusout` output and another delay chain before the `dqsenable` input. Figure 8–24 shows the delay chains in the DQS input path.

Figure 8–24. Delay Chains in the DQS Input Path



I/O Configuration Block and DQS Configuration Block

The I/O configuration block and the DQS configuration block are shift registers that you can use to dynamically change the settings of various device configuration bits. The shift registers power-up low. Every I/O pin contains one I/O configuration register, while every DQS pin contains one DQS configuration block in addition to the I/O configuration register. Figure 8–25 shows the I/O configuration block and the DQS configuration block circuitry.

Figure 8–25. I/O Configuration Block and DQS Configuration Block

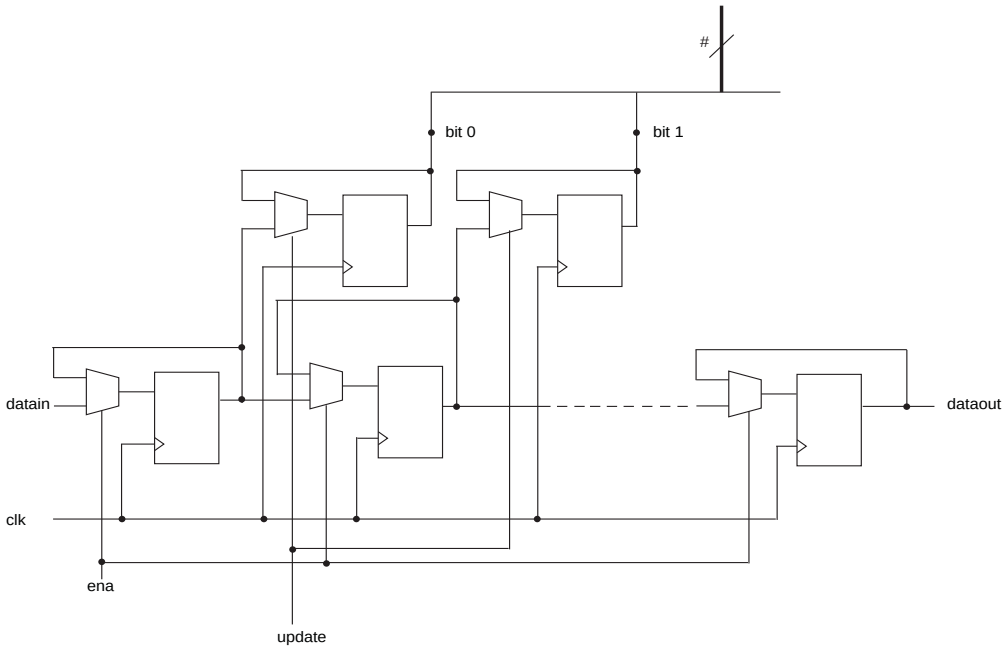


Table 8–11 lists the I/O configuration block bit sequence.

Table 8–11. I/O Configuration Block Bit Sequence

Bit	Bit Name
0..3	outputdelaysetting1[0..3]
4..6	outputdelaysetting2[0..2]
7..10	padtoinputregisterdelaysetting[0..3]
11	outputfinedelaysetting1
12	outputfinedelaysetting2
13	padtoinputregisterfinedelaysetting
14	outputonlyfinedelaysetting2
15..17	outputonlydelaysetting2[2..0]
18	dutycyclecorrectionmode
19..22	dutycyclecorrectionsetting[3..0]

Table 8–12 lists the DQS configuration block bit sequence.

Table 8–12. DQS Configuration Block Bit Sequence

Bit	Bit Name
0..3	dqsbusoutdelaysetting[0..3]
4..6	dqsinputphasesetting[0..2]
7..10	dqsenablectrlphasesetting[0..3]
11..14	dqsoutputphasesetting[0..3]
15..18	dqoutputphasesetting[0..3]
19..22	resyncinputphasesetting[0..3]
23	dividerphasesetting
24	enaoctcycledelaysetting
25	enainputcycledelaysetting
26	enaoutputcycledelaysetting
27..29	dqsenabledelaysetting[0..2]
30..33	octdelaysetting1[0..3]
34..36	octdelaysetting2[0..2]
37	enadataoutbypass
38	enadqsenablephasetransferreg
39	enaoctphasetransferreg
40	enaoutputphasetransferreg
41	enainputphasetransferreg
42	resyncinputphaseinvert
43	dqsenablectrlphaseinvert
44	dqoutputphaseinvert
45	dqsoutputphaseinvert
46	dqsbusoutfinedelaysetting
47	dqsenablefinedelaysetting

IOE Features

This section briefly describes how OCT, programmable delay chains, programmable output delay, slew rate adjustment, and programmable drive strength are useful in memory interfaces.

For more information about the features listed below, refer to the *Stratix III Device I/O Features* chapter.

OCT

Stratix III devices feature dynamic calibrated OCT, in which series termination (OCT R_S) is turned on when driving signals and turned off when receiving signals, while the parallel termination (OCT R_T) is turned off when driving signals and turned on when receiving signals. This feature complements the DDR3/DDR2 SDRAM on-die termination (ODT), whereby memory termination is turned off when the memory is sending data and turned on when receiving data. You can also use OCT for other memory interfaces to improve signal integrity.

☞ You cannot use the programmable drive strength and programmable slew rate features when using OCT R_S.

To use dynamic calibrated OCT, you must use the R_{UP} and R_{DN} pins to calibrate the OCT calibration block. You can use one OCT calibration block to calibrate one type of termination with the same V_{CCIO} on the entire device. There are up to ten OCT calibration blocks to allow for different types of terminations throughout the device. For more information, refer to "Dynamic OCT Control" on page 8–33.

☞ You have the option to use the OCT R_S feature with or without calibration. However, the OCT R_T feature is only available with calibration.

You can also use the R_{UP} and R_{DN} pins as DQ pins. However, you cannot use the ×4 DQS/DQ groups where the R_{UP} and R_{DN} pins are located if you are planning to use dynamic calibrated OCT. The R_{UP} and R_{DN} pins are located in the first and last ×4 DQS/DQ group on each side of the device.

Use the OCT RT/RS setting for uni-directional read and write data; use a dynamic OCT setting for bi-directional data signals.

Programmable IOE Delay Chains

You can use programmable delay chains in the Stratix III I/O registers as deskewing circuitry. Each pin can have a different input delay from the pin to input register or a delay from the output register to the output pin to ensure that the bus has the same delay going into or out of the FPGA. This feature helps read and write time margins as it minimizes the uncertainties between signals in the bus.

☞ Deskewing circuitry and programmable IOE delay chains are the same circuit.

Programmable Output Buffer Delay

In addition to allowing output buffer duty cycle adjustment, the programmable output buffer delay chain allows you to adjust the delays between data bits in your output bus to introduce or compensate channel-to-channel skew. Incorporating skew to the output bus helps to minimize simultaneous switching events by enabling smaller parts of the bus to switch simultaneously, instead of the whole bus. This feature is particularly useful in DDR3 SDRAM interfaces where the memory system clock delay can be much larger than the data and data clock/strobe delay. Use this delay chain to add delay to the data and data clock/strobe to better match the memory system clock delay.

Programmable Slew Rate Control

Stratix III devices provide four levels of static output slew rate control: 0, 1, 2, and 3, where 0 is the slowest slew rate setting and 3 is the fastest slew rate setting. The default setting for the HSTL and SSTL I/O standards is 3. A fast slew rate setting allows you to achieve higher I/O performance; a slow slew-rate setting reduces system noise and signal overshoot. This feature is disabled if you are using the OCT R_S features.

Programmable Drive Strength

You can choose the optimal drive strength required for your interface after performing a board simulation. Higher drive strength helps provide a larger voltage swing, which in turn provides bigger eye diagrams with greater timing margin. However, higher drive strengths typically require more power, faster slew rates, and add to simultaneous switching noise. You can use programmable slew rate control along with this feature to minimize simultaneous switching noise with higher drive strengths.

This feature is disabled if you use the OCT R_S feature, which is the default drive strength in Stratix III devices. Use the OCT R_T/R_S setting for uni-directional read/write data; use the dynamic OCT setting for bi-directional data signals. You must simulate the system to determine the drive strength required for command, address, and clock signals.

PLL

PLLs are used to generate the memory interface controller clocks, similar to the 0° system clock, the –90° or 270° phase-shifted write clock, the half-rate PHY clock, and the resynchronization clock. You can use the PLL reconfiguration feature to calibrate resynchronization phase shift to balance the setup and hold margin.

The VCO and counter setting combinations may be limited for high-performance memory interfaces.

For more information about the Stratix III PLL, refer to the *Clock Networks and PLLs in Stratix III Devices* chapter.

Chapter Revision History

Table 8–13 lists the revision history for this chapter.

Table 8–13. Chapter Revision History (Part 1 of 2)

Date and Revision	Changes Made	Summary of Changes
March 2010, version 1.9	■ Added "Delay Chain" section. ■ Updated "DLL", "DQS Logic Block", and "Dynamic OCT Control" sections. ■ Added Figure 8–22, Figure 8–23, Figure 8–24, and Figure 8–25. ■ Updated Figure 8–12, Figure 8–13,and Figure 8–20. ■ Added Table 8–11 and Table 8–12. ■ Updated Table 8–7. ■ Minor text edits.	Updated for the Quartus II software version 9.1 SP2 release.
May 2009, version 1.8	■ Updated Table 8–1, Table 8–2, Table 8–3, Table 8–2, Table 8–4, and Table 8–10. ■ Updated Figure 8–3, Figure 8–4, Figure 8–5, Figure 8–6, and Figure 8–7. ■ Updated "DLL", "Memory Interfaces Pin Support", and "Rules to Combine Groups"sections.	—
February 2009, version 1.7	■ Updated Table 8–1,Table 8–2, and Table 8–6. ■ Updated "Data and Data-Strobe/Clock Pins" section. ■ Removed "Referenced Document" section.	—
October 2008, version 1.6	■ Updated Table 8–1, Table 8–2, Table 8–3, Table 8–4, Table 8–5, Table 8–7, and Table 8–8. ■ Updated the "Rules to Combine Groups", "Phase Offset Control", "OCT", "Introduction", "Memory Interfaces Pin Support", "Combining ×16/×18 DQS/DQ groups for ×36 QDR II+/QDR II SRAM Interface", "Rules to Combine Groups", "DQS Phase-Shift Circuitry", "DLL", and "DQS Delay Chain" sections. ■ Updated Figure 8–2, Figure 8–4, Figure 8–10, Figure 8–21, and Figure 8–22. ■ Updated New Document Format. ■ Added (Note 3) to Table 8–5.	—
July 2008, version 1.5	Updated Table 8–1 and Table 8–2.	—
May 2008, version 1.4	■ Updated Figure 8–2, Figure 8–9, Figure 8–18, Figure 8–21, and Figure 8–22. ■ Updated Table 8–1, Table 8–2, Table 8–3, Table 8–4, Table 8–7, and Table 8–10. ■ Added Table 8–7 and Table 8–8. ■ Added Figure 8–19. ■ Added new "Supporting ×36 QDR II+/QDR II SRAM Interfaces in the F780 and F1152-Pin Packages" section. ■ Updated "Data and Data Clock/Strobe Pins". ■ Updated "Referenced Documents".	Text, Table, and Figure updates.

Table 8–13. Chapter Revision History (Part 2 of 2)

Date and Revision	Changes Made	Summary of Changes
November 2007, version 1.3	■ Updated Table 8–5. ■ Updated Figure 8–6.	Minor updates to content.
October 2007, version 1.2	■ Updated Table 8–1, Table 8–3, Table 8–4, Table 8–5. ■ Added Table 8–2. ■ Minor text edits. ■ Updated Figure 8–3, note 3 to Figure 8–4, note 3 to Figure 8–5, note 2 to Figure 8–6, added a note to Figure 8–7, added a note and updated Figure 8–10, notes to Figure 8–11, and updated Figure 8–12. ■ Added new material to “Memory Clock Pins” on page 8–21. ■ Added section “Referenced Documents”. ■ Added live links for references.	Minor updates to content.
May 2007, version 1.1	■ Updated Figure 8–5, Figure 8–8, Figure 8–14, Figure 8–18, Figure 8–19, Figure 8–20, and Figure 8–21. ■ Added new figure, Figure 8–17. ■ Added memory support information for -4L in Table 8–1, Table 8–8, Table 8–10, and Table 8–11. ■ Added new material to section “Phase Offset Control” on page 8–32.	Minor updates to content.
November 2006, version 1.0	Initial Release.	—

9. High-Speed Differential I/O Interfaces and DPA in Stratix III Devices

SIII51009-1.9

Stratix® III devices offers up to 1.6-Gbps differential I/O capabilities to support source-synchronous communication protocols such as Utopia, Rapid I/O®, XSBI, SGMII, SFI, and SPI.

Stratix III devices have the following dedicated circuitry for high-speed differential I/O support:

- Differential I/O buffer
- Transmitter serializer
- Receiver deserializer
- Data realignment
- Dynamic phase aligner (DPA)
- Synchronizer (FIFO buffer)
- Analog Phase-Locked Loops (PLLs) (located on left and right sides of the device)

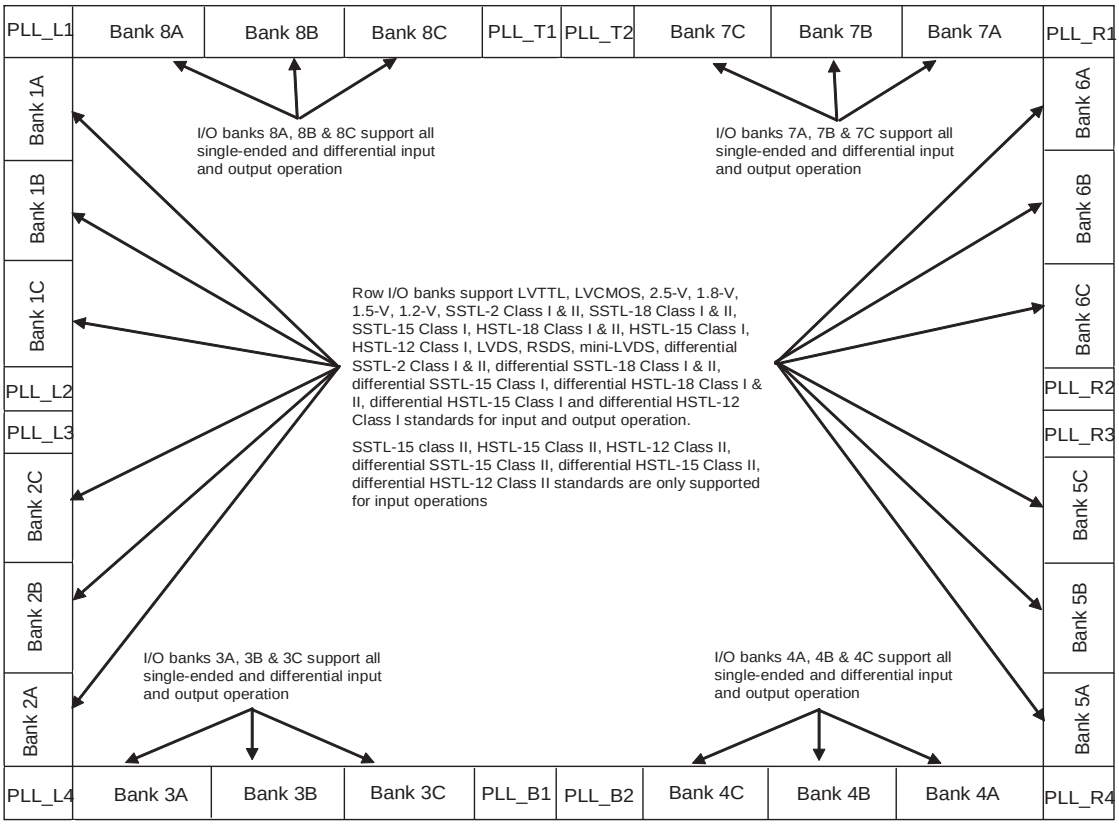
For high-speed differential interfaces, Stratix III devices support the following differential I/O standards:

- Low voltage differential signaling (LVDS)
- Mini-LVDS
- Reduced swing differential signaling (RSDS)
- High-speed Transceiver Logic (HSTL)
- Stub Series Terminated Logic (SSTL)

I/O Banks

Stratix III I/Os are divided into 16 to 24 I/O banks. The dedicated serializer and deserializer (SERDES) circuitry with DPA that supports high-speed differential I/Os is located in banks in the right side and left side of the device. Figure 9–1 shows the different banks and the I/O standards supported by the banks.

Figure 9–1. I/O Banks in Stratix III Devices *(Note 1)*, *(2)*, *(3)*, *(4)*, *(5)*, *(6)*



Notes to Figure 9–1:

(1) Figure 9–1 is a top view of the silicon die that corresponds to a reverse view for flip chip packages. It is a graphical representation only. For exact locations, refer to the pin list and the Quartus II software.

(2) Differential HSTL and SSTL outputs use two single-ended (SE) outputs with the second output programmed as inverted to support differential I/O operations.

(3) Column I/O differential HSTL and SSTL inputs use LVDS differential input buffers without on-chip differential termination (OCT R_D) support.

(4) Column I/O supports LVDS outputs using SE buffers and external resistor networks.

(5) Row I/O supports PCI/PCI-X without on-chip clamping diodes.

(6) The PLL blocks are shown for location purposes only and are not considered additional banks. The PLL input and output uses the I/Os in adjacent banks.

LVDS Channels

The Stratix III device supports LVDS on both side I/O banks and column I/O banks. Single-ended reference clocks are supported when using the source-synchronous SERDES in DPA and soft-CDR mode. There are true LVDS input and output buffers on the side I/O banks. On column I/O banks, there are true LVDS input buffers but do not have true LVDS output buffers. However, you can configure all column user I/Os—including I/Os with true LVDS input buffers—as emulated LVDS output buffers. When using emulated LVDS standards, you must implement the logic driving these pins in soft logic (logic elements) and not hard SERDES.

☞ Emulated differential output buffers support tri-state capability starting with the Quartus® II software version 9.1.

Table 9–1 lists the LVDS channels supported in Stratix III device side I/O banks.

Table 9–1. LVDS Channels Supported in Stratix III Device Side I/O Banks *(Note 1)*, *(2)*, *(3)*

Device	484-Pin FineLine BGA	780-Pin FineLine BGA	1152-Pin FineLine BGA	1517-Pin FineLine BGA	1780-Pin FineLine BGA
EP3SL50	48Rx/eTx + 48Tx/eTx	56Rx/eTx + 56Tx/eTx	—	—	—
EP3SL70	48Rx/eTx + 48Tx/eTx	56Rx/eTx + 56Tx/eTx	—	—	—
EP3SL110	—	56Rx/eTx + 56Tx/eTx	88Rx/eTx + 88Tx/eTx	—	—
EP3SL150	—	56Rx/eTx + 56Tx/eTx	88Rx/eTx + 88Tx/eTx	—	—
EP3SL200	—	56Rx/eTx + 56Tx/eTx *(4)*	88Rx/eTx + 88Tx/eTx	112Rx/eTx + 112Tx/eTx	—
EP3SL340	—	—	88Rx/eTx + 88Tx/eTx *(5)*	112Rx/eTx + 112Tx/eTx	132Rx/eTx + 132Tx/eTx
EP3SE50	48Rx/eTx + 48Tx/eTx	56Rx/eTx + 56Tx/eTx	—	—	—
EP3SE80	—	56Rx/eTx + 56Tx/eTx	88Rx/eTx + 88Tx/eTx	—	—
EP3SE110	—	56Rx/eTx + 56Tx/eTx	88Rx/eTx + 88Tx/eTx	—	—
EP3SE260	—	56Rx/eTx + 56Tx/eTx *(4)*	88Rx/eTx + 88Tx/eTx	112Rx/eTx + 112Tx/eTx	—

Notes to Table 9–1:

(1) Rx = true LVDS input buffers.
(2) Tx = true LVDS output buffers.
(3) eTx = emulated-LVDS output buffers, either `LVDS_E3R` or `LVDS_E1R`.
(4) The EP3SL200 and EP3SL260 FPGAs are offered in the H780 package, instead of the F780 package.
(5) The EP3SL340 FPGA is offered in the H1152 package, instead of the F1152 package.

Table 9–2 lists the LVDS channels (emulated) supported in Stratix III device column I/O banks.

Table 9–2. LVDS Channels (Emulated) Supported in Stratix III Device Column I/O Banks *(Note 1)*, *(2)*

Device	484-Pin FineLine BGA	780-Pin FineLine BGA	1152-Pin FineLine BGA	1517-Pin FineLine BGA	1780-Pin FineLine BGA
EP3SL50	24Rx/eTx + 24eTx	64Rx/eTx + 64eTx	—	—	—
EP3SL70	24Rx/eTx + 24eTx	64Rx/eTx + 64eTx	—	—	—
EP3SL110	—	64Rx/eTx + 64eTx	96Rx/eTx + 96eTx	—	—
EP3SL150	—	64Rx/eTx + 64eTx	96Rx/eTx + 96eTx	—	—
EP3SL200	—	64Rx/eTx + 64eTx *(3)*	96Rx/eTx + 96eTx	128Rx/eTx + 128eTx	—
EP3SL340	—	—	96Rx/eTx + 96eTx *(4)*	128Rx/eTx + 128eTx	144Rx/eTx + 144eTx
EP3SE50	24Rx/eTx + 24eTx	64Rx/eTx + 64eTx	—	—	—
EP3SE80	—	64Rx/eTx + 64eTx	96Rx/eTx + 96eTx	—	—
EP3SE110	—	64Rx/eTx + 64eTx	96Rx/eTx + 96eTx	—	—
EP3SE260	—	64Rx/eTx + 64eTx *(3)*	96Rx/eTx + 96eTx	128Rx/eTx + 128eTx	—

Notes to Table 9–2:

(1) Rx = true LVDS input buffers without on-chip differential input termination.

(2) eTx = emulated LVDS output buffers, either `LVDS_E3R` or `LVDS_E1R`.

(3) The EP3SL200 and EP3SE260 FPGAs are offered in the H780 package, instead of the F780 package.

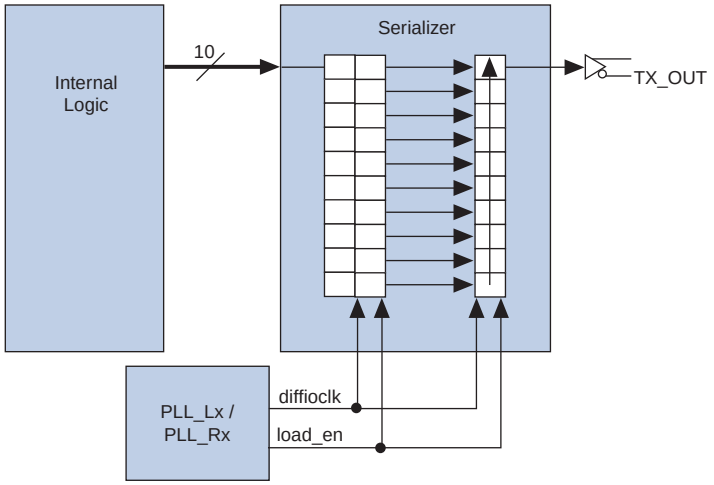
(4) The EP3SL340 FPGA is offered in the H1152 package, instead of the F1152 package.

Differential Transmitter

The Stratix III transmitter has dedicated circuitry to provide support for LVDS signaling. The dedicated circuitry consists of a differential buffer, a serializer, and a shared analog PLL (left/right PLL). The differential buffer can drive out LVDS, mini-LVDS, and RSDS signaling levels. The serializer takes up to 10-bits wide parallel data from the FPGA core, clocks it into the load registers, and serializes it using shift registers clocked by the left/right PLL before sending the data to the differential buffer. The most significant bit (MSB) of the parallel data is transmitted first.

The load and shift registers are clocked by the load enable (`load_en`) signal and the `diffioclk` (clock running at serial data rate) signal generated from PLL_Lx (left PLL) or PLL_Rx (right PLL). The serialization factor can be statically set to ×3, ×4, ×5, ×6, ×7, ×8, ×9, or ×10 with the Quartus II software. The load enable signal is derived from the serialization factor setting. Figure 9–2 shows a block diagram of the Stratix III transmitter.

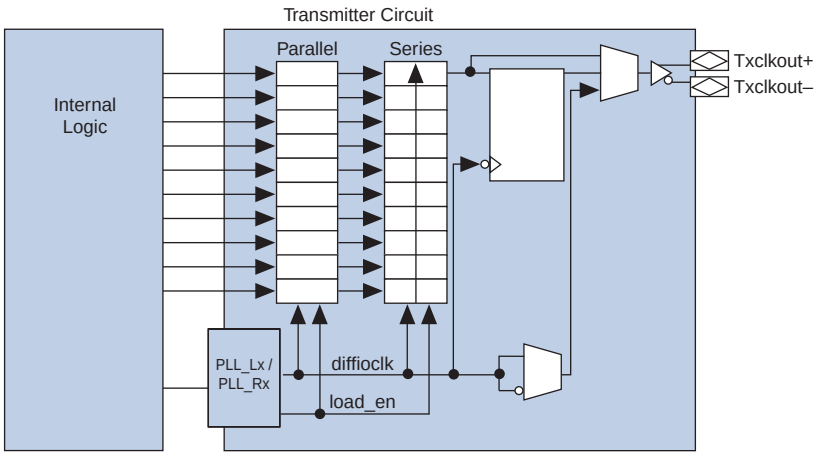
Figure 9–2. Transmitter Block Diagram for Stratix III Devices



You can configure any Stratix III transmitter data channel to generate a source-synchronous transmitter clock output. This flexibility allows placing the output clock near the data outputs to simplify board layout and reduce clock-to-data skew. Different applications often require specific clock-to-data alignments or specific data rate-to-clock rate factors. The transmitter can output a clock signal at the same rate as the data with a maximum frequency of 800 MHz. You can also divide the output clock by a factor of 2, 4, 8, or 10, depending on the serialization factor. The phase of the clock in relation to the data can be set at 0° or 180° (edge or center aligned). The left and right PLLs (PLL_Lx/PLL_Rx) provide additional support for other phase shifts in 45° increments. These settings are statically made in the MegaWizard® Plug-In Manager in the Quartus II software.

Figure 9–3 shows the Stratix III transmitter in clock output mode.

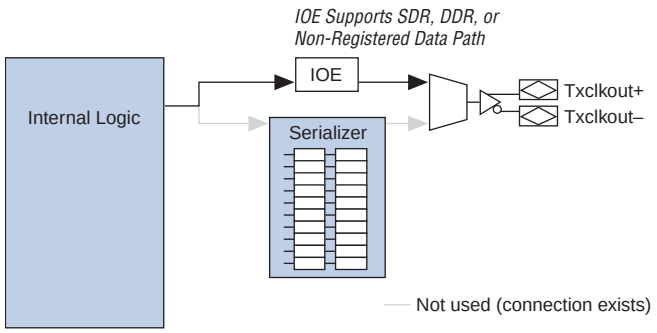
Figure 9–3. Transmitter in Clock Output Mode for Stratix III Devices



You can bypass the Stratix III serializer to support DDR (×2) and SDR (×1) operations to achieve a serialization factor of 2 and 1, respectively. The I/O element (IOE) contains two data output registers that can each operate in either DDR or SDR mode. The clock source for the registers in the IOE can come from any routing resource, from the left/right PLL (PLL_Lx/PLL_Rx) or from the top/bottom (PLL_Tx/PLL_Bx) PLL.

Figure 9–4 shows the serializer bypass path.

Figure 9–4. Serializer Bypass for Stratix III Devices



For more information about how to use the differential transmitter, refer to the *ALTLVDS Megafunction User Guide*.

Differential Receiver

The Stratix III device has dedicated circuitry to receive high-speed differential signals. The receiver has a differential buffer, a shared PLL_Lx/PLL_Rx, DPA block, synchronization FIFO buffer, data realignment block, and a deserializer. The differential buffer can receive LVDS, mini-LVDS, and RSDS signal levels, which are statically set in the Quartus II software Assignment Editor. The PLL receives the external source clock input that is transmitted with the data and generates different phases of the same clock. The DPA block chooses one of the clocks from the left/right PLL and aligns the incoming data on each channel.

The synchronizer circuit is a 1-bit wide by 6-bit deep FIFO buffer that compensates for any phase difference between the DPA clock and the data realignment block. If necessary, the data realignment circuit can insert a single bit of latency in the serial bit stream to align to the word boundary. The deserializer includes shift registers and parallel load registers, and sends a maximum of 10 bits to the internal logic. The data path in the Stratix III receiver is clocked by either a `dffioclk` signal or the DPA recovered clock. The deserialization factor can be statically set to ×3, ×4, ×5, ×6, ×7, ×8, ×9, or ×10 with the Quartus II software. The left/right PLLs (PLL_Lx/PLL_Rx) generate the load enable signal, which is derived from the deserialization factor setting.

To support DDR (×2) or SDR (×1) operations, you can bypass the Stratix III deserializer in MegaWizard Plug-In Manager in the Quartus II software. You cannot use the DPA and the data realignment circuit when the deserializer is bypassed. The IOE contains two data input registers that can operate in DDR or SDR mode. The clock source for the registers in the IOE can come from any routing resource, from the left/right PLLs or from the top/bottom PLLs. Figure 9–5 shows the block of the Stratix III receiver.

Figure 9–5. Receiver Block Diagram for Stratix III Devices

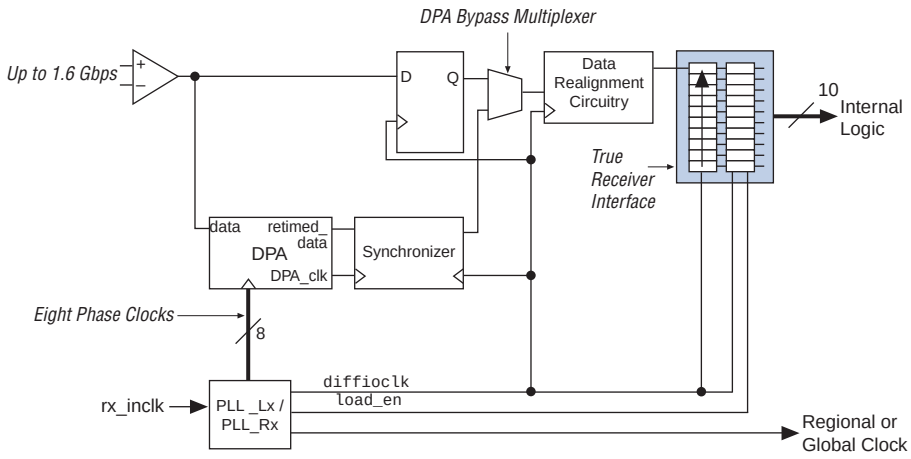
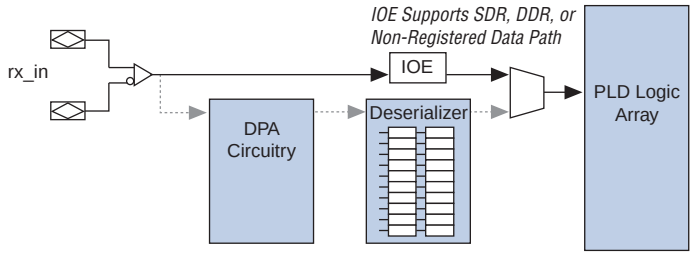


Figure 9–6 shows the deserializer bypass data path.

Figure 9–6. Deserializer Bypass for Stratix III Devices



Receiver Data Realignment Circuit (Bit Slip)

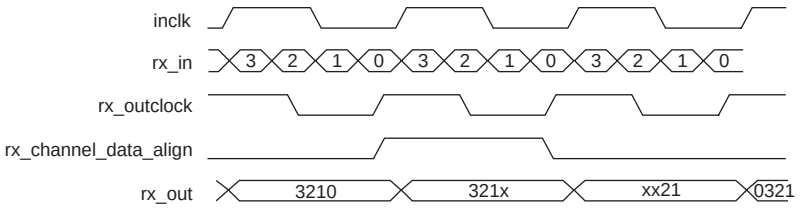
Skew in the transmitted data along with skew added by the link causes channel-to-channel skew on the received serial data streams. If DPA is enabled, the received data is captured with different clock phases on each channel. This may cause the received data to be misaligned from channel to channel. To compensate for this channel-to-channel skew and establish the correct received word boundary at each channel, each receiver channel has a dedicated data realignment circuit that realigns the data by inserting bit latencies into the serial stream.

An optional `RX_CHANNEL_DATA_ALIGN` port controls the bit insertion of each receiver independently controlled from the internal logic. The data slips one bit for every pulse on the `RX_CHANNEL_DATA_ALIGN`. The following are requirements for the `RX_CHANNEL_DATA_ALIGN` signal:

- The minimum pulse width is one period of the parallel clock in the logic array
- The minimum low time between pulses is one period of parallel clock
- There is no maximum high or low time
- Valid data is available two parallel clock cycles after the rising edge of `RX_CHANNEL_DATA_ALIGN`

Figure 9–7 shows receiver output (RX_OUT) after one bit slip pulse with the deserialization factor set to **4**.

Figure 9–7. Data Realignment Timing



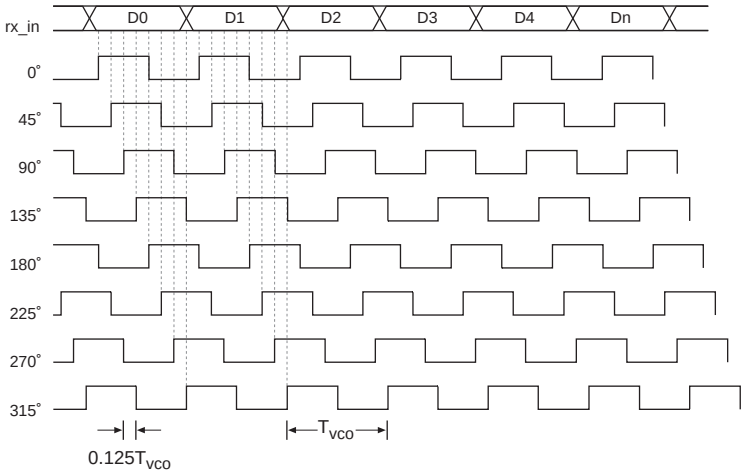
The data realignment circuit can have up to 11 bit-times of insertion before a rollover occurs. The programmable bit rollover point can be from 1 to 11 bit-times, independent of the deserialization factor. An optional status port, RX_CDA_MAX, is available to the FPGA from each channel to indicate when the preset rollover point is reached.

Dynamic Phase Aligner (DPA)

The DPA block takes in high-speed serial data from the differential input buffer and selects one of the eight phase clocks from the left/right PLL to sample the data. The DPA chooses the phase closest to the phase of the serial data. The maximum phase offset between the received data and the selected phase is 1/8 UI, which is the maximum quantization error of the DPA. The eight phases of the clock are equally divided, giving a 45° resolution.

Figure 9–8 shows the possible phase relationships between the DPA clocks and the incoming serial data.

Figure 9–8. DPA Clock Phase-to-Serial Data Timing Relationship



The DPA block continuously monitors the phase of the incoming serial data and selects a new clock phase if required. You can prevent the DPA from selecting a new clock phase by asserting the optional RX_DPLL_HOLD port, which is available for each channel.

The DPA circuitry does not require a fixed training pattern to lock to the optimum phase out of the 8 phases. After reset or power up, the DPA circuitry requires transitions on the received data to lock to the optimum phase. The ALTLVDS megafunction provides an optional output port, `rx_dpa_locked` to indicate if the DPA has locked to the optimum phase. When the DPA locks to the optimum phase, the `rx_dpa_locked` signal always stays high unless you assert the `rx_reset` signal of the associated LVDS channel or the `pll_areset` signal of the receiver PLL providing the 8 DPA clock phases.

The `rx_dpa_locked` signal only indicates an initial DPA lock condition to the optimum phase after power up or reset. You must not use the `rx_dpa_locked` signal to validate the integrity of the LVDS link. Use error checkers, for example cyclical redundancy check (CRC) and diagonal interleave parity (DIP4), to validate the integrity of the LVDS link.

An independent reset port (`RX_RESET`) is available to reset the DPA circuitry. You must retrain the DPA circuitry after reset.

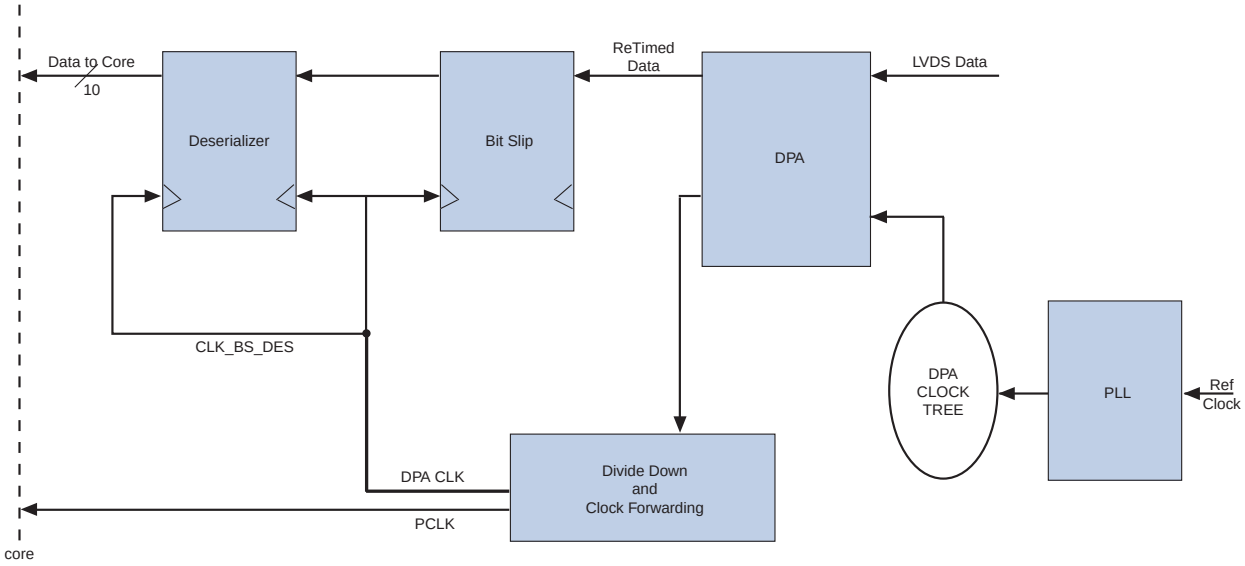
Soft-CDR Mode

The Stratix III LVDS channel offers the soft-CDR mode to support the Gigabit Ethernet/SGMII protocols. Clock-data recovery (CDR) is required to extract the clock out of the clock-embedded data to support SGMII. In Stratix III devices, the CDR circuit is implemented in soft-logic as an IP.

In soft-CDR mode, the DPA circuitry selects an optimal DPA clock phase to sample the data and carry on the bit-slip operation and deserialization. The selected DPA clock is also divided down by the deserialization factor, and then forwarded to the PLD core along with the deserialized data. The LVDS block has an output called `DIVCLKOUT` (`rx_divfwdclk` port of the ALTLVDS megafunction) for the forwarded clock signal. This signal is put on the newly introduced periphery clock (PCLK) network. When using soft-CDR mode, the `rx_reset` port should not be asserted when the `rx_dpa_lock` is asserted because the DPA will continually choose new phase taps from the PLL to track parts per million (PPM) differences between the reference clock and incoming data. In Stratix III devices, you can use every LVDS channel in soft-CDR mode and can drive the core via the PCLK network.

Figure 9–9 shows the path enabled in soft-CDR mode.

Figure 9–9. Soft-CDR Data and Clock Path for a Stratix III Devices



The synchronizer FIFO is bypassed in soft-CDR mode. The reference clock frequency must be suitable for the PLL to generate a clock that matches the data rate of the interface. The DPA circuitry can track PPM differences between the reference clock and the data stream.

Synchronizer

The synchronizer is a 1-bit × 6-bit deep FIFO buffer that compensates for the phase difference between the recovered clock from the DPA circuit and the `diffioclk` that clocks the rest of the logic in the receiver. The synchronizer can only compensate for phase differences, not frequency differences between the data and the receiver's `INCLK`.

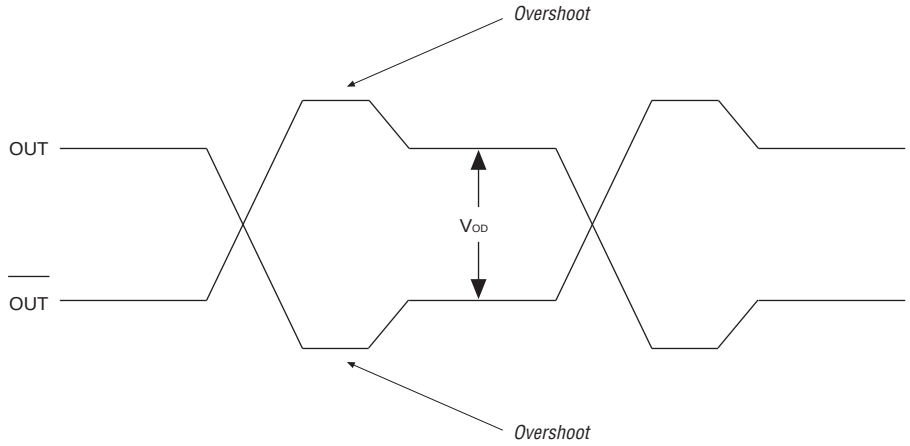
An optional port (`RX_FIFO_RESET`) is available to the internal logic to reset the synchronizer. Altera® recommends using `RX_FIFO_RESET` to reset the synchronizer once after the `RX_DPA_LOCKED` signal gets asserted and before valid data is received.

For more information about how to use the differential receiver, refer to the *ALTLVDS Megafunction User Guide*.

Programmable Pre-Emphasis and Programmable V_{OD}

Stratix III LVDS transmitters support programmable pre-emphasis and programmable voltage output differential (V_{OD}). Pre-emphasis increases the amplitude of the high frequency component of the output signal, and thus helps compensate for the frequency dependent attenuation along the transmission line. Figure 9–10 shows an LVDS output with pre-emphasis. The overshoot is produced by pre-emphasis. This overshoot should not be included in the V_{OD} voltage. The definition of V_{OD} is also shown in Figure 9–10.

Figure 9–10. Programmable V_{OD}



Pre-emphasis is an important feature for high-speed transmission. Without pre-emphasis, the output current is limited by the V_{OD} setting and the output impedance of the driver. At high frequency, the slew rate may not be fast enough to reach the full V_{OD} before the next edge, producing a pattern dependent jitter.

With pre-emphasis, the output current is boosted momentarily during switching to increase the output slew rate. The overshoot introduced by the extra current happens only during switching and does not ring, unlike the overshoot caused by signal reflection. The amount of pre-emphasis required depends on the attenuation of the high-frequency component along the transmission line.

Stratix III pre-emphasis is programmable to create the right amount of overshoot at different transmission conditions. There are four settings for pre-emphasis: zero, low, medium, and high. The default setting is low. In the Quartus II Assignment Editor, pre-emphasis settings are represented in numbers with 0 (zero), 1 (low), 2 (medium) and 3 (high). For a particular design, simulation with an LVDS buffer and transmission line can be used to determine the best pre-emphasis setting.

The V_{OD} is also programmable with four settings: low, medium low, medium high, and high. The default setting is medium low. In the Quartus II Assignment Editor, programmable V_{OD} settings are represented in numbers with 0 (low), 1 (medium low), 2 (medium high) and 3 (high).

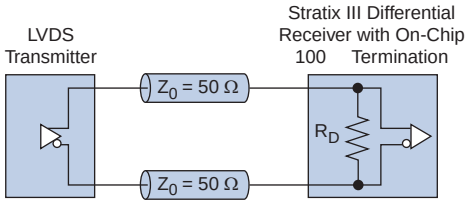
Differential I/O Termination

Stratix III devices provide a 100-ΩOCT R_D option on each differential receiver channel for LVDS standards. OCT saves board space by eliminating the need to add external resistors on the board. You can enable OCT in the Quartus II software Assignment Editor.

OCT R_D is supported on all row I/O pins and SERDES block clock pins: `CLK (0, 2, 9,` and `11)`. It is not supported for column I/O pins, high-speed clock pins `CLK [1, 3, 8, 10]`, or the corner PLL clock inputs.

Figure 9–11 shows the device OCT.

Figure 9–11. On-Chip Differential I/O Termination for Stratix III Devices



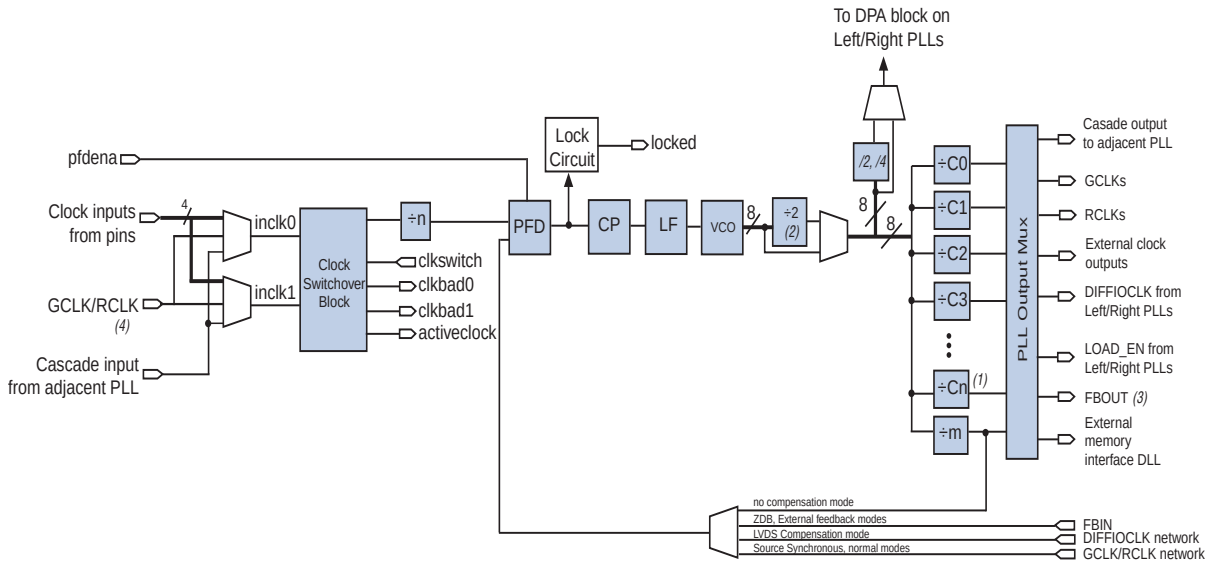
Left/Right PLLs (PLL_Lx/ PLL_Rx)

Stratix III devices contain up to eight left/right PLLs, with up to four PLLs located on the left side and four on the right side of the device. The left PLLs can support high-speed differential I/O banks on the left side and the right PLLs can support banks on the right side of the device. The high-speed differential I/O receiver and transmitter channels use these left/right PLLs to generate the parallel global clocks (`rx`- or `tx`-clock) and high-speed clocks (`diffioclk`). Figure 9–1 shows the locations of the left/right PLLs. The PLL VCO operates at the clock frequency of the data rate. Each left/right PLL offers a single serial data rate support, but up to two separate serialization and/or deserialization factors (from the C0 and C1 left/right PLL clock outputs). Clock switchover and dynamic left/right PLL reconfiguration is available in high-speed differential I/O support mode.

For more information, refer to the *Clock Network and PLLs in Stratix III Devices* chapter.

Figure 9–12 shows a simplified block diagram of the major components of the Stratix III PLL.

Figure 9–12. PLL Block Diagram for Stratix III Devices



Notes to Figure 9–12:

(1) n = 6 for Left/Right PLLs; n = 9 for Top/Bottom PLLs.

(2) This is the VCO post-scale counter K.

(3) The FBOUT port is fed by the M counter in Stratix III PLLs.

(4) The global or regional clock input can be driven by an output from another PLL, a pin-driven dedicated global or regional clock, or through a clock control block provided the clock control block is fed by an output from another PLL or a pin-driven dedicated global or regional clock. An internally generated global signal or general purpose I/O pin cannot drive the PLL.

Clocking

The left/right PLLs feed into the differential transmitter and receive channels through the LVDS and DPA clock network. The center left/right PLLs can clock the transmitter and receive channels above and below them. The corner left/right PLLs can drive I/Os in the banks adjacent to them. Figure 9–13 and Figure 9–14 show center and corner PLL clocking in Stratix III devices. You can find more information about PLL clocking restrictions in "Differential Pin Placement Guidelines" on page 9–19.

Figure 9–13. LVDS/DPA Clocks with Center PLLs for Stratix III Devices

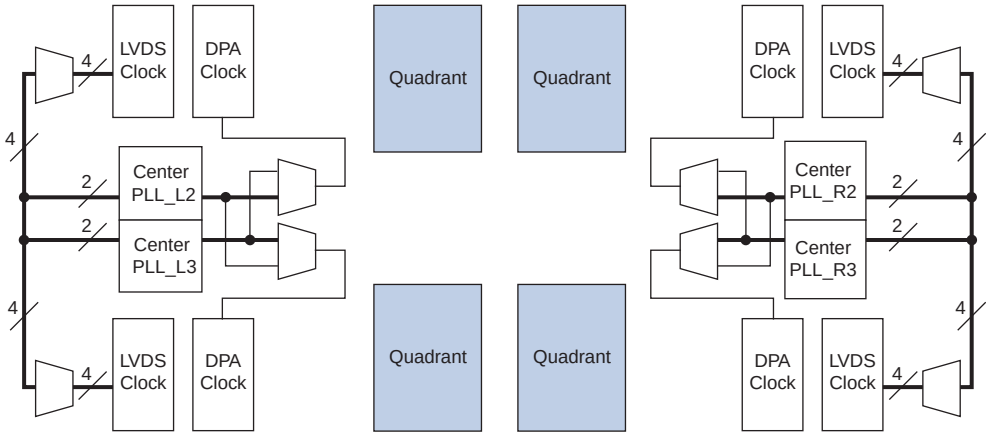
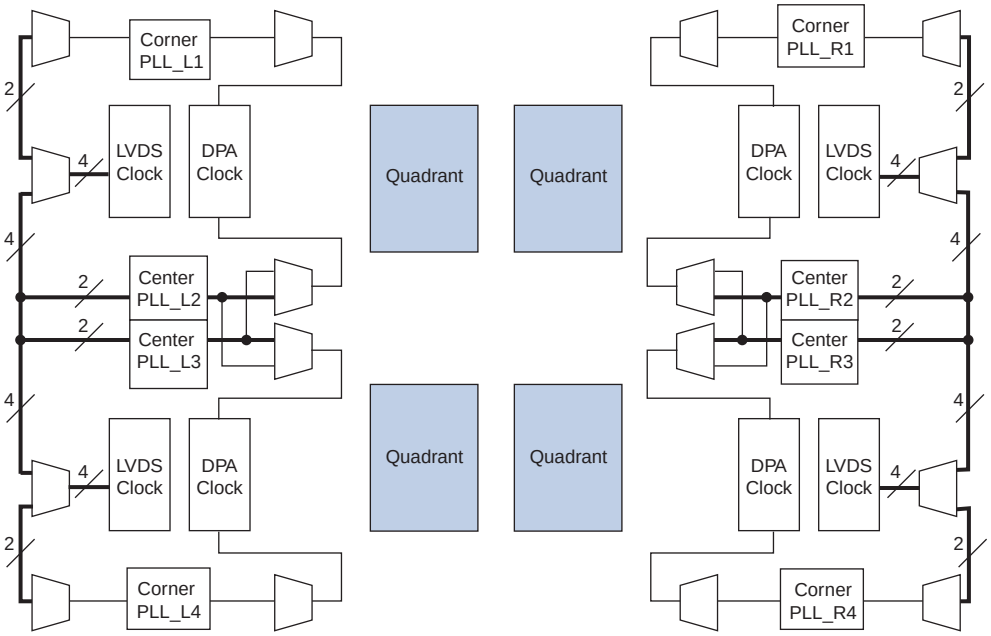


Figure 9–14. LVDS/DPA Clocks with Center and Corner PLLs for Stratix III Devices



Source-Synchronous Timing Budget

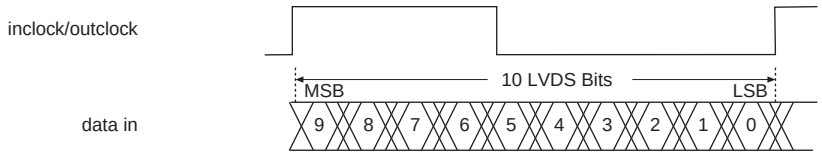
This section discusses the timing budget, waveforms, and specifications for source-synchronous signaling in Stratix III devices. LVDS I/O standards enable high-speed data transmission. This high data transmission rate results in better overall system performance. To take advantage of fast system performance, it is important to understand how to analyze timing for these high-speed signals. Timing analysis for the differential block is different from traditional synchronous timing analysis techniques.

Rather than focusing on clock-to-output and setup times, source synchronous timing analysis is based on the skew between the data and the clock signals. High-speed differential data transmission requires the use of timing parameters provided by IC vendors and is strongly influenced by board skew, cable skew, and clock jitter. This section defines the source-synchronous differential data orientation timing parameters, the timing budget definitions for Stratix III devices, and ways to use these timing parameters to determine the maximum performance of your design.

Differential Data Orientation

There is a set relationship between an external clock and the incoming data. For an operation at 1 Gbps and SERDES factor of 10, the external clock is multiplied by 10, and phase-alignment can be set in the PLL to coincide with the sampling window of each data bit. The data is sampled on the falling edge of the multiplied clock. Figure 9–15 shows the data bit orientation of the ×10 mode.

Figure 9–15. Bit Orientation in Quartus II Software



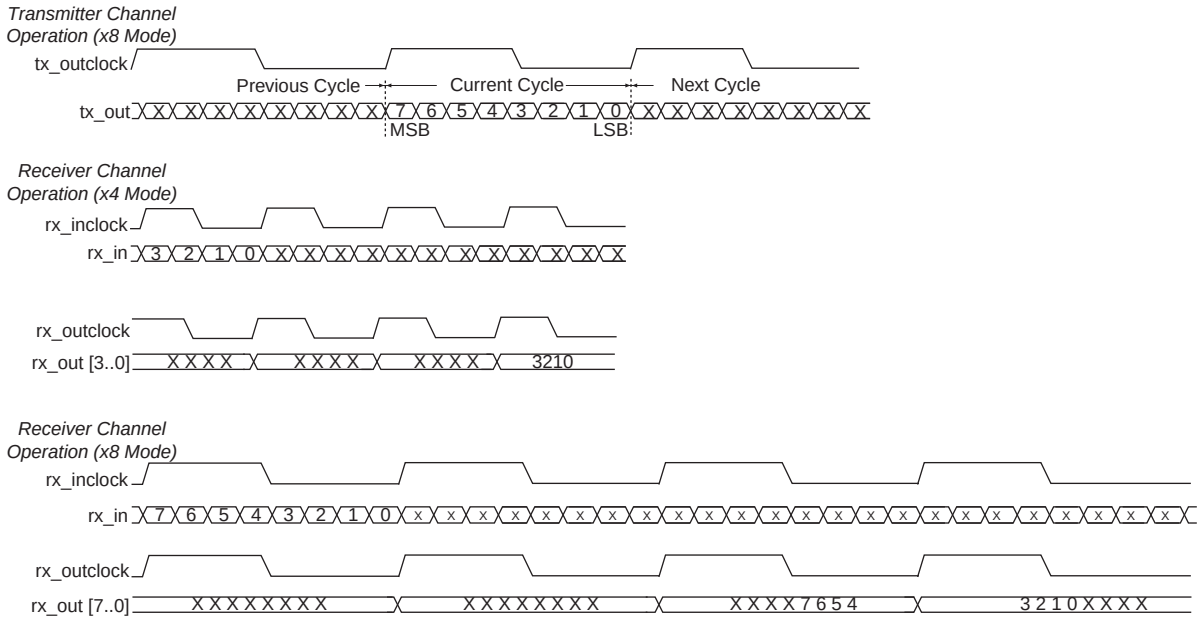
Differential I/O Bit Position

Data synchronization is necessary for successful data transmission at high frequencies. Figure 9–16 shows the data bit orientation for a channel operation. These figures are based on the following:

- SERDES factor equals clock multiplication factor
- Edge alignment is selected for phase alignment
- Implemented in hard SERDES

For other serialization factors, use the Quartus II software tools and find the bit position within the word and the bit positions after deserialization.

Figure 9–16. Bit-Order and Word Boundary for One Differential Channel *(Note 1)*



Note to Figure 9–16:

(1) These are only functional waveforms and are not intended to convey timing information.

Table 9–3 lists the conventions for differential bit naming for 18 differential channels. The MSB and LSB positions increase with the number of channels used in a system.

Table 9–3. Differential Bit Naming (Part 1 of 2)

Receiver Channel Data Number	Internal 8-Bit Parallel Data	
	MSB Position	LSB Position
1	7	0
2	15	8
3	23	16
4	31	24
5	39	32
6	47	40
7	55	48
8	63	56
9	71	64
10	79	72
11	87	80
12	95	88
13	103	96
14	111	104
15	119	112
16	127	120

Table 9–3. Differential Bit Naming (Part 2 of 2)

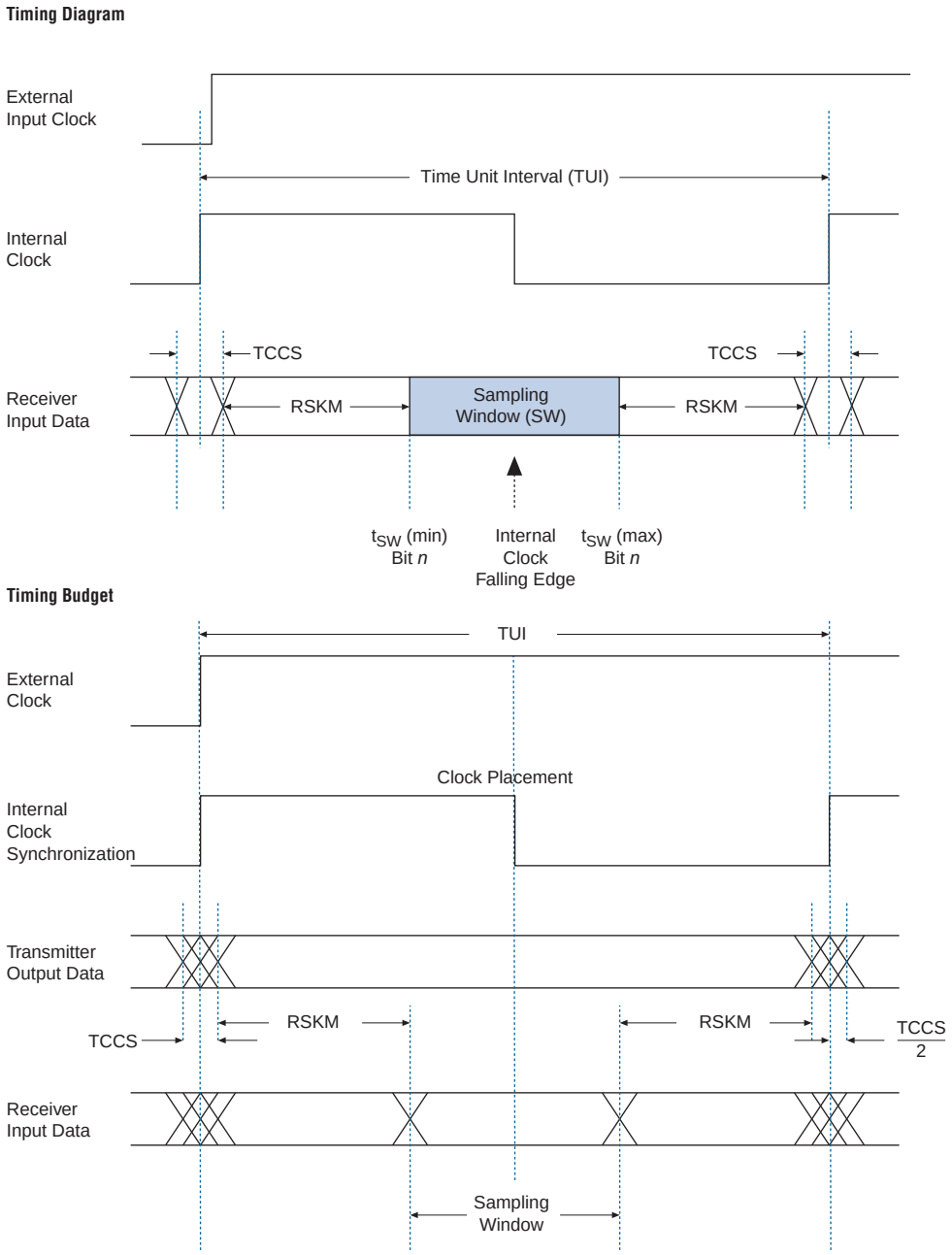
Receiver Channel Data Number	Internal 8-Bit Parallel Data	
	MSB Position	LSB Position
17	135	128
18	143	136

Receiver Skew Margin for Non-DPA

Changes in system environment, such as temperature, media (cable, connector, or PCB) loading effect, the receiver's setup and hold times, and internal skew, reduce the sampling window for the receiver. The timing margin between the receiver's clock input and the data input sampling window is called receiver skew margin (RSKM). Figure 9–17 shows the relationship between the RSKM and the sampling window of the receiver.

Transmit channel-to-channel skew (TCCS), RSKM, and the sampling window specifications are used for high-speed source-synchronous differential signals without DPA. When using DPA, these specifications are exchanged for the simpler single DPA jitter tolerance specification. For instance, the receiver skew is why each input with DPA selects a different phase of the clock, thus removing the requirement for this margin. In the timing diagram, TSW represents time for the sampling window.

Figure 9–17. Differential High-Speed Timing Diagram and Timing Budget for Non-DPA



Differential Pin Placement Guidelines

To ensure proper high-speed operation, differential pin placement guidelines have been established. The Quartus II compiler automatically checks that these guidelines are followed and issues an error message if they are not met.

Because DPA usage adds some constraints on the placement of high-speed differential channels, this section is divided into pin placement guidelines with and without DPA usage.

Guidelines for DPA-Enabled Differential Channels

The Stratix III device has differential receivers and transmitters in I/O banks on the left and right sides of the device. Each receiver has a dedicated DPA circuit to align the phase of the clock to the data phase of its associated channel. When DPA-enabled channels are used in differential banks, you must adhere to the guidelines listed in the following sections.

DPA-Enabled Channels and Single-Ended I/Os

When there is a DPA channel enabled in a bank, both single-ended I/Os and differential I/O standards are allowed in the bank.

Single-ended I/Os are allowed in the same I/O bank as long as the single-ended I/O standard uses the same V_{CCIO} as the DPA-enabled differential I/O bank.

DPA-Enabled Channel Driving Distance

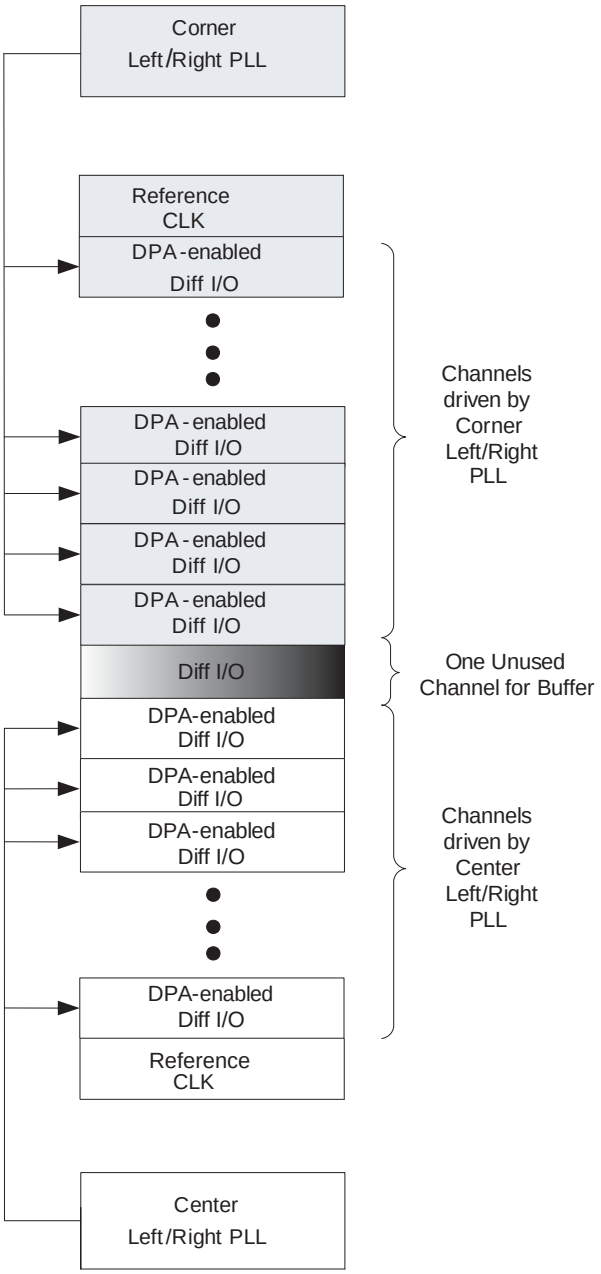
If the number of DPA channels driven by each left/right PLL exceeds 25 LAB rows, Altera recommends implementing data realignment (bit-slip) circuitry for all the DPA channels.

Using Corner and Center Left/Right PLLs

If a differential bank is being driven by two left/right PLLs, where the corner left/right PLL is driving one group and the center left/right PLL is driving another group, there must be at least one row of separation between the two groups of DPA-enabled channels (refer to Figure 9–18). The two groups can operate at independent frequencies.

No separation is necessary if a single left/right PLL is driving DPA-enabled channels as well as DPA-disabled channels.

Figure 9–18. Corner and Center Left/Right PLLs Driving DPA-Enabled Differential I/Os in the Same Bank



Using Both Center Left/Right PLLs

Both center left/right PLLs can be used to drive DPA-enabled channels simultaneously, as long as they drive these channels in their adjacent banks only, as shown in Figure 9–19.

If one of the center left/right PLLs drive the top and bottom banks, the other center left/right PLL cannot be used to drive the differential channels, as shown in Figure 9–19.

If the top PLL_L2/PLL_R2 drives DPA-enabled channels in the lower differential bank, the PLL_L3/PLL_R3 cannot drive DPA-enabled channels in the upper differential banks and vice versa. In other words, the center left/right PLLs cannot drive cross-banks simultaneously, as shown in Figure 9–20.

Figure 9–19. Center Left/Right PLLs Driving DPA-Enabled Differential I/Os

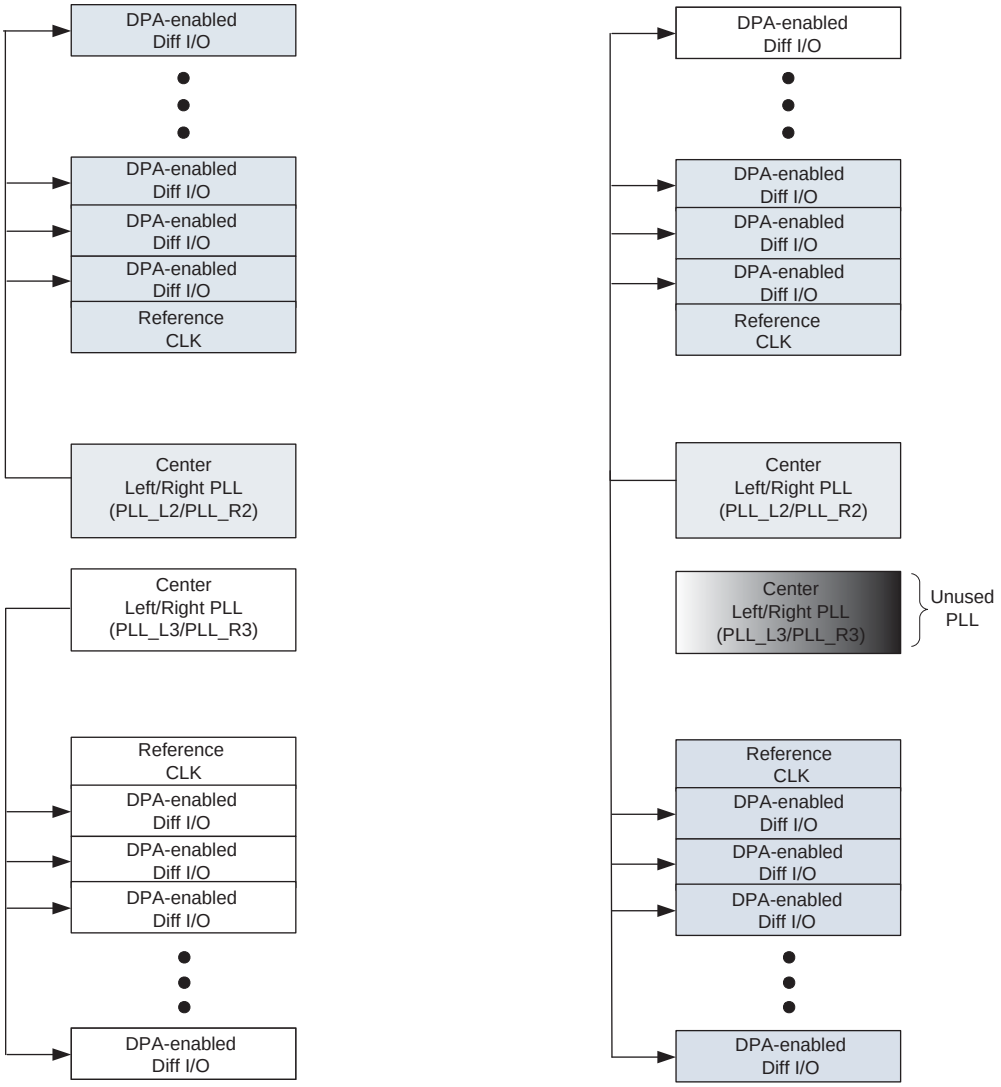
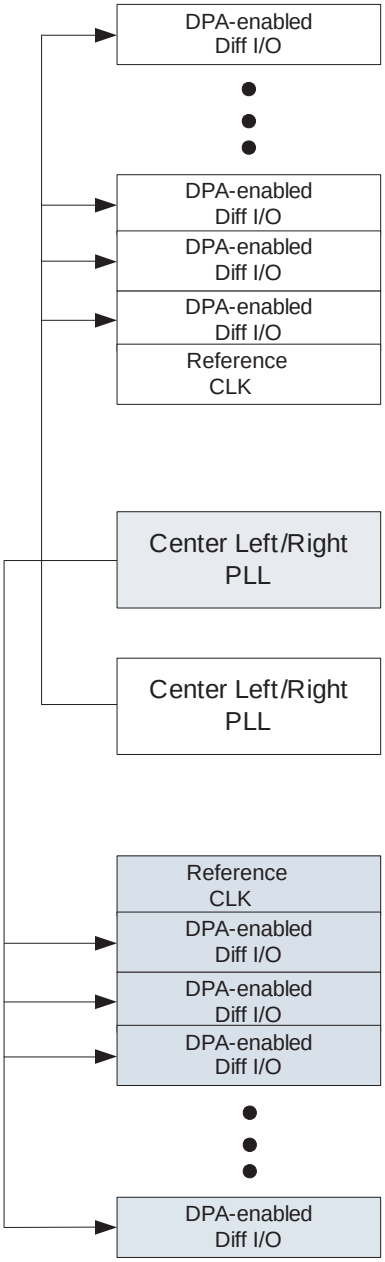


Figure 9–20. Invalid Placement of DPA-Enabled Differential I/Os Driven by Both Center Left/Right PLLs



Guidelines for DPA-Disabled Differential Channels

When DPA-disabled channels are used in the left and right banks of a Stratix III device, you must adhere to the guidelines in the following sections.

DPA-Disabled Channels and Single-Ended I/Os

The placement rules for DPA-disabled channels and single-ended I/Os are the same as those for DPA-enabled channels and single-ended I/Os.

DPA-Disabled Channel Driving Distance

Each left/right PLL can drive all the DPA-disabled channels in the entire bank.

Using Corner and Center Left/Right PLLs

A corner left/right PLL can be used to drive all transmitter channels and a center left/right PLL can be used to drive all DPA-disabled receiver channels within the same differential bank. In other words, a transmitter channel and a receiver channel in the same LAB row can be driven by two different PLLs, as shown in Figure 9–21.

A corner left/right PLL and a center left/right PLL can drive duplex channels in the same differential bank as long as the channels driven by each PLL are not interleaved. No separation is necessary between the group of channels driven by the corner and center left/right PLLs. Refer to Figure 9–21 and Figure 9–22.

Figure 9–21. Corner and Center Left/Right PLLs Driving DPA-Disabled Differential I/Os in the Same Bank

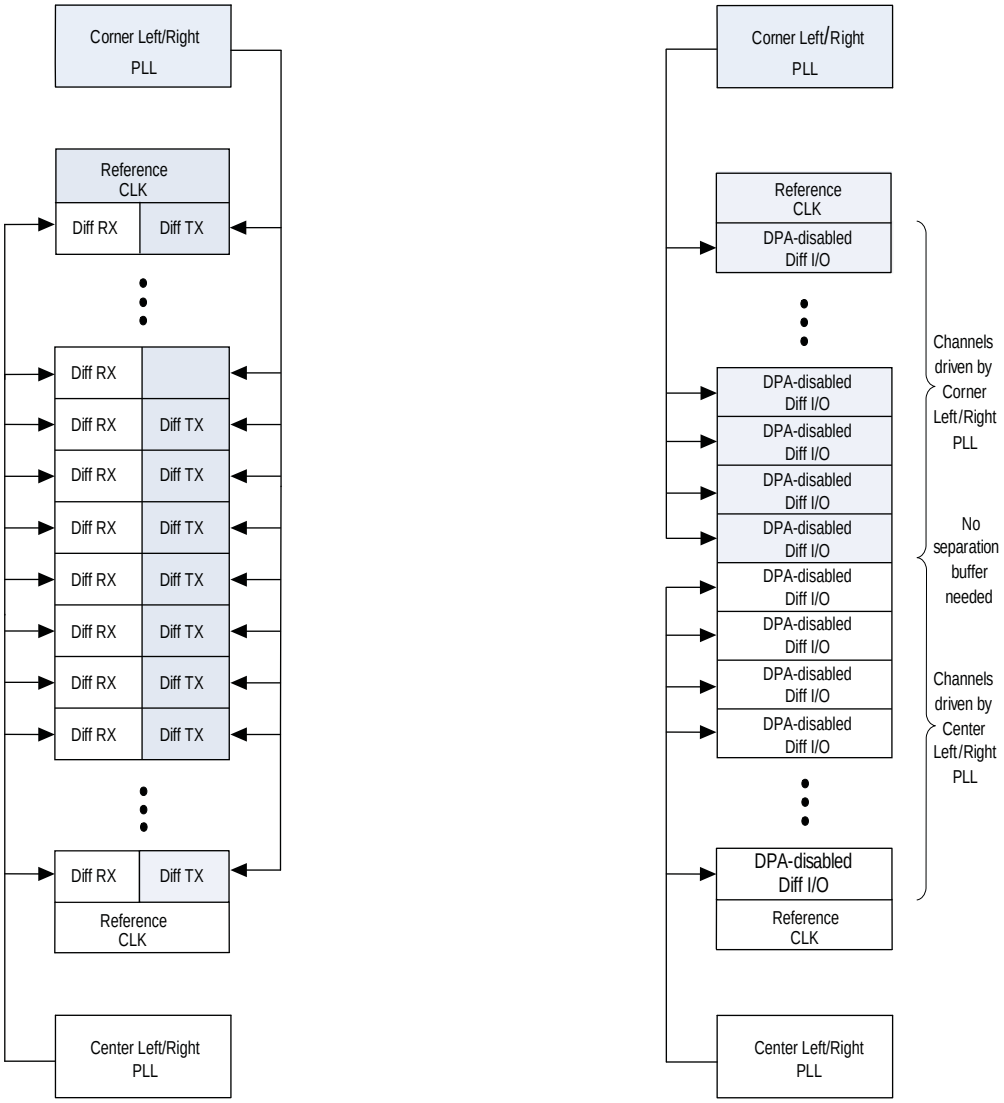
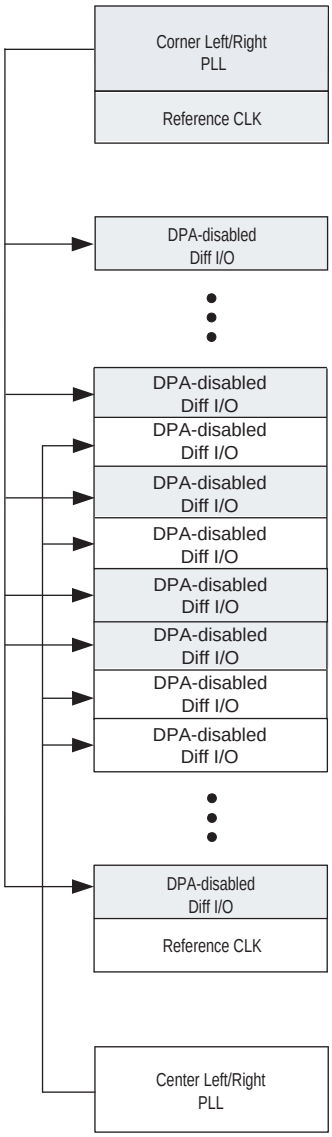


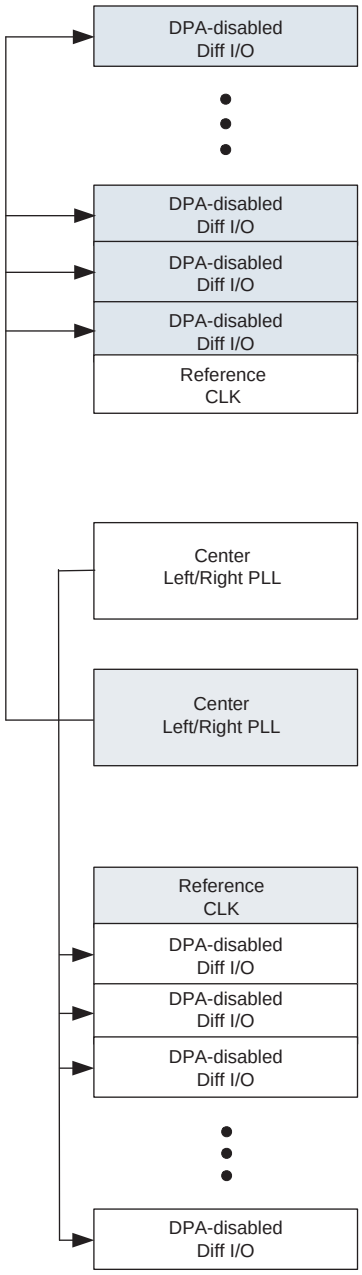
Figure 9–22. Invalid Placement of DPA-Disabled Differential I/Os Due to Interleaving of Channels Driven by the Corner and Center Left/Right PLLs



Using Both Center Left/Right PLLs

Both center left/right PLLs can be used simultaneously to drive DPA-disabled channels on upper and lower differential banks. Unlike DPA-enabled channels, the center left/right PLLs can drive cross-banks. For example, the upper center left/right PLL can drive the lower differential bank at the same time the lower center left/right PLL is driving the upper differential bank and vice versa, as shown in Figure 9–23.

Figure 9–23. Both Center Left/Right PLLs Driving Cross-Bank DPA-Disabled Channels Simultaneously



Chapter Revision History

Table 9–4 lists the revision history for this chapter.

Table 9–4. Chapter Revision History

Date	Revision	Changes Made
July 2010	1.9	Updated "Differential Transmitter" and "Differential Receiver" sections.
March 2010	1.8	Updated for the Quartus II software version 9.1 SP2 release: ■ Updated "LVDS Channels", "Differential Transmitter", and "Differential Receiver" section. ■ Minor changes to the text.
May 2009	1.7	■ Updated Table 9–1 and Table 9–2. ■ Updated Figure 9–5. ■ Updated "DPA-Enabled Channels and Single-Ended I/Os" section.
February 2009	1.6	■ Updated "DPA-Enabled Channels and Single-Ended I/Os" section. ■ Updated Table 9–2. ■ Removed "Reference Documents" section.
October 2008	1.5	■ Updated "Introduction", "Differential Receiver", and "Synchronizer" sections. ■ Updated Figure 9–5. ■ Updated New Document Format.
May 2008	1.4	■ Updated "Soft-CDR Mode", "Dynamic Phase Aligner (DPA)", "Programmable Pre-Emphasis and Programmable VOD", and "Guidelines for DPA-Enabled Differential Channels" sections. ■ Updated Table 9–1 and Table 9–2. ■ Removed "Figure 9–19. Left/Right PLL Driving Distance for DPA-Enabled Channels".
November 2007	1.3	■ Updated Table 9–1 and Table 9–2.
October 2007	1.2	■ Added material to "DPA-Enabled Channels and Single-Ended I/Os" on page 9–21 and removed material from "DPA-Disabled Channels and Single-Ended I/Os" on page 9–29. ■ Added new sections "Programmable Pre-Emphasis and Programmable VOD" on page 9–12, "Soft-CDR Mode", and "Referenced Documents". ■ Added live links for references. ■ Added Figure 9–10. ■ Minor edits to "DPA-Enabled Channel Driving Distance" section.
May 2007	1.1	■ Minor changes to second paragraph of the section "Differential I/O Termination". ■ Added Table 9–1 and Table 9–2.
November 2006	1.0	Initial release.

Section III. Hot Socketing, Configuration, Remote Upgrades, and Testing

This section provides information on hot socketing and power-on reset, configuring Stratix® III devices, remote system upgrades, and IEEE 1149.1 (JTAG) Boundary-Scan Testing in the following sections:

- Chapter 10, Hot Socketing and Power-On Reset in Stratix III Devices
- Chapter 11, Configuring Stratix III Devices
- Chapter 12, Remote System Upgrades with Stratix III Devices
- Chapter 13, IEEE 1149.1 (JTAG) Boundary-Scan Testing in Stratix III Devices

Revision History

Refer to each chapter for its own specific revision history. For information on when each chapter was updated, refer to the Chapter Revision Dates section, which appears in the full handbook.

10. Hot Socketing and Power-On Reset in Stratix III Devices

SIII51010-1.7

This chapter describes information about hot-socketing specifications, power-on reset (POR) requirements, and their implementation in Stratix® III devices.

Stratix III devices offer hot socketing, also known as hot plug-in or hot swap, and power sequencing support without the use of any external devices. You can insert or remove a Stratix III device or a board in a system during system operation without causing undesirable effects to the running system bus or board that is inserted into the system.

The hot socketing feature also removes some of the difficulty when you use Stratix III devices on PCBs that contain a mixture of 3.3-, 3.0-, 2.5-, 1.8-, 1.5-, and 1.2-V devices. With the Stratix III hot socketing feature, you no longer need to ensure a proper power-up sequence for each device on the board.

The Stratix III hot-socketing feature provides:

- Board or device insertion and removal without external components or board manipulation
- Support for any power-up sequence
- I/O buffers non-intrusive to system buses during hot insertion

This section also describes the POR circuitry in Stratix III devices. POR circuitry keeps the devices in the reset state until the power supplies are within operating range.

Stratix III Hot-Socketing Specifications

Stratix III devices are hot-socketing compliant without the need for external components or special design requirements. Hot socketing support in Stratix III devices has the following advantages:

- You can drive the device before power-up without damaging it.
- I/O pins remain tri-stated during power-up. The device does not drive out before or during power-up, thereby not affecting other buses in operation.
- You can insert a Stratix III device into or remove it from a powered-up system board without damaging or interfering with normal system/board operation.

Stratix III Devices Can Be Driven Before Power Up

You can drive signals into I/O pins, dedicated input pins, and dedicated clock pins of Stratix III devices before or during power up or power down without damaging the device. Stratix III devices support power up or power down of the power supplies in any sequence in order to simplify system-level design.

I/O Pins Remain Tri-Stated During Power Up

A device that does not support hot socketing can interrupt system operation or cause contention by driving out before or during power up. In a hot-socketing situation, the Stratix III device's output buffers are turned off during system power up or power down. Also, the Stratix III device does not drive out until the device is configured and working within recommended operating conditions.

Insertion or Removal of a Stratix III Device from a Powered-Up System

Devices that do not support hot socketing can short power supplies when powered up through the device signal pins. This irregular power up can damage both the driving and driven devices and can disrupt card power up.

You can insert a Stratix III device into or remove it from a powered-up system board without damaging the system board or interfering with its operation.

You can power up or power down the core voltage supplies (V_{CC}, V_{CCL}, V_{CCPT}, V_{CCA_PLL}, and V_{CCD_PLL}), V_{CCIO}, V_{CCPMG}, V_{CC_CLKIN}, and V_{CCPD} supplies in any sequence and at any time between them. The individual power supply ramp-up and ramp-down rates can range from 50 µs to 12 ms or 100 ms depending on the PORSEL setting. During hot socketing, the I/O pin capacitance is less than 15 pF and the clock pin capacitance is less than 20 pF.

For more information about the hot socketing specification, refer to the *DC and Switching Characteristics of Stratix III Devices* chapter and the *Hot-Socketing and Power-Sequencing Feature and Testing for Altera Devices White Paper*.

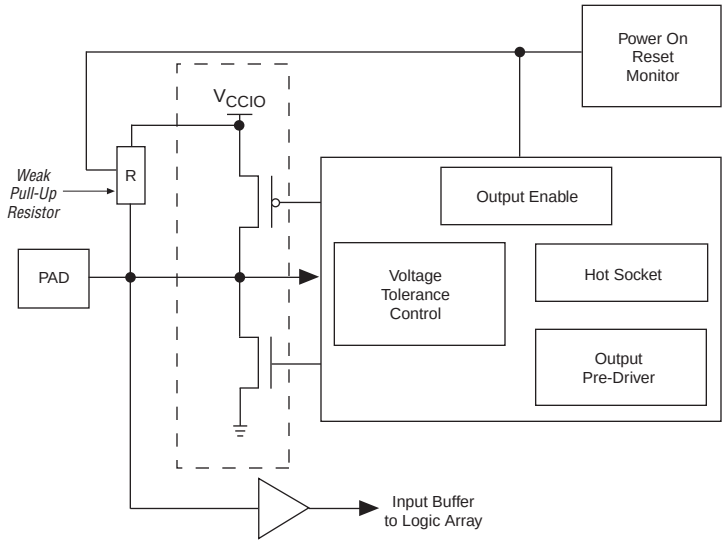
A possible concern regarding hot socketing is the potential for “latch-up”. Nevertheless, Stratix III devices are immune to latch-up when hot socketing. Latch-up can occur when electrical subsystems are hot socketed into an active system. During hot socketing, the signal pins can be connected and driven by the active system before the power supply can provide current to the device's power and ground planes. This condition can lead to latch-up and cause a low-impedance path from power to ground within the device. As a result, the device draws a large amount of current, possibly causing electrical damage.

Hot-Socketing Feature Implementation in Stratix III Devices

The hot-socketing feature turns off the output buffer during power up and power down of the V_{CC}, V_{CCIO}, V_{CCPGM}, or V_{CCPD} power supplies. The hot-socketing circuitry generates an internal `HOTSCKT` signal when the V_{CC}, V_{CCIO}, V_{CCPGM}, or V_{CCPD} power supplies are below the threshold voltage. Hot-socketing circuitry is designed to prevent excess I/O leakage during power up. When the voltage ramps up very slowly, it is still relatively low, even after the POR signal is released and the configuration is completed. The `CONF_DONE`, `nCEO`, and `nSTATUS` pins fail to respond, as the output buffer cannot flip from the state set by the hot-socketing circuit at this low voltage. Therefore, the hot-socketing circuit has been removed on these configuration pins to make sure that they are able to operate during configuration. Thus, it is expected behavior for these pins to drive out during power-up and power-down sequences.

Figure 10–1 shows the Stratix III device's I/O pin circuitry.

Figure 10–1. Hot-Socketing Circuitry for Stratix III Devices

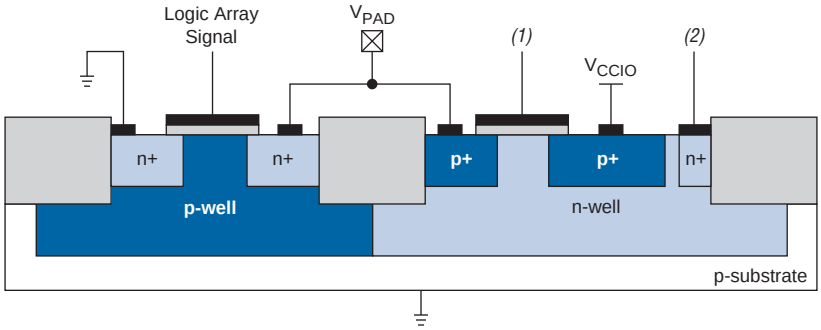


The POR circuit monitors the voltage level of power supplies (V_{CC}, V_{CCL}, V_{CCPD}, V_{CCPGM} and V_{CCPT}) and keeps the I/O pins tri-stated until the device is in user mode. The weak pull-up resistor (R) in the Stratix III input/output element (IOE) keeps the I/O pins from floating. The 3.3-V tolerance control circuit permits the I/O pins to be driven by 3.3 V before V_{CCIO}, V_{CC}, V_{CCPD}, and/or V_{CCPGM} supplies are powered, and it prevents the I/O pins from driving out when the device is not in user mode.

Altera uses GND as reference for hot-socketing operation and I/O buffer designs. To ensure proper operation, you must connect the GND between boards before connecting the power supplies. This will prevent the GND on your board from being pulled up inadvertently by a path to power through other components on your board. A pulled up GND could otherwise cause an out-of-specification I/O voltage or current condition with the Altera device.

Figure 10–2 shows a transistor-level cross section of the Stratix III device I/O buffers. This design prevents leakage current from I/O pins to the V_{CCIO} supply when V_{CCIO} is powered before the other voltage supplies or if the I/O pad voltage is higher than V_{CCIO}. This also applies for sudden voltage spikes during hot insertion. The V_{PAD} leakage current charges the 3.3-V tolerant circuit capacitance.

Figure 10–2. Transistor Level Diagram of a Stratix III Device I/O Buffers



Notes to Figure 10–2:

(1) This is the logic array signal or the larger of either the V_{CCIO} or V_{PAD} signal.

(2) This is the larger of either the V_{CCIO} or V_{PAD} signal.

Power-On Reset Circuitry

When power is applied to a Stratix III device, a POR event occurs when all the power supplies reach the recommended operating range within a certain period of time (specified as a maximum power supply ramp time; t_{RAMP}). Hot socketing feature in Stratix III allows the required power supplies to be powered up in any sequence and at any time between them with each individual power supply must reach the recommended operating range within t_{RAMP}.

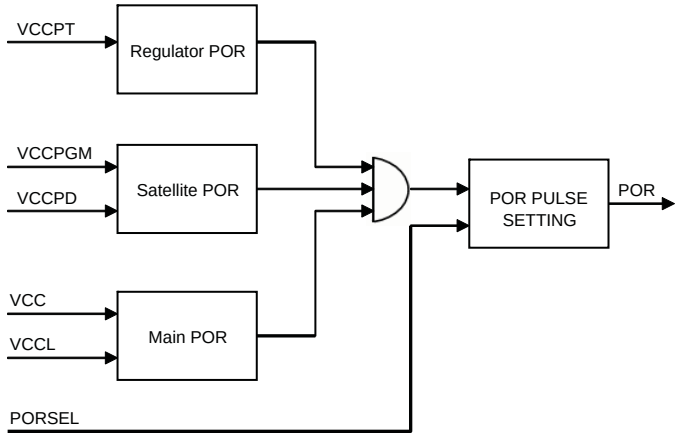
For maximum power supplies ramp-up time for Stratix III Devices, refer Table 10–1.

Stratix III devices provide a dedicated input pin (`PORSEL`) to select a POR delay time during power up. When the `PORSEL` pin is connected to ground, the POR delay time is 100 ms. When the `PORSEL` pin is set to high, the POR delay time is 12 ms.

The POR block consists of a regulator POR, satellite POR, and main POR to check the power supply levels for proper device configuration. The satellite POR monitors V_{CCPD} and V_{CCPGM} power supplies that are used in the configuration buffers for device programming. The POR block also checks for functionality of I/O level shifters powered by V_{CCPD} and V_{CCPGM} during power-up mode. The main POR checks the V_{CC} and V_{CCL} supplies used in core. The internal configuration memory supply, which is used during device configuration, is checked by the regulator POR block and is gated in the main POR block for the final POR trip. A simplified block diagram of the POR block is shown in Figure 10–3.

All configuration-related dedicated and dual function I/O pins must be powered by V_{CCPGM}.

Figure 10–3. Simplified POR Block Diagram



The ramp-up time specification for Stratix III devices is listed in Table 10–1.

Table 10–1. Power Supplies Ramp-Up Time (t_{RAMP}) Requirements

Power Supply	PORSEL setting	Ramp-up Time	
		Minimum	Maximum
V_{CCPT}	HIGH	50 µs	5 ms
	GND	50 µs	5 ms
V_{CC}, V_{CCL}, V_{CCPD}, V_{CCPGM}, V_{CCIO}, V_{CCA_PLL}, V_{CCD_PLL}, V_{CC_CLKIN}	HIGH	50 µs	12 ms
	GND	50 µs	100 ms

Power-On Reset Specifications

The POR circuit monitors the power supplies listed in Table 10–2.

Table 10–2. Power Supplies Monitored by the POR Circuitry

Power Supply	Description	Setting (V)
V_{CC}	I/O registers power supply	1.1
V_{CCL}	Selectable core voltage power supply	0.9, 1.1
V_{CCPT}	Power supply for the programmable power technology	2.5
V_{CCPD}	I/O pre-driver power supply	2.5, 3.0, 3.3
V_{CCPGM}	Configuration pins power supply	1.8, 2.5, 3.0, 3.3

To ensure proper device operation, all power supplies listed in Table 10–2 are required to be powered up at all times during device operation.

The POR circuit does not monitor the power supplies listed in Table 10–3.

Table 10–3. Power Supplies That Are Not Monitored by the POR Circuitry

Voltage Supply	Description	Setting (V)
V_{CCIO}	I/O power supply	1.2, 1.5, 1.8, 2.5, 3.0, 3.3
V_{CCA_PLL}	PLL analog global power supply	2.5
V_{CCD_PLL}	PLL digital power supply	1.1
V_{CC_CLKIN}	PLL differential clock input power supply (top and bottom I/O banks only)	2.5
V_{CCBAT}	Battery back-up power supply for design security volatile key storage	1.0 – 3.3 *(1)*

Note to Table 10–3:

(1) The nominal voltage for V_{CCBAT} is 3.0-V.

During power up, all power supplies listed in Table 10–2 and Table 10–3 are required to monotonically reach their full-rail values within t_{RAMP}.

The POR specification is designed to ensure that all the circuits in the Stratix III device are at certain known states during power up.

The POR signal pulse width is programmable using the PORSEL input pin. When PORSEL is set to low, the POR signal pulse width is set to 100 ms. A POR pulse width of 100 ms allows serial flash devices with 65 ms to 100 ms internal POR delay to be powered up and ready to receive the nSTATUS signal from Stratix III. When the PORSEL is set to high, the POR signal pulse width is set to 12 ms. A POR pulse width of 12 ms allows time for power supplies to ramp-up to full rail.

For more information about the POR specification, refer to the *DC and Switching Characteristics* chapter.

Chapter Revision History

Table 10–4 lists the revision history for this chapter.

Table 10–4. Chapter Revision History

Date	Version	Changes Made
March 2010	1.7	■ Updated for the Quartus II software version 9.1 SP2 release. ■ Minor text edits.
February 2009	1.6	■ Updated "Hot Socketing Feature Implementation in Stratix III Devices" section. ■ Removed "Referenced Documents" section.
October 2008	1.5	■ Updated Table 10–3. ■ Updated "Insertion or Removal of a Stratix III Device from a Powered-Up System" and "Power-On Reset Circuitry" sections. ■ Updated Figure 10–3. ■ Added Table 10–1. ■ Updated New Document Format.
July 2008	1.4	Updated Table 10–2.
May 2008	1.3	■ Updated "Insertion or Removal of a Stratix III Device from a Powered-Up System", "Hot Socketing Feature Implementation in Stratix III Devices", and "Power-On Reset Circuitry" sections. ■ Updated "Power-On Reset Specifications" section tables.
October 2007	1.2	■ Added section "Referenced Documents". ■ Added live links for references.
May 2007	1.1	All instances of VCCR changed to VCCPT in text, and in Figure 10–3, and Table 10–1.
November 2006	1.0	Initial Release.

11. Configuring Stratix III Devices

SIII51011-2.0

This chapter contains complete information about Stratix® III supported configuration schemes, how to execute the required configuration schemes, and all necessary option pin settings.

Stratix III devices use SRAM cells to store configuration data. Because SRAM memory is volatile, you must download configuration data to the Stratix III device each time the device powers up. You can configure Stratix III devices using one of four configuration schemes:

- Fast passive parallel (FPP)
- Fast active serial (AS)
- Passive serial (PS)
- Joint Test Action Group (JTAG)

All configuration schemes use an external controller (for example, a MAX® II device or microprocessor), a configuration device, or a download cable. Refer to "Configuration Features" on page 11–3 for more information.

Configuration Devices

The Altera® serial configuration devices (EPCS128, EPCS64, and EPCS16) support a single-device and multi-device configuration solution for Stratix III devices and are used in the fast AS configuration scheme. Serial configuration devices offer a low-cost, low-pin count configuration solution.

For information about serial configuration devices, refer to the *Serial Configuration Devices (EPCS1, EPCS4, EPCS16, EPCS64, and EPCS128) Data Sheet* in volume 2 of the *Configuration Handbook*.

All minimum timing information in this handbook covers the entire Stratix III family. Some devices may work at less than the minimum timing stated in this handbook due to process variation.

Configuration Schemes

Select the configuration scheme by driving the Stratix III device `MSEL` pins either high or low, as detailed in Table 11–1. The `MSEL` pins are powered by the V_{CCPGM} power supply of the bank they reside in. The `MSEL[2..0]` pins have 5-kΩ internal pull-down resistors that are always active. During power-on reset (POR) and reconfiguration, the `MSEL` pins must be at LVTTL V_{IL} and V_{IH} levels to be considered a logic low and logic high.

To avoid any problems with detecting an incorrect configuration scheme, hard-wire the `MSEL[]` pins to V_{CCPGM} and GND, without any pull-up or pull-down resistors. Do not drive the `MSEL[]` pins with a microprocessor or another device.

Table 11–1. Stratix III Configuration Schemes

Configuration Scheme	MSEL2	MSEL1	MSEL0
Fast passive parallel (FPP)	0	0	0
Passive serial (PS)	0	1	0
Fast AS (40 MHz) *(1)*	0	1	1
Remote system upgrade fast AS (40 MHz) *(1)*	0	1	1
FPP with design security feature, decompression, or both enabled *(2)*	0	0	1
JTAG-based configuration *(4)*	*(3)*	*(3)*	*(3)*

Notes to Table 11–1:

(1) To support fast AS configuration for Stratix III, you must use EPCS16, EPCS64, or EPCS128 devices. For more information, refer to *Serial Configuration Devices Data Sheet* chapter.

(2) These modes are only supported when using a MAX® II device or a microprocessor with flash memory for configuration. In these modes, the host system must output a DCLK that is ×4 the data rate.

(3) Do not leave the MSEL pins floating. Connect them to VCCPGM or ground. These pins support the non-JTAG configuration scheme used in production. If you only use JTAG configuration, connect the MSEL pins to ground.

(4) JTAG-based configuration takes precedence over other configuration schemes, which means MSEL pin settings are ignored.

Table 11–2 lists the uncompressed raw binary file (**.rbf**) configuration file sizes for Stratix III devices.

Table 11–2. Stratix III Uncompressed Raw Binary File (**.rbf**) Sizes

Device	Data Size (Bits)
EP3SL50	22, 178, 792
EP3SL70	22, 178, 792
EP3SL110	47, 413, 312
EP3SL150	47, 413, 312
EP3SL200	93, 324, 656
EP3SL340	117, 387, 664
EP3SE50	25, 891, 968
EP3SE80	48, 225, 392
EP3SE110	48, 225, 392
EP3SE260	93, 324, 656

Use the data in Table 11–2 to estimate the file size before design compilation. Different configuration file formats, such as a hexadecimal (**.hex**) or tabular text file (**.ttf**) format, have different file sizes. Refer to the Quartus® II software for the different types of configuration file and the file sizes. However, for any specific version of the Quartus II software, any design targeted for the same device will have the same uncompressed configuration file size. If you are using compression, the file size can vary after each compilation because the compression ratio is dependent on the design.

For more information about setting device configuration options or creating configuration files, refer to the *Device Configuration Options* and *Configuration File Formats chapters* in volume 2 of the *Configuration Handbook*.

Configuration Features

Stratix III devices offer design security, decompression, and remote system upgrade features. Design security using configuration bitstream encryption is available in Stratix III devices, which protects your designs. Stratix III devices can receive a compressed configuration bitstream and decompress this data in real-time, reducing storage requirements and configuration time. You can make real-time system upgrades from remote locations of your Stratix III designs with the remote system upgrade feature.

Table 11–3 summarizes which configuration features you can use in each configuration scheme.

Table 11–3. Stratix III Configuration Features

Configuration Scheme	Configuration Method	Decompression	Design Security	Remote System Upgrade
FPP	MAX II device or a microprocessor with flash memory	✓ *(1)*	✓ *(1)*	—
Fast AS	Serial configuration device	✓	✓	✓ *(2)*
PS	MAX II device or a microprocessor with flash memory	✓	✓	—
	Download cable	✓	✓	—
JTAG	MAX II device or a microprocessor with flash memory	—	—	—
	Download cable	—	—	—

Notes to Table 11–3:

(1) In these modes, the host system must send a DCLK that is ×4 the data rate.

(2) Remote system upgrade is only available in the fast AS configuration scheme. Only remote update mode is supported when using the fast AS configuration scheme. Local update mode is not supported.

If your system already contains a common flash interface (CFI) flash memory, you can use it for the Stratix III device configuration storage as well. The MAX II parallel flash loader (PFL) feature in MAX II devices provides an efficient method to program CFI flash memory devices through the JTAG interface, and the logic to control configuration from the flash memory device to the Stratix III device. Both PS and FPP configuration modes are supported using the PFL feature.

For more information about PFL, refer to *AN 386: Using the MAX II Parallel Flash Loader with the Quartus II Software*.

For more information about programming Altera serial configuration devices, refer to "Programming Serial Configuration Devices" on page 11–25.

Configuration Data Decompression

Stratix III devices support configuration data decompression, which saves configuration memory space and time. This feature allows you to store compressed configuration data in configuration devices or other memory and transmit this compressed bitstream to Stratix III devices. During configuration, the Stratix III device decompresses the bitstream in real time and programs its SRAM cells.

☞ Preliminary data indicates that compression typically reduces the configuration bitstream size by 35 to 55%, based on the designs used.

Stratix III devices support decompression in the FPP (when using a MAX II device/microprocessor + flash), fast AS, and PS configuration schemes. The Stratix III decompression feature is not available in the JTAG configuration scheme.

☞ When using FPP mode, the intelligent host must provide a `DCLK` that is ×4 the data rate. Therefore, the configuration data must be valid for four `DCLK` cycles.

In PS mode, use the Stratix III decompression feature, because sending compressed configuration data reduces configuration time.

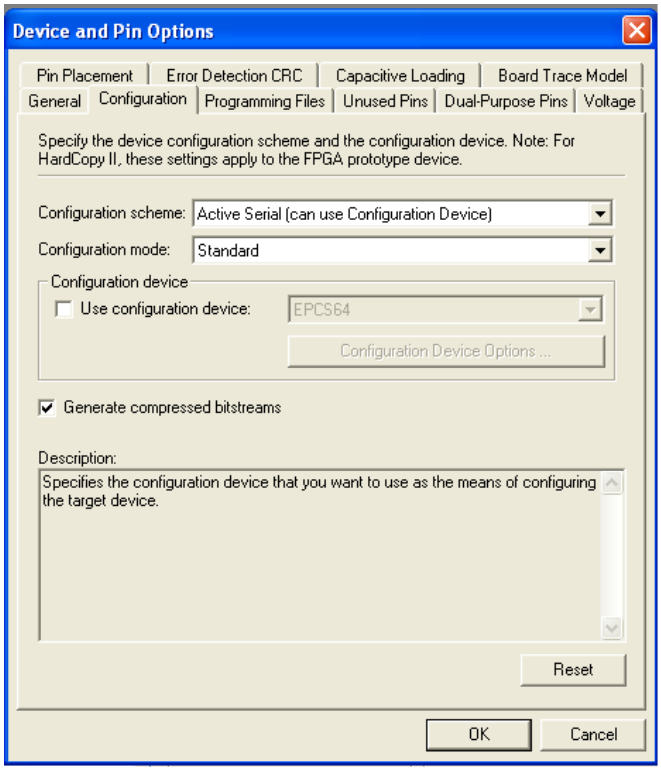
When you enable compression, the Quartus II software generates configuration files with compressed configuration data. This compressed file reduces the storage requirements in the configuration device or flash memory, and decreases the time needed to transmit the bitstream to the Stratix III device. The time required by a Stratix III device to decompress a configuration file is less than the time needed to transmit the configuration data to the device.

There are two ways to enable compression for Stratix III bitstreams: before design compilation (in the Compiler Settings menu) and after design compilation (in the Convert Programming Files window).

To enable compression in the project's Compiler Settings menu, perform the following steps:

1. On the Assignments menu, click **Device**. The **Settings** dialog box appears.
2. In the **Family** list, select **Stratix III** and then click the **Device and Pin Options** button.
3. On the **Configuration** tab, turn on the **Generate compressed bitstreams** option(Figure 11–1).

Figure 11–1. Enabling Compression for Stratix III Bitstreams in Compiler Settings



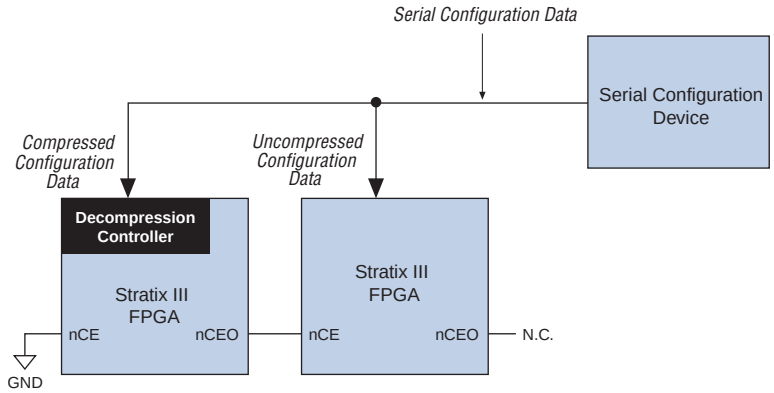
You can also enable compression when creating programming files from the Convert Programming Files window.

1. On the File menu, click **Convert Programming Files**.
2. In the (**.pof**, **.sram**, **.hex**, **.rbf**, or **.ttf**) list, select the programming file type.
3. For POF output files, select a configuration device from (**.pof**, **.sram**, **.hex**, **.rbf**, or **.ttf**).
4. Under **Input files to convert**, select **SOF Data**.
5. Select **Add File** and add a Stratix III device SOF or SOFs.
6. Select the name of the file you added to the **SOF Data** area and click **Properties**.
7. Check the **Compression** check box.

When multiple Stratix III devices are cascaded, you can selectively enable the compression feature for each device in the chain if you are using a serial configuration scheme. Figure 11–2 shows a chain of two Stratix III devices. The first Stratix III device has compression enabled, and receives a compressed bitstream from the configuration device. The second Stratix III device has the compression feature disabled, and receives uncompressed data.

In a multi-device FPP configuration chain (with a MAX II device/microprocessor + flash), all Stratix III devices in the chain must either enable or disable the decompression feature. You cannot selectively enable the compression feature for each device in the chain because of the DATA and `DCLK` relationship.

Figure 11–2. Compressed and Uncompressed Configuration Data in the Same Configuration File



To generate programming files for this setup in the Quartus II software, on the File menu, click **Convert Programming Files**.

Design Security Using Configuration Bitstream Encryption

Stratix III devices support decryption of configuration bitstreams using the advanced encryption standard (AES) algorithm—the most advanced encryption algorithm available today. Both non-volatile and volatile key programming are supported using Stratix III devices. When using the design security feature, a 256-bit security key is stored in the Stratix III device. To successfully configure a Stratix III device that has the design security feature enabled, the device must be configured with a configuration file that was encrypted using the same 256-bit security key. Non-volatile key programming does not require any external devices, such as a battery backup, for storage. However, for certain applications, you can store the security keys in volatile memory in the Stratix III device. An external battery is needed for this volatile key storage.

When using a serial configuration scheme such as PS or fast AS, configuration time is the same whether or not the design security feature is enabled. If the FPP scheme is used with the design security or decompression feature, a ×4 DCLK is required. This results in a slower configuration time when compared to the configuration time of a Stratix III device that has neither the design security nor the decompression feature enabled.

For more information about this feature, refer to the *Design Security in Stratix III Devices* chapter in volume 1 of the *Stratix III Device Handbook*.

Remote System Upgrade

Stratix III devices contain the remote update feature. For more information about this feature, refer to the *Remote System Upgrades with Stratix III Devices* in volume 1 of the *Stratix III Device Handbook*.

Power-On Reset Circuit

The POR circuit keeps the entire system in reset until the power supply voltage levels have stabilized on power-up. On power-up, the device does not release `nSTATUS` until V_{CCPT}, V_{CCL}, V_{CC}, V_{CCPD}, and V_{CCPGM} are above the device's POR trip point. On power down, brown-out occurs if V_{CC} or V_{CCL} ramps down below the POR trip point and V_{CC}, V_{CCPD}, or V_{CCPGM} drops below the threshold voltage.

In Stratix III devices, a pin-selectable option (`PORSEL`) is provided that allows you to select a typical POR time setting of 12 ms or 100 ms. In both cases, you can extend the POR time by using an external component to assert the `nSTATUS` pin low.

V_{CCPGM} Pins

Stratix III devices offer a new power supply, V_{CCPGM}, for all the dedicated configuration pins and dual function pins. The configuration voltages supported are 1.8 V, 2.5 V, 3.0 V, and 3.3 V. Stratix III devices do not support the 1.5 V configuration.

Use this pin to power all dedicated configuration inputs, dedicated configuration outputs, dedicated configuration bi-directional pins, and some of the dual functional pins that you use for configuration. With V_{CCPGM}, configuration input buffers do not have to share power lines with the regular I/O buffer in Stratix III devices.

The operating voltage for the configuration input pin is independent of the I/O bank's power supply V_{CCIO} during the configuration. Therefore, no configuration voltage constraints on V_{CCIO} are needed in Stratix III devices.

V_{CCPD} Pins

Stratix III devices have a dedicated programming power supply, V_{CCPD}, which must be connected to 3.3 V/3.0 V/2.5 V to power the I/O pre-drivers, the JTAG input and output pins (`TCK`, `TMS`, `TDI`, `TDO`, and `TRST`), and the design security circuitry.

V_{CCPGM} and V_{CCPD} must ramp up from 0 V to the desired voltage level within 100 ms. If these supplies are not ramped up within this specified time, your Stratix III device will not configure successfully. If your system does not allow ramp-up time of 100 ms or less, you must hold `nCONFIG` low until all power supplies are stable.

For more information about the configuration pins power supply, refer to "Device Configuration Pins" on page 11–43.

Fast Passive Parallel Configuration

Fast passive parallel (FPP) configuration in Stratix III devices is designed to meet the continuously increasing demand for faster configuration times. Stratix III devices are designed with the capability of receiving byte-wide configuration data per clock cycle. Table 11–4 lists the MSEL pin settings when using the FPP configuration scheme.

Table 11–4. Stratix III MSEL Pin Settings for FPP Configuration Schemes

Configuration Scheme	MSEL2	MSEL1	MSEL0
Fast Passive Parallel (FPP)	0	0	0
FPP with the design security feature, decompression feature, or both enabled *(1)*	0	0	1

Note to Table 11–4:

(1) These modes are only supported when using a MAX II device or a microprocessor with flash memory for configuration. In these modes, the host system must output a DCLK that is ×4 the data rate.

You can perform FPP configuration of Stratix III devices using an intelligent host, such as a MAX II device, or a microprocessor.

FPP Configuration Using a MAX II Device as an External Host

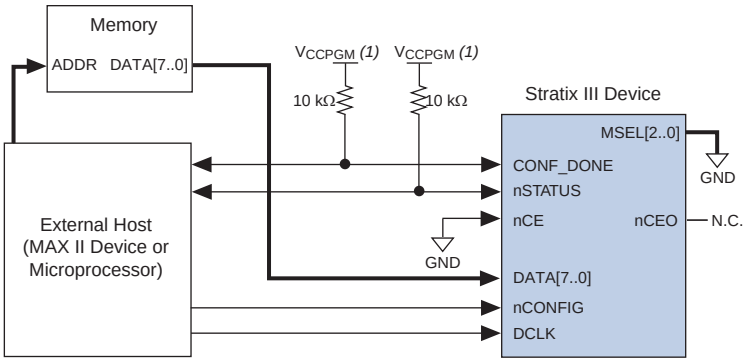
FPP configuration using compression and an external host provides the fastest method to configure Stratix III devices. In this configuration scheme, you can use a MAX II device as an intelligent host that controls the transfer of configuration data from a storage device, such as flash memory, to the target Stratix III device. You can store configuration data in **.rbf**, **.hex**, or **.ttf** format. When using the MAX II device as an intelligent host, a design that controls the configuration process, such as fetching the data from flash memory and sending it to the device, must be stored in the MAX II device.

If you are using the Stratix III decompression feature, design security feature or both, the external host must be able to send a DCLK frequency that is four times the data rate.

The ×4 DCLK signal does not require an additional pin and is sent on the DCLK pin. The maximum DCLK frequency is 100 MHz, which results in a maximum data rate of 200 Mbps. If you are not using the Stratix III decompression or design security features, the data rate is the same as the DCLK frequency.

Figure 11–3 shows the configuration interface connections between the Stratix III device and a MAX II device for single device configuration.

Figure 11–3. Single Device FPP Configuration Using an External Host



Note to Figure 11–3:

(1) Connect the resistor to a supply that provides an acceptable input signal for the Stratix III device. V_{CCPGM} should be high enough to meet the V_{IH} specification of the I/O on the external host. It is recommended to power up all configuration system's I/O with V_{CCPGM}.

Upon power-up, the Stratix III device goes through a POR. The POR delay is dependent on the `PORSEL` pin setting. When `PORSEL` is driven low, the POR time is approximately 100 ms. When `PORSEL` is driven high, the POR time is approximately 12 ms. During POR, the device resets, holds `nSTATUS` low, and tri-states all user I/O pins. After the device successfully exits POR, all user I/O pins continue to be tri-stated. If `nIO_pullup` is driven low during power-up and configuration, the user I/O pins and dual-purpose I/O pins have weak pull-up resistors, which are on (after POR) before and during configuration. If `nIO_pullup` is driven high, the weak pull-up resistors are disabled.

The configuration cycle consists of three stages: reset, configuration, and initialization. While `nCONFIG` or `nSTATUS` is low, the device is in the reset stage. To initiate configuration, the MAX II device must drive the `nCONFIG` pin from low to high.

☞ V_{CC}, V_{CCIO}, V_{CCPGM}, and V_{CCPD} of the banks where the configuration and JTAG pins reside must be fully powered to the appropriate voltage levels to begin the configuration process.

When `nCONFIG` goes high, the device comes out of reset and releases the open-drain `nSTATUS` pin, which is then pulled high by an external 10-kΩ pull-up resistor. After `nSTATUS` is released, the device is ready to receive configuration data and the configuration stage begins. When `nSTATUS` is pulled high, the MAX II device places the configuration data one byte at a time on the `DATA[7..0]` pins.

☞ Stratix III devices receive configuration data on the `DATA[7..0]` pins and the clock is received on the `DCLK` pin. Data is latched into the device on the rising edge of `DCLK`. If you are using the Stratix III decompression feature, design security feature, or both, the configuration data is latched on the rising edge of every fourth `DCLK` cycle. After the configuration data is latched in, it is processed during the following three `DCLK` cycles.

Data is continuously clocked into the target device until CONF_DONE goes high. A low-to-high transition on CONF_DONE indicates configuration is complete and initialization of the device can begin. The CONF_DONE pin must have an external 10-kΩ pull-up resistor for the device to initialize.

In Stratix III devices, the initialization clock source is either the internal oscillator (typically 10 MHz) or the optional CLKUSR pin. By default, the internal oscillator is the clock source for initialization. If you use the internal oscillator, the Stratix III device receives enough clock cycles for proper initialization. Therefore, if the internal oscillator is the initialization clock source, sending the entire configuration file to the device is sufficient to configure and initialize the device. Driving DCLK to the device after configuration is complete does not affect device operation.

You can also synchronize initialization of multiple devices or delay initialization with the CLKUSR option. You can turn on the **Enable user-supplied start-up clock** (**CLKUSR**) option in the Quartus II software from the **General** tab of the **Device and Pin Options** dialog box. Supplying a clock on CLKUSR does not affect the configuration process. After the CONF_DONE pin transitions high, CLKUSR is enabled after the time specified as t_{CD2CU}. When this time period elapses, Stratix III devices require 4,436 clock cycles to initialize properly and enter user mode. Stratix III devices support a CLKUSR f_{MAX} of 100 MHz.

An optional INIT_DONE pin is available, which signals the end of initialization and the start of user mode with a low-to-high transition. The **Enable INIT_DONE Output** option is available in the Quartus II software on the **General** tab of the **Device and Pin Options** dialog box. If you use the INIT_DONE pin, it is high because of an external 10-kΩ pull-up resistor when nCONFIG is low and during the beginning of configuration. After the option bit to enable INIT_DONE is programmed into the device (during the first frame of configuration data), the INIT_DONE pin goes low. When initialization is complete, the INIT_DONE pin is released and pulled high. The MAX II device must be able to detect this low-to-high transition, which signals the device has entered user mode. When initialization is complete, the device enters user mode. In user mode, the user I/O pins no longer have weak pull-up resistors and function as assigned in your design.

To ensure DCLK and DATA[7..0] are not left floating at the end of configuration, the MAX II device must drive them either high or low, whichever is convenient on your board. During configuration, DATA[7..0] pins are powered by V_{CCPGM}. After entering user mode, these pins are available as user I/O pins that are powered by V_{CCIO}. When you select the FPP scheme as a default in the Quartus II software, these I/O pins are tri-stated in user mode. To change this default option in the Quartus II software, select the **Dual-Purpose Pins** tab of the **Device and Pin Options** dialog box.

The configuration clock (DCLK) speed must be below the specified frequency to ensure correct configuration. No maximum DCLK period exists, which means you can pause configuration by halting DCLK for an indefinite amount of time.

If you are using the Stratix III decompression feature, design security feature, or both and need to stop DCLK, it can only be stopped three clock cycles after the last data byte was latched into the Stratix III device. If you are using the Stratix III device without decompression or design security feature, the DCLK can only be stopped two clock cycles after the last data byte was latched into the Stratix III device.

By stopping DCLK, the configuration circuit allows enough clock cycles to process the last byte of latched configuration data. When the clock restarts, the MAX II device must provide data on the DATA[7..0] pins prior to sending the first DCLK rising edge.

If an error occurs during configuration, the device drives its nSTATUS pin low, resetting itself internally. The low signal on the nSTATUS pin also alerts the MAX II device that there is an error. If the **Auto-restart configuration after error** option (available in the Quartus II software on the **General** tab of the **Device and Pin Options** dialog box) is turned on, the device releases nSTATUS after a reset time-out period (maximum of 100 µs). After nSTATUS is released and pulled high by a pull-up resistor, the MAX II device can try to reconfigure the target device without needing to pulse nCONFIG low. If this option is turned off, the MAX II device must generate a low-to-high transition (with a low pulse of at least 2 µs) on nCONFIG to restart the configuration process.

If you have enabled the **Auto-restart configuration after error** option, the nSTATUS pin transitions from high to low and back again to high when a configuration error is detected. This appears as a low pulse at the nSTATUS pin with a minimum pulse width of 10 µs to a maximum pulse width of 500 µs, as defined in the t_{STATUS} specification.

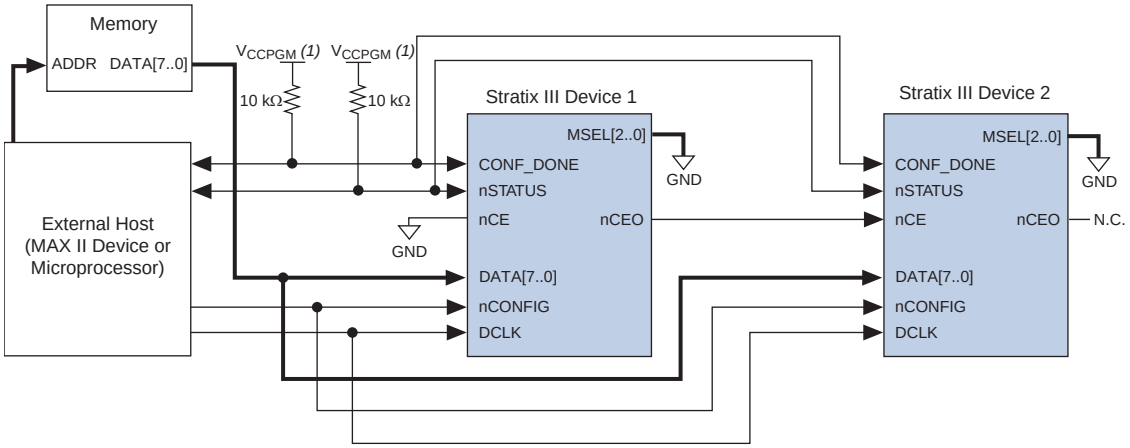
The MAX II device can also monitor the CONF_DONE and INIT_DONE pins to ensure successful configuration. The MAX II device must monitor the CONF_DONE pin to detect errors and determine when programming completes. If all configuration data is sent, but the CONF_DONE or INIT_DONE signals have not gone high, the MAX II device will reconfigure the target device.

If you use the optional CLKUSR pin and the nCONFIG is pulled low to restart configuration during device initialization, you must ensure CLKUSR continues toggling during the time nSTATUS is low (maximum of 100 µs).

When the device is in user mode, transitioning the nCONFIG pin low to high initiates a reconfiguration. The nCONFIG pin should be low for at least 2 µs. When nCONFIG is pulled low, the device also pulls nSTATUS and CONF_DONE low and all I/O pins are tri-stated. After nCONFIG returns to a logic high level and nSTATUS is released by the device, reconfiguration begins.

Figure 11–4 shows how to configure multiple devices using a MAX II device. This circuit is similar to the FPP configuration circuit for a single device, except the Stratix III devices are cascaded for multi-device configuration.

Figure 11–4. Multi-Device FPP Configuration Using an External Host



Note to Figure 11–4:

(1) Connect the resistor to a supply that provides an acceptable input signal for all Stratix III devices on the chain. V_{CCPGM} should be high enough to meet the V_{IH} specification of the I/O on the external host. It is recommended to power up all configuration system's I/O with V_{CCPGM}.

In a multi-device FPP configuration, the first device's nCE pin is connected to GND while its nCEO pin is connected to nCE of the next device in the chain. The last device's nCE input comes from the previous device, while its nCEO pin is left floating. After the first device completes configuration in a multi-device configuration chain, its nCEO pin drives low to activate the second device's nCE pin, which prompts the second device to begin configuration. The second device in the chain begins configuration within one clock cycle; therefore, the transfer of data destinations is transparent to the MAX II device. All other configuration pins (nCONFIG, nSTATUS, DCLK, DATA[7..0], and CONF_DONE) are connected to every device in the chain. The configuration signals may require buffering to ensure signal integrity and prevent clock skew problems. Ensure that the DCLK and DATA lines are buffered for every fourth device. Because all device CONF_DONE pins are tied together, all devices initialize and enter user mode at the same time.

All nSTATUS and CONF_DONE pins are tied together. If any device detects an error, configuration stops for the entire chain and you must reconfigure the entire chain. For example, if the first device flags an error on nSTATUS, it resets the chain by pulling its nSTATUS pin low. This behavior is similar to a single device detecting an error.

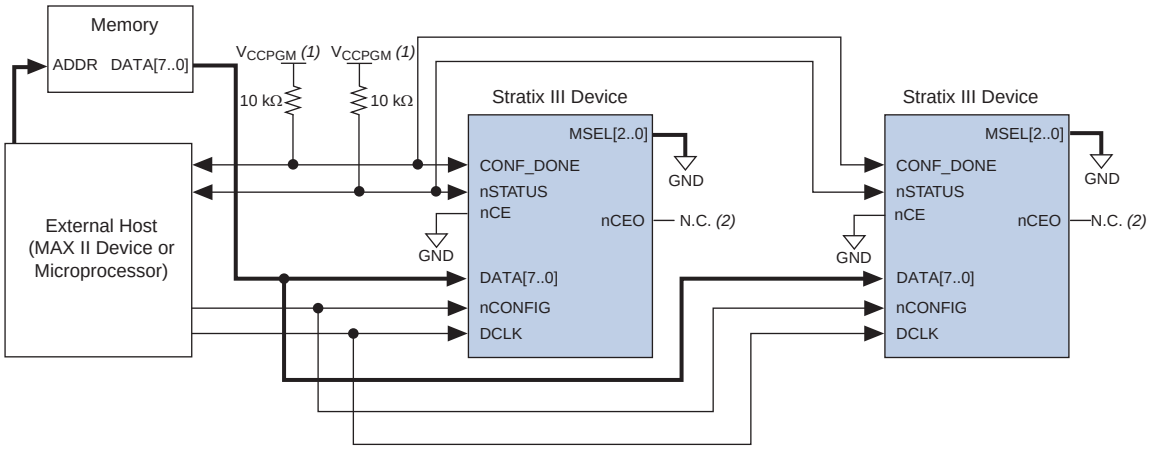
If the **Auto-restart configuration after error** option is turned on, the devices release their nSTATUS pins after a reset time-out period (maximum of 100 µs). After all nSTATUS pins are released and pulled high, the MAX II device tries to reconfigure the chain without pulsing nCONFIG low. If this option is turned off, the MAX II device must generate a low-to-high transition (with a low pulse of at least 2 µs) on nCONFIG to restart the configuration process.

If you have enabled the **Auto-restart configuration after error** option, the nSTATUS pin transitions from high to low and back again to high when a configuration error is detected. This appears as a low pulse at the nSTATUS pin with a minimum pulse width of 10 µs to a maximum pulse width of 500 µs, as defined in the t_{STATUS} specification.

In a multi-device FPP configuration chain, all Stratix III devices in the chain must either enable or disable the decompression feature, design security feature, or both. You cannot selectively enable the decompression feature, design security feature, or both for each device in the chain because of the DATA and DCLK relationship. If the chain contains devices that do not support design security, you should use a serial configuration scheme.

If a system has multiple devices that contain the same configuration data, tie all device nCE inputs to GND, and leave nCEO pins floating. All other configuration pins (nCONFIG, nSTATUS, DCLK, DATA[7..0], and CONF_DONE) are connected to every device in the chain. Configuration signals may require buffering to ensure signal integrity and prevent clock skew problems. Ensure that the DCLK and DATA lines are buffered for every fourth device. Devices must be the same density and package. All devices start and complete configuration at the same time. Figure 11–5 shows a multi-device FPP configuration when both Stratix III devices are receiving the same configuration data.

Figure 11–5. Multiple-Device FPP Configuration Using an External Host When Both Devices Receive the Same Data



Notes to Figure 11–5:

(1) Connect the resistor to a supply that provides an acceptable input signal for all Stratix III devices on the chain. V_{CCPGM} should be high enough to meet the V_{IH} specification of the I/O on the external host. It is recommended to power up all configuration system's I/O with V_{CCPGM}.

(2) The nCEO pins of both devices are left unconnected when configuring the same configuration data into multiple devices.

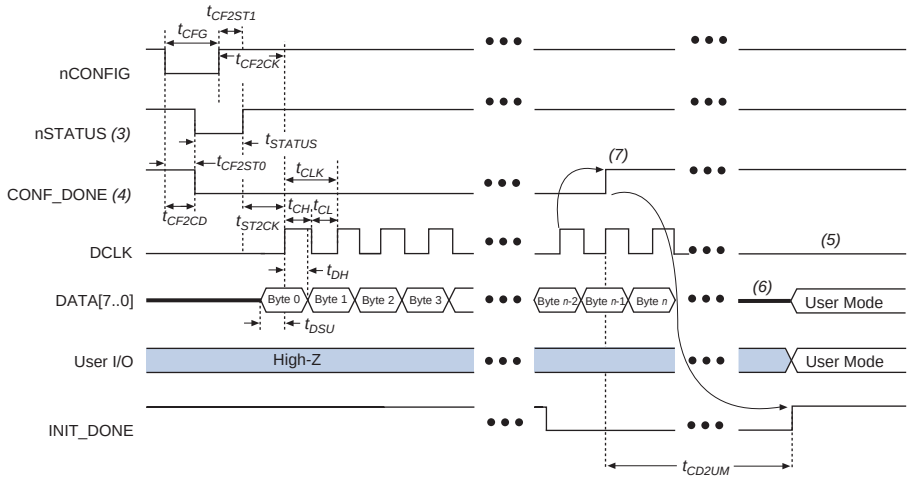
You can use a single configuration chain to configure Stratix III devices with other Altera devices that support FPP configuration, such as other types of Stratix devices. To ensure that all devices in the chain complete configuration at the same time, or that an error flagged by one device initiates reconfiguration in all devices, tie all of the device CONF_DONE and nSTATUS pins together.

For more information about configuring multiple Altera devices in the same configuration chain, refer to *Configuring Mixed Altera FPGA Chains* in the *Configuration Handbook*.

FPP Configuration Timing

Figure 11–6 shows the timing waveform for FPP configuration using a MAX II device as an external host. This waveform shows the timing when the decompression and the design security feature are not enabled.

Figure 11–6. FPP Configuration Timing Waveform *(Note 1)*, *(2)*



Notes to Figure 11–6:

(1) You should use this timing waveform when the decompression and design security features are not used.

(2) The beginning of this waveform shows the device in user mode. In user mode, nCONFIG, nSTATUS, and CONF_DONE are at logic high levels. When nCONFIG is pulled low, a reconfiguration cycle begins.

(3) Upon power-up, the Stratix III device holds nSTATUS low for the time of the POR delay.

(4) Upon power-up, before and during configuration, CONF_DONE is low.

(5) You should not leave DCLK floating after configuration. You should drive it high or low, whichever is more convenient.

(6) DATA[7..0] are available as user I/O pins after configuration. The state of these pins depends on the dual-purpose pin settings.

(7) Two DCLK falling edges are required after CONF_DONE goes high to begin the initialization of the device.

Table 11–5 defines the timing parameters for Stratix III devices for FPP configuration when the decompression and the design security features are not enabled.

Table 11–5. FPP Timing Parameters for Stratix III Devices *(Note 1)* (Part 1 of 2)

Symbol	Parameter	Minimum	Maximum	Units
t_{CF2CD}	nCONFIG low to CONF_DONE low	—	800	ns
t_{CF2ST0}	nCONFIG low to nSTATUS low	—	800	ns
t_{CFG}	nCONFIG low pulse width	2	—	µs
t_{STATUS}	nSTATUS low pulse width	10	100 *(2)*	µs
t_{CF2ST1}	nCONFIG high to nSTATUS high	—	100 *(2)*	µs
t_{CF2CK}	nCONFIG high to first rising edge on DCLK	100	—	µs
t_{ST2CK}	nSTATUS high to first rising edge of DCLK	2	—	µs
t_{DSU}	Data setup time before rising edge on DCLK	5	—	ns
t_{DH}	Data hold time after rising edge on DCLK	0	—	ns
t_{CH}	DCLK high time	4	—	ns
t_{CL}	DCLK low time	4	—	ns

Table 11–5. FPP Timing Parameters for Stratix III Devices *(Note 1)* (Part 2 of 2)

Symbol	Parameter	Minimum	Maximum	Units
t_{CLK}	DCLK period	10	—	ns
f_{MAX}	DCLK frequency	—	100	MHz
t_R	Input rise time	—	40	ns
t	Input fall time	—	40	ns
t_{CD2UM}	CONF_DONE high to user mode *(3)*	20	100	µs
t_{CD2CU}	CONF_DONE high to CLKUSR enabled	4 × maximum DCLK period	—	—
t_{CD2UMC}	CONF_DONE high to user mode with CLKUSR option on	tCD2CU + (4,436 × CLKUSR period)	—	—

Notes to Table 11–5:

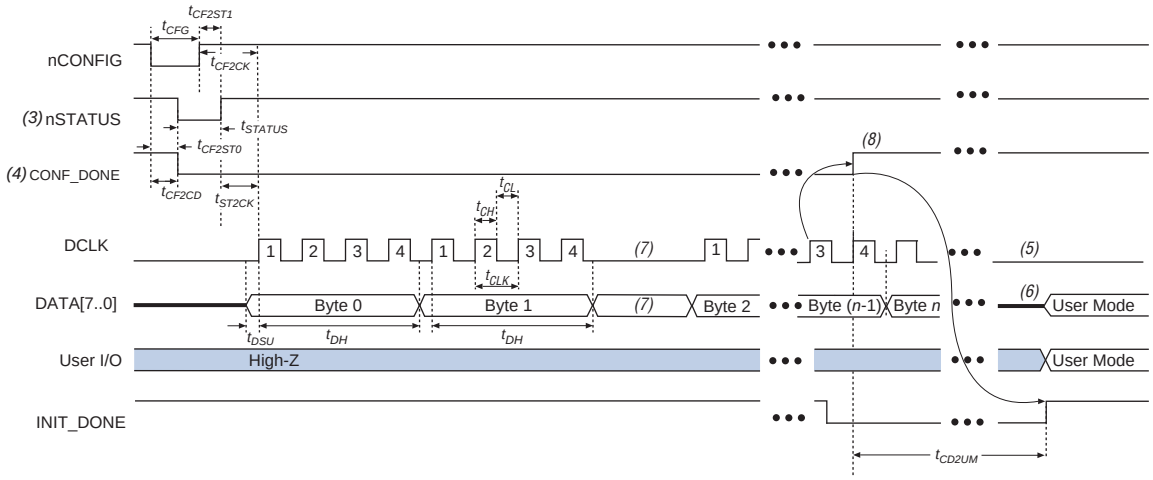
(1) Use these timing parameters when the decompression and design security features are not used.

(2) This value is obtainable if you do not delay configuration by extending the nCONFIG or nSTATUS low pulse width.

(3) The minimum and maximum numbers apply only if you chose the internal oscillator as the clock source for starting up the device.

Figure 11–7 shows the timing waveform for FPP configuration when using a MAX II device as an external host. This waveform shows the timing when the decompression feature, design security feature, or both are enabled.

Figure 11–7. FPP Configuration Timing Waveform with Decompression or Design Security Feature Enabled *(Note 1), (2)*



Notes to Figure 11–7:

(1) Use this timing waveform when the decompression feature, design security feature, or both are used.

(2) The beginning of this waveform shows the device in user mode. In user mode, nCONFIG, nSTATUS, and CONF_DONE are at logic high levels. When nCONFIG is pulled low, a reconfiguration cycle begins.

(3) Upon power-up, the Stratix III device holds nSTATUS low for the time of the POR delay.

(4) Upon power-up, before and during configuration, CONF_DONE is low.

(5) Do not leave DCLK floating after configuration. Drive it high or low, whichever is more convenient.

(6) DATA[7..0] are available as user I/O pins after configuration. The state of these pins depends on the dual-purpose pin settings.

(7) If needed, pause DCLK by holding it low. When DCLK restarts, the external host must provide data on the DATA[7..0] pins prior to sending the first DCLK rising edge.

(8) Two DCLK falling edges are required after CONF_DONE goes high to begin the initialization of the device.

Table 11–6 defines the timing parameters for Stratix III devices for FPP configuration when the decompression feature, design security feature, or both are enabled.

Table 11–6. FPP Timing Parameters for Stratix III Devices with Decompression or Design Security Feature Enabled *(Note 1)*

Symbol	Parameter	Minimum	Maximum	Units
t_{CF2CD}	nCONFIG low to CONF_DONE low	—	800	ns
t_{CF2ST0}	nCONFIG low to nSTATUS low	—	800	ns
t_{CFG}	nCONFIG low pulse width	2	—	µs
t_{STATUS}	nSTATUS low pulse width	10	100 *(2)*	µs
t_{CF2ST1}	nCONFIG high to nSTATUS high	—	100 *(2)*	µs
t_{CF2CK}	nCONFIG high to first rising edge on DCLK	100	—	µs
t_{ST2CK}	nSTATUS high to first rising edge of DCLK	2	—	µs
t_{DSU}	Data setup time before rising edge on DCLK	5	—	ns
t_{DH}	Data hold time after rising edge on DCLK	30	—	ns
t_{CH}	DCLK high time	4	—	ns
t_{CL}	DCLK low time	4	—	ns
t_{CLK}	DCLK period	10	—	ns
f_{MAX}	DCLK frequency	—	100	MHz
t_{DATA}	Data rate	—	200	Mbps
t_R	Input rise time	—	40	ns
t	Input fall time	—	40	ns
t_{CD2UM}	CONF_DONE high to user mode *(3)*	20	100	µs
t_{CD2CU}	CONF_DONE high to CLKUSR enabled	4 × maximum DCLK period	—	—
t_{CD2UMC}	CONF_DONE high to user mode with CLKUSR option on	t_{CD2CU} + (4,436 × CLKUSR period)	—	—

Notes to Table 11–6:

(1) Use these timing parameters when the decompression and design security features are used.

(2) This value is obtainable if you do not delay configuration by extending the nCONFIG or nSTATUS low pulse width.

(3) The minimum and maximum numbers apply only if you chose the internal oscillator as the clock source for starting up the device.

Device configuration options and how to create configuration files are discussed further in the *Device Configuration Options* and *Configuration File Formats* chapters in volume 2 of the *Configuration Handbook*.

FPP Configuration Using a Microprocessor

In this configuration scheme, a microprocessor can control the transfer of configuration data from a storage device, such as flash memory, to the target Stratix III device.

All information in "FPP Configuration Using a MAX II Device as an External Host" on page 11–8 is also applicable when using a microprocessor as an external host. Refer to this section for all configuration and timing information.

Fast Active Serial Configuration (Serial Configuration Devices)

In the fast AS configuration scheme, Stratix III devices are configured using a serial configuration device. These configuration devices are low-cost devices with non-volatile memory that feature a simple four-pin interface and a small form factor. These features make serial configuration devices an ideal low-cost configuration solution.

For more information about serial configuration devices, refer to the *Serial Configuration Devices Data Sheet* in the *Configuration Handbook*.

Serial configuration devices provide a serial interface to access configuration data. During device configuration, Stratix III devices read configuration data through the serial interface, decompress data if necessary, and configure their SRAM cells. This scheme is referred to as the AS configuration scheme, because the Stratix III device controls the configuration interface. This scheme contrasts with the PS configuration scheme, where the configuration device controls the interface.

The Stratix III decompression and design security features are fully available when configuring your Stratix III device using fast AS mode.

Table 11–7 lists the MSEL pin settings for the AS configuration scheme.

Table 11–7. Stratix III MSEL Pin Settings for AS Configuration Schemes *(Note 1)*

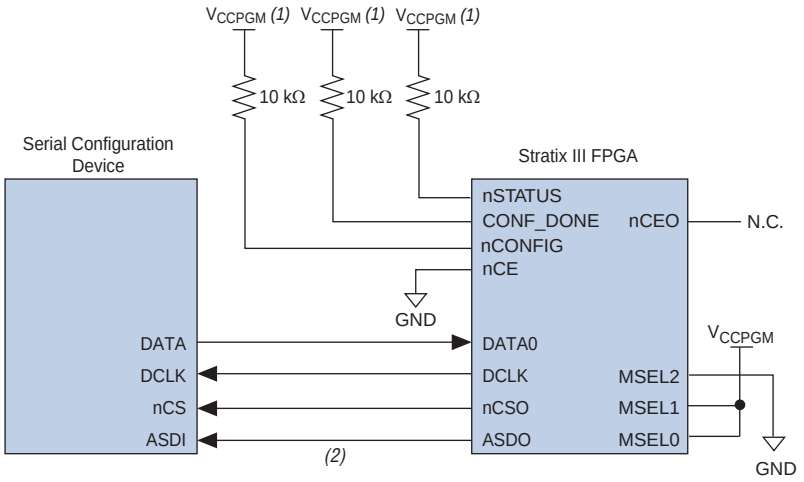
Configuration Scheme	MSEL2	MSEL1	MSEL0
Fast AS (40 MHz)	0	1	1
Remote system upgrade fast AS (40 MHz)	0	1	1

Note to Table 11–7:

(1) Use EPCS16, EPCS64, or EPCS128 devices.

Serial configuration devices have a four-pin interface: serial clock input (DCLK), serial data output (DATA), AS data input (ASDI), and an active-low chip select (nCS). This four-pin interface connects to Stratix III device pins, as shown in Figure 11–8.

Figure 11–8. Single Device Fast AS Configuration



Notes to Figure 11–8:

(1) Connect the pull-up resistors to V_{CCPGM} at 3.3-V supply.

(2) Stratix III devices use the ASDO-to-ASDI path to control the configuration device.

Upon power-up, the Stratix III devices go through a POR. The POR delay is dependent on the PORSEL pin setting. When PORSEL is driven low, the POR time is approximately 100 ms. If PORSEL is driven high, the POR time is approximately 12 ms. During POR, the device will reset, hold nSTATUS and CONF_DONE low, and tri-state all user I/O pins. After the device successfully exits POR, all user I/O pins continue to be tri-stated. If nIO_pullup is driven low during power-up and configuration, the user I/O pins and dual-purpose I/O pins will have weak pull-up resistors which are on (after POR) before and during configuration. If nIO_pullup is driven high, the weak pull-up resistors are disabled.

The configuration cycle consists of three stages: reset, configuration, and initialization. While nCONFIG or nSTATUS are low, the device is in reset. After POR, the Stratix III device releases nSTATUS, which is pulled high by an external 10-kΩ pull-up resistor and enters configuration mode.

To begin configuration, power the V_{CC}, V_{CCIO}, V_{CCPGM}, and V_{CCPD} voltages (for the banks where the configuration and JTAG pins reside) to the appropriate voltage levels.

The serial clock (DCLK) generated by the Stratix III device controls the entire configuration cycle and provides the timing for the serial interface. Stratix III devices use an internal oscillator to generate DCLK. Using the MSEL[] pins, you can select to use a 40 MHz oscillator.

In fast AS configuration schemes, Stratix III devices drive out control signals on the falling edge of DCLK. The serial configuration device responds to the instructions by driving out configuration data on the falling edge of DCLK. Then the data is latched into the Stratix III device on the following falling edge of DCLK.

In configuration mode, Stratix III devices enable the serial configuration device by driving the nCSO output pin low, which connects to the chip select (nCS) pin of the configuration device. The Stratix III device uses the serial clock (DCLK) and serial data output (ASDO) pins to send operation commands, read address signals, or both to the serial configuration device. The configuration device provides data on its serial data output (DATA) pin, which connects to the DATA0 input of the Stratix III devices.

After all configuration bits are received by the Stratix III device, it releases the open-drain CONF_DONE pin, which is pulled high by an external 10-kΩ resistor. Initialization begins only after the CONF_DONE signal reaches a logic high level. All AS configuration pins (DATA0, DCLK, nCSO, and ASDO) have weak internal pull-up resistors that are always active. After configuration, these pins are set as input tri-stated and are driven high by the weak internal pull-up resistors. The CONF_DONE pin must have an external 10-kΩ pull-up resistor for the device to initialize.

In Stratix III devices, the initialization clock source is either the 10 MHz (typical) internal oscillator (separate from the active serial internal oscillator) or the optional CLKUSR pin. By default, the internal oscillator is the clock source for initialization. If you use the internal oscillator, the Stratix III device has enough clock cycles for proper initialization. You also have the flexibility to synchronize initialization of multiple devices or to delay initialization with the CLKUSR option. You can turn on the **Enable user-supplied start-up clock** (**CLKUSR**) option in the Quartus II software on the **General** tab of the **Device and Pin Options** dialog box. When you enable the **user supplied start-up clock** option, the CLKUSR pin is the initialization clock source. Supplying a clock on CLKUSR does not affect the configuration process. When all configuration data has been accepted and CONF_DONE goes high, CLKUSR is enabled after 600 ns. After this time period elapses, Stratix III devices require 4,436 clock cycles to initialize properly and enter user mode. Stratix III devices support a CLKUSR f_{MAX} of 100 MHz.

An optional INIT_DONE pin is available, which signals the end of initialization and the start of user-mode with a low-to-high transition. The **Enable INIT_DONE Output** option is available in the Quartus II software on the **General** tab of the **Device and Pin Options** dialog box. If you use the INIT_DONE pin, it will be high due to an external 10-kΩ pull-up resistor when nCONFIG is low and during the beginning of configuration. When the option bit to enable INIT_DONE is programmed into the device (during the first frame of configuration data), the INIT_DONE pin goes low. When initialization is complete, the INIT_DONE pin is released and pulled high. This low-to-high transition signals that the device has entered user mode. When initialization is complete, the device enters user mode. In user mode, the user I/O pins no longer have weak pull-up resistors and function as assigned in your design.

If an error occurs during configuration, Stratix III devices assert the nSTATUS signal low, indicating a data frame error, and the CONF_DONE signal stays low. If the **Auto-restart configuration after error** option (available in the Quartus II software on the **General** tab of the **Device and Pin Options** dialog box) is turned on, the Stratix III device resets the configuration device by pulsing nCSO, releases nSTATUS after a reset time-out period (maximum of 100 µs), and retries configuration. If this option is turned off, the system must monitor nSTATUS for errors and then pulse nCONFIG low for at least 2 µs to restart configuration.

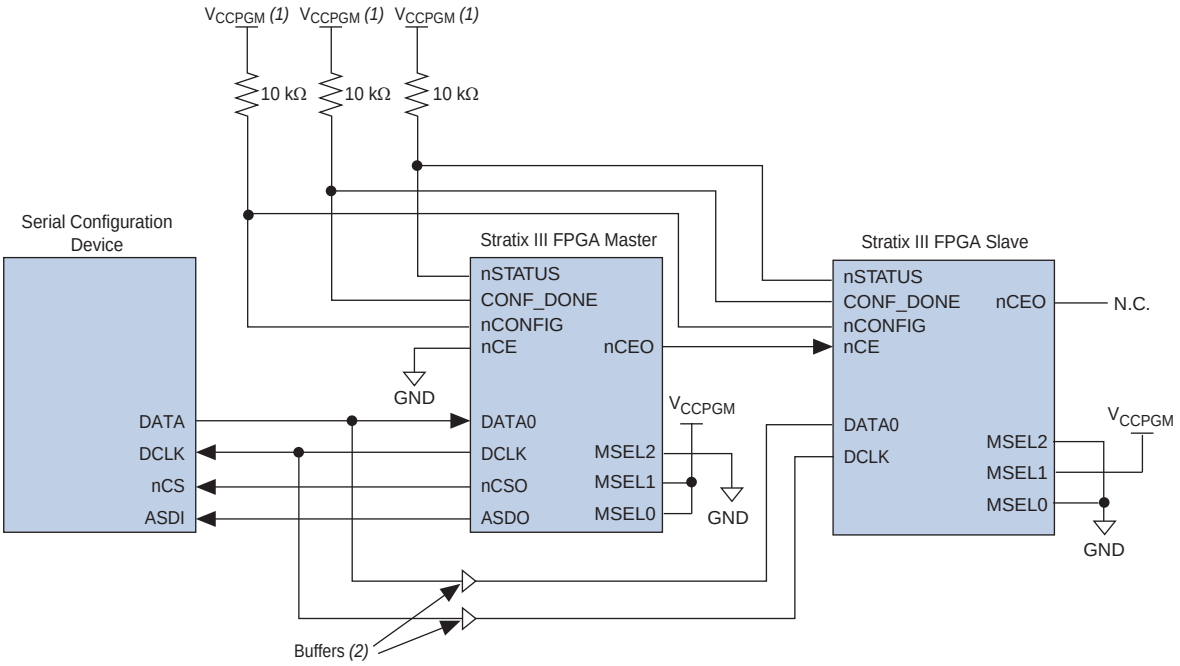
If you have enabled the **Auto-restart configuration after error** option, the nSTATUS pin transitions from high to low and back again to high when a configuration error is detected. This appears as a low pulse at the nSTATUS pin with a minimum pulse width of 10 µs to a maximum pulse width of 500 µs, as defined in the t_{STATUS} specification.

When the Stratix III device is in user mode, you can initiate reconfiguration by pulling the nCONFIG pin low. The nCONFIG pin should be low for at least 2 µs. When nCONFIG is pulled low, the device also pulls nSTATUS and CONF_DONE low and all I/O pins are tri-stated. When nCONFIG returns to a logic high level and nSTATUS is released by the Stratix III device, reconfiguration begins.

You can configure multiple Stratix III devices using a single serial configuration device. You can cascade multiple Stratix III devices using the chip-enable (nCE) and chip-enable-out (nCEO) pins. The first device in the chain must have its nCE pin connected to ground. You must connect its nCEO pin to the nCE pin of the next device in the chain. When the first device captures all of its configuration data from the bitstream, it drives the nCEO pin low, enabling the next device in the chain. You must leave the nCEO pin of the last device unconnected. The nCONFIG, nSTATUS, CONF_DONE, DCLK, and DATA0 pins of each device in the chain are connected (refer to Figure 11–9).

This first Stratix III device in the chain is the configuration master and controls configuration of the entire chain. You must connect its MSEL pins to select the AS configuration scheme. The remaining Stratix III devices are configuration slaves. You must connect their MSEL pins to select the PS configuration scheme. Any other Altera device that supports PS configuration can also be part of the chain as a configuration slave. Figure 11–9 shows the pin connections for this setup.

Figure 11–9. Multi-Device Fast AS Configuration



Notes to Figure 11–9:

(1) Connect the pull-up resistors to V_{CCPGM} at 3.3-V supply.

(2) Connect the repeater buffers between the Stratix III master and slave device(s) for `DATA[0]` and `DCLK`. This prevents any potential signal integrity and clock skew problems.

As shown in Figure 11–9, the `nSTATUS` and `CONF_DONE` pins on all target devices are connected with external pull-up resistors. These pins are open-drain bi-directional pins on the devices. When the first device asserts `nCEO` (after receiving all of its configuration data), it releases its `CONF_DONE` pin. The subsequent devices in the chain keep this shared `CONF_DONE` line low until they have received their configuration data. When all target devices in the chain have received their configuration data and released `CONF_DONE`, the pull-up resistor drives a high level on this line and all devices simultaneously enter initialization mode.

If an error occurs at any point during configuration, the `nSTATUS` line is driven low by the failing device. If you enable the **Auto-restart configuration after error** option, reconfiguration of the entire chain begins after a reset time-out period (maximum of 100 µs). If the **Auto-restart configuration after error** option is turned off, the external system must monitor `nSTATUS` for errors and then pulse `nCONFIG` low to restart configuration. The external system can pulse `nCONFIG` if it is under system control rather than tied to V_{CCPGM}.

If you have enabled the **Auto-restart configuration after error** option, the `nSTATUS` pin transitions from high to low and back again to high when a configuration error is detected. This appears as a low pulse at the `nSTATUS` pin with a minimum pulse width of 10 µs to a maximum pulse width of 500 µs, as defined in the t_{STATUS} specification.

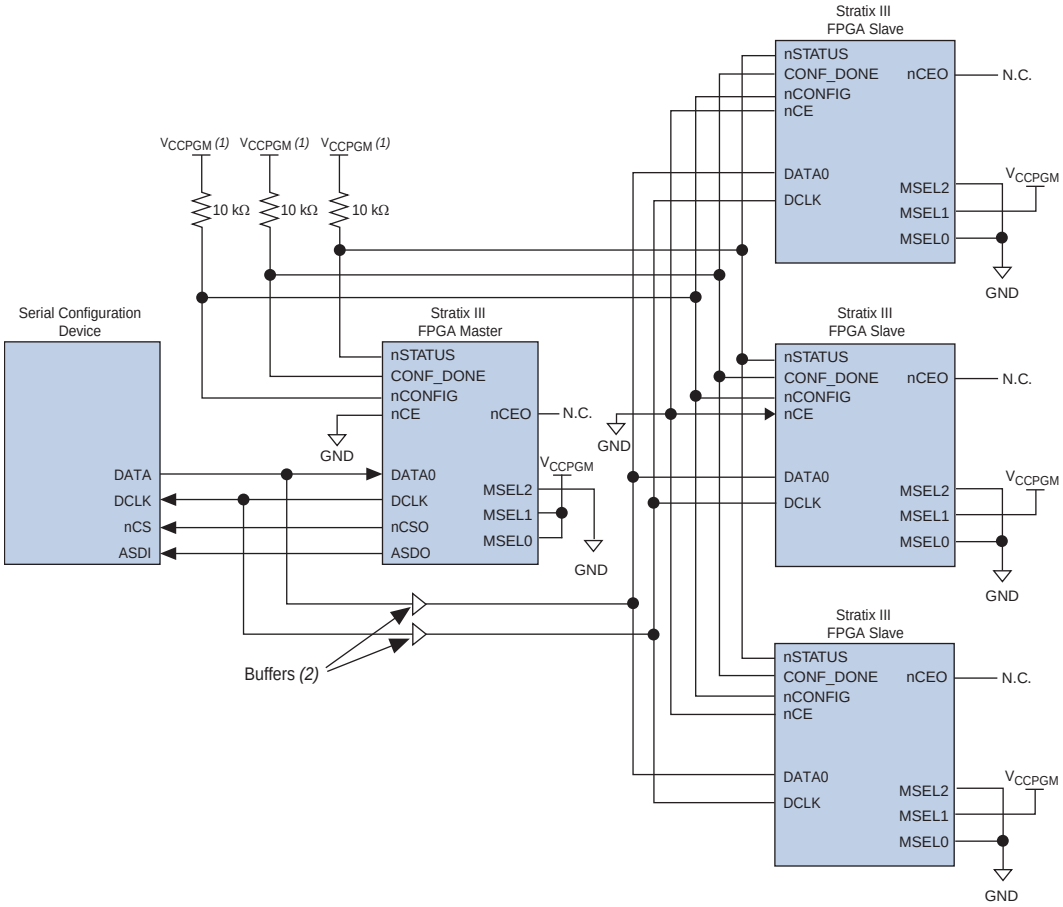
While you can cascade Stratix III devices, you cannot cascade or chain together serial configuration devices.

If the configuration bitstream size exceeds the capacity of a serial configuration device, you must select a larger configuration device, enable the compression feature, or both. When configuring multiple devices, the size of the bitstream is the sum of the individual devices' configuration bitstreams.

A system may have multiple devices that contain the same configuration data. In active serial chains, you can implement this by storing one copy of the SOF in the serial configuration device. The same copy of the SOF configures the master Stratix III device and all remaining slave devices concurrently. All Stratix III devices must be the same density and package. The master device is set up in active serial mode and the slave devices are set up in passive serial mode.

To configure four identical Stratix III devices with the same SOF, you could set up the chain similar to the example shown in Figure 11–10. The first device is the master device, and its `MSEL` pins should be set to select AS configuration. The other three slave devices are set up for concurrent configuration, and their `MSEL` pins should be set to select PS configuration. The `nCE` input pins from the master and slave are connected to `GND`, and the `DATA` and `DCLK` pins connect in parallel to all four devices. During the configuration cycle, the master device reads its configuration data from the serial configuration device and transmits the second copy of the configuration data to all three slave devices, configuring all of them simultaneously.

Figure 11–10. Multi-Device Fast AS Configuration When the Devices Receive the Same Data Using a Single SOF



Notes to Figure 11–10:

(1) Connect the pull-up resistors to V_{CCPGM} at 3.3-V supply.

(2) Connect the repeater buffers between the Stratix III master and slave device(s) for `DATA[0]` and `DCLK`. This prevents any potential signal integrity and clock skew problems.

Estimating Active Serial Configuration Time

Active serial configuration time is dominated by the time it takes to transfer data from the serial configuration device to the Stratix III device. This serial interface is clocked by the Stratix III `DCLK` output (generated from an internal oscillator). Because the Stratix III device only supports fast AS configuration, the `DCLK` frequency needs to be set to 40 MHz (25 ns). The minimum configuration time estimate for an EP3SL50 device (15 MBits of uncompressed data) is shown in Equation 11–1 and Example 11–1.

Equation 11–1.

$$\text{RBF Size} \times (\text{minimum DCLK period} / 1 \text{ bit per DCLK cycle}) = \text{estimated minimum configuration time}$$

Example 11–1.

$$15 \text{ Mbits} \times (25 \text{ ns} / 1 \text{ bit}) = 375 \text{ ms}$$

Enabling compression reduces the amount of configuration data that is transmitted to the Stratix III device, which also reduces configuration time. On average, compression reduces configuration time, depending on the design.

Figure 11–11 shows the timing waveform for AS configuration.

Figure 11–11. Fast AS Configuration Timing

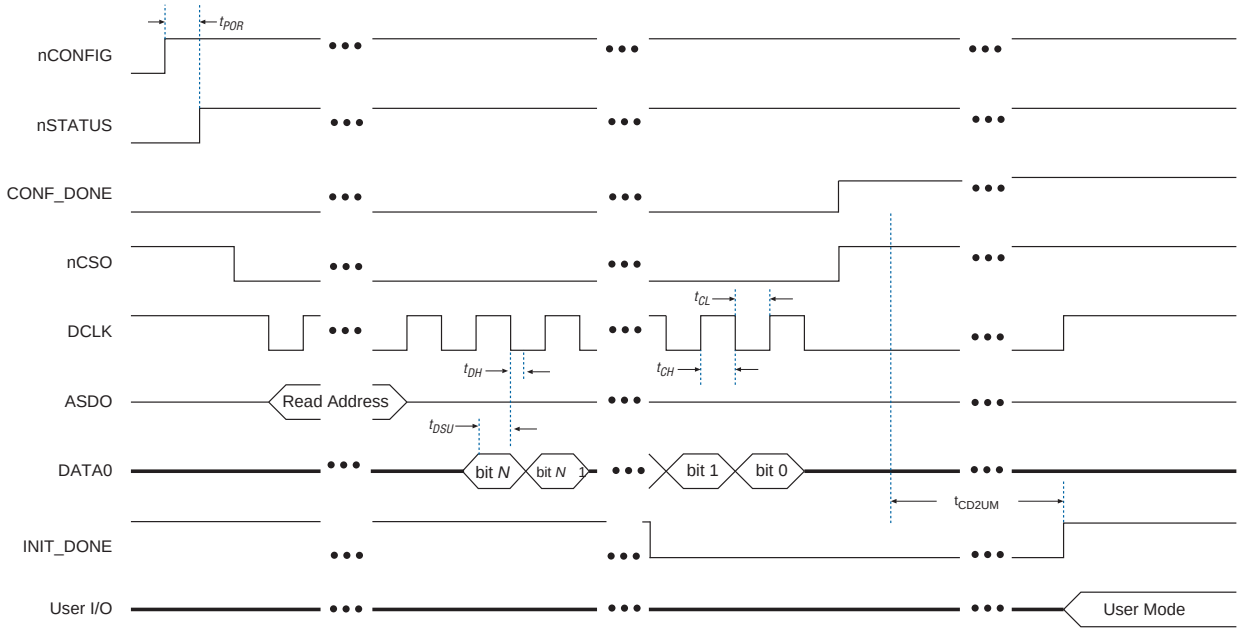


Table 11–8 defines the timing parameters for Stratix III devices for fast AS configuration.

Table 11–8. Fast AS Timing Parameters for Stratix III Devices

Symbol	Parameter	Min	Typ	Max	Units
f_{CLK}	DCLK frequency from Stratix III	15	25	40	MHz
t_{CF2ST1}	nCONFIG high to nSTATUS high	—	—	100	µs
t_{DSU}	Data setup time before falling edge on DCLK	7	—	—	ns
t_{DH}	Data hold time after falling edge on DCLK	0	—	—	ns
t_{CH}	DCLK high time	10	—	—	ns
t_{CL}	DCLK low time	10	—	—	ns
t_{CD2UM}	CONF_DONE high to user mode	20	—	100	µs

Programming Serial Configuration Devices

Serial configuration devices are non-volatile, flash-memory-based devices. You can program these devices in-system using the USB-Blaster™ or ByteBlaster™ II download cable. Alternatively, you can program them using the Altera programming unit (APU), supported third-party programmers, or a microprocessor with the SRunner software driver.

You can perform in-system programming of serial configuration devices by using the conventional AS programming interface or JTAG interface solution.

Because serial configuration devices do not support the JTAG interface, the conventional method to program them is by using the AS programming interface. The configuration data used to program serial configuration devices is downloaded by using the programming hardware.

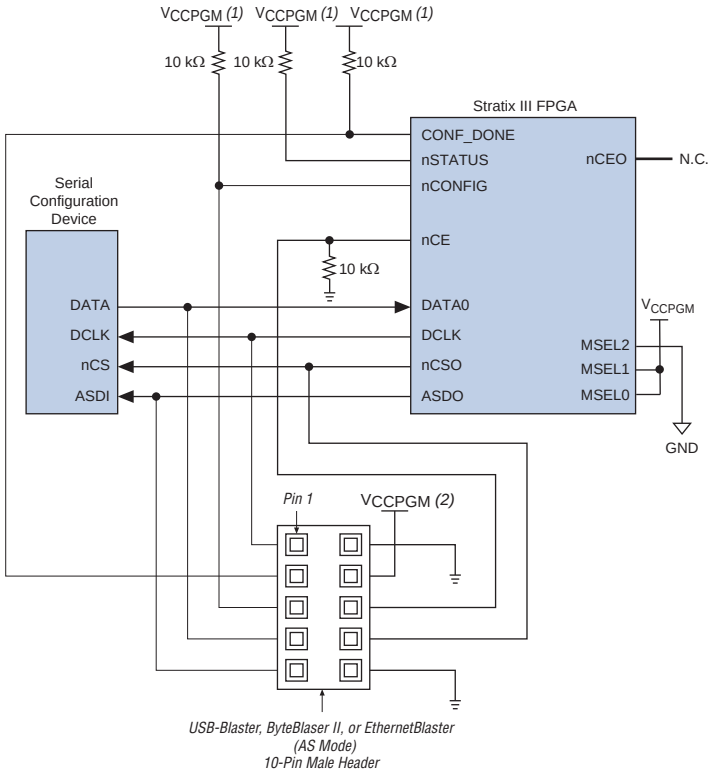
During in-system programming, the download cable disables device access to the AS interface by driving the `nCE` pin high. Stratix III devices are also held in reset by a low level on `nCONFIG`. After programming is complete, the download cable releases `nCE` and `nCONFIG`, allowing the pull-down and pull-up resistors to drive GND and V_{CCPGM}, respectively. Figure 11–12 shows the download cable connections to the serial configuration device.

Altera has developed the Serial FlashLoader (SFL), which is an in-system programming solution for serial configuration devices using the JTAG interface. This solution requires the Stratix III device to be a bridge between the JTAG interface and the serial configuration device.

For more information about the SFL, refer to *AN 370: Using the Serial FlashLoader with the Quartus II Software*.

For more information about the USB-Blaster download cable, refer to the *USB-Blaster Download Cable User Guide*. For more information about the ByteBlaster II cable, refer to the *ByteBlaster II Download Cable User Guide*.

Figure 11–12. In-System Programming of Serial Configuration Devices



Notes to Figure 11–12:

(1) Connect the pull-up resistors to V_{CCPGM} at 3.3-V supply.

(2) Power up the USB-Blaster, ByteBlaster II, or EthernetBlaster cable's $V_{CC(TRGT)}$ with V_{CCPGM}.

You can program serial configuration devices with the Quartus II software using the Altera programming hardware and the appropriate configuration device programming adapter.

In production environments, you can program serial configuration devices using multiple methods. You can use Altera programming hardware or other third-party programming hardware to program blank serial configuration devices before they are mounted onto PCBs. Alternatively, you can use an on-board microprocessor to program the serial configuration device in-system using C-based software drivers provided by Altera.

You can program a serial configuration device in-system by an external microprocessor using SRunner. SRunner is a software driver developed for embedded serial configuration device programming that can be easily customized to fit in different embedded systems. SRunner is able to read a raw programming data (.**rpd**) file and write to the serial configuration devices. The serial configuration device programming time using SRunner is comparable to the programming time with the Quartus II software.

For more information about SRunner, refer to *AN 418: SRunner: An Embedded Solution for EPCS Programming* and the source code on the Altera website at www.altera.com.

For more information about programming serial configuration devices and fast AS Configuration Timing, refer to the *Serial Configuration Devices (EPCS1, EPCS4, EPCS16, EPCS64, and EPCS128) Data Sheet* in the *Configuration Handbook*.

Passive Serial Configuration

You can program PS configuration of Stratix III devices using an intelligent host, such as a MAX II device or microprocessor with flash memory, or a download cable. In the PS scheme, an external host (a MAX II device, embedded processor, or host PC) controls configuration. Configuration data is clocked into the target Stratix III device by using the `DATA0` pin at each rising edge of `DCLK`.

The Stratix III decompression and design security features are fully available when configuring your Stratix III device using PS mode.

Table 11–9 lists the `MSEL` pin settings when using the PS configuration scheme.

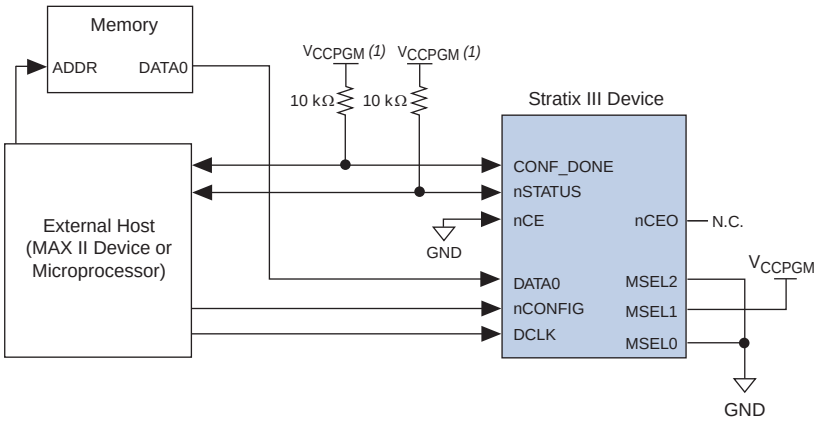
Table 11–9. Stratix III MSEL Pin Settings for PS Configuration Scheme

Configuration Scheme	MSEL2	MSEL1	MSEL0
PS	0	1	0

PS Configuration Using a MAX II Device as an External Host

In this configuration scheme, you can use a MAX II device as an intelligent host that controls the transfer of configuration data from a storage device, such as flash memory, to the target Stratix III device. You can store configuration data in **.rbf**, **.hex**, or **.ttf** format. Figure 11–13 shows the configuration interface connections between a Stratix III device and a MAX II device for single device configuration.

Figure 11–13. Single Device PS Configuration Using an External Host



Note to Figure 11–13:

(1) Connect the resistor to a supply that provides an acceptable input signal for the Stratix III device. V_{CCPGM} must be high enough to meet the V_{IH} specification of the I/O on the external host. It is recommended to power up all configuration systems' I/O with V_{CCPGM}.

Upon power-up, Stratix III devices go through a POR. The POR delay is dependent on the PORSEL pin setting. When PORSEL is driven low, the POR time is approximately 100 ms. When PORSEL is driven high, the POR time is approximately 12 ms. During POR, the device resets, holds nSTATUS low, and tri-states all user I/O pins. When the device successfully exits POR, all user I/O pins continue to be tri-stated. If nIO_pullup is driven low during power-up and configuration, the user I/O pins and dual-purpose I/O pins will have weak pull-up resistors that are on (after POR) before and during configuration. If nIO_pullup is driven high, the weak pull-up resistors are disabled.

The configuration cycle consists of three stages: reset, configuration, and initialization. While nCONFIG or nSTATUS are low, the device is in reset. To initiate configuration, the MAX II device must generate a low-to-high transition on the nCONFIG pin.

☞ V_{CC}, V_{CCIO}, V_{CCPGM}, and V_{CCPD} of the banks where the configuration and JTAG pins reside must be fully powered to the appropriate voltage levels to begin the configuration process.

When nCONFIG goes high, the device comes out of reset and releases the open-drain nSTATUS pin, which is then pulled high by an external 10-kΩ pull-up resistor. When nSTATUS is released, the device is ready to receive configuration data and the configuration stage begins. When nSTATUS is pulled high, the MAX II device should place the configuration data one bit at a time on the DATA0 pin. If you are using configuration data in **.rbf**, **.hex**, or **.ttf** format, you must send the least significant bit (LSB) of each data byte first. For example, if the RBF contains the byte sequence 02 1B EE 01 FA, the serial bitstream you must transmit to the device is 0100-0000 1101-1000 0111-0111 1000-0000 0101-1111.

The Stratix III device receives configuration data on the DATA0 pin and the clock is received on the DCLK pin. Data is latched into the device on the rising edge of DCLK. Data is continuously clocked into the target device until CONF_DONE goes high. After the device has received all configuration data successfully, it releases the open-drain CONF_DONE pin, which is pulled high by an external 10-kΩ pull-up resistor. A low-to-high transition on CONF_DONE indicates configuration is complete and initialization of the device can begin. The CONF_DONE pin must have an external 10-kΩ pull-up resistor for the device to initialize.

In Stratix III devices, the initialization clock source is either the internal oscillator (typically 10 MHz) or the optional CLKUSR pin. By default, the internal oscillator is the clock source for initialization. If you use the internal oscillator, the Stratix III device has enough clock cycles for proper initialization. Therefore, if the internal oscillator is the initialization clock source, sending the entire configuration file to the device is sufficient to configure and initialize the device. Driving DCLK to the device after configuration is complete does not affect device operation.

You also have the flexibility to synchronize initialization of multiple devices or to delay initialization with the CLKUSR option. You can turn on the **Enable user-supplied start-up clock** (**CLKUSR**) option in the Quartus II software on the **General** tab of the **Device and Pin Options** dialog box. If you supply a clock on CLKUSR, it will not affect the configuration process. After all configuration data has been accepted and CONF_DONE goes high, CLKUSR will be enabled after the time specified as t_{CD2CU}. After this time period elapses, Stratix III devices require 4,436 clock cycles to initialize properly and enter user mode. Stratix III devices support a CLKUSR f_{MAX} of 100 MHz.

An optional INIT_DONE pin is available, which signals the end of initialization and the start of user-mode with a low-to-high transition. The **Enable INIT_DONE Output** option is available in the Quartus II software on the **General** tab of the **Device and Pin Options** dialog box. If you use the INIT_DONE pin, it will be high due to an external 10-kΩ pull-up resistor when nCONFIG is low and during the beginning of configuration. When the option bit to enable INIT_DONE is programmed into the device (during the first frame of configuration data), the INIT_DONE pin will go low. When initialization is complete, the INIT_DONE pin will be released and pulled high. The MAX II device must be able to detect this low-to-high transition, which signals the device has entered user mode. When initialization is complete, the device enters user mode. In user mode, the user I/O pins will no longer have weak pull-up resistors and will function as assigned in your design.

To ensure DCLK and DATA0 are not left floating at the end of configuration, the MAX II device must drive them either high or low, whichever is convenient on your board. The DATA[0] pin is available as a user I/O pin after configuration. When you choose the PS scheme as a default in the Quartus II software, this I/O pin is tri-stated in user mode and should be driven by the MAX II device. To change this default option in the Quartus II software, click the **Dual-Purpose Pins** tab of the **Device and Pin Options** dialog box.

The configuration clock (DCLK) speed must be below the specified frequency to ensure correct configuration. No maximum DCLK period exists, which means you can pause configuration by halting DCLK for an indefinite amount of time.

If an error occurs during configuration, the device drives its nSTATUS pin low, resetting itself internally. The low signal on the nSTATUS pin also alerts the MAX II device that there is an error. If the **Auto-restart configuration after error** option (available in the Quartus II software on the **General** tab of the **Device and Pin Options** dialog box) is turned on, the Stratix III device releases nSTATUS after a reset time-out period (maximum of 100 µs). After nSTATUS is released and pulled high by a pull-up resistor, the MAX II device can attempt to reconfigure the target device without needing to pulse nCONFIG low. If this option is turned off, the MAX II device must generate a low-to-high transition (with a low pulse of at least 2 µs) on nCONFIG to restart the configuration process.

☞ If you have enabled the **Auto-restart configuration after error** option, the nSTATUS pin transitions from high to low and back again to high when a configuration error is detected. This appears as a low pulse at the nSTATUS pin with a minimum pulse width of 10 µs to a maximum pulse width of 500 µs, as defined in the t_{STATUS} specification.

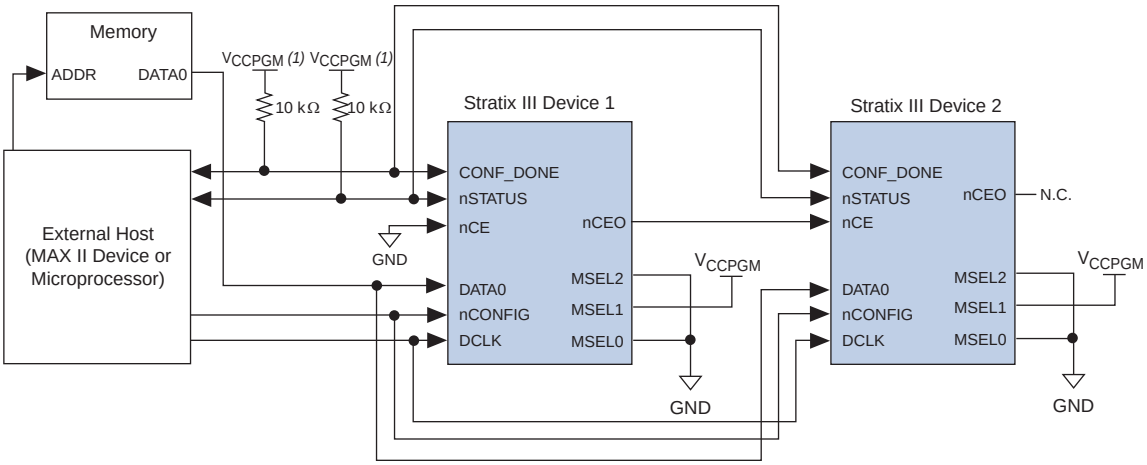
The MAX II device can also monitor the CONF_DONE and INIT_DONE pins to ensure successful configuration. The CONF_DONE pin must be monitored by the MAX II device to detect errors and determine when programming completes. If all configuration data is sent, but CONF_DONE or INIT_DONE have not gone high, the MAX II device must reconfigure the target device.

☞ If you use the optional CLKUSR pin and nCONFIG is pulled low to restart configuration during device initialization, you must ensure that CLKUSR continues toggling during the time nSTATUS is low (maximum of 100 µs).

When the device is in user mode, you can initiate a reconfiguration by transitioning the nCONFIG pin low-to-high. The nCONFIG pin must be low for at least 2 µs. When nCONFIG is pulled low, the device also pulls nSTATUS and CONF_DONE low and all I/O pins are tri-stated. When nCONFIG returns to a logic high level and nSTATUS is released by the device, reconfiguration begins.

Figure 11–14 shows how to configure multiple devices using a MAX II device. This circuit is similar to the PS configuration circuit for a single device, except Stratix III devices are cascaded for multi-device configuration.

Figure 11–14. Multi-Device PS Configuration Using an External Host



Note to Figure 11–14:

(1) Connect the resistor to a supply that provides an acceptable input signal for all Stratix III devices on the chain. V_{CCPGM} must be high enough to meet the V_{IH} specification of the I/O on the external host. It is recommended to power up all configuration systems' I/O with V_{CCPGM}.

In multi-device PS configuration, the first device's nCE pin is connected to GND while its nCEO pin is connected to nCE of the next device in the chain. The last device's nCE input comes from the previous device, while its nCEO pin is left floating. After the first device completes configuration in a multi-device configuration chain, its nCEO pin drives low to activate the second device's nCE pin, which prompts the second device to begin configuration. The second device in the chain begins configuration within one clock cycle. Therefore, the transfer of data destinations is transparent to the MAX II device. All other configuration pins (nCONFIG, nSTATUS, DCLK, DATA0, and CONF_DONE) are connected to every device in the chain. Configuration signals can require buffering to ensure signal integrity and prevent clock skew problems. Ensure that the DCLK and DATA lines are buffered for every fourth device. Because all device CONF_DONE pins are tied together, all devices initialize and enter user mode at the same time.

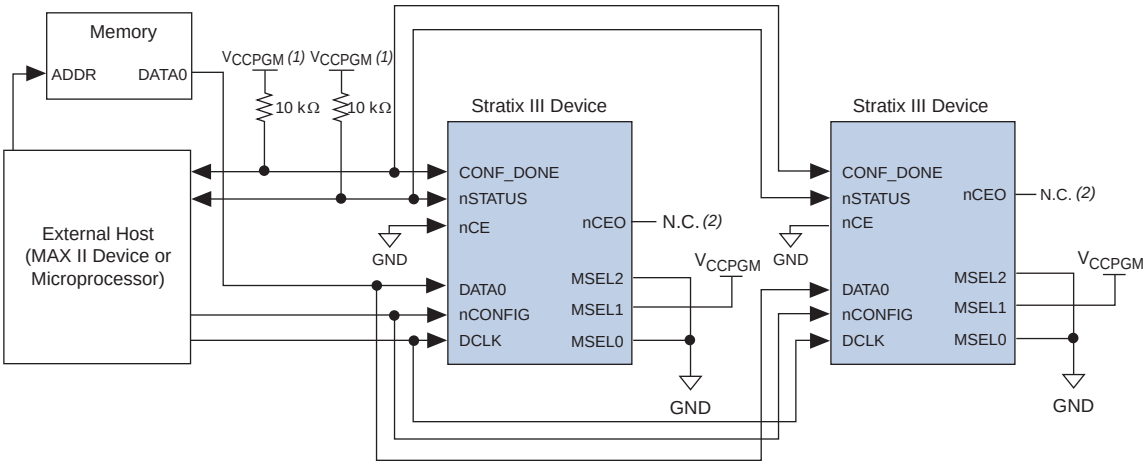
Since all nSTATUS and CONF_DONE pins are tied together, if any device detects an error, configuration stops for the entire chain and you must reconfigure the entire chain. For example, if the first device flags an error on nSTATUS, it resets the chain by pulling its nSTATUS pin low. This behavior is similar to a single device detecting an error.

If the **Auto-restart configuration after error** option is turned on, the devices release their nSTATUS pins after a reset time-out period (maximum of 100 µs). After all nSTATUS pins are released and pulled high, the MAX II device can attempt to reconfigure the chain without needing to pulse nCONFIG low. If this option is turned off, the MAX II device must generate a low-to-high transition (with a low pulse of at least 2 µs) on nCONFIG to restart the configuration process.

If you have enabled the **Auto-restart configuration after error** option, the nSTATUS pin transitions from high to low and back again to high when a configuration error is detected. This appears as a low pulse at the nSTATUS pin with a minimum pulse width of 10 µs to a maximum pulse width of 500 µs, as defined in the t_{STATUS} specification.

In your system, you can have multiple devices that contain the same configuration data. To support this configuration scheme, all device nCE inputs are tied to GND, while nCEO pins are left floating. All other configuration pins (nCONFIG, nSTATUS, DCLK, DATA0, and CONF_DONE) are connected to every device in the chain. Configuration signals can require buffering to ensure signal integrity and prevent clock skew problems. Ensure that the DCLK and DATA lines are buffered for every fourth device. Devices must be the same density and package. All devices will start and complete configuration at the same time. Figure 11–15 shows multi-device PS configuration when both Stratix III devices are receiving the same configuration data.

Figure 11–15. Multiple-Device PS Configuration When Both Devices Receive the Same Data



Notes to Figure 11–15:

(1) Connect the resistor to a supply that provides an acceptable input signal for all Stratix III devices on the chain. V_{CCPGM} must be high enough to meet the V_{IH} specification of the I/O on the external host. It is recommended to power up all configuration systems' I/O with V_{CCPGM}.

(2) The nCEO pins of both devices are left unconnected when configuring the same configuration data into multiple devices.

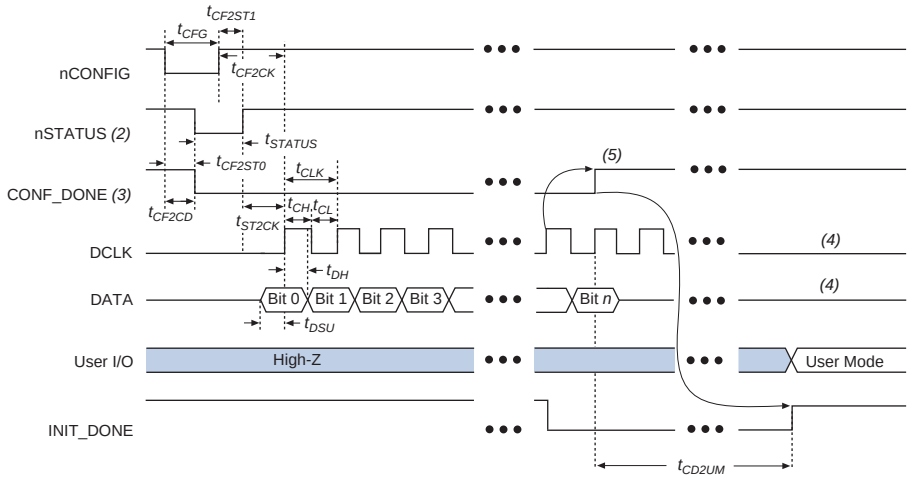
You can use a single configuration chain to configure Stratix III devices with other Altera devices. To ensure that all devices in the chain complete configuration at the same time, or that an error flagged by one device initiates reconfiguration in all devices, all of the device CONF_DONE and nSTATUS pins must be tied together.

For more information about configuring multiple Altera devices in the same configuration chain, refer to the *Configuring Mixed Altera FPGA Chains* chapter in the *Configuration Handbook*.

PS Configuration Timing

Figure 11–16 shows the timing waveform for PS configuration when using a MAX II device as an external host.

Figure 11–16. PS Configuration Timing Waveform *(Note 1)*



Notes to Figure 11–16:

(1) The beginning of this waveform shows the device in user-mode. In user-mode, nCONFIG, nSTATUS, and CONF_DONE are at logic high levels. When nCONFIG is pulled low, a reconfiguration cycle begins.

(2) Upon power-up, the Stratix III device holds nSTATUS low for the time of the POR delay.

(3) Upon power-up, before and during configuration, CONF_DONE is low.

(4) Do not leave DCLK floating after configuration. You should drive it high or low, whichever is more convenient. DATA[0] is available as a user I/O pin after configuration. The state of this pin depends on the dual-purpose pin settings.

(5) Two DCLK falling edges are required after CONF_DONE goes high to begin the initialization of the device.

Table 11–10 defines the timing parameters for Stratix III devices for PS configuration.

Table 11–10. PS Timing Parameters for Stratix III Devices (Part 1 of 2)

Symbol	Parameter	Minimum	Maximum	Units
t_{CF2CD}	nCONFIG low to CONF_DONE low	—	800	ns
t_{CF2ST0}	nCONFIG low to nSTATUS low	—	800	ns
t_{CFG}	nCONFIG low pulse width	2	—	µs
t_{STATUS}	nSTATUS low pulse width	10	100 *(1)*	µs
t_{CF2ST1}	nCONFIG high to nSTATUS high	—	100 *(1)*	µs
t_{CF2CK}	nCONFIG high to first rising edge on DCLK	100	—	µs
t_{ST2CK}	nSTATUS high to first rising edge of DCLK	2	—	µs
t_{DSU}	Data setup time before rising edge on DCLK	5	—	ns
t_{DH}	Data hold time after rising edge on DCLK	0	—	ns
t_{CH}	DCLK high time	4	—	ns
t_{CL}	DCLK low time	4	—	ns
t_{CLK}	DCLK period	10	—	ns
f_{MAX}	DCLK frequency	—	100	MHz
t_R	Input rise time	—	40	ns

Table 11–10. PS Timing Parameters for Stratix III Devices (Part 2 of 2)

Symbol	Parameter	Minimum	Maximum	Units
t	Input fall time	—	40	ns
t_{CD2UM}	CONF_DONE high to user mode *(2)*	20	100	µs
t_{CD2CU}	CONF_DONE high to CLKUSR enabled	4 × maximum DCLK period	—	—
t_{CD2UMC}	CONF_DONE high to user mode with CLKUSR option on	tCD2CU + (4,436 × CLKUSR period)	—	—

Notes to Table 11–10:

(1) This value is applicable if you do not delay configuration by extending the nCONFIG or nSTATUS low pulse width.

(2) The minimum and maximum numbers apply only if you choose the internal oscillator as the clock source for starting the device.

Device configuration options and how to create configuration files are discussed further in the *Device Configuration Options and Configuration File Formats* chapters in volume 2 of the *Configuration Handbook*.

PS Configuration Using a Microprocessor

In this PS configuration scheme, a microprocessor can control the transfer of configuration data from a storage device, such as flash memory, to the target Stratix III device.

You can do a PS configuration using MicroBlaster™ Passive Serial Software Driver. For more information, refer to *AN423: Configuring the MicroBlaster Passive Serial Software Driver*.

For all configuration and timing information, refer to "PS Configuration Using a MAX II Device as an External Host" on page 11–27. This section is also applicable when using a microprocessor as an external host.

PS Configuration Using a Download Cable

In this section, the generic term *download cable* includes the Altera USB-Blaster USB port download cable, MasterBlaster™ serial/USB communications cable, ByteBlaster II parallel port download cable, ByteBlasterMV™ parallel port download cable, and the EthernetBlaster download cable.

In PS configuration with a download cable, an intelligent host (such as a PC) transfers data from a storage device to the device by using the USB-Blaster, MasterBlaster, ByteBlaster II, EthernetBlaster, or ByteBlasterMV cable.

Upon power-up, the Stratix III devices go through a POR. The POR delay is dependent on the PORSEL pin setting. When PORSEL is driven low, the POR time is approximately 100 ms. If PORSEL is driven high, the POR time is approximately 12 ms. During POR, the device will reset, hold nSTATUS low, and tri-state all user I/O pins. After the device successfully exits POR, all user I/O pins continue to be tri-stated. If nIO_pullup is driven low during power-up and configuration, the user I/O pins and dual-purpose I/O pins will have weak pull-up resistors which are on (after POR) before and during configuration. If nIO_pullup is driven high, the weak pull-up resistors are disabled.

The configuration cycle consists of three stages: reset, configuration and initialization. While nCONFIG or nSTATUS are low, the device is in reset. To initiate configuration in this scheme, the download cable generates a low-to-high transition on the nCONFIG pin.

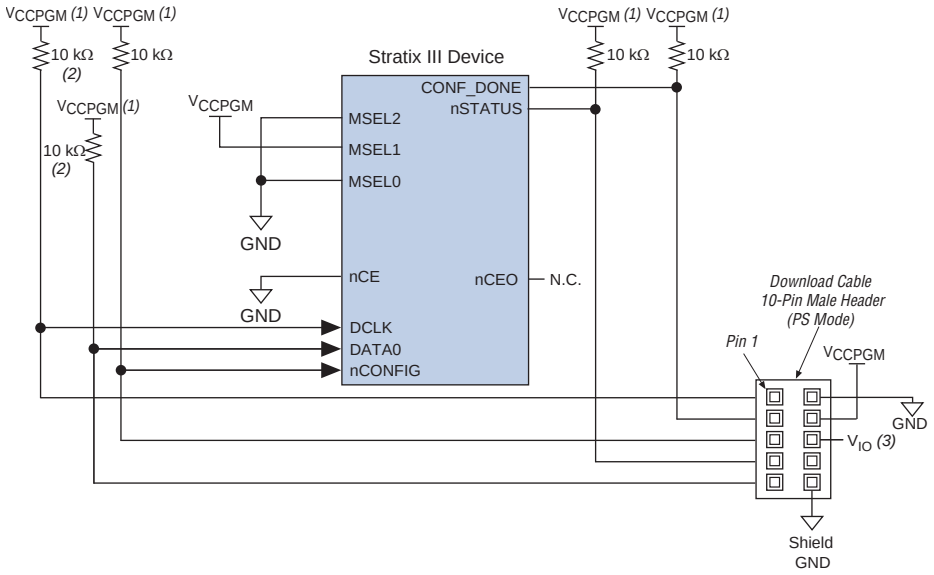
To begin configuration, power the V_{CC}, V_{CCIO}, V_{CCPGM}, and V_{CCPD} voltages (for the banks where the configuration and JTAG pins reside) to the appropriate voltage levels.

When nCONFIG goes high, the device comes out of reset and releases the open-drain nSTATUS pin, which is then pulled high by an external 10-kΩ pull-up resistor. After nSTATUS is released, the device is ready to receive configuration data and the configuration stage begins. The programming hardware or download cable then places the configuration data one bit at a time on the device's DATA0 pin. The configuration data is clocked into the target device until CONF_DONE goes high. The CONF_DONE pin must have an external 10-kΩ pull-up resistor in order for the device to initialize.

When using a download cable, setting the **Auto-restart configuration after error** option does not affect the configuration cycle because you must manually restart configuration in the Quartus II software when an error occurs. Additionally, the **Enable user-supplied start-up clock** (**CLKUSR**) option has no affect on the device initialization since this option is disabled in the SOF when programming the device using the Quartus II programmer and download cable. Therefore, if you turn on the CLKUSR option, you do not need to provide a clock on CLKUSR when you are configuring the device with the Quartus II programmer and a download cable.

Figure 11–17 shows PS configuration for Stratix III devices using a USB-Blaster, MasterBlaster, ByteBlaster II, ByteBlasterMV, or EthernetBlaster cable.

Figure 11–17. PS Configuration Using a Download Cable



Notes to Figure 11–17:

(1) You should connect the pull-up resistor to the same supply voltage (V_{CCPGM}) as the USB-Blaster, MasterBlaster (VIO pin), ByteBlaster II, ByteBlasterMV, or EthernetBlaster cable.

(2) You only need the pull-up resistors on DATA0 and DCLK if the download cable is the only configuration scheme used on your board. This ensures that DATA0 and DCLK are not left floating after configuration. For example, if you are also using a configuration device, you do not need the pull-up resistors on DATA0 and DCLK.

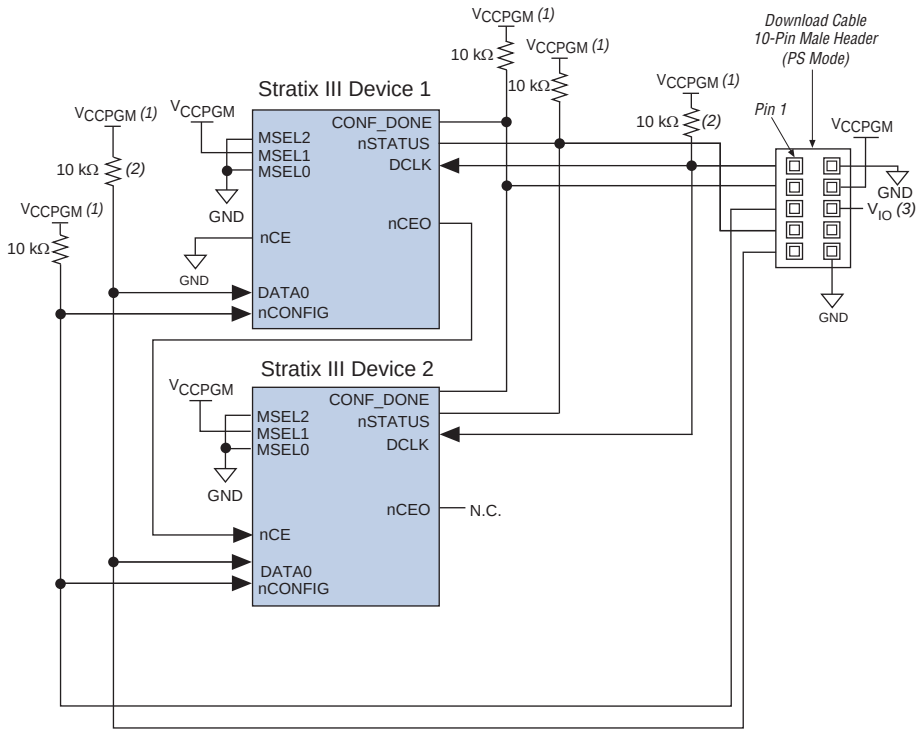
(3) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. V_{IO} should match the device's V_{CCPGM}. Refer to the *MasterBlaster Serial/USB Communications Cable Data Sheet* for this value. In the USB-Blaster, ByteBlaster II, ByteBlasterMV, or EthernetBlaster cable, this pin is a no connect.

You can use a download cable to configure multiple Stratix III devices by connecting each device's nCEO pin to the subsequent device's nCE pin. The first device's nCE pin is connected to GND while its nCEO pin is connected to the nCE of the next device in the chain. The last device's nCE input comes from the previous device, while its nCEO pin is left floating. All other configuration pins (nCONFIG, nSTATUS, DCLK, DATA0, and CONF_DONE) are connected to every device in the chain. Because all CONF_DONE pins are tied together, all devices in the chain initialize and enter user mode at the same time.

In addition, because the nSTATUS pins are tied together, the entire chain halts configuration if any device detects an error. The **Auto-restart configuration after error** option does not affect the configuration cycle because you must manually restart configuration in the Quartus II software when an error occurs.

Figure 11–18 shows how to configure multiple Stratix III devices with a download cable.

Figure 11–18. Multi-Device PS Configuration using a Download Cable



Notes to Figure 11–18:

(1) Connect the pull-up resistor to the same supply voltage (V_{CCPGM}) as the USB-Blaster, MasterBlaster (V_{IO} pin), ByteBlaster II, ByteBlasterMV, or EthernetBlaster cable.

(2) You only need the pull-up resistors on DATA0 and DCLK if the download cable is the only configuration scheme used on your board. This is to ensure that DATA0 and DCLK are not left floating after configuration. For example, if you are also using a configuration device, you do not need the pull-up resistors on DATA0 and DCLK.

(3) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. V_{IO} should match the device's V_{CCPGM}. Refer to the *MasterBlaster Serial/USB Communications Cable Data Sheet* for this value. In the USB-Blaster, ByteBlaster II, ByteBlasterMV, or EthernetBlaster cable, this pin is a no connect.

For more information about how to use the USB-Blaster, MasterBlaster, ByteBlaster II, ByteBlasterMV, or EthernetBlaster cable, refer to the following user guides:

- *USB-Blaster USB Port Download Cable User Guide*
- *MasterBlaster Serial/USB Communications Cable User Guide*
- *ByteBlaster II Parallel Port Download Cable User Guide*
- *ByteBlasterMV Parallel Port Download Cable User Guide*
- *EthernetBlaster Download Cable User Guide*

JTAG Configuration

The JTAG has developed a specification for boundary-scan testing. This boundary-scan test (BST) architecture offers the capability to efficiently test components on PCBs with tight lead spacing. The BST architecture can test pin connections without using physical test probes and capture functional data while a device is operating normally. You can also use the JTAG circuitry to shift configuration data into the device. The Quartus II software automatically generates SOFs that can be used for JTAG configuration with a download cable in the Quartus II software programmer.

For more information about JTAG boundary-scan testing and commands available using Stratix III devices, refer to the following documents:

- *IEEE 1149.1 (JTAG) Boundary Scan Testing in Stratix III Device* chapter of the *Stratix III Device Handbook*
- *AN 425: Using the Command-Line Jam STAPL Solution for Device Programming*

Stratix III devices are designed such that JTAG instructions have precedence over any device configuration modes. Therefore, JTAG configuration can take place without waiting for other configuration modes to complete. For example, if you attempt JTAG configuration of Stratix III devices during PS configuration, PS configuration is terminated and JTAG configuration begins.

You cannot use the Stratix III decompression or design security features if you are configuring your Stratix III device when using JTAG-based configuration.

A device operating in JTAG mode uses four required pins, `TDI`, `TDO`, `TMS`, and `TCK`, and one optional pin, `TRST`. The `TCK` pin has an internal weak pull-down resistor, while the `TDI`, `TMS`, and `TRST` pins have weak internal pull-up resistors (typically 25 kΩ). JTAG output pin `TDO` and all JTAG input pins are powered by the 2.5 V/3.0 V/3.3 V V_{CCPD} power supply of I/O bank 1A. All the JTAG pins support only LVTTL I/O standard.

All user I/O pins are tri-stated during JTAG configuration. Table 11–11 explains each JTAG pin's function.

The `TDO` output is powered by the V_{CCPD} power supply of I/O bank 1A. For recommendations on how to connect a JTAG chain with multiple voltages across the devices in the chain, refer to the *IEEE 1149.1 (JTAG) Boundary Scan Testing in Stratix III Devices* chapter of the *Stratix III Device Handbook.*

Table 11–11. Dedicated JTAG Pins

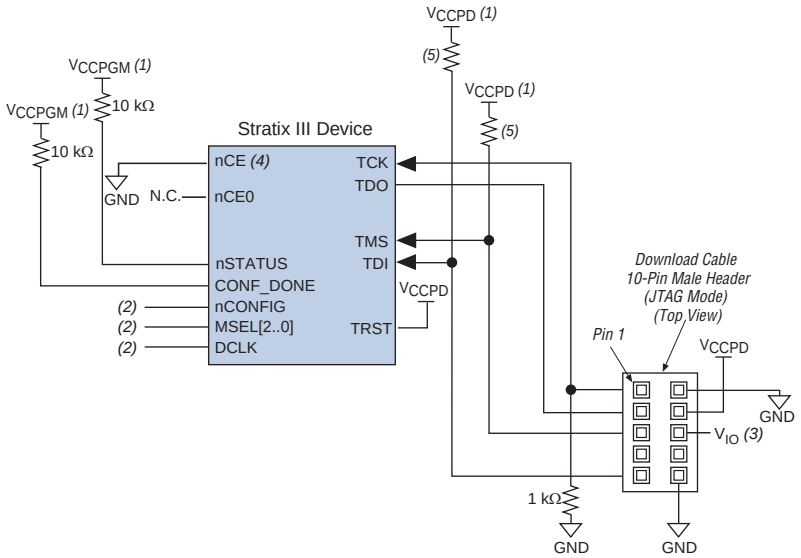
Pin Name	Pin Type	Description
TDI	Test data input	Serial input pin for instructions as well as test and programming data. Data is shifted in the rising edge of TCK. If the JTAG interface is not required on the board, you can disable the JTAG circuitry by connecting this pin to V_{CCPD}.
TDO	Test data output	Serial data output pin for instructions as well as test and programming data. Data is shifted out on the falling edge of TCK. The pin is tri-stated if data is not being shifted out of the device. If the JTAG interface is not required on the board, you can disable the JTAG circuitry by leaving this pin unconnected.
TMS	Test mode select	Input pin that provides the control signal to determine the transitions of the TAP controller state machine. Transitions within the state machine occur on the rising edge of TCK. Therefore, you must set up TMS before the rising edge of TCK. TMS is evaluated on the rising edge of TCK. If the JTAG interface is not required on the board, you can disable the JTAG circuitry by connecting this pin to VCCPD.
TCK	Test clock input	The clock input to the BST circuitry. Some operations occur at the rising edge while others occur at the falling edge. If the JTAG interface is not required on the board, you can disable the JTAG circuitry by connecting this pin to GND.
TRST	Test reset input (optional)	Active-low input to asynchronously reset the boundary-scan circuit. The TRST pin is optional according to IEEE Std. 1149.1. If the JTAG interface is not required on the board, you can disable the JTAG circuitry by connecting this pin to GND.

During JTAG configuration, you can download data to the device on the PCB through the USB-Blaster, MasterBlaster, ByteBlaster II, ByteBlasterMV, or EthernetBlaster download cables. Configuring devices through a cable is similar to programming devices in-system, except you should connect the TRST pin to V_{CCPD}. This ensures that the TAP controller is not reset.

☞ The JRunner™ software driver is developed to configure Altera FPGA devices in JTAG mode through the ByteBlaster II or ByteBlasterMV download cables for embedded configurations. For more information, refer to *AN 414: The JRunner Software Driver*.

Figure 11–19 shows JTAG configuration of a single Stratix III device.

Figure 11–19. JTAG Configuration of a Single Device Using a Download Cable



Notes to Figure 11–19:

(1) You should connect the pull-up resistor to the same supply voltage as the USB-Blaster, MasterBlaster (`VIO` pin), ByteBlaster II, ByteBlasterMV, or EthernetBlaster cables. The voltage supply can be connected to the V_{CCPD} of the device.

(2) You should connect the `nCONFIG` and `MSEL[2..0]` pins to support a non-JTAG configuration scheme. If you only use the JTAG configuration, connect `nCONFIG` to V_{CCPGM}, and `MSEL[2..0]` to ground. Pull `DCLK` either high or low, whichever is convenient on your board.

(3) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. V_{IO} should match the device's V_{CCPD}. Refer to the *MasterBlaster Serial/USB Communications Cable Data Sheet* for this value. In the USB-Blaster, ByteBlaster II, ByteBlasterMV, and EthernetBlaster, this pin is a no connect.

(4) You must connect `nCE` to GND or drive it low for successful JTAG configuration.

(5) Pull-up resistor values can vary from 1 kΩ to 10 kΩ

To configure a single device in a JTAG chain, the programming software places all other devices in bypass mode. In bypass mode, devices pass programming data from the `TDI` pin to the `TDO` pin through a single bypass register without being affected internally. This scheme enables the programming software to program or verify the target device. Configuration data driven into the device appears on the `TDO` pin one clock cycle later.

The Quartus II software verifies successful JTAG configuration upon completion. At the end of configuration, the software checks the state of `CONF_DONE` through the JTAG port. When Quartus II generates a JAM file (**.jam**) for a multi-device chain, it contains instructions so that all the devices in the chain will be initialized at the same time. If `CONF_DONE` is not high, the Quartus II software indicates that configuration has failed. If `CONF_DONE` is high, the software indicates that configuration was successful. After the configuration bitstream is transmitted serially through the JTAG `TDI` port, the `TCK` port is clocked an additional 1,094 cycles to perform device initialization.

Stratix III devices have dedicated JTAG pins that always function as JTAG pins. Not only can you perform JTAG testing on Stratix III devices before and after, but also during configuration. While other device families do not support JTAG testing during configuration, Stratix III devices support the bypass, id code, and sample instructions during configuration without interrupting configuration. All other JTAG instructions may only be issued by first interrupting configuration and reprogramming I/O pins using the CONFIG_IO instruction.

The CONFIG_IO instruction allows I/O buffers to be configured by using the JTAG port and when issued, interrupts configuration. This instruction allows you to perform board-level testing prior to configuring the Stratix III device or waiting for a configuration device to complete configuration. When configuration has been interrupted and JTAG testing is complete, you must reconfigure the part by using JTAG (PULSE_CONFIG instruction) or by pulsing nCONFIG low.

The chip-wide reset (DEV_CLRn) and chip-wide output enable (DEV_OE) pins on Stratix III devices do not affect JTAG boundary-scan or programming operations. Toggling these pins does not affect JTAG operations (other than the usual boundary-scan operation).

When designing a board for JTAG configuration of Stratix III devices, consider the dedicated configuration pins. Table 11–12 lists how these pins should be connected during JTAG configuration.

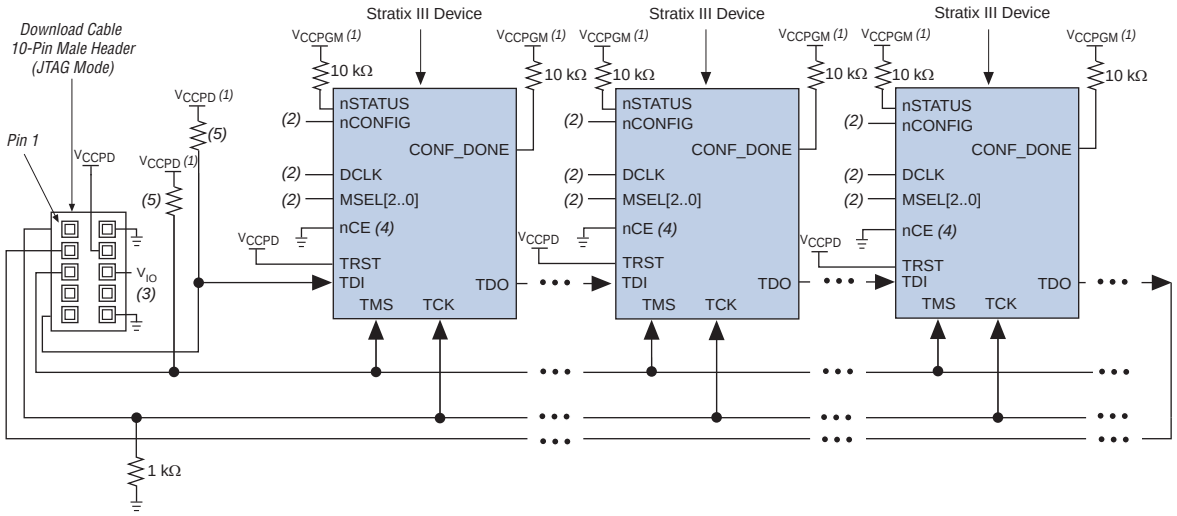
Table 11–12. Dedicated Configuration Pin Connections During JTAG Configuration

Signal	Description
nCE	On all Stratix III devices in the chain, nCE should be driven low by connecting it to ground, pulling it low by using a resistor, or driving it by some control circuitry. For devices that are also in multi-device FPP, AS, or PS configuration chains, the nCE pins should be connected to GND during JTAG configuration or JTAG configured in the same order as the configuration chain.
nCEO	On all Stratix III devices in the chain, you can leave nCEO floating or connected to the nCE of the next device.
MSEL	These pins must not be left floating. These pins support whichever non-JTAG configuration is used in production. If you only use JTAG configuration, tie these pins to ground.
nCONFIG	Driven high by connecting to V_{CCPGM}, pull up by using a resistor, or driven high by some control circuitry.
nSTATUS	Pull to V_{CCPGM} by using a 10-kΩ resistor. When configuring multiple devices in the same JTAG chain, each nSTATUS pin should be pulled up to V_{CCPGM} individually.
CONF_DONE	Pull to V_{CCPGM} by using a 10-kΩ resistor. When configuring multiple devices in the same JTAG chain, each CONF_DONE pin should be pulled up to V_{CCPGM} individually. CONF_DONE going high at the end of JTAG configuration indicates successful configuration.
DCLK	Should not be left floating. Drive low or high, whichever is more convenient on your board.

When programming a JTAG device chain, one JTAG-compatible header is connected to several devices. The number of devices in the JTAG chain is limited only by the drive capability of the download cable. When four or more devices are connected in a JTAG chain, Altera recommends buffering the TCK, TDI, and TMS pins with an on-board buffer.

JTAG-chain device programming is ideal when the system contains multiple devices, or when testing your system using JTAG BST circuitry. Figure 11–20 shows multi-device JTAG configuration.

Figure 11–20. JTAG Configuration of Multiple Devices Using a Download Cable



Notes to Figure 11–20:

(1) Connect the pull-up resistor to the same supply voltage as the USB-Blaster, MasterBlaster (V_{IO} pin), ByteBlaster II, ByteBlasterMV, or EthernetBlaster cables. The voltage supply can be connected to the V_{CCPD} of the device.

(2) Connect the `nCONFIG`, `MSEL[2..0]` pins to support a non-JTAG configuration scheme. If you only use JTAG configuration, connect `nCONFIG` to V_{CCPGM}, and `MSEL[2..0]` to ground. Pull `DCLK` either high or low, whichever is convenient on your board.

(3) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. V_{IO} should match the device's V_{CCPD}. Refer to the *MasterBlaster Serial/USB Communications Cable Data Sheet* for this value. In the USB-Blaster, ByteBlaster II, ByteBlasterMV, and EthernetBlaster, this pin is a no connect.

(4) You must connect `nCE` to GND or drive it low for successful JTAG configuration.

(5) Pull-up resistor values can vary from 1 kΩ to 10 kΩ

You must connect the `nCE` pin to GND or drive it low during JTAG configuration. In multi-device FPP, AS, and PS configuration chains, the first device's `nCE` pin is connected to GND while its `nCEO` pin is connected to `nCE` of the next device in the chain. The last device's `nCE` input comes from the previous device, while its `nCEO` pin is left floating. In addition, the `CONF_DONE` and `nSTATUS` signals are all shared in multi-device FPP, AS, or PS configuration chains so the devices can enter user mode at the same time after configuration is complete. When the `CONF_DONE` and `nSTATUS` signals are shared among all the devices, you must configure every device when JTAG configuration is performed.

If you only use JTAG configuration, Altera recommends that you connect the circuitry as shown in Figure 11–20, where each of the `CONF_DONE` and `nSTATUS` signals are isolated, so that each device can enter user mode individually.

After the first device completes configuration in a multi-device configuration chain, its `nCEO` pin drives low to activate the second device's `nCE` pin, which prompts the second device to begin configuration. Therefore, if these devices are also in a JTAG chain, make sure the `nCE` pins are connected to GND during JTAG configuration or that the devices are JTAG configured in the same order as the configuration chain. As long as the devices are JTAG configured in the same order as the multi-device configuration chain, the `nCEO` of the previous device will drive the `nCE` of the next device low when it has successfully been JTAG configured.

You can place other Altera devices that have JTAG support in the same JTAG chain for device programming and configuration.

JTAG configuration support has been enhanced and allows more than 17 Stratix III devices to be cascaded in a JTAG chain.

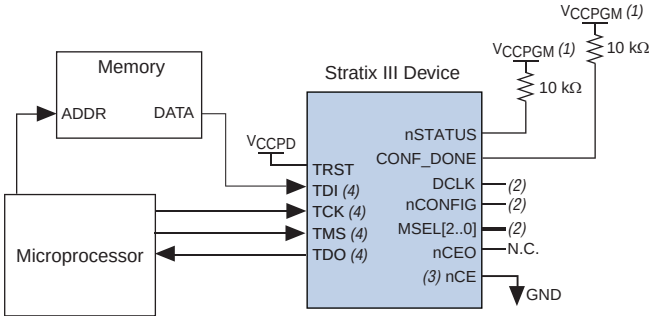
For more information about configuring multiple Altera devices in the same configuration chain, refer to the *Configuring Mixed Altera Device Chains* chapter in the *Configuration Handbook*.

You can configure Stratix III devices using multiple configuration schemes on the same board. Combining JTAG configuration with passive serial (PS) or active serial (AS) configuration on your board is useful in the prototyping environment because it allows multiple methods to configure your FPGA.

For more information about combining JTAG configuration with other configuration schemes, refer to the *Combining Different Configuration Schemes* chapter in the Configuration Handbook.

Figure 11–21 shows JTAG configuration of a Stratix III device with a microprocessor.

Figure 11–21. JTAG Configuration of a Single Device Using a Microprocessor



Notes to Figure 11–21:

(1) Connect the pull-up resistor to a supply that provides an acceptable input signal for all devices in the chain. V_{CCPGM} should be high enough to meet the V_{IH} specification of the I/O on the device.

(2) Connect the `nCONFIG` and `MSEL[2..0]` pins to support a non-JTAG configuration scheme. If you use only the JTAG configuration, connect `nCONFIG` to V_{CCPGM}, and `MSEL[2..0]` to ground. Pull `DCLK` either high or low, whichever is convenient on your board.

(3) You must connect `nCE` to `GND` or drive it low for successful JTAG configuration.

(4) Microprocessor should use the same I/O standard as V_{CCPD} to drive the JTAG pins.

Jam STAPL

Jam STAPL, JEDEC standard JESD-71, is a standard file format for in-system programmability (ISP) purposes. Jam STAPL supports programming or configuration of programmable devices and testing of electronic systems, using the IEEE 1149.1 JTAG interface. Jam STAPL is a freely licensed open standard.

The Jam Player provides an interface for manipulating the IEEE Std. 1149.1 JTAG TAP state machine.

For more information about JTAG and Jam STAPL in embedded environments, refer to *AN 425: Using Command-Line Jam STAPL Solution for Device Programming*. To download the jam player, visit the Altera web site at www.altera.com.

Device Configuration Pins

The following tables describe the connections and functionality of all the configuration-related pins on the Stratix III devices. Table 11–13 summarizes the Stratix III configuration pins and their power supply.

Table 11–13. Stratix III Configuration Pin Summary *(Note 1)* (Part 1 of 2)

Description	Input/Output	Dedicated	Powered By	Configuration Mode
TDI	Input	Yes	V_{CCPD} *(2)*	JTAG
TMS	Input	Yes	V_{CCPD} *(2)*	JTAG
TCK	Input	Yes	V_{CCPD} *(2)*	JTAG
TRST	Input	Yes	V_{CCPD} *(2)*	JTAG
TDO	Output	Yes	V_{CCPD} *(2)*	JTAG
CRC_ERROR	Output	—	Pull-up	Optional, all modes
DATA0	Input	—	V_{CCPGM}/V_{CCIO} *(3)*	All modes except JTAG
DATA[7..1]	Input	—	V_{CCPGM}/V_{CCIO} *(3)*	FPP
INIT_DONE	Output	—	Pull-up	Optional, all modes
CLKUSR	Input	—	V_{CCPGM}/V_{CCIO} *(3)*	Optional
nSTATUS	bi-directional	Yes	Pull-up	All modes
nCE	Input	Yes	V_{CCPGM}	All modes
CONF_DONE	bi-directional	Yes	Pull-up	All modes
nCONFIG	Input	Yes	V_{CCPGM}	All modes
PORSEL	Input	Yes	V_{CCPGM}	All modes
ASDO	Output	Yes	V_{CCPGM}	AS
nCSO	Output	Yes	V_{CCPGM}	AS
DCLK	Input	Yes	V_{CCPGM}	PS, FPP
—	Output	—	V_{CCPGM}	AS
nIO_PULLUP	Input	Yes	V_{CCPGM}	All modes
nCEO	Output	Yes	V_{CCPGM}	All modes

Table 11–13. Stratix III Configuration Pin Summary *(Note 1)* (Part 2 of 2)

Description	Input/Output	Dedicated	Powered By	Configuration Mode
MSEL[2..0]	Input	Yes	V_{CCPGM}	All modes

Notes to Table 11–13:

(1) The total number of pins is 30. The total number of dedicated pins is 19.

(2) The JTAG output pin TDO and all JTAG input pins are powered by the 2.5 V/3.0 V/3.3-V V_{CCPD} power supply of I/O bank 1A.

(3) These dual purpose pins are powered by V_{CCPGM} during configuration, then are powered by V_{CCIO} while in user mode.This applies for all configuration modes.

Table 11–14 describes the dedicated configuration pins, which are required to be connected properly on your board for successful configuration. Some of these pins may not be required for your configuration schemes.

Table 11–14. Dedicated Configuration Pins on the Stratix III Device (Part 1 of 5)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
VCCPGM	N/A	All	Power	Dedicated power pin. Use this pin to power all dedicated configuration inputs, dedicated configuration outputs, dedicated configuration bi-direction pins, and some of the dual functional pins that are used for configuration. You must connect this pin to 1.8-V, 2.5-V, 3.0-V, or 3.3-V. V_{CCPGM} must ramp-up from 0-V to 3.3-V within 100 ms. If V_{CCPGM} is not ramped up within this specified time, your Stratix III device will not configure successfully. If your system does not allow for a VCCPGM ramp-up time of 100 ms or less, you must hold nCONFIG low until all power supplies are stable.
VCCPD	N/A	All	Power	Dedicated power pin. Use this pin to power the I/O pre-drivers, the JTAG input and output pins, and the design security circuitry. You must connect this pin to 2.5-V, 3.0-V, or 3.3-V depending on the I/O standards selected. For 3.3-V I/O standards, VCCPD=3.3-V, for 3.0-V I/O standards, V_{CCPD} = 3.0 V, for 2.5-V or below I/O standards, V_{CCPD} = 2.5 V. V_{CCPD} must ramp-up from 0-V to 2.5-V / 3.0-V/3.3-V within 100 ms. If V_{CCPD} is not ramped up within this specified time, your Stratix III device will not configure successfully. If your system does not allow for a V_{CCPD} ramp-up time of 100 ms or less, you must hold nCONFIG low until all power supplies are stable.
PORSEL	N/A	All	Input	Dedicated input which selects either a POR time of 12 ms or 100 ms. A logic high (1.8 V, 2.5 V, 3.0 V, 3.3 V) selects a POR time of approximately 12 ms and a logic low selects a POR time of approximately 100 ms. The PORSEL input buffer is powered by VCCPGM and has an internal 5-kΩ pull-down resistor that is always active. You should tie the PORSEL pin directly to VCCPGM or GND.

Table 11–14. Dedicated Configuration Pins on the Stratix III Device (Part 2 of 5)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
nIO_PULLUP	N/A	All	Input	Dedicated input that chooses whether the internal pull-up resistor on the user I/O pins and dual-purpose I/O pins (nCSO, nASDO, DATA[7..0], nWS, nRS, RDYnBSY, nCS, CLKUSR, INIT_DONE) are on or off before and during configuration. A logic high (1.8 V, 2.5 V, 3.0 V, 3.3 V) turns off the weak internal pull-up resistors, while a logic low turns them on. The nIO-PULLUP input buffer is powered by VCCPGM and has an internal 5-kΩ pull-down resistor that is always active. You can tie the nIO-PULLUP directly to VCCPGM or use a 1-kΩ pull-up resistor or tie it directly to GND.
MSEL[2..0]	N/A	All	Input	3-bit configuration input that sets the Stratix III device configuration scheme. Refer to Table 11–1 for the appropriate connections. You must hard-wire these pins to VCCPGM or GND. The MSEL[2..0] pins have internal 5-kΩ pull-down resistors that are always active.
nCONFIG	N/A	All	Input	Configuration control input. Pulling this pin low during user-mode will cause the device to lose its configuration data, enter a reset state, and tri-state all I/O pins. Returning this pin to a logic high level will initiate a reconfiguration. Configuration is possible only if this pin is high, except in JTAG programming mode when nCONFIG is ignored.

Table 11–14. Dedicated Configuration Pins on the Stratix III Device (Part 3 of 5)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
nSTATUS	N/A	All	Bi-directional open-drain	The device drives nSTATUS low immediately after power-up and releases it after the POR time. During user mode and regular configuration, this pin is pulled high by an external 10-kΩ resistor. This pin, when driven low by Stratix III, indicates that the device is being initialized and has encountered an error during configuration. Status output. If an error occurs during configuration, nSTATUS is pulled low by the target device. Status input. If an external source drives the nSTATUS pin low during configuration or initialization, the target device enters an error state. Driving nSTATUS low after configuration and initialization does not affect the configured device. If you use a configuration device, driving nSTATUS low will cause the configuration device to attempt to configure the device, but since the device ignores transitions on nSTATUS in user-mode, the device does not reconfigure. To initiate a reconfiguration, nCONFIG must be pulled low. If you have enabled the **Auto-restart configuration after error** option, the nSTATUS pin transitions from high to low and back again to high when a configuration error is detected. This appears as a low pulse at the nSTATUS pin with a minimum pulse width of 10 µs to a maximum pulse width of 500 µs, as defined in the t_{STATUS} specification.
nSTATUS (continued)	—	—	—	If V_{CCPGM} and V_{CCIO} are not fully powered up, the following could occur: ■ V_{CCPGM} and V_{CCIO} are powered high enough for the nSTATUS buffer to function properly, and nSTATUS is driven low. When V_{CCPGM} and V_{CCIO} are ramped up, POR trips and nSTATUS is released after POR expires. ■ V_{CCPGM} and V_{CCIO} are not powered high enough for the nSTATUS buffer to function properly. In this situation, nSTATUS might appear logic high, triggering a configuration attempt that would fail because POR did not yet trip. When V_{CCPD} and V_{CCIO} are powered up, nSTATUS is pulled low because POR did not yet trip. When POR trips after V_{CCPGM} and V_{CCIO} are powered up, nSTATUS is released and pulled high. At that point, reconfiguration is triggered and the device is configured.

Table 11–14. Dedicated Configuration Pins on the Stratix III Device (Part 4 of 5)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
CONF_DONE	N/A	All	Bi-directional open-drain	Status output. The target device drives the CONF_DONE pin low before and during configuration. After all configuration data is received without error and the initialization cycle starts, the target device releases CONF_DONE. Status input. After all data is received and CONF_DONE goes high, the target device initializes and enters user mode. The CONF_DONE pin must have an external 10-kΩ pull-up resistor in order for the device to initialize. Driving CONF_DONE low after configuration and initialization does not affect the configured device. Do not connect bus holds or ADC to CONF_DONE pin.
nCE	N/A	All	Input	Active-low chip enable. The nCE pin activates the device with a low signal to allow configuration. The nCE pin must be held low during configuration, initialization, and user mode. In single device configuration, it should be tied low. In multi-device configuration, nCE of the first device is tied low while its nCEO pin is connected to nCE of the next device in the chain. The nCE pin must also be held low for successful JTAG programming of the device.
nCEO	N/A	All	Output	Output that drives low when device configuration is complete. In single device configuration, this pin is left floating. In multi-device configuration, this pin feeds the next device's nCE pin. The nCEO of the last device in the chain is left floating. The nCEO pin is powered by V_{CCPGM}.
ASDO *(1)*	N/A	AS	Output	Control signal from the Stratix III device to the serial configuration device in AS mode used to read out configuration data. In AS mode, ASDO has an internal pull-up resistor that is always active.
nCSO *(1)*	N/A	AS	Output	Output control signal from the Stratix III device to the serial configuration device in AS mode that enables the configuration device. In AS mode, nCSO has an internal pull-up resistor that is always active.

Table 11–14. Dedicated Configuration Pins on the Stratix III Device (Part 5 of 5)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
DCLK *(1)*	N/A	Synchronous configuration schemes (PS, FPP, AS)	Input (PS, FPP) Output (AS)	DCLK has an internal pull-up resistor (typically 25 kΩ) that is always active. In AS mode, DCLK is an output from the Stratix III device that provides timing for the configuration interface. After AS configuration, this pin is driven to an inactive state. In schemes that use a configuration device, DCLK will be driven low after configuration is done. In schemes that use a control host, DCLK should be driven either high or low, whichever is more convenient. Toggling this pin after configuration does not affect the configured device.
DATA0 *(1)*	N/A in AS mode. I/O in PS or FPP mode	PS, FPP, AS	Input	Data input. In serial configuration modes, bit-wide configuration data is presented to the target device on the DATA0 pin. In AS mode, DATA0 has an internal pull-up resistor that is always active. After PS or FPP configuration, DATA0 is available as a user I/O pin and the state of this pin depends on the **Dual-Purpose Pin** settings.
DATA[7..1]	I/O	Parallel configuration schemes (FPP)	Inputs	Data inputs. Byte-wide configuration data is presented to the target device on DATA[7..0]. In serial configuration schemes, they function as user I/O pins during configuration, which means they are tri-stated. After configuration, DATA[7..1] are available as user I/O pins and the state of these pin depends on the **Dual-Purpose Pin** settings.

Note to Table 11–14:

(1) To tri-state AS configuration pins in AS configuration scheme, turn on **Enable input tri-state on active configuration pins in user mode** option from the **Device and Pin Options** dialog box. This tri-states DCLK, nCSO, Data0, and ASDO pins. Dual-purpose Pins Setting for Data0 is ignored. To set Data0 to a different setting, for example to use Data0 pin as a regular I/O in user mode, turn off **Enable input tri-state on active configuration pins in user mode** option and set your desired setting from the Dual-purpose Pins Setting menu.

Table 11–15 describes the optional configuration pins. If these optional configuration pins are not enabled in the Quartus II software, they are available as general-purpose user I/O pins. Therefore, during configuration, these pins function as user I/O pins and are tri-stated with weak pull-up resistors.

Table 11–15. Optional Configuration Pins

Pin Name	User Mode	Pin Type	Description
CLKUSR	N/A if option is on. I/O if option is off.	Input	Optional user-supplied clock input synchronizes the initialization of one or more devices. Enable this pin by turning on the **Enable user-supplied start-up clock** (**CLKUSR**) option in the Quartus II software.
INIT_DONE	N/A if option is on. I/O if option is off.	Output open-drain	Use the Status pin to indicate when the device has initialized and is in user mode. When nCONFIG is low and during the beginning of configuration, the INIT_DONE pin is tri-stated and pulled high due to an external 10-kΩ pull-up resistor. After the option bit to enable INIT_DONE is programmed into the device (during the first frame of configuration data), the INIT_DONE pin will go low. When initialization is complete, the INIT_DONE pin is released and pulled high and the device enters user mode. Thus, the monitoring circuitry must be able to detect a low-to-high transition. This pin is enabled by turning on the **Enable INIT_DONE output** option in the Quartus II software.
DEV_OE	N/A if option is on. I/O if option is off.	Input	Optional pin that allows you to override all tri-states on the device. When this pin is driven low, all I/O pins are tri-stated, when this pin is driven high, all I/O pins behave as programmed. Enable this pin by turning on the **Enable device-wide output enable** (**DEV_OE**) option in the Quartus II software.
DEV_CLRn	N/A if option is on. I/O if option is off.	Input	Optional pin that allows you to override all clears on all device registers. When this pin is driven low, all registers are cleared. When this pin is driven high, all registers behave as programmed. This pin is enabled by turning on the **Enable device-wide reset** (**DEV_CLRn**) option in the Quartus II software.

Table 11–16 describes the dedicated JTAG pins. JTAG pins must be kept stable before and during configuration to prevent accidental loading of JTAG instructions. The TDI, TMS, and TRST have weak internal pull-up resistors while TCK has a weak internal pull-down resistor (typically 25 kΩ). If you plan to use the SignalTap® embedded logic array analyzer, you must connect the JTAG pins of the Stratix III device to a JTAG header on your board.

Table 11–16. Dedicated JTAG Pins

Pin Name	User Mode	Pin Type	Description
TDI	N/A	Input	Serial input pin for instructions as well as test and programming data. Data is shifted in on the rising edge of TCK. The TDI pin is powered by the 2.5-V/3.0-V/3.3-V V_{CCPD} supply. If the JTAG interface is not required on the board, you can disable the JTAG circuitry by connecting this pin to V_{CCPD}.
TDO	N/A	Output	Serial data output pin for instructions as well as test and programming data. Data is shifted out on the falling edge of TCK. The pin is tri-stated if data is not being shifted out of the device. The TDO pin is powered by V_{CCPD} For recommendations on connecting a JTAG chain with multiple voltages across the devices in the chain, refer to the chapter *IEEE 1149.1 (JTAG) Boundary Scan Testing in Stratix III Devices* chapter in volume 1 of the *Stratix III Device Handbook*. If the JTAG interface is not required on the board, you can disable the JTAG circuitry by leaving this pin unconnected.
TMS	N/A	Input	Input pin that provides the control signal to determine the transitions of the TAP controller state machine. Transitions within the state machine occur on the rising edge of TCK. Therefore, TMS must be set up before the rising edge of TCK. TMS is evaluated on the rising edge of TCK. The TMS pin is powered by the 2.5-V/3.0-V/3.3-V V_{CCPD}. If the JTAG interface is not required on the board, you can disable the JTAG circuitry by connecting this pin to V_{CCPD}.
TCK	N/A	Input	The clock input to the BST circuitry. Some operations occur at the rising edge, while others occur at the falling edge. The TCK pin is powered by the 2.5-V/3.0-V/3.3-V V_{CCPD} supply. It is expected that the clock input waveform have a nominal 50% duty cycle. If the JTAG interface is not required on the board, you can disable the JTAG circuitry by connecting TCK to GND.
TRST	N/A	Input	Active-low input to asynchronously reset the boundary-scan circuit. The TRST pin is optional according to IEEE Std. 1149.1. The TRST pin is powered by the 2.5-V/3.0-V/3.3-V V_{CCPD} supply. You should hold TMS at 1 or you should keep TCK static while TRST is changed from 0 to 1. If the JTAG interface is not required on the board, you can disable the JTAG circuitry by connecting the TRST pin to GND.

Chapter Revision History

Table 11–17 lists the revision history for this chapter.

Table 11–17. Chapter Revision History (Part 1 of 2)

Date	Version	Changes Made
March 2011	2.0	■ Updated the "FPP Configuration Using a MAX II Device as an External Host", "Fast Active Serial Configuration (Serial Configuration Devices)", and "PS Configuration Using a MAX II Device as an External Host" ■ Updated Table 11–14.
July 2010	1.9	■ Updated Table 11–14. ■ Updated "FPP Configuration Using a MAX II Device as an External Host" on page 11–8.
March 2010	1.8	Updated for the Quartus II software version 9.1 SP2 release: ■ Added Figure 11–11. ■ Updated Figure 11–6, Figure 11–7, Figure 11–16, Figure 11–19, and Figure 11–20. ■ Updated "Estimating Active Serial Configuration Time" section. ■ Added Table 11–8. ■ Updated Table 11–8, Table 11–13, and Table 11–14. ■ Removed "Conclusion" section.
May 2009	1.7	■ Updated Table 11–1, Table 11–2, Table 11–5, Table 11–6, Table 11–9, and Table 11–13. ■ Updated Figure 11–6, Figure 11–16, Figure 11–17, Figure 11–18, Figure 11–19, and Figure 11–20. ■ Updated "PS Configuration Using a Microprocessor", "PS Configuration Using a Download Cable", and "JTAG Configuration" sections. ■ Removed Figure 11-12 Fast AS Configuration Timing. ■ Removed Table 11-8 Fast AS Timing Parameters for Stratix III devices.
February 2009	1.6	■ Updated Figure 11–6, Figure 11–7, Figure 11–12, and Figure 11–16. ■ Removed "Referenced Documents" section.
October 2008	1.5	■ Updated "FPP Configuration Using a MAX II Device as an External Host", "Fast Active Serial Configuration (Serial Configuration Devices)", "JTAG Configuration", "Power-On Reset Circuit", "PS Configuration Using a MAX II Device as an External Host", and "PS Configuration Using a Download Cable" sections. ■ Updated Table 11–13 and Table 11–14. ■ Updated New Document Format. ■ Updated (Note 3) to Figure 11–17. ■ Updated (Note 3) to Figure 11–18. ■ Updated (Note 3) to Figure 11–19. ■ Updated (Note 3) to Figure 11–20.

Table 11–17. Chapter Revision History (Part 2 of 2)

Date	Version	Changes Made
May 2008	1.4	■ Updated "Power-On Reset Circuit" section. ■ Updated "Fast Active Serial Configuration (Serial Configuration Devices)" section. ■ Updated Table 11–4, Table 11–11, Table 11–12, Table 11–13, Table 11–14, and Table 11–16. ■ Updated Figure 11–3, Figure 11–4, Figure 11–5, Figure 11–6, Figure 11–7, Figure 11–8, Figure 11–9, Figure 11–10, Figure 11–11, Figure 11–13, Figure 11–14, Figure 11–15, Figure 11–17, Figure 11–18, Figure 11–19, Figure 11–20, and Figure 11–21.
November 2007	1.3	■ Updated Table 11–2. ■ Updated Figure 11–8, Figure 11–14, Figure 11–15, Figure 11–17, and Figure 11–18.
October 2007	1.2	■ Updated Table 11–13, Table 11–14. ■ Updated Figure 11–6, Figure 11–7, Figure 11–9, Figure 11–10, Figure 11–11, and Figure 11–13. ■ Removed text regarding enhanced configuration device support. Removed Figure 11–19. ■ Added live links for references. ■ Added section "Referenced Documents"
May 2007	1.1	Removed Bank Column from Table 11–13.
November 2006	1.0	Initial Release

12. Remote System Upgrades with Stratix III Devices

SIII51012-1.5

This chapter describes the functionality and implementation of the dedicated remote system upgrade circuitry. It also defines several concepts related to remote system upgrade, including factory configuration, application configuration, remote update mode, and user watchdog timer. Additionally, this chapter provides design guidelines for implementing remote system upgrades with the supported configuration schemes.

System designers sometimes face challenges such as shortened design cycles, evolving standards, and system deployments in remote locations. Stratix® III devices help overcome these challenges with their inherent re-programmability and dedicated circuitry to perform remote system upgrades. Remote system upgrades help deliver feature enhancements and bug fixes without costly recalls, reduce time-to-market, and extend product life.

Stratix III devices feature dedicated remote system upgrade circuitry. Soft logic (either the Nios® II embedded processor or user logic) implemented in a Stratix III device can download a new configuration image from a remote location, store it in configuration memory, and direct the dedicated remote system upgrade circuitry to initiate a reconfiguration cycle. The dedicated circuitry performs error detection during and after the configuration process, recovers from any error condition by reverting back to a safe configuration image, and provides error status information. This dedicated remote system upgrade circuitry is unique to the Stratix series and helps to avoid system downtime.

Remote system upgrade is supported in fast active serial (FAS) Stratix III configuration schemes. You can also implement remote system upgrade in conjunction with advanced Stratix III features such as real-time decompression of configuration data and design security using the advanced encryption standard (AES) for secure and efficient field upgrades.

Functional Description

The dedicated remote system upgrade circuitry in Stratix III devices manage remote configuration and provides error detection, recovery, and status information. User logic or a Nios II processor implemented in the Stratix III device logic array provides access to the remote configuration data source and an interface to the system's configuration memory.

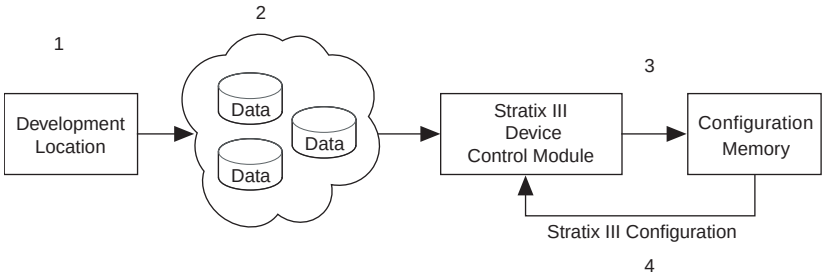
Stratix III devices have remote system upgrade processes that involves the following steps:

1. A Nios II processor (or user logic) implemented in the Stratix III device logic array receives new configuration data from a remote location. The connection to the remote source uses a communication protocol such as the transmission control protocol/Internet protocol (TCP/IP), peripheral component interconnect (PCI), user datagram protocol (UDP), universal asynchronous receiver/transmitter (UART), or a proprietary interface.
2. The Nios II processor (or user logic) stores this new configuration data in non-volatile configuration memory.

3. The Nios II processor (or user logic) initiates a reconfiguration cycle with the new or updated configuration data.
4. The dedicated remote system upgrade circuitry detects and recovers from any error(s) that might occur during or after the reconfiguration cycle, and provides error status information to the user design.

Figure 12–1 shows the steps required for performing remote configuration updates. (The numbers in the figure below coincide with the steps above.)

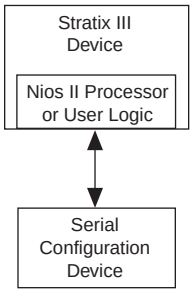
Figure 12–1. Functional Diagram of Stratix III Remote System Upgrade



Stratix III devices only support remote system upgrade in the single device Fast AS configuration scheme.

Figure 12–2 shows the block diagrams for implementing a remote system upgrade with the Stratix III Fast AS configuration scheme.

Figure 12–2. Remote System Upgrade Block Diagram for Stratix III Fast AS Configuration Scheme



You must set the mode select pins (`MSEL[2..0]`) to Fast AS mode to use the remote system upgrade in your system. Table 12–1 lists the `MSEL` pin settings for Stratix III devices in standard configuration mode and remote system upgrade mode. The following sections describe the remote update of remote system upgrade mode.

For more information about standard configuration schemes supported in Stratix III devices, refer to the *Configuring Stratix III Devices* chapter in volume 1 of the *Stratix III Device Handbook*.

Table 12–1. Stratix III Remote System Upgrade Modes

Configuration Scheme	MSEL[2..0]	Remote System Upgrade Mode
Fast AS (40 MHz) *(1)*	011	Standard
	011	Remote update

Note to Table 12–1:

(1) The EPCS16, EPCS64, and EPCS128 serial configuration devices support a DCLK up to 40 MHz. For more information, refer to the *Serial Configuration Devices (EPCS1, EPCS4, EPCS16, EPCS64, and EPCS128) Data Sheet* chapter in volume 1 of the *Configuration Handbook*.

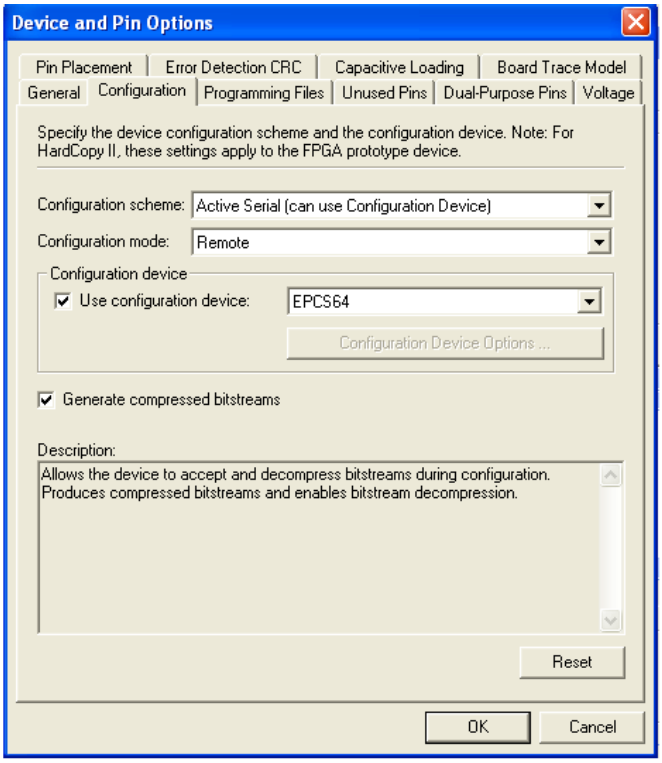
When using the Fast AS mode, you must select the **Remote Update** mode in the Quartus® II software and insert the ALTREMOTE_UPDATE megafunction to access the circuitry. Refer to "ALTREMOTE_UPDATE Megafunction" on page 12–13 for more information.

Enabling Remote Update

You can enable remote update for Stratix III devices in the Quartus II software before design compilation (in the Compiler Settings menu). To enable remote update in the project's compiler settings, perform the following steps in the Quartus II software:

1. On the Assignment menu, click **Device**. The **Settings** dialog box appears.
2. Click **Device and Pin Options**. The **Device and Pin Options** dialog box appears.
3. Click the **Configuration** tab.
4. From the Configuration scheme list, select **Active Serial** (can use **Configuration Device**) (Figure 12–3).
5. From the Configuration Mode list, select **Remote**. (Figure 12–3).
6. Click **OK**.
7. In the **Setting** dialog box, click **OK**.

Figure 12–3. Enabling Remote Update for Stratix III Devices in Compiler Settings



Configuration Image Types

When using a remote system upgrade, Stratix III device configuration bitstreams are classified as factory configuration images or application configuration images. An image, also referred to as a configuration, is a design loaded into the Stratix III device that performs certain user-defined functions.

Each Stratix III device in your system requires one factory configuration image or the addition of one or more application configuration images. The factory configuration image is a user-defined fall-back, or safe configuration, and is responsible for administering remote updates in conjunction with the dedicated circuitry. Application configuration images implement user-defined functionality in the target Stratix III device. You may include the default application configuration image functionality in the factory configuration image.

A remote system upgrade involves storing a new application configuration image or updating an existing one through the remote communication interface. After an application configuration image is stored or updated remotely, the user design in the Stratix III device initiates a reconfiguration cycle with the new image. Any errors during or after this cycle are detected by the dedicated remote system upgrade circuitry and cause the device to automatically revert to the factory configuration image. The factory configuration image then performs error processing and recovery. The factory configuration is written to the serial configuration device only once by the system manufacturer and should not be remotely updated. On the other hand, application configurations may be remotely updated in the system. Both images can initiate system reconfiguration.

Remote System Upgrade Mode

Remote system upgrade has one mode of operation: remote update mode. The remote update mode allows you to determine the functionality of your system upon power-up and offers different features.

In remote update mode, Stratix III devices load the factory configuration image upon power up. The user-defined factory configuration determines which application configuration is to be loaded and triggers a reconfiguration cycle. The factory configuration may also contain application logics.

When used with serial configuration devices, the remote update mode allows an application configuration to start at any flash sector boundary. This translates to a maximum of 128 pages in the EPCS64 device and 32 pages in the EPCS16 device, where the minimum size of each page is 512 KBits. Additionally, the remote update mode features a user watchdog timer that determines the validity of an application configuration.

Remote Update Mode

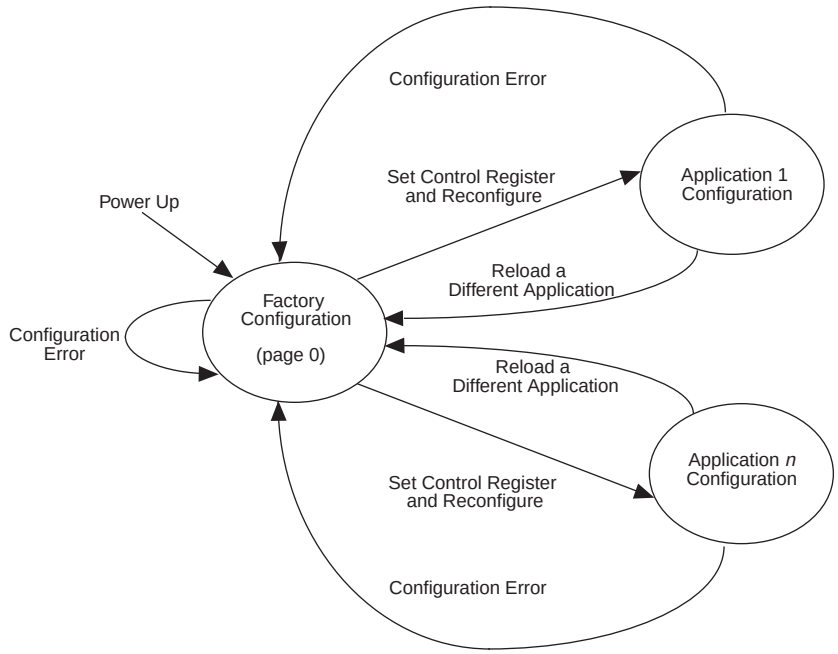
When a Stratix III device is first powered-up in remote update mode, it loads the factory configuration located at page zero (page registers `PGM[23:0] = 24'b0`). You should always store the factory configuration image for your system at page address zero. This corresponds to the start address location 0×000000 in the serial configuration device.

The factory configuration image is user-designed and contains soft logic to do the following:

- Process any errors based on status information from the dedicated remote system upgrade circuitry
- Communicate with the remote host and receive new application configurations and store this new configuration data in the local non-volatile memory device
- Determine which application configuration is to be loaded into the Stratix III device
- Enable or disable the user watchdog timer and load its time-out value (optional)
- Instruct the dedicated remote system upgrade circuitry to initiate a reconfiguration cycle

Figure 12–4 shows the transitions between the factory and application configurations in remote update mode.

Figure 12–4. Transitions Between Configurations in Remote Update Mode



After power up or a configuration error, the factory configuration logic is loaded automatically. The factory configuration also needs to specify whether to enable the user watchdog timer for the application configuration and if enabled, to include the timer setting information as well.

The user watchdog timer ensures that the application configuration is valid and functional. The timer must be continually reset within a specific amount of time during user mode operation of an application configuration. Only valid application configurations contain the logic to reset the timer in user mode. This timer reset logic should be part of a user-designed hardware and/or software health monitoring signal that indicates error-free system operation. If the timer is not reset in a specific amount of time, for example, the user application configuration detects a functional problem or if the system hangs, the dedicated circuitry updates the remote system upgrade status register, triggering the loading of the factory configuration.

☞ The user watchdog timer is automatically disabled for factory configurations. For more information about the user watchdog timer, refer to "User Watchdog Timer" on page 12–11.

If there is an error while loading the application configuration, the cause of the reconfiguration is written by the dedicated circuitry to the remote system upgrade status register. Actions that cause the remote system upgrade status register to be written:

- nSTATUS driven low externally
- Internal CRC error
- User watchdog timer time out

- A configuration reset (logic array `nCONFIG` signal or external `nCONFIG` pin assertion to low)

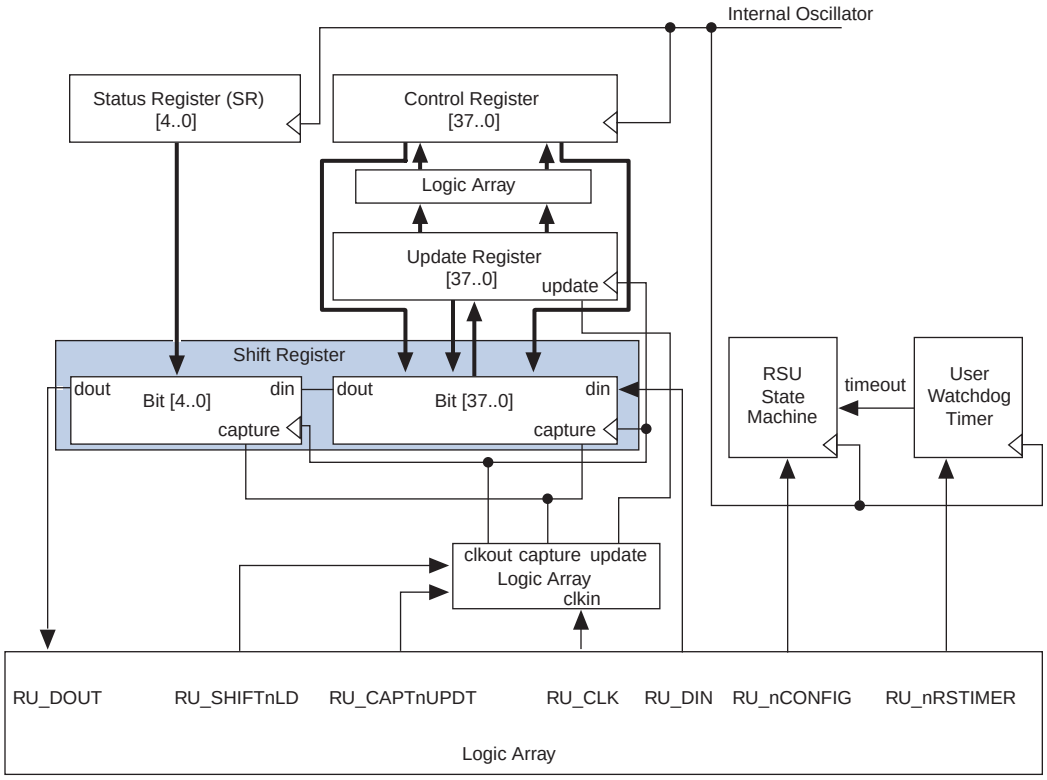
Stratix III devices automatically load the factory configuration located at page address zero. This user-designed factory configuration reads the remote system upgrade status register to determine the reason for the reconfiguration. The factory configuration then takes appropriate error recovery steps and writes to the remote system upgrade control register to determine the next application configuration to be loaded.

When Stratix III devices successfully load the application configuration, they enter into user mode. In user mode, the soft logic (Nios II processor or state machine and the remote communication interface) assists the Stratix III device in determining when a remote system update is arriving. When a remote system update arrives, the soft logic receives the incoming data, writes it to the configuration memory device, and triggers the device to load the factory configuration. The factory configuration reads the remote system upgrade status register and control register, determines the valid application configuration to load, writes the remote system upgrade control register accordingly, and initiates system reconfiguration.

Dedicated Remote System Upgrade Circuitry

This section explains the implementation of the Stratix III remote system upgrade dedicated circuitry. The remote system upgrade circuitry is implemented in hard logic. This dedicated circuitry interfaces to the user-defined factory and application configurations implemented in the Stratix III device logic array to provide the complete remote configuration solution. The remote system upgrade circuitry contains the remote system upgrade registers, a watchdog timer, and a state machine that controls those components. Figure 12–5 shows the remote system upgrade block's data path.

Figure 12–5. Remote System Upgrade Circuit Data Path *(Note 1)*



Note to Figure 12–4:

(1) `RU_DOUT`, `RU_SHIFTnLD`, `RU_CAPTnUPDT`, `RU_CLK`, `RU_DIN`, `RU_nCONFIG`, and `RU_nRSTIMER` signals are internally controlled by the ALTREMOTE_UPDATE megafunction. For more information about the ALTREMOTE_UPDATE megafunction, refer to the *ALTREMOTE_MUPDATE Megafunction User Guide*.

Remote System Upgrade Registers

The remote system upgrade block contains a series of registers that store the page addresses, watchdog timer settings, and status information. These registers are detailed in Table 12–2.

Table 12–2. Remote System Upgrade Registers (Part 1 of 2)

Register	Description
Shift register	Accessible by the logic array and allows the update, status, and control registers to be written and sampled by user logic. Write access is enabled in remote update mode for factory configurations to allow writes to the update register.
Control register	Contains the current page address, the user watchdog timer settings, and one bit specifying whether the current configuration is a factory configuration or an application configuration. During a read operation of an application configuration, this register is read into the shift register. When a reconfiguration cycle is initiated, the contents of the update register are written into the control register.

Table 12–2. Remote System Upgrade Registers (Part 2 of 2)

Register	Description
Update register	Contains data similar to that in the control register. However, it can only be updated by the factory configuration by shifting data into the shift register and issuing an update operation. When a reconfiguration cycle is triggered by the factory configuration, the control register is updated with the contents of the update register. During a capture in a factory configuration, this register is read into the shift register.
Status register	Written to by the remote system upgrade circuitry on every reconfiguration to record the cause of the reconfiguration. This information is used by the factory configuration to determine the appropriate action following a reconfiguration. During a capture cycle, this register is read into the shift register.

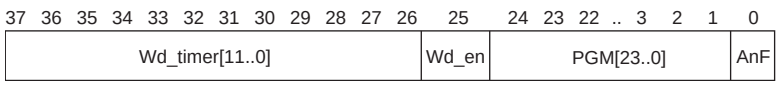
The remote system upgrade control and status registers are clocked by the 10-MHz internal oscillator (the same oscillator that controls the user watchdog timer). However, the remote system upgrade shift and update registers are clocked by the user clock input (`RU_CLK`).

Remote System Upgrade Control Register

The remote system upgrade control register stores the application configuration page address and user watchdog timer settings. The control register functionality depends on the remote system upgrade mode selection. In remote update mode, the control register page address bits are set to all zeros (`24'b0` = `0×000000`) at power up in order to load the factory configuration. A factory configuration in remote update mode has write access to this register.

The control register bit positions are shown in Figure 12–6 and defined in Table 12–3. In the figure, the numbers show the bit position of a setting within a register. For example, bit number 8 is the enable bit for the watchdog timer.

Figure 12–6. Remote System Upgrade Control Register



The application-not-factory (`AnF`) bit indicates whether the current configuration loaded in the Stratix III device is the factory configuration or an application configuration. This bit is set low by the remote system upgrade circuitry when an error condition causes a fall-back to the factory configuration. When the `AnF` bit is high, the control register access is limited to read operations. When the `AnF` bit is low, the register allows write operations and disables the watchdog timer.

In remote update mode, factory configuration design sets this bit high (1'b1) when updating the contents of the update register with the application page address and watchdog timer settings.

Table 12–3. Remote System Upgrade Control Register Contents (Part 1 of 2)

Control Register Bit	Remote System Upgrade Mode	Value *(2)*	Definition
`AnF` *(1)*	Remote update	1'b0	Application not factory
`PGM[23..0]`	Remote update	24'b0×000000	AS configuration start address (`StAdd[23..0]`)

Table 12–3. Remote System Upgrade Control Register Contents (Part 2 of 2)

Control Register Bit	Remote System Upgrade Mode	Value *(2)*	Definition
`Wd_en`	Remote update	1'b0	User watchdog timer enable bit
`Wd_timer[11..0]`	Remote update	12'b000000000000	User watchdog time-out value (most significant 12 bits of 29-bit count value: `{Wd_timer[11..0], 17'b0}`)

Notes to Table 12–3:

(1) In remote update mode, the remote configuration block does not update the `AnF` bit automatically (you can update it manually).

(2) This is the default value of the control register bit.

Remote System Upgrade Status Register

The remote system upgrade status register specifies the reconfiguration trigger condition. The various trigger and error conditions include in the following:

- Cyclic redundancy check (CRC) error during application configuration
- `nSTATUS` assertion by an external device due to an error
- Stratix III device logic array triggers a reconfiguration cycle, possibly after downloading a new application configuration image
- External configuration reset (`nCONFIG`) assertion
- User watchdog timer time out

Figure 12–7 and Table 12–4 specify the contents of the status register. The numbers in the figure show the bit positions within a 5-bit register.

Figure 12–7. Remote System Upgrade Status Register

4	3	2	1	0
Wd	nCONFIG	Core_nCONFIG	nSTATUS	CRC

Table 12–4. Remote System Upgrade Status Register Contents

Status Register Bit	Definition	POR Reset Value
`CRC` (from configuration)	`CRC` error caused reconfiguration	1 bit '0'
`nSTATUS`	`nSTATUS` caused reconfiguration	1 bit '0'
`CORE_nCONFIG` *(1)*	Device logic array caused reconfiguration	1 bit '0'
`nCONFIG`	`nCONFIG` caused reconfiguration	1 bit '0'
`Wd`	Watchdog timer caused reconfiguration	1 bit '0'

Note to Table 12–4:

(1) Logic array reconfiguration forces the system to load the application configuration data into the Stratix III device. This occurs after the factory configuration specifies the appropriate application configuration page address by updating the update register.

Remote System Upgrade State Machine

The remote system upgrade control and update registers have identical bit definitions, but serve different roles (refer to Table 12–2). While both registers can only be updated when the device is loaded with a factory configuration image, the update register writes are controlled by the user logic; the control register writes are controlled by the remote system upgrade state machine.

In factory configurations, the user logic sends the `AnF` bit (set high), the page address, and the watchdog timer settings for the next application configuration bit to the update register. When the logic array configuration reset (`RU_nCONFIG`) goes high, the remote system upgrade state machine updates the control register with the contents of the update register and initiates system reconfiguration from the new application page.

To ensure the successful reconfiguration between the pages, assert `RU_nCONFIG` signal for a minimum of 250 ns. This is equivalent to strobing the `reconfig` input of the ALTREMOTE_UPDATE megafunction high for a minimum of 250 ns.

In the event of an error or reconfiguration trigger condition, the remote system upgrade state machine directs the system to load a factory or application configuration (page zero or page one, based on the mode and error condition) by setting the control register accordingly. Table 12–5 lists the contents of the control register after such an event occurs for all possible error or trigger conditions.

The remote system upgrade status register is updated by the dedicated error monitoring circuitry after an error condition but before the factory configuration is loaded.

Table 12–5. Control Register Contents After an Error or Reconfiguration Trigger Condition

Reconfiguration Error/Trigger	Control Register Setting Remote Update
`nCONFIG` reset	All bits are 0
`nSTATUS` error	All bits are 0
`CORE` triggered reconfiguration	Update register
`CRC` error	All bits are 0
`Wd` time out	All bits are 0

Capture operations during factory configuration access the contents of the update register. This feature is used by the user logic to verify that the page address and watchdog timer settings were written correctly. Read operations in application configurations access the contents of the control register. This information is used by the user logic in the application configuration.

User Watchdog Timer

The user watchdog timer prevents a faulty application configuration from stalling the device indefinitely. The system uses the timer to detect functional errors after an application configuration is successfully loaded into the Stratix III device.

The user watchdog timer is a counter that counts down from the initial value loaded into the remote system upgrade control register by the factory configuration. The counter is 29-bits wide and has a maximum count value of 2^{29}. When specifying the user watchdog timer value, specify only the most significant 12 bits. The granularity of the timer setting is 2^{15} cycles. The cycle time is based on the frequency of the 10-MHz internal oscillator. Table 12–6 specifies the operating range of the 10-MHz internal oscillator.

Table 12–6. 10-MHz Internal Oscillator Specifications

Minimum	Typical	Maximum	Unit
5	6.5	10	MHz

The user watchdog timer begins counting once the application configuration enters device user mode. This timer must be periodically reloaded or reset by the application configuration before the timer expires by asserting `RU_nRSTIMER`. If the application configuration does not reload the user watchdog timer before the count expires, a time-out signal is generated by the remote system upgrade dedicated circuitry. The time-out signal tells the remote system upgrade circuitry to set the user watchdog timer status bit (`Wd`) in the remote system upgrade status register and reconfigures the device by loading the factory configuration.

To allow remote system upgrade dedicated circuitry to reset the watchdog timer, you must assert the `RU_nRSTIMER` signal active for a minimum of 250 ns. This is equivalent to strobing the `reset_timer` input of the ALTREMOTE_UPDATE megafunction high for a minimum of 250 ns.

The user watchdog timer is disabled during the configuration cycle of the device. Errors during configuration are detected by the CRC engine. Also, the timer is disabled for factory configurations. Functional errors should not exist in the factory configuration since it is stored and validated during production and is never updated remotely.

The user watchdog timer is disabled in factory configurations and during the configuration cycle of the application configuration. It is enabled after the application configuration enters user mode. If you do not wish to use the user watchdog timer feature during application configuration user mode operation, turn this feature off by setting `Wd_en` bit to `1'b0` in the update register during factory configuration user mode operation.

Quartus II Software Support

The Quartus II software provides the flexibility to include the remote system upgrade interface between the Stratix III device logic array and the dedicated circuitry, generate configuration files for productions, and remote programming of the system configuration memory.

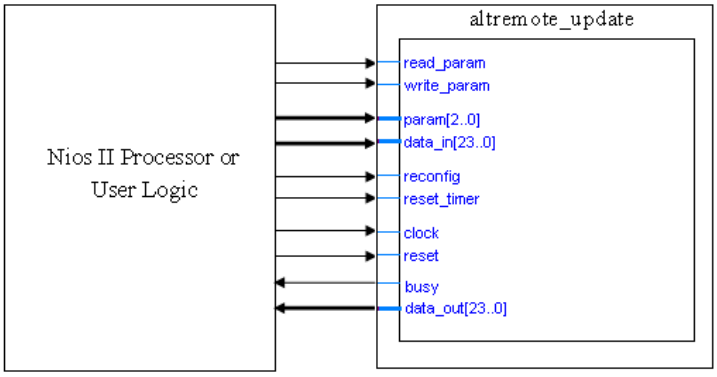
The implementation of the **ALTREMOTE_UPDATE megafunction** option in the Quartus II software is for the interface between the remote system upgrade circuitry and the device logic array interface. Using the megafunction block instead of creating your own logic saves design time and offers more efficient logic synthesis and device implementation.

ALTREMOTE_UPDATE Megafunction

The ALTREMOTE_UPDATE megafunction provides a memory-like interface to the remote system upgrade circuitry and handles the shift register read/write protocol in Stratix III device logic. This implementation is suitable for designs that implement the factory configuration functions using a Nios II processor or user logic in the device.

Figure 12–8 shows the interface signals between the ALTREMOTE_UPDATE megafunction and Nios II processor / user logic.

Figure 12–8. Interface Signals Between the ALTREMOTE_UPDATE Megafunction and the Nios II Processor



For more information about the ALTREMOTE_UPDATE Megafunction and the description of ports listed in Figure 12–8, refer to the *ALTREMOTE_UPDATE Megafunction User Guide*.

Chapter Revision History

Table 12–7 lists the revision history for this chapter.

Table 12–7. Chapter Revision History

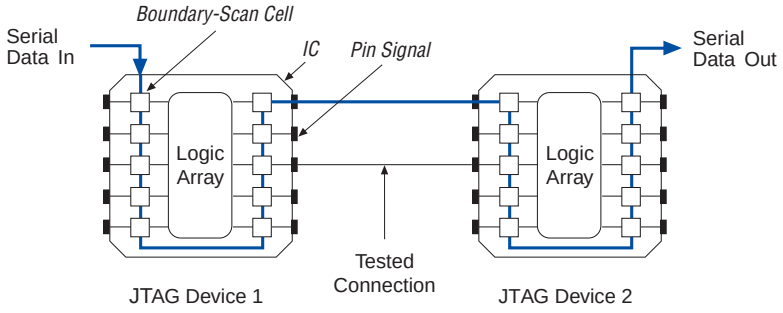
Date	Version	Changes Made
March 2010	1.5	Updated for the Quartus II version 9.1 SP2 release: ■ Updated "Remote System Upgrade State Machine" and "User Watchdog Timer" sections. ■ Updated Table 12–6. ■ Updated Figure 12–1. ■ Removed "Conclusion" section. ■ Minor text edits.
February 2009	1.4	Removed "Referenced Documents" section.
October 2008	1.3	■ Updated "Introduction" section. ■ Updated New Document Format.
October 2007	1.2	■ Added new section "Referenced Documents". ■ Added live links for references.
May 2007	1.1	■ Minor text edits to page 4 and 5. ■ Changes to Figure 12–2. Added Figure 12–3. Added a note to Figure 12–5. Added Figure 12–8. ■ Added new section, "Enabling Remote Update" on page 12–4. ■ Removed references to "Remote System Upgrade atom" and section of same title. Removed "Interface Signals Between Remote System Upgrade Circuitry and Stratix III Device Logic Array" section. Removed Table titled "Interface Signals between Remote System Upgrade Circuitry and Stratix III Device Logic Array." Removed footnote, table titled "Input Ports of the altremote_update Megafunction," table titled "Output Ports of the altremote_update Megafunction," and table titled "Parameter Settings for the altremote_update Megafunction" in section "altremote_update Megafunction" on page 12–15. Removed "System Design Guidelines Using Remote System Upgrade With Serial Configuration Devices" section.
November 2006	1.0	Initial Release.

13. IEEE 1149.1 (JTAG) Boundary-Scan Testing in Stratix III Devices

SIII51013-1.9

This chapter discusses how to use the IEEE Std. 1149.1 boundary-scan test (BST) circuitry in Stratix® III devices. The BST architecture offers the capability to test efficiently components on PCBs with tight lead spacing. BST architecture tests pin connections without using physical test probes and captures functional data while a device is operating normally. Boundary-scan cells in a device can force signals onto pins or capture data from pin or logic array signals. Forced test data is serially shifted into the boundary-scan cells. Captured data is serially shifted out and externally compared to expected results. Figure 13–1 shows the concept of BST.

Figure 13–1. IEEE Std. 1149.1 Boundary-Scan Testing



In addition to BST, you can use the IEEE Std. 1149.1 controller for Stratix III device in-circuit reconfiguration (ICR). However, this chapter only discusses the BST feature of the IEEE Std. 1149.1 circuitry.

For information about configuring Stratix III devices by using the IEEE Std. 1149.1 circuitry, refer to the *Configuring Stratix III Devices*, *Hot Socketing and Power-On Reset in Stratix III Devices*, and the *Remote System Upgrades with Stratix III Devices* chapters.

IEEE Std. 1149.1 BST Architecture

A Stratix III device operating in IEEE Std. 1149.1 BST mode uses four required pins, `TDI`, `TDO`, `TMS`, and `TCK`, and one optional pin, `TRST`. The `TCK` pin has an internal weak pull-down resistor; the `TDI`, `TMS`, and `TRST` pins have weak internal pull-ups. The `TDO` output pin and all of the JTAG input pins are powered by the 2.5-V/3.0-V/3.3-V V_{CCPD} supply of I/O Bank 1A. All user I/O pins are tri-stated during JTAG configuration.

For recommendations about how to connect a JTAG chain with multiple voltages across the devices in the chain, refer to "I/O Voltage Support in JTAG Chain" on page 13–17.

Table 13–1 summarizes the functions of each of these pins.

Table 13–1. IEEE Std. 1149.1 Pin Descriptions

Pin	Description	Function
TDI	Test data input	Serial input pin for instructions as well as test and programming data. Signal applied to TDI is expected to change state at the falling edge of TCK. Data is shifted in on the rising edge of TCK.
TDO	Test data output	Serial data output pin for instructions as well as test and programming data. Data is shifted out on the falling edge of TCK. The pin is tri-stated if data is not being shifted out of the device.
TMS	Test mode select	Input pin that provides the control signal to determine the transitions of the test access port (TAP) controller state machine. Transitions within the state machine occur at the rising edge of TCK. Therefore, you must set up TMS before the rising edge of TCK. TMS is evaluated on the rising edge of TCK. During non-JTAG operation, Altera recommends you drive TMS high.
TCK	Test clock input	The clock input to the BST circuitry. Some operations occur at the rising edge, while others occur at the falling edge.
TRST *(1)*	Test reset input (optional)	Active-low input to asynchronously reset the boundary-scan circuit. For non-JTAG users, you should permanently tie the pin to GND.

Note to Table 13–1:

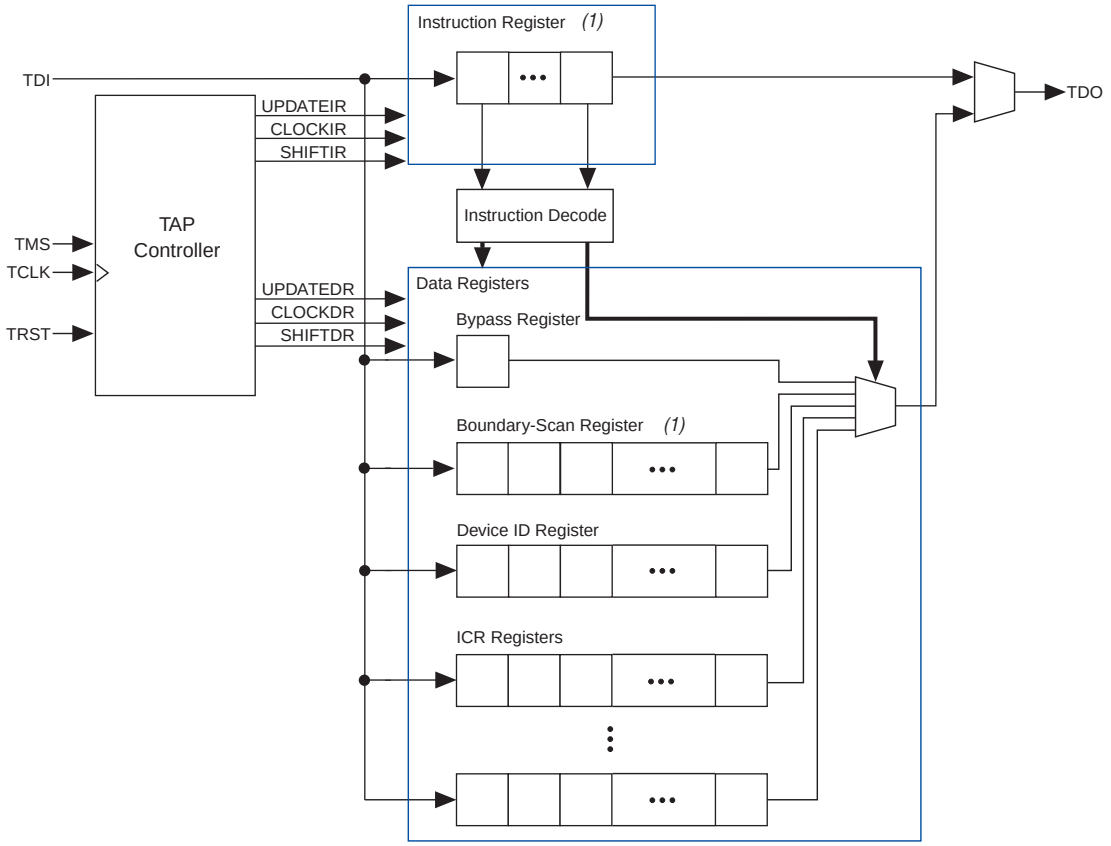
(1) The minimum TRST pulse width to reset the JTAG TAP controller is 60 ns.

The IEEE Std. 1149.1 BST circuitry requires the following registers:

- The instruction register determines the action to be performed and the data register to be accessed.
- The bypass register is a one-bit-long data register that provides a minimum-length serial path between TDI and TDO.
- The boundary-scan register is a shift register composed of all the boundary-scan cells of the device.

Figure 13–2 shows a functional model of the IEEE Std. 1149.1 circuitry.

Figure 13–2. IEEE Std. 1149.1 Circuitry



Note to Figure 13–2:

(1) For register lengths, refer to the device datasheet in the *Configuring Stratix III Devices* chapter in volume 1 of the *Stratix III Device Handbook*.

IEEE Std. 1149.1 boundary-scan testing is controlled by a TAP controller. For more information about the TAP controller, refer to "IEEE Std. 1149.1 BST Operation Control" on page 13–7. The `TMS` and `TCK` pins operate the TAP controller. The `TDI` and `TDO` pins provide the serial path for the data registers. The `TDI` pin also provides data to the instruction register, which then generates control logic for the data registers.

IEEE Std. 1149.1 Boundary-Scan Register

The boundary-scan register is a large serial shift register that uses the TDI pin as an input and the TDO pin as an output. The boundary-scan register consists of three-bit peripheral elements that are associated with Stratix III I/O pins. You can use the boundary-scan register to test external pin connections or to capture internal data.

For the Stratix III family device boundary-scan register lengths, refer to the *Configuring Stratix III Devices* chapter in volume 1 of the *Stratix III Device Handbook*.

Figure 13–3 shows how test data is serially shifted around the periphery of the IEEE Std. 1149.1 device.

Figure 13–3. Boundary-Scan Register

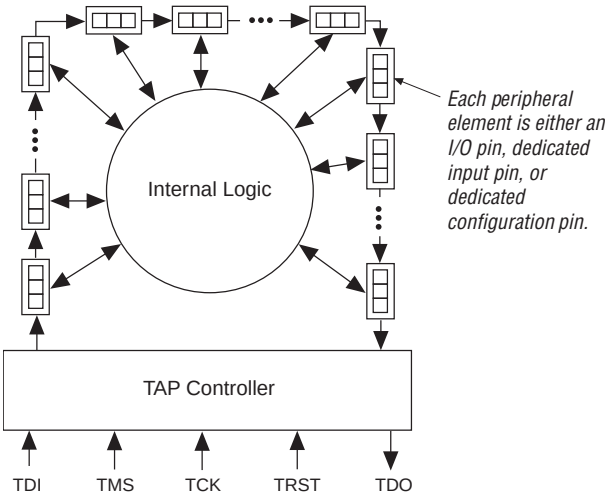


Table 13–2 lists the boundary-scan register length for Stratix III devices.

Table 13–2. Stratix III Boundary-Scan Register Length

Device	Boundary-Scan Register Length
EP3SL50	1506
EP3SL70	1506
EP3SL110	2274
EP3SL150ES	2274
EP3SL150	2274
EP3SL200	2970
EP3SL340	3402
EP3SE50	1506
EP3SE80	2274
EP3SE110	2274
EP3SE260	2970

Boundary-Scan Cells of a Stratix III Device I/O Pin

The Stratix III device three-bit boundary-scan cell (BSC) consists of a set of capture registers and a set of update registers. The capture registers can connect to internal device data through the `OUTJ`, `OEJ`, and `PIN_IN` signals, while the update registers connect to external data through the `PIN_OUT` and `PIN_OE` signals.

The global control signals for the IEEE Std. 1149.1 BST registers (such as `shift`, `clock`, and `update`) are generated internally by the TAP controller. The `MODE` signal is generated by a decode of the instruction register. The `HIGHZ` signal is high when executing the `HIGHZ` instruction. The data signal path for the boundary-scan register runs from the serial data in (`SDI`) signal to the serial data out (`SDO`) signal. The scan register begins at the `TDI` pin and ends at the `TDO` pin of the device.

Figure 13–4 shows the Stratix III device's user I/O boundary-scan cell.

Figure 13–4. Stratix III Device's User I/O BSC with IEEE Std. 1149.1 BST Circuitry

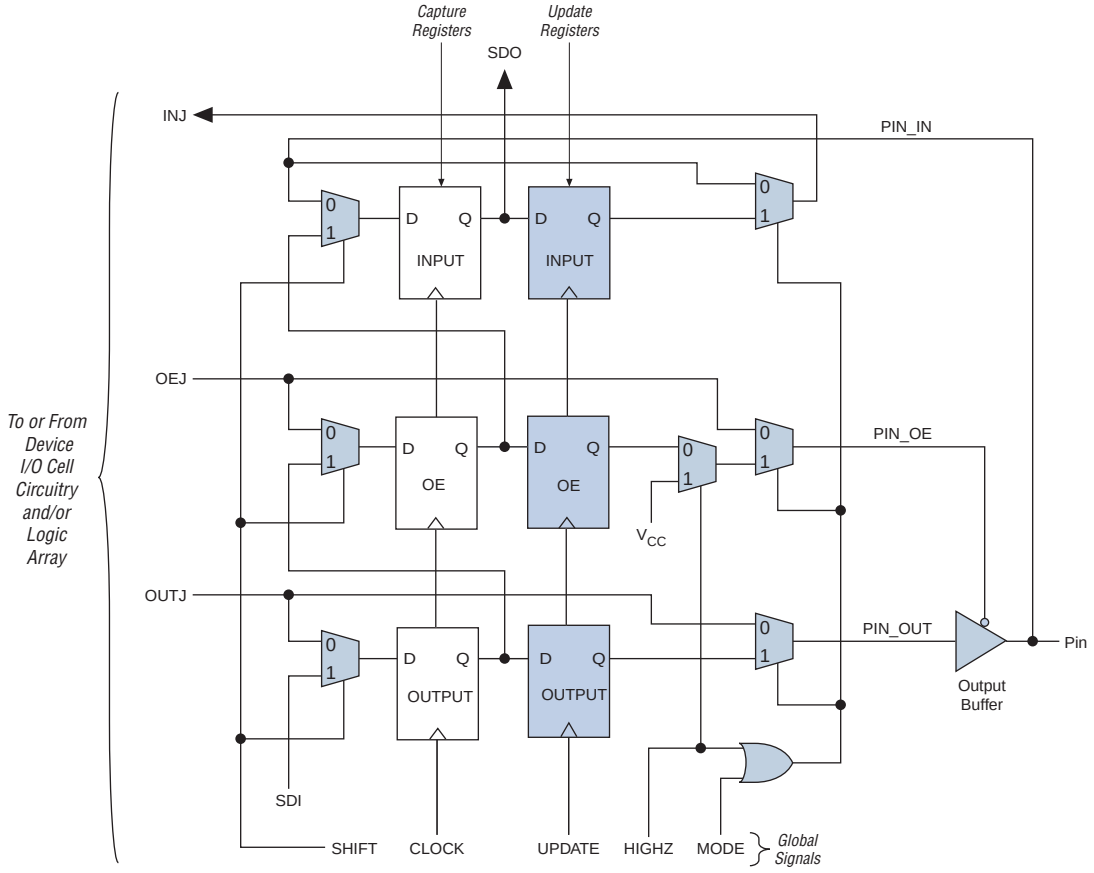


Table 13–3 lists the capture and update register capabilities of all boundary-scan cells within Stratix III devices.

Table 13–3. Stratix III Device Boundary Scan Cell Descriptions *(Note 1)*

Pin Type	Captures			Drives			Comments
	Output Capture Register	OE Capture Register	Input Capture Register	Output Update Register	OE Update Register	Input Update Register	
User I/O pins	OUTJ	OEJ	PIN_IN	PIN_OUT	PIN_OE	INJ	—
Dedicated clock input	0	1	PIN_IN	—	—	—	PIN_IN drives to clock network or logic array
Dedicated input *(2)*	0	1	PIN_IN	—	—	—	PIN_IN drives to control logic
Dedicated bi-directional (open drain) *(3)*	0	OEJ	PIN_IN	—	—	—	PIN_IN drives to configuration control
Dedicated bi-directional *(4)*	OUTJ	OEJ	PIN_IN	—	—	—	PIN_IN drives to configuration control and OUTJ drives to output buffer
Dedicated output *(5)*	OUTJ	0	0	—	—	—	OUTJ drives to output buffer

Notes to Table 13–3:

(1) TDI, TDO, TMS, TCK, TRST, all V_{CC} and GND pin types, VREF, and TEMP_DIODE pins do not have BSCs.
(2) This includes pins PLL_ENA, nCONFIG, MSEL0, MSEL1, MSEL2, nCE, PORSEL, and nIO_PULLUP.
(3) This includes pins CONF_DONE and nSTATUS.
(4) This includes pin DCLK.
(5) This includes pin nCEO.

IEEE Std. 1149.1 BST Operation Control

Stratix III devices support the IEEE Std. 1149.1 (JTAG) instructions listed in Table 13–4.

Table 13–4. Stratix III JTAG Instructions

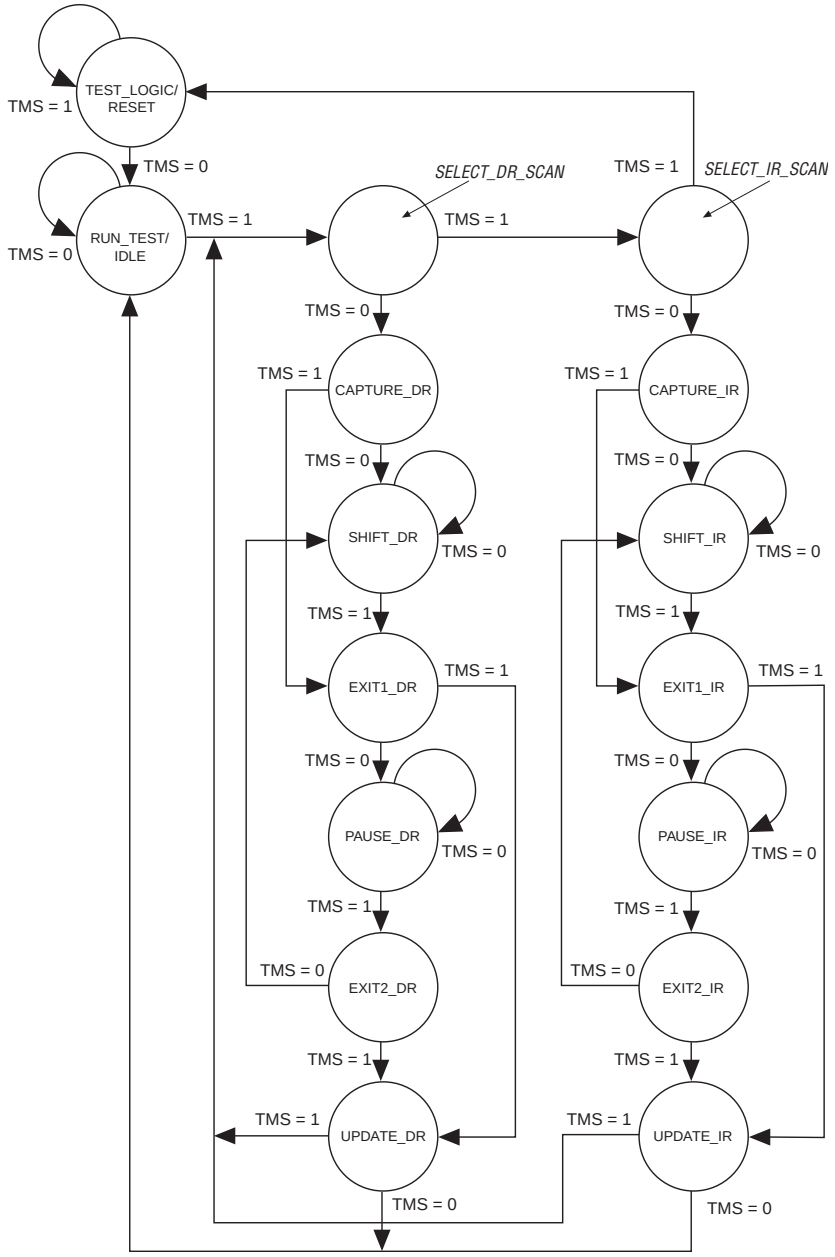
JTAG Instruction	Instruction Code	Description
SAMPLE / PRELOAD	00 0000 0101	Allows a snapshot of signals at the device pins to be captured and examined during normal device operation, and permits an initial data pattern to be output at the device pins. Also used by the SignalTap® II embedded logic analyzer.
EXTEST *(1)*	00 0000 1111	Allows the external circuitry and board-level interconnects to be tested by forcing a test pattern at the output pins and capturing test results at the input pins.
BYPASS	11 1111 1111	Places the 1-bit bypass register between the TDI and TDO pins, which allows the BST data to pass synchronously through selected devices to adjacent devices during normal device operation.
USERCODE	00 0000 0111	Places the 32-bit device identification register between TDI and TDO. The USERCODE value are loaded into this Device ID register for shifting out through TDO.
IDCODE	00 0000 0110	Selects the IDCODE register and places it between TDI and TDO, allowing the IDCODE to be serially shifted out of TDO.
HIGHZ *(1)*	00 0000 1011	Places the 1-bit bypass register between the TDI and TDO pins, which allows the BST data to pass synchronously through selected devices to adjacent devices during normal device operation, while tri-stating all of the I/O pins.
CLAMP *(1)*	00 0000 1010	Places the 1-bit bypass register between the TDI and TDO pins, which allows the BST data to pass synchronously through selected devices to adjacent devices during normal device operation while holding I/O pins to a state defined by the data in the boundary-scan register.
ICR instructions	—	Used when configuring a Stratix III device through the JTAG port with a USB-Blaster™, ByteBlaster™ II, MasterBlaster™ or ByteBlasterMV™ download cable, or when using a Jam File or Jam Byte-Code (JBC) File through an embedded processor.
PULSE_NCONFIG	00 0000 0001	Emulates pulsing the nCONFIG pin low to trigger reconfiguration even though the physical pin is unaffected.
CONFIG_IO	00 0000 1101	Allows I/O reconfiguration through JTAG ports using the IOCSR for JTAG testing. Can be executed before, after, or during configurations.

Note to Table 13–4:

(1) Bus hold and weak pull-up resistor features override the high-impedance state of the HIGHZ, CLAMP, and EXTEST instructions.

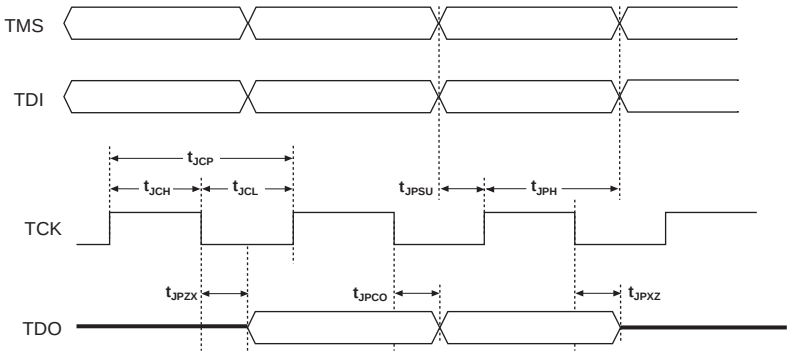
The IEEE Std. 1149.1 TAP controller, a 16-state machine clocked on the rising edge of TCK, uses the TMS pin to control IEEE Std. 1149.1 operation in the device. Figure 13–5 shows the TAP controller state machine.

Figure 13–5. IEEE Std. 1149.1 TAP Controller State Machine



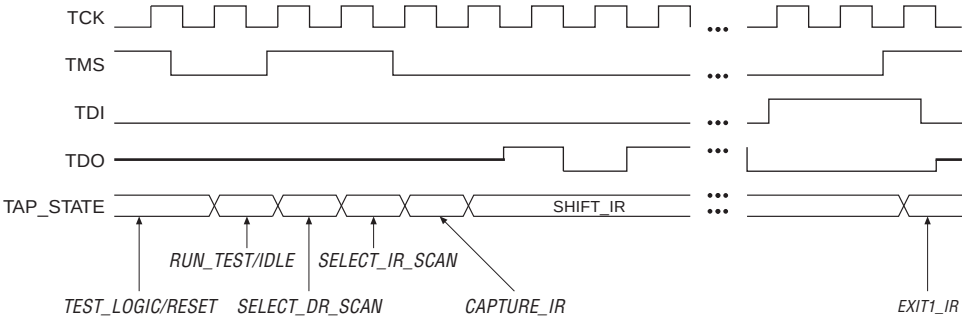
When the TAP controller is in the `TEST_LOGIC/RESET` state, the BST circuitry is disabled, the device is in normal operation, and the instruction register is initialized with `IDCODE` as the initial instruction. At device power-up, the TAP controller starts in this `TEST_LOGIC/RESET` state. In addition, forcing the TAP controller to the `TEST_LOGIC/RESET` state is achieved by holding `TMS` high for five `TCK` clock cycles, or by holding the `TRST` pin low. In the `TEST_LOGIC/RESET` state, the TAP controller remains in this state as long as `TMS` is held high (while `TCK` is clocked) or `TRST` is held low. Figure 13–6 shows the timing requirements for the IEEE Std. 1149.1 signals.

Figure 13–6. IEEE Std. 1149.1 Timing Waveforms



To start IEEE Std. 1149.1 operation, select an instruction mode by advancing the TAP controller to the shift instruction register (`SHIFT_IR`) state and shift in the appropriate instruction code on the `TDI` pin. The waveform diagram in Figure 13–7 shows the entry of the instruction code into the instruction register. It also shows the values of `TCK`, `TMS`, `TDI`, `TDO`, and the states of the TAP controller. From the `RESET` state, `TMS` is clocked with the pattern `01100` to advance the TAP controller to `SHIFT_IR`.

Figure 13–7. Selecting the Instruction Mode



The `TDO` pin is tri-stated in all states except in the `SHIFT_IR` and `SHIFT_DR` states. The `TDO` pin is activated at the first falling edge of `TCK` after entering either of the shift states and is tri-stated at the first falling edge of `TCK` after leaving either of the shift states.

When the `SHIFT_IR` state is activated, `TDO` is no longer tri-stated, and the initial state of the instruction register is shifted out on the falling edge of `TCK`. `TDO` continues to shift out the contents of the instruction register as long as the `SHIFT_IR` state is active. The TAP controller remains in the `SHIFT_IR` state as long as `TMS` remains low.

During the SHIFT_IR state, an instruction code is entered by shifting data on the TDI pin on the rising edge of TCK. The last bit of the instruction code is clocked at the same time that the next state, EXIT1_IR, is activated. Set TMS high to activate the EXIT1_IR state. After the EXIT1_IR state is activated, TDO becomes tri-stated again. TDO is always tri-stated except in the SHIFT_IR and SHIFT_DR states. After an instruction code is entered correctly, the TAP controller advances to serially shift test data in one of three modes.

The three serially shift test data instruction modes are discussed in the following sections:

- "SAMPLE/PRELOAD Instruction Mode" on page 13–11
- "EXTEST Instruction Mode" on page 13–13
- "BYPASS Instruction Mode" on page 13–15

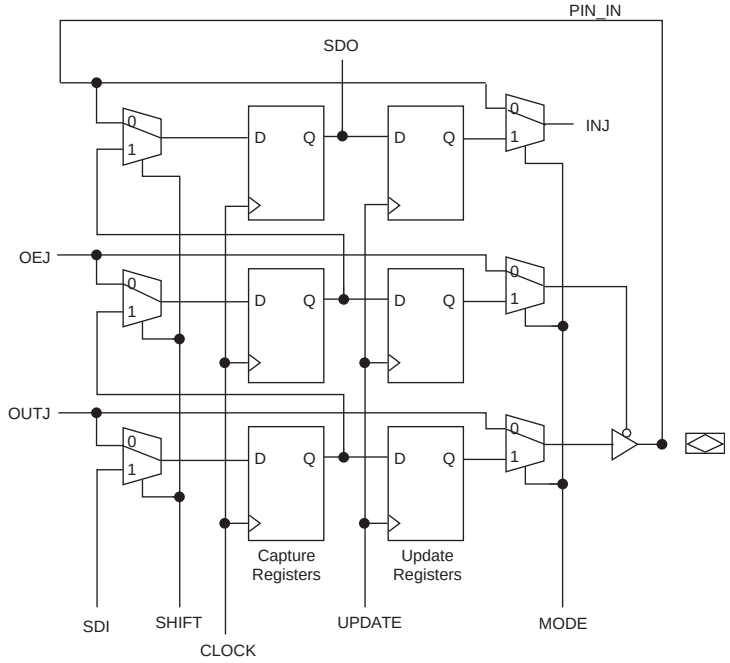
SAMPLE/PRELOAD Instruction Mode

The SAMPLE/PRELOAD instruction mode allows you to take a snapshot of device data without interrupting normal device operation. However, this instruction is most often used to preload the test data into the update registers prior to loading the EXTEST instruction. Figure 13–8 shows the capture, shift, and update phases of the SAMPLE/PRELOAD mode.

Figure 13–8. IEEE Std. 1149.1 BST SAMPLE/PRELOAD Mode

Capture Phase

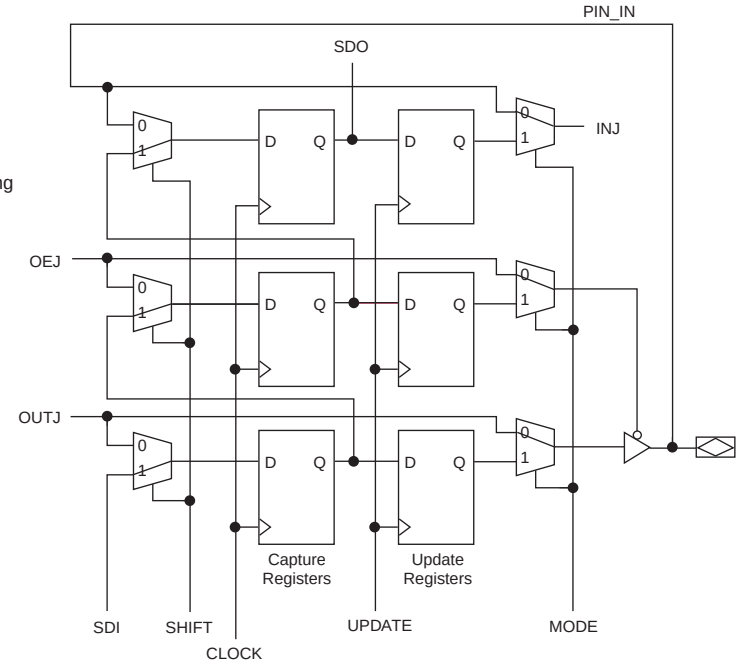
In the capture phase, the signals at the pin, OEJ and OUTJ, are loaded into the capture registers. The CLOCK signals are supplied by the TAP controller's CLOCKDR output. The data retained in these registers consists of signals from normal device operation.



Shift and Update Phases

In the shift phase, the previously captured signals at the pin, OEJ and OUTJ, are shifted out of the boundary-scan register through the TDO pin using CLOCK. As data is shifted out, the patterns for the next test can be shifted in through the TDI pin.

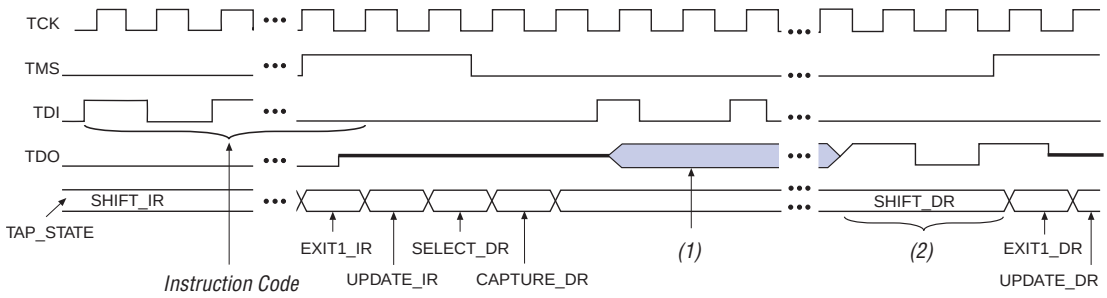
In the update phase, data is transferred from the capture registers to the update registers using the UPDATE clock. The data stored in the update registers can be used for the EXTEST instruction.



During the capture phase, multiplexers preceding the capture registers select the active device data signals. This data is then clocked into the capture registers. The multiplexers at the outputs of the update registers also select active device data to prevent functional interruptions to the device. During the shift phase, the boundary-scan shift register is formed by clocking data through the capture registers around the device periphery and then out of the TDO pin. The device can simultaneously shift new test data into TDI and replace the contents of the capture registers. During the update phase, data in the capture registers is transferred to the update registers. You can then use this data in the EXTEST instruction mode. Refer to "EXTEST Instruction Mode" on page 13–13 for more information.

Figure 13–9 shows the SAMPLE/PRELOAD waveforms. The SAMPLE/PRELOAD instruction code is shifted in through the TDI pin. The TAP controller advances to the CAPTURE_DR state and then to the SHIFT_DR state, where it remains if TMS is held low. The data that was present in the capture registers after the capture phase is shifted out of the TDO pin. New test data shifted into the TDI pin appears at the TDO pin after being clocked through the entire boundary-scan register. If TMS is held high on two consecutive TCK clock cycles, the TAP controller advances to the UPDATE_DR state for the update phase.

Figure 13–9. SAMPLE/PRELOAD Shift Data Register Waveforms



Note to Figure 13–9:

(1) Data stored in boundary-scan register is shifted out of TDO.

(2) After boundary-scan register data has been shifted out, data entered into TDI will shift out of TDO.

EXTEST Instruction Mode

Use the EXTEST instruction mode primarily to check external pin connections between devices. Unlike the SAMPLE/PRELOAD mode, EXTEST allows test data to be forced onto the pin signals. By forcing known logic high and low levels on output pins, you can detect opens and shorts at pins of any device in the scan chain.

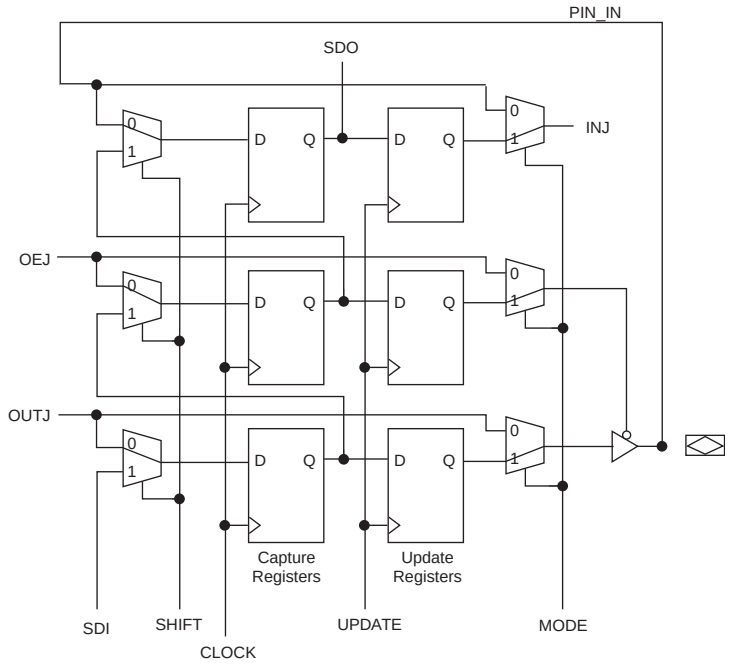
Figure 13–10 shows the capture, shift, and update phases of the EXTEST mode.

Figure 13–10. IEEE Std. 1149.1 BST EXTEST Mode

Capture Phase

In the capture phase, the signals at the pin, OEJ and OUTJ, are loaded into the capture registers. The CLOCK signals are supplied by the TAP controller's CLOCKDR output. Previously retained data in the update registers drive the PIN_OUT, INJ, and allows the I/O pin to tri-state or drive a signal out.

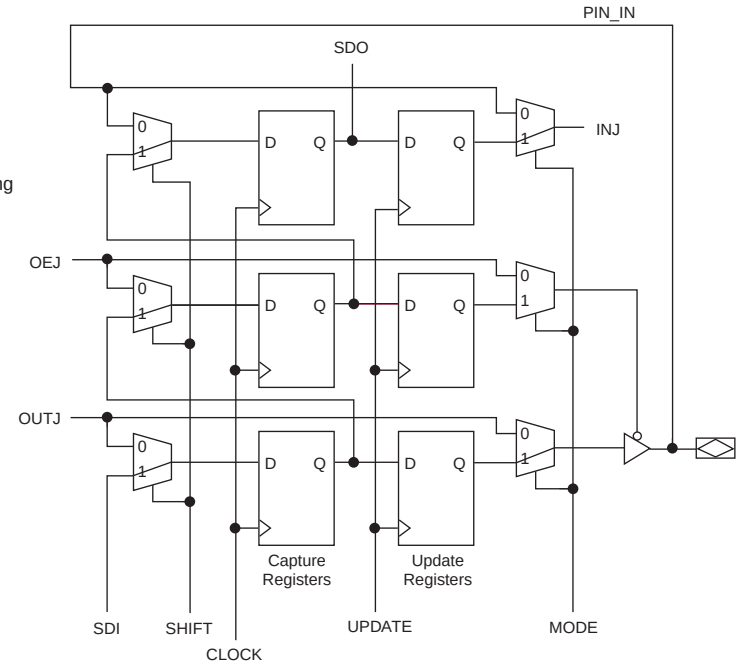
A "1" in the OEJ update register tri-states the output buffer.



Shift and Update Phases

In the shift phase, the previously captured signals at the pin, OEJ and OUTJ, are shifted out of the boundary-scan register through the TDO pin using CLOCK. As data is shifted out, the patterns for the next test can be shifted in through the TDI pin.

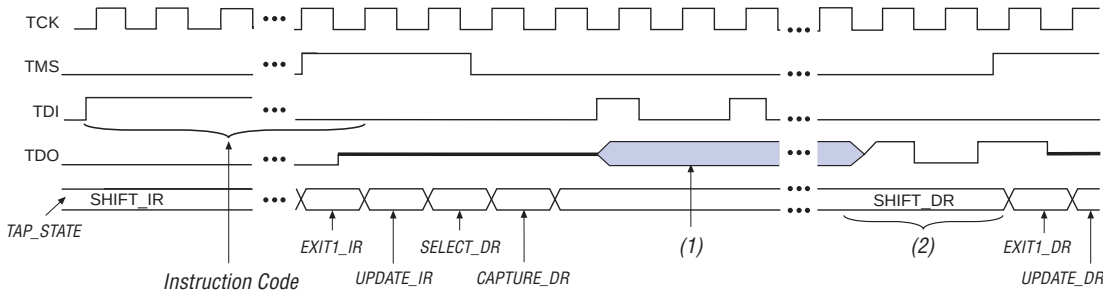
In the update phase, data is transferred from the capture registers to the update registers using the UPDATE clock. The update registers then drive the PIN_OUT, INJ, and allow the I/O pin to tri-state or drive a signal out.



EXTEST mode selects data differently than SAMPLE/PRELOAD mode. EXTEST chooses data from the update registers as the source of the output and output-enable signals. After the EXTEST instruction code is entered, the multiplexers select the update register data. Therefore, data stored in these registers from a previous EXTEST or SAMPLE/PRELOAD test cycle can be forced onto the pin signals. In the capture phase, the results of this test data are stored in the capture registers and then shifted out of TDO during the shift phase. You can then store new test data in the update registers during the update phase.

The EXTEST waveform diagram in Figure 13–11 resembles the SAMPLE/PRELOAD waveform diagram, except for the instruction code. The data shifted out of TDO consists of the data that was present in the capture registers after the capture phase. New test data shifted into the TDI pin appears at the TDO pin after being clocked through the entire boundary-scan register.

Figure 13–11. EXTEST Shift Data Register Waveforms



Note to Figure 13–11:

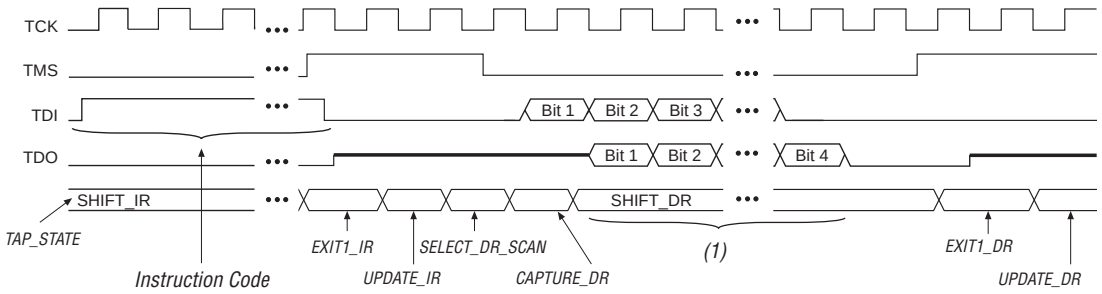
(1) Data stored in the boundary-scan register is shifted out of TDO.

(2) After boundary-scan register data has been shifted out, data entered into TDI will shift out of TDO.

BYPASS Instruction Mode

The BYPASS mode is activated when an instruction code of all ones is loaded in the instruction register. This mode allows the boundary scan data to pass the selected device synchronously to adjacent devices when no test operation of the device is needed at the board level. The waveforms in Figure 13–12 show how scan data passes through a device after the TAP controller is in the SHIFT_DR state. In this state, data signals are clocked into the bypass register from TDI on the rising edge of TCK and out of TDO on the falling edge of the same clock pulse.

Figure 13–12. BYPASS Shift Data Register Waveforms



Note to Figure 13–12:

(1) Data shifted into TDI on the rising edge of TCK is shifted out of TDO on the falling edge of the same TCK pulse.

IDCODE Instruction Mode

Use the IDCODE instruction mode to identify the devices in an IEEE Std. 1149.1 chain. When IDCODE is selected, the device identification register is loaded with the 32-bit vendor-defined identification code. The device ID register is connected between the TDI and TDO ports, and the device IDCODE is shifted out. Table 13–5 lists the IDCODE information for Stratix III devices.

Table 13–5. 32-Bit Stratix III Device IDCODE

Device	IDCODE (32 Bits) *(1)*			
	Version (4 Bits)	Part Number (16 Bits)	Manufacturer Identity (11 Bits)	LSB (1 Bit) *(2)*
EP3SL50	0000	0010 0001 0000 1000	000 0110 1110	1
EP3SL70	0000	0010 0001 0000 0001	000 0110 1110	1
EP3SL110	0001	0010 0001 0000 1001	000 0110 1110	1
EP3SL150ES	0000	0010 0001 0000 0010	000 0110 1110	1
EP3SL150	0001	0010 0001 0000 0010	000 0110 1110	1
EP3SL200	0000	0010 0001 0000 0011	000 0110 1110	1
EP3SL340	0000	0010 0001 0000 0101	000 0110 1110	1
EP3SE50	0000	0010 0001 0000 0110	000 0110 1110	1
EP3SE80	0000	0010 0001 0000 1010	000 0110 1110	1
EP3SE110	0000	0010 0001 0000 0111	000 0110 1110	1
EP3SE260	0000	0010 0001 0000 0100	000 0110 1110	1

Notes to Table 13–5:

(1) The most significant bit (MSB) is on the left.

(2) The IDCODE's least significant bit (LSB) is always 1.

USERCODE Instruction Mode

Use the USERCODE instruction mode to examine the user electronic signature (UES) within the devices along an IEEE Std. 1149.1 chain. When you select this instruction, the device identification register is connected between the TDI and TDO ports. The user-defined UES is shifted into the device ID register in parallel from the 32-bit USERCODE register. The UES is then shifted out through the device ID register.

The UES value is not user defined until after the device is configured. This value is stored in the programmer object file (**.pof**) and only loaded to the device during configuration. Before configuration, the UES value is set to the default value.

CLAMP Instruction Mode

Use the CLAMP instruction mode to allow the state of the signals driven from the pins to be determined from the boundary-scan register while the bypass register is selected as the serial path between the TDI and TDO ports. The states of all signals driven from the pins are completely defined by the data held in the boundary-scan register.

HIGHZ Instruction Mode

The HIGHZ instruction mode sets all of the user I/O pins to an inactive drive state. These pins are tri-stated until a new JTAG instruction is executed. When this instruction is loaded into the instruction register, the bypass register is connected between the TDI and TDO ports.

I/O Voltage Support in JTAG Chain

The JTAG chain supports several devices. However, you should use caution if the chain contains devices that have different V_{CCIO} levels. The output voltage level of the TDO pin must meet the specifications of the TDI pin it drives. The TDI and TDO pins of Stratix III device are powered by the V_{CCPD} (2.5 V / 3.0 V / 3.3 V) of I/O Bank 1A. You should connect V_{CCPD} according to the I/O standard used in the same bank. For 3.3-V I/O standards, you should connect V_{CCPD} to 3.3 V. For 3.0-V I/O standards, you should connect V_{CCPD} to 3.0 V; for 2.5-V and below I/O standards, you should connect V_{CCPD} to 2.5 V. Table 13–6 lists board design recommendations to ensure proper JTAG chain operation.

Table 13–6. Supported TDO/TDI Voltage Combinations

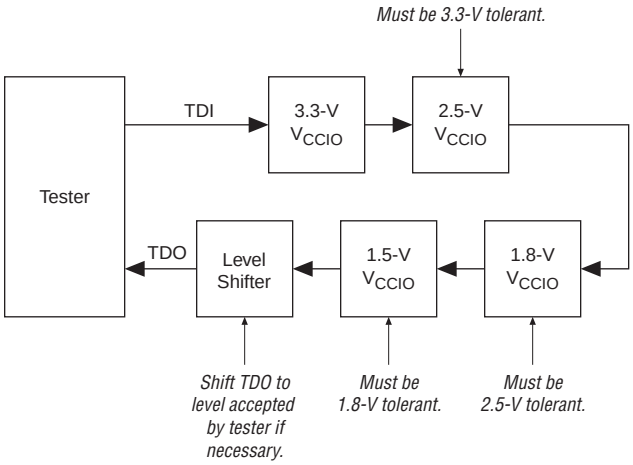
Device	TDI Input Buffer Power	Stratix III TDO V_{CCPD}		
		V_{CCPD} = 3.3 V *(1)*	V_{CCPD} = 3.0 V *(1)*	V_{CCPD} = 2.5 V *(2)*
Stratix III	V_{CCPD} = 3.3V	✓	✓	✓
	V_{CCPD} = 3.0V	✓	✓	✓
	V_{CCPD} = 2.5V	✓	✓	✓
Non-Stratix III	V_{CC} = 3.3 V	✓ *(3)*	✓ *(4)*	✓ *(5)*
	V_{CC} = 2.5 V	✓ *(3)*	✓ *(4)*	✓ *(5)*
	V_{CC} = 1.8 V	✓ *(3)*	✓ *(4)*	✓ *(5)*
	V_{CC} = 1.5 V	✓ *(3)*	✓ *(4)*	✓ *(5)*

Notes to Table 13–6:

(1) The TDO output buffer meets V_{OH} (MIN) = 2.4 V.
(2) The TDO output buffer meets V_{OH} (MIN) = 2.0 V.
(3) Input buffer must be 3.3-V tolerant.
(4) Input buffer must be 3.0-V tolerant.
(5) Input buffer must be 2.5-V tolerant.

You can interface the TDI and TDO lines of the devices that have different V_{CCIO} levels by inserting a level shifter between the devices. If possible, you should build the JTAG chain in such a way that a device with a higher V_{CCIO} level drives to a device with an equal or lower V_{CCIO} level. This way, a level shifter is used only to shift the TDO level to a level acceptable to the JTAG tester. Figure 13–13 shows the JTAG chain of mixed voltages and how a level shifter is inserted in the chain.

Figure 13–13. JTAG Chain of Mixed Voltages



IEEE Std. 1149.1 BST Circuitry

Stratix III devices have dedicated JTAG pins and the IEEE Std. 1149.1 BST circuitry is enabled upon device power-up. You can perform BST on Stratix III FPGAs before, during, and after configuration. Stratix III FPGAs support the `BYPASS`, `IDCODE`, and `SAMPLE` instructions during configuration without interrupting configuration. To send all other JTAG instructions, you must interrupt configuration using the `CONFIG_IO` instruction.

The `CONFIG_IO` instruction allows you to configure I/O buffers through the JTAG port, and when issued, interrupts configuration. This instruction allows you to perform board-level testing prior to configuring the Stratix III FPGA or you can wait for the configuration device to complete configuration. After configuration is interrupted and JTAG BST is complete, you must reconfigure the part through JTAG (`PULSE_CONFIG` instruction) or by pulsing `nCONFIG` low.

When you perform JTAG boundary-scan testing before configuration, the `nCONFIG` pin must be held low.

The chip-wide reset (`DEV_CLRn`) and chip-wide output enable (`DEV_OE`) pins on Stratix III devices do not affect JTAG boundary-scan or configuration operations. Toggling these pins does not disrupt BST operation (other than the expected BST behavior).

When you design a board for JTAG configuration of Stratix III devices, you must consider the connections for the dedicated configuration pins.

For more information about using the IEEE Std.1149.1 circuitry for device configuration, refer to the *Configuring Stratix III Devices* chapter in volume 1 of the *Stratix III Device Handbook*.

IEEE Std. 1149.1 BST Circuitry (Disabling)

The IEEE Std. 1149.1 BST circuitry for Stratix III devices is enabled upon device power-up. Because the IEEE Std. 1149.1 BST circuitry is used for BST or in-circuit reconfiguration, you must enable the circuitry only at specific times as mentioned in, "IEEE Std. 1149.1 BST Circuitry" on page 13–18.

If you are not using the IEEE Std. 1149.1 circuitry in Stratix III, you should permanently disable the circuitry to ensure that you do not inadvertently enable it when it is not required.

Table 13–7 lists the pin connections necessary for disabling the IEEE Std. 1149.1 circuitry in Stratix III devices.

Table 13–7. Disabling IEEE Std. 1149.1 Circuitry

JTAG Pins *(1)*	Connection for Disabling
TMS	V_{CCPD} supply of Bank 1A
TCK	GND
TDI	V_{CCPD} supply of Bank 1A
TDO	Leave open
TRST	GND

Note to Table 13–7:

(1) There is no software option to disable JTAG in Stratix III devices. The JTAG pins are dedicated.

IEEE Std. 1149.1 BST Guidelines

Use the following guidelines when performing boundary-scan testing with IEEE Std. 1149.1 devices:

- If the "10..." pattern does not shift out of the instruction register through the TDO pin during the first clock cycle of the SHIFT_IR state, the TAP controller did not reach the proper state. To solve this problem, try one of the following procedures:
  - Verify that the TAP controller has reached the SHIFT_IR state correctly. To advance the TAP controller to the SHIFT_IR state, return to the RESET state and send the code 01100 to the TMS pin.
  - Check the connections to the V_{CC}, GND, JTAG, and dedicated configuration pins on the device.

  Do NOT use the following private instructions as they may render the device inoperable:
  11 0001 0000
  00 1100 1001
  11 0001 0011
  11 0001 0111

  You should take precautions to avoid invoking these instructions at any time.

- Perform a `SAMPLE/PRELOAD` test cycle prior to the first `EXTEST` test cycle to ensure that known data is present at the device pins when you enter the `EXTEST` mode. If the `OEJ` update register contains a 0, the data in the `OUTJ` update register is driven out. The state must be known and correct to avoid contention with other devices in the system.
- Do not perform `EXTEST` testing during ICR. This instruction is supported before or after ICR, but not during ICR. Use the `CONFIG_IO` instruction to interrupt configuration and then perform testing, or wait for configuration to complete.
- If performing testing before configuration, hold the `nCONFIG` pin low.

For more information about boundary scan testing, contact Altera® Application at www.altera.com.

Boundary-Scan Description Language (BSDL) Support

The Boundary-Scan Description Language (BSDL), a subset of VHDL, provides a syntax that allows you to describe the features of an IEEE Std. 1149.1 BST-capable device that can be tested. Test software development systems then use the BSDL files for test generation, analysis, and failure diagnostics.

For more information about BSDL files for IEEE Std. 1149.1-compliant Stratix III devices, refer to the *Stratix III BSDL Files* page on the Altera website.

To perform BST on a configured device, you will require a post configuration BSDL file that is customized to your design. This file can be generated with the BSDL Customizer script.

For more information about the BSDL Customizer, refer to the *Altera BSDL Support* page on the Altera website.

Chapter Revision History

Table 13–8 lists the revision history for this chapter.

Table 13–8. Chapter Revision History

Date	Version	Changes Made
July 2010	1.9	Updated Table 13–1.
March 2010	1.8	Updated for the Quartus II software version 9.1 SP2 release: ■ Removed "IEEE Std. 1149.1 BST for Configured Devices" and "Conclusion" section. ■ Updated Table 13–5.
May 2009	1.7	■ Updated Table 13–1. ■ Updated "I/O Voltage Support in JTAG Chain" and "IEEE Std. 1149.1 BST Circuitry" sections.
February 2009	1.6	Removed "Referenced Documents" section.
October 2008	1.5	■ Updated Table 13–4. ■ Updated New Document Format.
May 2008	1.4	■ Updated Table 13–2 and Table 13–5 with EP3SL150ES information. ■ Updated Table 13–6. ■ Updated Figure 13–6.
November 2007	1.3	Updated Table 13–2.
October 2007	1.2	■ Added new section "Referenced Documents". ■ Added live links for references.
May 2007	1.1	■ Updated Note 3 to Table 13–3. Updated Figure 13–6. ■ Added Table 13–2, Table 13–4, Table 13–5, and Table 13–7. ■ Removed opening paragraph and footnote for "IEEE Std. 1149.1 BST Operation Control" on page 13–9. ■ Added warning on page 13-22.
November 2006	1.0	Initial Release.

Section IV. Design Security and Single Event Upset (SEU) Mitigation

This section provides information on Design Security and Single Event Upset (SEU) Mitigation in Stratix® III devices.

- Chapter 14, Design Security in Stratix III Devices
- Chapter 15, SEU Mitigation in Stratix III Devices

Revision History

Refer to each chapter for its own specific revision history. For information on when each chapter was updated, refer to the Chapter Revision Dates section, which appears in the full handbook.

14. Design Security in Stratix III Devices

SIII51014-1.5

Introduction

This chapter provides an overview of the design security feature and its implementation on Stratix® III devices using advanced encryption standard (AES) as well as security modes available in Stratix III devices.

As Stratix III devices start to play a role in larger and more critical designs in competitive commercial and military environments, it is increasingly important to protect the designs from copying, reverse engineering, and tampering. Stratix III devices address these concerns and are the industry's only high-density, high-performance devices with both volatile and non-volatile security feature support. Stratix III devices have the ability to decrypt configuration bitstreams using the AES algorithm, an industry standard encryption algorithm that is FIPS-197 certified. They also have a design security feature that utilizes a 256-bit security key.

Altera® Stratix III devices store configuration data in static random access memory (SRAM) configuration cells during device operation. Because SRAM memory is volatile, SRAM cells must be loaded with configuration data each time the device powers-up. It is possible to intercept configuration data when it is being transmitted from the memory source (flash memory or a configuration device) to the device. The intercepted configuration data could then be used to configure another device.

When using the Stratix III design security feature, the security key is stored in the Stratix III device. Depending on the security mode, you can configure the Stratix III device using a configuration file that is encrypted with the same key, or for board testing, configured with a normal configuration file.

The design security feature is available when configuring Stratix III devices using the fast passive parallel (FPP) configuration mode with an external host (such as a MAX® II device or microprocessor), or when using fast active serial (AS) or passive serial (PS) configuration schemes. However, the design security feature is also available in remote update with fast AS configuration mode. The design security feature is not available when you are configuring your Stratix III device using Joint Test Action Group (JTAG)-based configuration. For more information, refer to "Supported Configuration Schemes" on page 14–5.

Stratix III Security Protection

Stratix III device designs are protected from copying, reverse engineering, and tampering using configuration bitstream encryption.

Security Against Copying

The security key is securely stored in the Stratix III device and cannot be read out through any interfaces. In addition, as configuration file read-back is not supported in Stratix III devices, the design information cannot be copied.

Security Against Reverse Engineering

Reverse engineering from an encrypted configuration file is very difficult and time consuming because the Stratix III configuration file formats are proprietary and the file contains million of bits which require specific decryption. Reverse engineering the Stratix III device is just as difficult because the device is manufactured on the most advanced 65-nm process technology.

Security Against Tampering

The non-volatile keys are one-time programmable. Once the tamper protection bit is set in the key programming file generated by the Quartus® II software, the Stratix III device can only be configured with configuration files encrypted with the same key.

For more information about why this feature is secured, refer to the *Design Security in Stratix III Devices white paper*.

AES Decryption Block

The main purpose of the AES decryption block is to decrypt the configuration bitstream prior to entering data decompression or configuration.

Prior to receiving encrypted data, you must enter and store the 256-bit security key in the device. You can choose between a non-volatile security key and a volatile security key with battery backup.

The security key is scrambled prior to storing it in key storage in order to make it more difficult for anyone to retrieve the stored key using de-capsulation of the device.

Flexible Security Key Storage

Stratix III devices support two types of security key programming: volatile and non-volatile. Table 14–1 shows the differences between volatile keys and non-volatile keys.

Table 14–1. Security Keys Options

Options	Volatile Key	Non-Volatile Key
Key programmability	Reprogrammable and erasable	One-time programmable
External battery	Required	Not required
Key programming method *(1)*	On-board	On and off board
Design protection	Secure against copying and reverse engineering	Secure against copying and reverse engineering. Tamper resistant if tamper protection bit is set.

Note to Table 14–1:

(1) Key programming is carried out using JTAG interface.

You can program the non-volatile key to the Stratix III device without an external battery. Also, there are no additional requirements to any of the Stratix III power supply inputs.

V_{CCBAT} is a dedicated power supply for volatile key storage and not shared with other on-chip power supplies, such as V_{CCIO} or V_{CC}. V_{CCBAT} continuously supplies power to the volatile register regardless of the on-chip supply condition.

Table 14–2. Key Retention Time of Coin-Cell Type Batteries used for Volatile Key Storage

Battery	Typical	Worst-case *(1)*
35mAh	49 years	6 years
1000mAh	1429 years	190 years

Note to Table 14–2:

(1) Worst-case refers to worst-case process and 100°C junction temperature.

After power-up, you need to wait 100 ms (PORSEL = 0) or 12 ms (PORSEL = 1) before beginning the key programming to ensure that V_{CCBAT} is at its full rail.

As an example, here are some lithium coin-cell type batteries used for volatile key storage purposes: BR1220 (-30° to +80°C) and BR2477A (-40°C to +125°C).

For more information about battery specifications, refer to the *DC and Switching Characteristics of Stratix III Devices* chapter in volume 2 of the *Stratix III Device Handbook*.

Stratix III Design Security Solution

Stratix III devices are SRAM-based devices. To provide design security, Stratix III devices require a 256-bit security key for configuration bitstream encryption.

To carry out secure configuration, complete the following steps. Figure 14–1 also describes secure configuration.

1. Program the security key into the Stratix III device.

   Program the user-defined 256-bit AES keys to the Stratix III device through the JTAG interface.

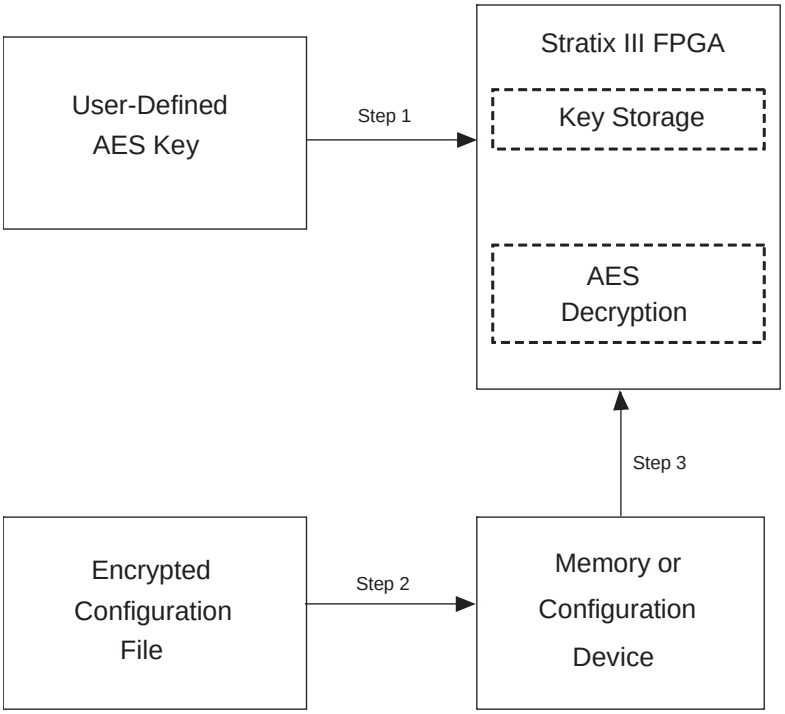
2. Encrypt the configuration file and store it in the external memory.

   Encrypt the configuration file with the same 256-bit keys used to program the Stratix III device. Encryption of the configuration file is done using the Quartus II software. The encrypted configuration file is then loaded into external memory, such as a configuration or flash device.

3. Configure the Stratix III device.

   At system power-up, the external memory device sends the encrypted configuration file to the Stratix III device.

Figure 14–1. Design Security *(Note 1)*



Note to Figure 14–1:

(1) Step 1, Step 2, and Step 3 correspond to the procedure detailed in the "Stratix III Design Security Solution" section.

Security Modes Available

There are several security modes available on the Stratix III device, which are described as follows:

Volatile Key

Secure operation with volatile key programmed and required external battery—this mode accepts both encrypted and unencrypted configuration bitstreams. Use the unencrypted configuration bitstream support for board-level testing only.

Non-Volatile Key

Secure operation with one time programmable (OTP) security key programmed—this mode accepts both encrypted and unencrypted configuration bitstreams. Use the unencrypted configuration bitstream support for board-level testing only.

Non-Volatile Key with Tamper Protection Bit Set

Secure operation in tamper resistant mode with OTP security key programmed—only encrypted configuration bitstreams are allowed to configure the device.
Tamper protection disables JTAG configuration with unencrypted configuration bitstream.

Setting the tamper protection bit disables test mode in Stratix III devices. This process is irreversible and prevents Altera from carrying-out failure analysis if test mode is disabled. Contact Altera Technical Support to set the tamper protection bit.

You can perform Boundary Scan testing or use the SignalTap II logic analyzer to analyze functional data with the tamper-protection bit set programmed into the Stratix III FPGA.

No Key Operation

Only unencrypted configuration bitstreams are allowed to configure the device.

Table 14–3 summarizes the different security modes and the configuration bitstream supported for each mode.

Table 14–3. Security Modes Supported

Mode *(1)*	Function	Configuration File
Volatile key	Secure	Encrypted
	Board-level testing	Unencrypted
Non-volatile key	Secure	Encrypted
	Board-level testing	Unencrypted
Non-volatile key with tamper protection bit set	Secure (tamper resistant) *(2)*	Encrypted

Notes to Table 14–3:

(1) In No key operation, only unencrypted configuration file is supported.

(2) The tamper protection bit setting does not prevent the device from being reconfigured.

Supported Configuration Schemes

The Stratix III device supports only selected configuration schemes, depending on the security mode you select when you encrypt the Stratix III device.

Figure 14–2 shows the restrictions of each security mode when encrypting Stratix III devices.

Figure 14–2. Stratix III Security Modes - Sequence and Restrictions

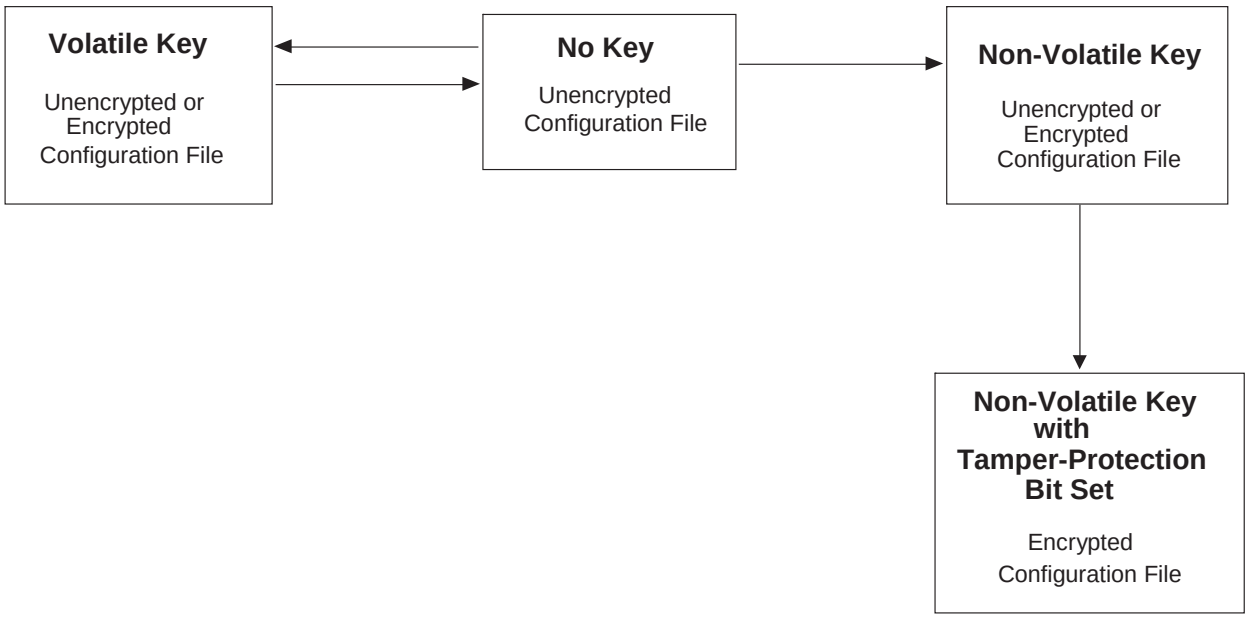


Table 14–4 shows the configuration modes allowed in each of the security modes.

Table 14–4. Allowed Configuration Modes for Various Security Modes *(Note 1)* (Part 1 of 2)

Security Mode	Configuration File	Allowed Configuration Modes
No key	Unencrypted	All configuration modes that do not engage the design security feature.
Secure with volatile key	Encrypted	■ Passive serial with AES (and/or with decompression) ■ Fast passive parallel with AES (and/or with decompression) ■ Remote update fast AS with AES (and/or with decompression) ■ Fast AS (and/or with decompression)
Board-level testing with volatile key	Unencrypted	All configuration modes that do not engage the design security feature.
Secure with non-volatile key	Encrypted	■ Passive serial with AES (and/or with decompression) ■ Fast passive parallel with AES (and/or with decompression) ■ Remote update fast AS with AES (and/or with decompression) ■ Fast AS (and/or with decompression)
Board-level testing with non-volatile key	Unencrypted	All configuration modes that do not engage the design security feature.

Table 14–4. Allowed Configuration Modes for Various Security Modes *(Note 1)* (Part 2 of 2)

Security Mode	Configuration File	Allowed Configuration Modes
Secure in tamper resistant mode using non-volatile key with tamper protection set	Encrypted	■ Passive serial with AES (and/or with decompression) ■ Fast passive parallel with AES (and/or with decompression) ■ Remote update fast AS with AES (and/or with decompression) ■ Fast AS (and/or with decompression)

Note to Table 14–4:

(1) There is no impact to the configuration time required compared to unencrypted configuration modes except fast passive parallel with AES (and/or decompression) which requires DCLK of 4× the data rate.

The design security feature with encrypted configuration file is available in all configuration methods, except JTAG. Therefore, use the design security feature in FPP mode (when using external controller, such as a MAX II device or a microprocessor and a flash memory), or in fast AS and PS configuration schemes.

Table 14–5 summarizes the configuration schemes that support the design security feature both for volatile and non-volatile key programming.

Table 14–5. Design Security Configuration Schemes Availability

Configuration Scheme	Configuration Method	Design Security
FPP	MAX II device or microprocessor and flash memory	✓ *(1)*
Fast AS	Serial configuration device	✓
PS	MAX II device or microprocessor and flash memory	✓
	Download cable	✓
JTAG *(2)*	MAX II device or microprocessor and flash memory	
	Download cable	—

Notes to Table 14–5:

(1) In this mode, the host system must send a DCLK that is 4× the data rate.

(2) JTAG configuration supports only unencrypted configuration file.

Use the design security feature with other configuration features, such as compression and remote system upgrade features. When you use compression with the design security feature, the configuration file is first compressed and then encrypted using the Quartus II software. During configuration, the Stratix III device first decrypts and then decompresses the configuration file.

Conclusion

The need for design security is increasing as devices move from glue logic to implementing critical system functions. Stratix III devices address this concern by providing built-in design security. These devices not only offer high density, fast performance, and cutting-edge features to meet your design needs, but also protect your designs against IP theft and tampering of your configuration files.

Chapter Revision History

Table 14–6 shows the revision history for this document.

Table 14–6. Chapter Revision History

Date and Revision	Changes Made	Summary of Changes
May 2009, version 1.5	Updated "Flexible Security Key Storage" and "Non-Volatile Key with Tamper Protection Bit Set" sections.	—
February 2009, version 1.4	■ Updated "Flexible Security Key Storage" section. ■ Removed "Referenced Documents" section.	—
October 2008, version 1.3	■ Updated "Non-Volatile Key with Tamper Protection Bit Set" section. ■ Added Table 14–2. ■ Updated New Document Format.	—
May 2008, version 1.2	■ Updated "Introduction" section. ■ Updated "Flexible Security Key Storage" section. ■ Updated Table 14–1 and Table 14–4. ■ Updated "Security Modes Available" section.	—
October 2007, version 1.1	■ Added new section "Referenced Documents". ■ Added live links for references.	Minor update
November 2006, version 1.0	Initial Release.	—

15. SEU Mitigation in Stratix III Devices

SIII51015-1.7

This chapter describes how to use the error detection cyclical redundancy check (CRC) feature when a Stratix® III device is in user mode and recovers from CRC errors. The purpose of the error detection CRC feature is to detect a flip in any of the configuration CRAM bits in Stratix III devices due to a soft error. By using the error detection circuitry, you can continuously verify the integrity of the configuration CRAM bits.

In critical applications such as avionics, telecommunications, system control, and military applications, it is important to be able to do the following:

- Confirm that the configuration data stored in a Stratix III device is correct.
- Alert the system to the occurrence of a configuration error.

The error detection feature has been enhanced in the Stratix III device family. In addition, the error detection and recovery time for single event upset (SEU) in Stratix III devices is reduced compared to Stratix II devices.

For Stratix III devices, use of the error detection CRC feature is provided in the Quartus® II software version 6.1 and onwards.

Stratix III devices only support the error detection CRC feature at 1.1 V for V_{CCL}. This feature is not supported in Stratix III devices operating at 0.9 V for V_{CCL}.

Dedicated circuitry is built into Stratix III devices and consists of a CRC error detection feature that can optionally check for SEUs continuously and automatically.

This section describes how to activate and use the error detection CRC feature when your Stratix III device is in user mode and describes how to recover from configuration errors caused by CRC errors.

Information about SEU is located on the Products page of the Altera® website at www.altera.com.

For more information regarding the test methodology for the enhanced error detection in Stratix III, refer to *AN 539: Test Methodology of Error Detection and Recovery using CRC in Altera FPGA Devices*.

For more information, refer to the *Robust SEU Mitigation with Stratix III FPGAs White Paper*.

Using CRC error detection for the Stratix III family has no impact on fitting or performance of your device.

Error Detection Fundamentals

Error detection determines if the data received through a medium is corrupted during transmission. To accomplish this, the transmitter uses a function to calculate a checksum value for the data and appends the checksum to the original data frame. The receiver uses the same calculation methodology to generate a checksum for the received data frame and compares the received checksum to the transmitted checksum. If the two checksum values are equal, the received data frame is correct and no data corruption occurred during transmission or storage.

The error detection CRC feature uses the same concept. When Stratix III devices have been configured successfully and are in user mode, the error detection CRC feature ensures the integrity of the configuration data.

There are two CRC error checks. One always runs during configuration, the second optional CRC error check runs in the background in user mode. Both CRC error checks use the same CRC polynomial but different error detection implementations.

For more information, refer to "Configuration Error Detection" and "User Mode Error Detection".

Configuration Error Detection

In configuration mode, a frame-based CRC is stored within the configuration data and contains the CRC value for each data frame.

During configuration, the Stratix III device calculates the CRC value based on the frame of data that is received and compares it against the frame CRC value in the data stream. Configuration continues until either the device detects an error or configuration is complete.

In Stratix III devices, the CRC value is calculated during the configuration stage. A parallel CRC engine generates 16 CRC check bits per frame and stores them into CRAM. The CRAM chain used for storing CRC check bits is 16 bits wide; its length is equal to the number of frames in the device.

User Mode Error Detection

Stratix III devices have built-in error detection circuitry to detect data corruption by soft errors in the CRAM cells. This feature allows all CRAM contents to be read and verified to match a configuration-computed CRC value. Soft errors are changes in a CRAM's bit state due to an ionizing particle.

The error detection capability continuously computes the CRC of the configured CRAM bits and compares it with the pre-calculated CRC. If the CRCs match, there is no error in the current configuration CRAM bits. The process of error detection continues until the device is reset (by setting `nCONFIG` low).

As soon as the device transitions into user mode, you can enable the error detection process if you enable the CRC error detection option. The internal 100-MHz configuration oscillator is divided down by a factor of 2 to 256 (at powers of 2) to be used as the clock source during the error detection process. Set the clock divide factor in the option setting in the Quartus II software.

A single 16-bit CRC calculation is done on a per-frame basis. Once it has finished the CRC calculation for a frame, the resulting 16-bit signature is `hex 0000` if there are no detected CRAM bit errors in a frame by the error detection circuitry and the output signal `CRC_ERROR` is `0`. If a CRAM bit error is detected by the circuitry within a frame in the device, the resulting signature is non-zero. This causes the CRC engine to start searching the error bit location.

Error detection in Stratix III devices calculates CRC check bits for each frame and pulls the `CRC_ERROR` pin high when it detects bit errors in the chip. Within a frame, it can detect all single-bit, double-bit, and three-bit errors. The probability of more than three CRAM bits being flipped by an SEU event is very low. In general, for all error patterns the probability of detection is 99.998%.

The CRC engine reports the bit location and determines the type of error for all single-bit errors and over 99.641% of double-adjacent errors. The probability of other error patterns is very low and the report of the bit flips error location is not guaranteed by the CRC engine.

You can also read-out the error bit location through the Joint Test Action Group (JTAG) and the core interface. You must shift these bits out through either the JTAG instruction, `SHIFT_EDERROR_REG`, or the core interface before the CRC detects the next error in another frame. If the next frame also has an error, you have to shift these bits out within the amount of time of one frame CRC verification. You can choose to extend this time interval by slowing down the error detection clock frequency, but this slows down the error recovery time for the SEU event. Refer to Table 15–6 on page 15–10 for the minimum update interval for Stratix III devices. If these bits are not shifted out before the next error location is found, the previous error location and error message is overwritten by the new information. The CRC circuit continues to run, and if an error is detected, you must decide whether to complete a reconfiguration or to ignore the CRC error.

The error detection logic continues to calculate the `CRC_ERROR` and 16-bit signatures for the next frame of data regardless if any error has occurred in the current frame or not. You must monitor these signals and take the appropriate actions if a soft error occurs.

Error detection circuitry in Stratix III devices uses a 16-bit CRC-ANSI standard (16-bit polynomial) as the CRC generator.

The computed 16-bit CRC signature for each frame is stored in registers within the core. The total storage register size is 16 (number of bits per frame) × the number of frames.

The Stratix III device error detection feature does not check memory blocks and I/O buffers. These memory blocks support parity bits that are used to check the contents of memory blocks for any error. The I/O buffers are not verified during error detection because these bits use flip-flops as storage elements that are more resistant to soft errors compared to CRAM cells.

The M144K TriMatrix memory block has a built-in error correction code block that checks and corrects errors in the block. However, for logic array blocks (LABs) that are used as MLAB memory blocks, they are ignored during error detection verification. Thus, the `CRC_ERROR` signal may stay solid high or low depending on the error status of the previous checked CRAM frame.

For more information about error detection in the Stratix III TriMatrix memory blocks, refer to the *TriMatrix Embedded Memory Blocks in Stratix III Devices* chapter.

In order to provide testing capability of the error detection block, a JTAG instruction `EDERROR_INJECT` is provided. This instruction can change the content of the 21-bit JTAG fault injection register, used for error injection in Stratix III devices, hence enabling testing of the error detection block.

You can only execute the `EDERROR_INJECT` JTAG instruction when the device is in user mode.

Table 15–1 lists the `EDERROR_INJECT` JTAG instruction.

Table 15–1. EDERROR_INJECT JTAG Instruction

JTAG Instruction	Instruction Code	Description
`EDERROR_INJECT`	00 0001 0101	This instruction controls the 21-bit JTAG fault injection register, which is used for error injection.

You can only execute the `EDERROR_INJECT` JTAG instruction at error detection frequency 50 MHz. Refer to "Error Detection Timing" on page 15–9 for instructions about how to set the error detection frequency in the Quartus II software. For the testing of the CRC detection block with the frequency lower than 50 MHz, contact Altera Technical Support at www.altera.com/support.

You can create Jam™ files (**.jam**) to automate the testing and verification process. This allows you to verify the CRC functionality in-system, on-the-fly, without having to reconfigure the device. You can then switch to the CRC circuit to check for real errors induced by an SEU.

You can introduce a single error, double errors, or double errors adjacent to each other to the configuration memory. This provides an extra way to facilitate design verification and system fault tolerance characterization. Use the JTAG fault injection register with `EDERROR_INJECT` instruction to flip the readback bits. The Stratix III device is then forced into error test mode.

The content of the JTAG fault injection register is not loaded into the fault injection register during the processing of the last and the first frame. It is only loaded at the end of this period.

You can only introduce error injection in the first data frame, but you can monitor the error information at any time.

For more information about the JTAG fault injection register and fault injection register, refer to"Error Detection Registers" on page 15–7.

Table 15–2 lists how the fault injection register is implemented and describes error injection.

Table 15–2. Fault Injection Register and Error Injection

Bit	Bit[20..19]			Bit[18..8]	Bit[7..0]
Description	**Error Type**			**Byte Location of the Injected Error**	**Error Byte Value**
Content	Error Type *(1)*		Error Injection Type	Depicts the location of the injected error in the first data frame.	Depicts the location of the bit error and corresponds to the error injection type selection.
	Bit[20]	Bit[19]			
	0	1	Single byte error injection		
	1	0	Double-adjacent byte error injection		
	0	0	No error injection		

Note to Table 15–2:

(1) Bit[20] and Bit[19] cannot both be set to 1 as this is not a valid selection. The error detection circuitry decodes it as no error injection.

After the test completes, Altera recommends that you reconfigure the device.

Automated Single Event Upset Detection

Stratix III devices offer on-chip circuitry for automated checking of single-event upset detection. Some applications that require the device to operate error-free in high-neutron flux environments require periodic checks to ensure continued data integrity. The error detection CRC feature ensures data reliability and is one of the best options for mitigating SEU.

You can implement the error detection CRC feature with existing circuitry in Stratix III devices, eliminating the need for external logic. The CRC_ERROR pin reports a soft error when configuration CRAM data is corrupted and you would have to decide whether to reconfigure the device or to ignore the error.

Error Detection Pin Description

Depending on the type of error detection feature you choose, you will need to use different error detection pins to monitor the data during user mode. The different error detection pins available are described in the following sections.

CRC_ERROR Pin

Table 15–3 lists the CRC_ERROR pin.

Table 15–3. CRC_ERROR Pin Description

Pin Name	Pin Type	Description
CRC_ERROR	I/O, output, or open-drain output (optional)	Active high signal that indicates the error detection circuit has detected errors in the configuration CRAM bits. This pin is optional and is used when the error detection CRC circuit is enabled. When the error detection CRC circuit is disabled, it is a user I/O pin. The CRC error output, when using the WYSIWYG function, is a dedicated path to the CRC_ERROR pin. By default, the Quartus II software sets the CRC_ERROR pin as a dedicated output. If CRC_ERROR is used as a dedicated output, make sure V_{CCIO} of the bank where the pin resides meets the input voltage specification of the system receiving the signal. Optionally, you can set this pin to be an open-drain output by enabling the option in the Quartus II software from the **Error Detection CRC** tab of the **Device & Pin Options** dialog box. Using this pin as open-drain provides advantage on voltage leveling. To use this pin as open-drain, tie the pin to V_{CCPGM} through a 10-kΩ resistor. Alternatively, depending on the voltage input voltage specification of the system receiving the signal, you can tie the pull-up resistor to a different pull-up voltage.

WYSIWYG is a design primitive that corresponds to device features and can be directly instantiated into your RTL design.

The CRC_ERROR pin information for Stratix III devices is reported in *Device Pin-Outs* on the Literature page of the Altera website (www.altera.com).

Error Detection Block

You can enable the Stratix III device error detection block in the Quartus II software (refer to"Software Support" on page 15–11). This block contains the logic necessary to calculate the 16-bit CRC signature for the configuration CRAM bits in the device.

The CRC circuit continues running even if an error occurs. When a soft error occurs, the device sets the `CRC_ERROR` pin high. Two types of CRC detection check the configuration bits:

- The CRAM error checking ability (16-bit CRC) during user mode, for use by the `CRC_ERROR` pin.
  - For each frame of data, the pre-calculated 16-bit CRC enters the CRC circuit right at the end of the frame data and determines whether or not there is an error.
  - If an error occurs, the search engine starts to find the location of the error.
  - You can shift the error messages out through the JTAG instruction or core interface logic while the error detection block continues running.
  - The JTAG interface reads out the 16-bit CRC result for the first frame and also shifts the 16-bit CRC bits to the 16-bit CRC storage registers for test purposes.
  - You can deliberately introduce single error, double errors, or double errors adjacent to each other to configuration memory for testing and design verification.

The "Error Detection Registers" section focuses on the first type, the 16-bit CRC only when the device is in user mode.

- The 16-bit CRC that is embedded in every configuration data frame.
  - During configuration, after a frame of data is loaded into the Stratix III device, the pre-computed CRC is shifted into the CRC circuitry.
  - At the same time, the CRC value for the data frame shifted-in is calculated. If the pre-computed CRC and calculated CRC values do not match, `nSTATUS` is set low. Every data frame has a 16-bit CRC; therefore, there are many 16-bit CRC values for the whole configuration bitstream. Every device has different lengths of the configuration data frame.

Error Detection Registers

There is one set of 16-bit registers in the error detection circuitry that stores the computed CRC signature. A non-zero value on the syndrome register causes the `CRC_ERROR` pin to be set high. Figure 15–1 shows the block diagram of the error detection circuitry, syndrome registers, and error injection block.

Figure 15–1. Error Detection Block Diagram

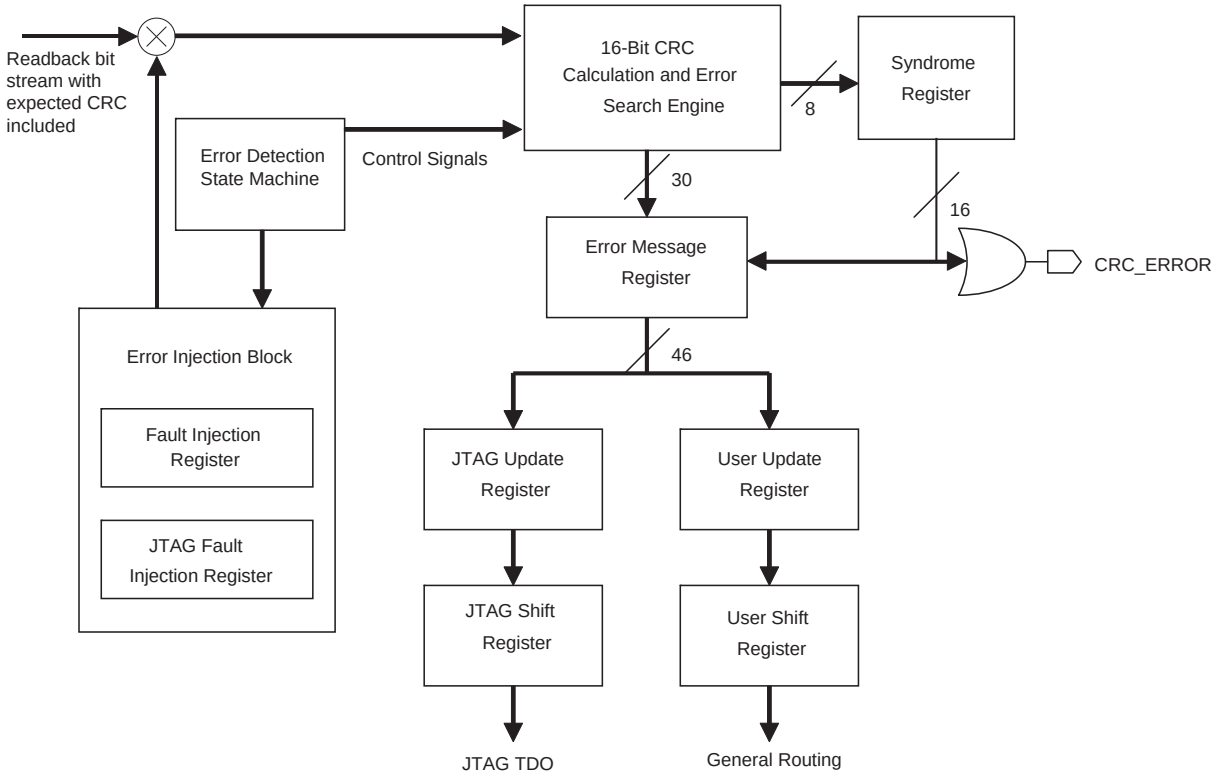


Table 15–5 lists the registers shown in Figure 15–1.

Table 15–4. Error Detection Registers (Part 1 of 2)

Register	Description
Syndrome Register	This register contains the CRC signature of the current frame through the error detection verification cycle. The `CRC_ERROR` signal is derived from the contents of this register.
Error Message Register	This 46-bit register contains information about the error type, location of the error, and the actual syndrome. The types of errors and location reported are single and double adjacent bit errors. The location bits for other types of errors are not identified by the Error Message Register. You can shift out the content of the register through the JTAG `SHIFT_EDERROR_REG` instruction or to the core through the core interface.
JTAG Update Register	This register is automatically updated with the contents of the Error Message Register one cycle after the 46-bit register content is validated. It includes a clock enable which needs to be asserted prior to being sampled into the JTAG Shift Register. This requirement ensures that the JTAG Update Register is not being written into by the contents of the Error Message Register at exactly the same time that the JTAG Shift Register is reading its contents.
User Update Register	This register is automatically updated with the contents of the Error Message Register, one cycle after the 46-bit register content is validated. It includes a clock enable which needs to be asserted prior to being sampled into the User Shift Register. This requirement ensures that the User Update Register is not being written into by the contents of the Error Message Register at exactly the same time that the User Shift Register is reading its contents.

Table 15–4. Error Detection Registers (Part 2 of 2)

Register	Description
JTAG Shift Register	This register is accessible by the JTAG interface and allows the contents of the JTAG Update Register to be sampled and read out by the JTAG instruction `SHIFT_EDERROR_REG`.
User Shift Register	This register is accessible by the core logic and allows the contents of the User Update Register to be sampled and read by the user logic.
JTAG Fault Injection Register	This 21-bit register is fully controlled by the JTAG instruction `EDERROR_INJECT`. This register holds the information of the error injection that you want in the bitstream.
Fault Injection Register	The content of the JTAG Fault Injection Register is loaded in this 21-bit register when it is being updated.

Error Detection Timing

When the CRC feature is enabled through the Quartus II software, the device automatically activates the CRC process upon entering user mode, after configuration, and after initialization is complete.

If an error is detected within a frame, `CRC_ERROR` is driven high at the end of the error location search, and after the Error Message Register gets updated. At the end of this cycle, the `CRC_ERROR` pin is pulled low for a minimum 32 clock cycles. If the next frame also contains an error, the `CRC_ERROR` is driven high again after the Error Message Register gets overwritten by the new value. You can start to unload the error message on each rising edge of `CRC_ERROR` pin. The error detection runs until the device is reset.

Error detection circuitry runs off an internal configuration oscillator with a divisor that sets the maximum frequency. Table 15–5 lists the minimum and maximum error detection frequencies.

Table 15–5. Minimum and Maximum Error Detection Frequencies

Device Type	Error Detection Frequency	Maximum Error Detection Frequency	Minimum Error Detection Frequency	Valid Exponents (*n*)
Stratix III	100 MHz / 2^n	50 MHz	390 kHz	1, 2, 3, 4, 5, 6, 7, 8

You can set a lower clock frequency by specifying a division factor in the Quartus II software (refer to "Software Support" on page 15–11). The divisor is a power of two (2), where *n* is between 1 and 8. The divisor ranges from 2 through 256. Refer to Equation 15–1.

Equation 15–1.

$$\text{Error detection frequency} = \frac{100\,MHz}{2^n}$$

The error detection frequency reflects the frequency of the error detection process for a frame because the CRC calculation in Stratix III devices is done on a per-frame basis.

You must monitor the error message to avoid missing information in the Error Message Register. The Error Message Register is updated whenever an error occurs. The minimum interval time between each update for the Error Message Register depends on the device and the error detection clock frequency. Table 15–6 lists the estimated minimum interval time between each update for the Error Message Register for Stratix III devices.

Table 15–6. Minimum Update Interval for Error Message Register *(Note 1)*

Device	Timing Interval (µs)
EP3SL50	9.8
EP3SL70	9.8
EP3SL110	14.8
EP3SL150	14.8
EP3SL200	19.8
EP3SE260	19.8
EP3SL340	21.8
EP3SE50	9.8
EP3SE80	14.8
EP3SE110	14.8

Note to Table 15–6:

(1) These timing numbers are preliminary.

The CRC calculation time for the error detection circuitry to check from the first until the last frame depends on the device and the error detection clock frequency.

Table 15–7 lists the estimated time for each CRC calculation with minimum and maximum clock frequencies for Stratix III devices. The minimum CRC calculation time is calculated by using the maximum error detection frequency with divisor factor 1 while the maximum CRC calculation time is calculated by using the minimum error detection frequency with divisor factor 8.

Table 15–7. CRC Calculation Time

Device	Minimum Time (ms)	Maximum Time (s)
EP3SL50	52.00	14.36
EP3SL70	52.00	14.36
EP3SL110	110.00	30.38
EP3SL150	110.00	30.38
EP3SL200	212.00	58.72
EP3SL260	212.00	58.72
EP3SL340	270.00	74.87
EP3SE50	59.00	16.41
EP3SE80	113.00	31.28
EP3SE110	113.00	31.28

Software Support

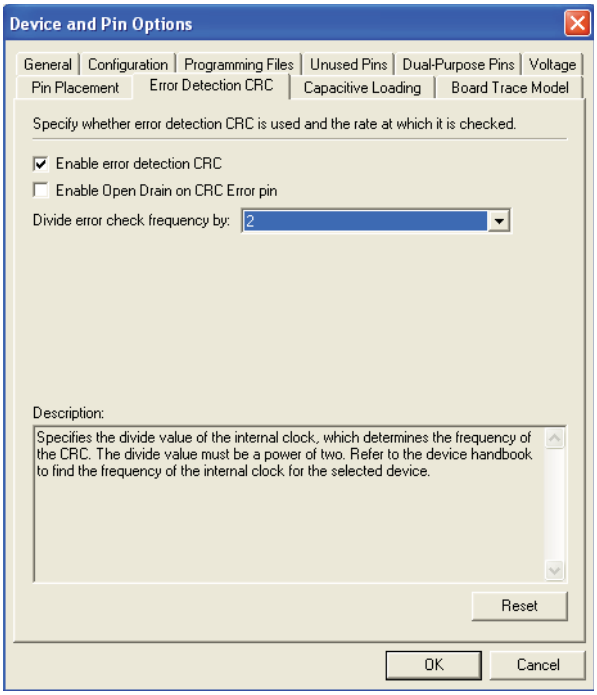
The Quartus II software, starting with version 6.1, supports the error detection CRC feature for Stratix III devices. Enabling this feature generates the `CRC_ERROR` output to the optional dual purpose `CRC_ERROR` pin.

The error detection CRC feature is controlled by the **Device and Pin Options** dialog box in the Quartus II software.

Enable the error detection feature using CRC by performing the following steps:

1. Open the Quartus II software and load a project that uses a Stratix III device.
2. On the Assignments menu, click **Settings**. The **Settings** dialog box is shown.
3. In the **Category** list, select **Device**. The **Device** page is shown.
4. Click **Device and Pin Options**. The **Device and Pin Options** dialog box is shown (Figure 15–2).
5. In the **Device and Pin Options** dialog box, click the **Error Detection CRC** tab.
6. Turn on **Enable error detection CRC** (Figure 15–2).

Figure 15–2. Enabling the Error Detection CRC Feature in the Quartus II Software



7. In the **Divide error check frequency by** box, enter a valid divisor as documented in Table 15–5 on page 15–9.

The divide value divides the frequency of the configuration oscillator output clock that clocks the CRC circuitry.

8. Click **OK**.

Recovering From CRC Errors

The system that contains the Stratix III device must control the device reconfiguration. After detecting an error on the `CRC_ERROR` pin, strobing the `nCONFIG` signal low directs the system to perform the reconfiguration at a time when it is safe for the system to reconfigure the device.

When the data bit is rewritten with the correct value by reconfiguring the device, the device functions correctly.

While soft errors are uncommon in Altera devices, certain high-reliability applications may require a design to account for these errors.

Chapter Revision History

Table 15–8 lists the revision history for this chapter.

Table 15–8. Chapter Revision History

Date	Version	Changes Made
March 2010	1.7	Updated for the Quartus II software version 9.1 SP2 release: ■ Updated Table 15–6. ■ Minor text edits.
May 2009	1.6	Updated “User Mode Error Detection” and “CRC_ERROR Pin” sections.
February 2009	1.5	■ Updated “Error Detection Timing” section. ■ Removed “Referenced Documents”, Critical Error Detection”, and “CRITICAL ERROR Pin” sections.
October 2008	1.4	■ Updated “Introduction” and “Referenced Documents” sections. ■ Updated New Document Format.
May 2008	1.3	■ Updated “Configuration Error Detection”, “User Mode Error Detection”, and “Error Detection Timing” sections. ■ Updated Table 15–3, Table 15–6, and Table 15–7. ■ Updated Figure 15–2 and Figure 15–3.
October 2007	1.2	■ Minor edits to Table 15–3. ■ Added new section “Referenced Documents”. ■ Added live links for references.
May 2007	1.1	■ Minor edits to page 2, 3, 4, and 14. ■ Updated Table 15–5.
November 2006	1.0	Initial Release.

Section V. Power and Thermal Management

This section provides information on Power and Thermal Management for the Stratix® III devices.

- Chapter 16, Programmable Power and Temperature-Sensing Diodes in Stratix III Devices

Revision History

Refer to each chapter for its own specific revision history. For information on when each chapter was updated, refer to the Chapter Revision Dates section, which appears in the full handbook.

16. Programmable Power and Temperature-Sensing Diodes in Stratix III Devices

SIII51016-1.5

Introduction

The total power of an FPGA includes static power and dynamic power. Static power is the power consumed by the FPGA when it is programmed but no clocks are operating. Dynamic power is comprised of switching power when the device is configured and running. Dynamic power is calculated with the Equation 16–1:

Equation 16–1. Dynamic Power Equation

$$P = \frac{1}{2}CV^2 \times \text{frequency} \times \text{toggle rate}$$

Equation 16–1 shows that the frequency and toggle rate are design-dependent. However, voltage can be varied to lower dynamic power consumption by the square value of the voltage difference. Stratix® III devices minimize static and dynamic power with advanced process optimizations, selectable core voltage, and programmable power technology. These technologies enable Stratix III designs to optimally meet design-specific performance requirements with the lowest possible power.

The Quartus® II software optimizes all designs with Stratix III power technology to ensure performance is met at the lowest power consumption. This automatic process allows you to concentrate on the functionality of your design, instead of the power consumption of the design.

Power consumption also affects thermal management. Stratix III offers a temperature sensing diode (TSD), which you can use with external circuitry to monitor the device junction temperature for activities such as controlling air flow to the FPGA.

Stratix III Power Technology

The following section provides details about Stratix III selectable core voltage and programmable power technology.

Selectable Core Voltage

Altera offers a series of low-voltage Stratix products that have the ability to power the core logic of the device with either a 0.9-V or 1.1-V power supply. This power supply, called V_{CCL}, powers the logic array block (LAB), memory logic array block (MLAB), digital signal processing (DSP) blocks, TriMatrix™ memory blocks, clock networks, and routing lines. The periphery, consisting of the I/O registers and their routing connections are powered by V_{CC} with a 1.1-V power supply. You can use the same 1.1-V power supply if you want both V_{CC} and V_{CCL} to be 1.1 V.

Lowering the core voltage reduces both static and dynamic power, but causes a reduction in performance. You need to set the correct core supply voltage in the Quartus II software settings under **Operating Conditions**, since the Quartus II software analyzes the core power consumption and timing delays based on this selection. When you compile a design, you can select either 0.9-V or 1.1-V core voltage. You can compare the power and performance trade-offs of a 0.9-V core voltage compilation result and a 1.1-V core voltage compilation result and then choose the most desirable core voltage for your design. By default, the Quartus II software sets the core voltage to 1.1 V.

Ensure that the board has a separate 0.9-V power supply to utilize the lower voltage option and be sure to connect V_{CCL} to the voltage level that you set in the Quartus II software. The Stratix III device cannot distinguish which core voltage level is used on the board. Connecting to the wrong voltage level gives you different timing delays and power consumption than what is reported by the Quartus II software.

For information about selectable core voltage performance and power effects on sample designs, refer to *AN 437: Power Optimization Techniques*.

Programmable Power Technology

In addition to the variable core voltage, Stratix III devices also offer the ability to configure portions of the core, called tiles, for high-speed or low-power mode of operation performed by the Quartus II software without user intervention. This programmable power technology, used to reduce static power, uses an on-chip voltage regulator powered by V_{CCPT}. In a design compilation, the Quartus II software determines whether a tile needs to be in high-speed or low-power mode based on the timing constraints of the design.

For more information about how the Quartus II software uses programmable power technology when compiling a design, refer to *AN 437: Power Optimization Techniques*.

A Stratix III tile can consist of the following:

- MLAB/LAB pairs with routing to the pair
- MLAB/LAB pairs with routing to the pair and to adjacent DSP/memory block routing
- TriMatrix memory blocks
- DSP blocks
- I/O interfaces

All blocks and routing associated with the tile share the same setting of either high speed or low power. Tiles that include DSP blocks, memory blocks, or I/O interfaces are set to high-speed mode by default for optimum performance when used in the design. Unused DSP blocks, memory blocks, and I/O elements are set to low-power mode to minimize static power. Clock networks do not support programmable power technology.

With programmable power technology, faster speed grade FPGAs may require less power, as there are fewer high-speed MLAB and LAB pairs, compared to slower speed grade FPGAs. The slower speed grade device may need to use more high-speed MLAB and LAB pairs to meet the performance requirements, while the faster speed grade device can meet the performance requirements with MLAB and LAB pairs in low-power mode.

The Quartus II software sets unshared inputs and unused device resources in the design to low-power mode to reduce static and dynamic power. The Quartus II software sets the following resources to low power when they are not used in the design:

- LABs and MLABs
- TriMatrix memory blocks
- External memory interface circuitry
- DSP blocks
- phase-locked loop (PLL)
- serializer/deserializer (SERDES) and DPA blocks

If the PLL is instantiated in the design, asserting a reset high keeps the PLL in low power.

Relationship Between Selectable Core Voltage and Programmable Power Technology

Table 16–1 shows available Stratix III programmable power capabilities. You can speed grade considerations to the permutations to give you flexibility in designing your system.

Table 16–1. Stratix III Programmable Power Capabilities

	Selectable Core Voltage	Programmable Power Technology
LAB	Yes	Yes
Routing	Yes	Yes
Memory Blocks	Yes	Fixed setting *(1)*
DSP Blocks	Yes	Fixed setting *(1)*
Global Clock Networks	Yes	No
I/O Elements (IOE)	No	Fixed setting *(1)*

Note to Table 16–1:

(1) Tiles with DSP blocks, memory blocks, and I/O elements that are used in the design are always set to high-speed mode. Unused DSP blocks, memory blocks, and I/O interfaces are set to low-power mode by default.

Stratix III External Power Supply Requirements

This section describes the different external power supplies needed to power Stratix III devices. Table 16–2 lists the external power supply pins for Stratix III devices. Some of the power supply pins can be supplied with the same external power supply, provided they need the same voltage level, as noted in the recommended board connection column.

For possible values of each power supply, refer to the *DC and Switching Characteristics of Stratix III Devices* chapter in volume 2 of the *Stratix III Device Handbook*.

For detailed guidelines about how to connect and isolate VCCL and VCC power supply pins, refer to the *Stratix III Device Family Pin Connections Guidelines*.

Table 16–2. Stratix III Power Supply Requirements

Power Supply Pin	Recommended Board Connection	Description
VCCL	VCCL	Selectable core voltage power supply
VCC	VCC	I/O registers power supply
VCCD_PLL	VCCD_PLL	PLL digital power supply
VCCA_PLL	VCCA_PLL *(1)*	PLL analog power supply
VCCPT		Power supply for programmable power technology
VCCPGM	VCCPGM	Configuration pins power supply
VCCPD	VCCPD *(2)*	I/O pre-driver power supply
VCCIO	VCCIO *(3)*	I/O power supply
VCC_CLKIN		Differential clock input pins power supply (top and bottom I/O banks only)
VCCBAT	VCCBAT	Battery back-up power supply for design security volatile key register
VREF	VREF *(4)*	Power supply for voltage-referenced I/O standards
GND	GND	Ground

Notes to Table 16–2:

(1) You can minimize the number of external power sources by driving the left column and supplies with the same voltage regulator. Note that separate power planes, decoupling capacitors, and ferrite beads are required for VCCA_PLL and VCCPT when implementing this scheme.

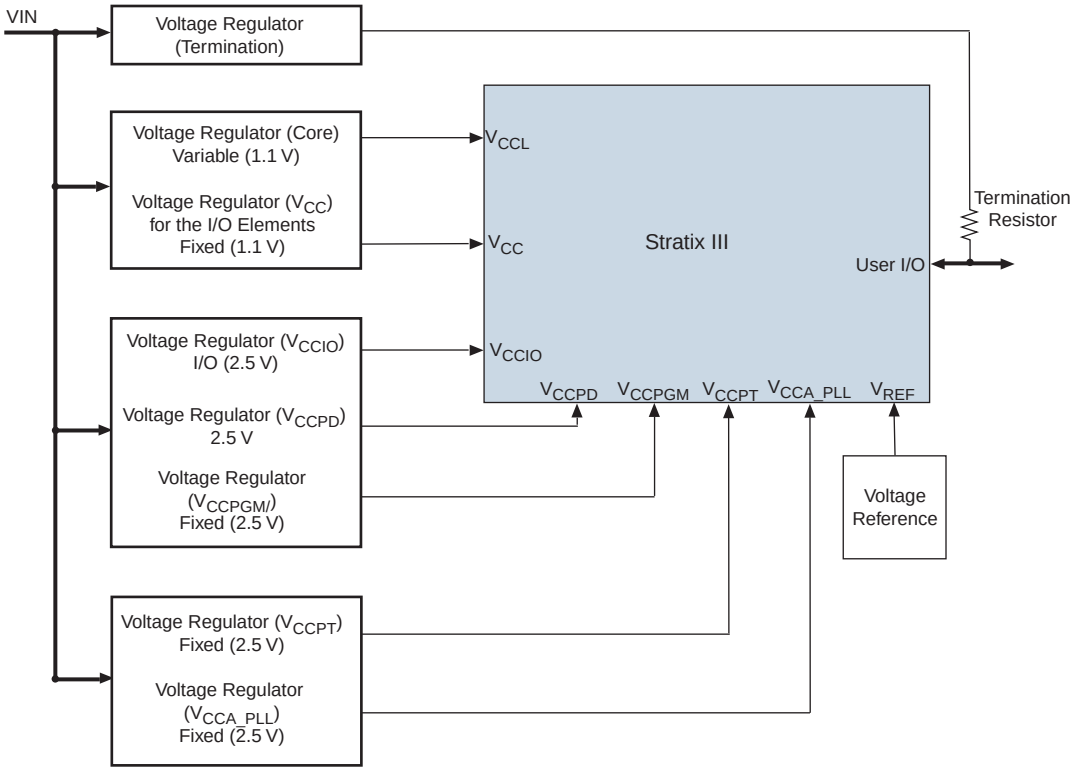
(2) V_{CCPD} can be either 2.5 V, 3.0 V, or 3.3 V. For a 3.3-V standard, V_{CCPD} = 3.3 V. For a 3.0-V I/O standard, V_{CCPD} = 3.0 V. For 2.5 V and below I/O standards, V_{CCPD} = 2.5 V.

(3) This scheme is for VCCIO = 2.5 V.

(4) There is one VREF pin per I/O bank. Use an external power supply or a resistor divider network to supply this voltage.

Figure 16–1 shows an example of power management for Stratix III devices.

Figure 16–1. Stratix III Power Management Example *(Note 1)*, *(2)*



Notes to Figure 16–1:

(1) When V_{CCL} = 0.9 V, you need a separate voltage regulator.

(2) When V_{CCL} = 0.9 V, V_{CCPT} and V_{CC} must be ramped before V_{CCL} to minimize V_{CCL} standby current during V_{CCPT} and V_{CC} ramping to full rail.

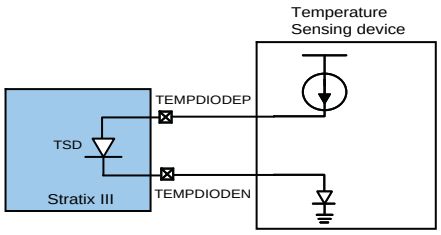
Temperature Sensing Diode

Knowing the junction temperature is crucial for thermal management. A Stratix III device monitors its die temperature with an embedded temperature sensing diode (TSD). This is done by sensing the voltage level across the TSD. Each temperature level produces a unique voltage across the diode. Use an external analog-to-digital converter that measures the voltage difference across the TSD and then converts it to a temperature reading.

External Pin Connections

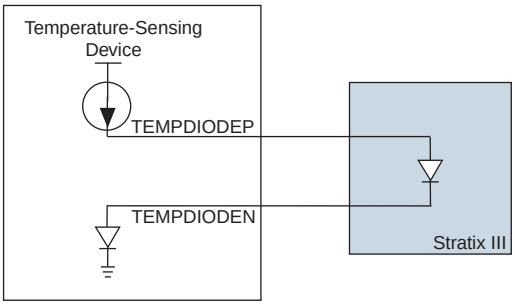
The Stratix III TSD, located in the top-right corner of the die, requires two pins for voltage reference. Connect the `TEMPDIODEP` and `TEMPDIODEN` pins to the external analog-to-digital converter, as shown in Figure 16–2.

Figure 16–2. TEMPDIODEP and TEMPDIODEN External Pin Connections



The TSD is a very sensitive circuit which can be influenced by the noise coupled from traces on the board, and possibly within the device package itself, depending on device usage. The interfacing device registers temperature based on milivolts of difference as seen at the TSD. Switching I/O near the TSD pins can affect the temperature reading. Altera recommends you take temperature readings during periods of no activity in the device (for example, standby mode where no clocks are toggling in the device), such as when the nearby I/Os are at a DC state and the clock networks in the device are disabled.

Figure 16–3. TSD Connections



Conclusion

As process geometries get smaller, power and thermal management is becoming more crucial in FPGA designs. Stratix III devices offer programmable power technology and selectable core voltage options for low-power operation. Use these features, along with speed grade choices, in different permutations to get the best power and performance combination. Taking advantage of the silicon, the Quartus II software is able to manipulate designs to use the best combination to achieve the lowest power at the required performance.

For thermal management, use the Stratix III temperature sensing diode with an external analog-to-digital converter in production devices. This allows you to easily incorporate this feature in your designs. Being able to monitor the junction temperature of the device at any time also allows you to control air flow to the device and save power for the whole system.

Chapter Revision History

Table 16–3 shows the revision history for this document.

Table 16–3. Chapter Revision History

Date and Revision	Changes Made	Summary of Changes
February 2009, version 1.5	Removed “Referenced Documents” section.	—
October 2008, version 1.4	■ Updated “Introduction”, “Temperature Sensing Diode”, “External Pin Connections”, and “Conclusion” sections. ■ Updated new Document Format.	—
May 2008, version 1.3	■ Updated Figure 16–1. ■ Updated Table 16–2. ■ Updated “External Pin Connections” section.	—
October 2007, version 1.2	■ Added material to note 3 of Table 16–2. ■ Updated Figure 16–1 and Figure 16–3. ■ Removed old version of Figure 16-2. ■ Removed section “Architecture Description”. ■ Removed material from the sections “Introduction”, “Temperature Sensing Diode”, “External Pin Connections”, and “Conclusion”. ■ Added new section “Referenced Documents”. ■ Added live links for references.	Minor update.
May 2007, version 1.1	Replaced all instances of VCCR with VCCPT	Minor update.
November 2006, version 1.0	Initial Release.	—

Section VI. Packaging Information

This section provides packaging information for the Stratix® III device.

- Chapter 17, Stratix III Device Packaging Information

Revision History

Refer to each chapter for its own specific revision history. For information on when each chapter was updated, refer to the Chapter Revision Dates section, which appears in the full handbook.

17. Stratix III Device Packaging Information

SIII51017-1.7

This chapter provides thermal resistance values and package information for Altera® Stratix® III devices, including:

- "Thermal Resistance" on page 17–2
- "Package Outlines" on page 17–2

Table 17–1 lists which Stratix III device, are available in FineLine BGA or Hybrid FineLine BGA packages.

Table 17–1. FineLine and Hybrid FineLine BGA Packages for Stratix III Devices

Device	Package	Pins
EP3SL50	FineLine BGA - Flip Chip (Option 1)	484
	FineLine BGA - Flip Chip (Option 1)	780
EP3SL70	FineLine BGA - Flip Chip (Option 1)	484
	FineLine BGA - Flip Chip (Option 1)	780
EP3SL110	FineLine BGA - Flip Chip (Option 1)	780
	FineLine BGA - Flip Chip (Option 1)	1152
EP3SL150	FineLine BGA - Flip Chip (Option 1)	780
	FineLine BGA - Flip Chip (Option 1)	1152
EP3SL200	Hybrid FineLine BGA - Flip Chip (Option 1)	780
	FineLine BGA - Flip Chip (Option 1)	1152
	FineLine BGA - Flip Chip (Option 1)	1517
EP3SL340	Hybrid FineLine BGA - Flip Chip (Option 1)	1152
	FineLine BGA - Flip Chip (Option 1)	1517
	FineLine BGA - Flip Chip (Option 1)	1760
EP3SE50	FineLine BGA - Flip Chip (Option 1)	484
	FineLine BGA - Flip Chip (Option 1)	780
EP3SE80	FineLine BGA - Flip Chip (Option 1)	780
	FineLine BGA - Flip Chip (Option 1)	1152
EP3SE110	FineLine BGA - Flip Chip (Option 1)	780
	FineLine BGA - Flip Chip (Option 1)	1152
EP3SE260	Hybrid FineLine BGA - Flip Chip (Option 1)	780
	FineLine BGA - Flip Chip (Option 1)	1152
	FineLine BGA - Flip Chip (Option 1)	1517

Thermal Resistance

For Stratix III devices thermal resistance specifications, refer to the *Stratix Series Device Thermal Resistance Data Sheet*.

Package Outlines

You can download Stratix III device package outlines from the *Device Packaging Specifications* web page.

Chapter Revision History

Table 17–2 lists the revision history for this chapter.

Table 17–2. Chapter Revision History

Date	Version	Changes Made
March 2010	1.7	Updated for the Quartus II software version 9.1 SP2 release: ■ Updated Table 17–1. ■ Minor text edits.
February 2009	1.6	Removed “Referenced Documents” section.
October 2008	1.5	Updated New Document Format.
May 2008	1.4	Updated “Package Outlines” section hyperlink.
November 2007	1.3	Updated Table 17–1.
October 2007	1.2	■ Added new section “Referenced Documents”. ■ Added live links for references.
May 2007	1.1	Removed thermal resistance and package outline information and replaced with links referencing this information.
November 2006	1.0	Initial Release.

© March 2010 Altera Corporation

Компания «ЭлектроПласт» предлагает заключение долгосрочных отношений при поставках импортных электронных компонентов на взаимовыгодных условиях!

Наши преимущества:

- Оперативные поставки широкого спектра электронных компонентов отечественного и импортного производства напрямую от производителей и с крупнейших мировых складов;
- Поставка более 17-ти миллионов наименований электронных компонентов;
- Поставка сложных, дефицитных, либо снятых с производства позиций;
- Оперативные сроки поставки под заказ (от 5 рабочих дней);
- Экспресс доставка в любую точку России;
- Техническая поддержка проекта, помощь в подборе аналогов, поставка прототипов;
- Система менеджмента качества сертифицирована по Международному стандарту ISO 9001;
- Лицензия ФСБ на осуществление работ с использованием сведений, составляющих государственную тайну;
- Поставка специализированных компонентов (Xilinx, Altera, Analog Devices, Intersil, Interpoint, Microsemi, Aeroflex, Peregrine, Syfer, Eurofarad, Texas Instrument, Miteq, Cobham, E2V, MA-COM, Hittite, Mini-Circuits,General Dynamics и др.);

Помимо этого, одним из направлений компании «ЭлектроПласт» является направление «Источники питания». Мы предлагаем Вам помощь Конструкторского отдела:

- Подбор оптимального решения, техническое обоснование при выборе компонента;
- Подбор аналогов;
- Консультации по применению компонента;
- Поставка образцов и прототипов;
- Техническая поддержка проекта;
- Защита от снятия компонента с производства.

Как с нами связаться

Телефон: 8 (812) 309 58 32 (многоканальный)
Факс: 8 (812) 320-02-42
Электронная почта: org@eplast1.ru
Адрес: 198099, г. Санкт-Петербург, ул. Калинина, дом 2, корпус 4, литера А.