I3C Controller IP

IP Version: v3.7.2

User Guide

FPGA-IPUG-02228-1.7

October 2026

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS, with all faults, and all associated risk is the responsibility entirely of the Buyer. The information provided herein is for informational purposes only and may contain technical inaccuracies or omissions, and may be otherwise rendered inaccurate for many reasons, and Lattice assumes no obligation to update or otherwise correct or revise this information. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. LATTICE PRODUCTS AND SERVICES ARE NOT DESIGNED, MANUFACTURED, OR TESTED FOR USE IN LIFE OR SAFETY CRITICAL SYSTEMS, HAZARDOUS ENVIRONMENTS, OR ANY OTHER ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, INCLUDING ANY APPLICATION IN WHICH THE FAILURE OF THE PRODUCT OR SERVICE COULD LEAD TO DEATH, PERSONAL INJURY, SEVERE PROPERTY DAMAGE OR ENVIRONMENTAL HARM (COLLECTIVELY, "HIGH-RISK USES"). FURTHER, BUYER MUST TAKE PRUDENT STEPS TO PROTECT AGAINST PRODUCT AND SERVICE FAILURES, INCLUDING PROVIDING APPROPRIATE REDUNDANCIES, FAIL-SAFE FEATURES, AND/OR SHUT-DOWN MECHANISMS. LATTICE EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS OF THE PRODUCTS OR SERVICES FOR HIGH-RISK USES. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Inclusive Language

This document was created consistent with Lattice Semiconductor’s inclusive language policy. In some cases, the language in underlying tools and other items may not yet have been updated. Please refer to Lattice’s inclusive language FAQ 6878 for a cross reference of terms. Note in some cases such as register names and state names it has been necessary to continue to utilize older terminology for compatibility.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Contents

Contents....3
Abbreviations in This Document....8
1. Introduction....10
1.1. Overview of the IP....10
1.2. Quick Facts....10
1.3. IP Support Summary....10
1.4. Features....11
1.5. Licensing and Ordering Information....11
1.6. Minimum Device Requirements....11
1.7. Naming Conventions....12
1.7.1. Nomenclature....12
1.7.2. Signal Names....12
1.7.3. Attribute Names....12
2. Functional Description....13
2.1. IP Architecture Overview....13
2.2. Clocking....14
2.2.1. Clocking Overview....15
2.3. Reset....15
2.4. User Interfaces....16
2.5. User Packet Frame Structure....16
2.6. Performing Various I3C Transfers (SDR)....18
2.6.1. I3C Private Write....18
2.6.2. I3C Private Read....18
2.6.3. I3C Consecutive Private Write....19
2.6.4. I3C Consecutive Private Write and Private Read....20
2.6.5. Enter Dynamic Address Assignment (DAA) CCC....21
2.6.6. Reset Dynamic Address (RSTDAA) CCC....22
2.6.7. Generic Broadcast CCC Format....22
2.6.8. Generic Direct CCC Format....23
2.6.9. Common Command Codes....23
2.7. Performing Various I2C Transfers....24
2.7.1. I2C Private Write....24
2.7.2. I2C Private Read....25
2.7.3. I2C Consecutive Private Write and Private Read....25
2.8. Performing HDR-DDR Mode Transfers....27
2.8.1. HDR-DDR Write....28
2.8.2. HDR-DDR Read....29
2.8.3. HDR-DDR Consecutive Write and Read....30
2.8.4. Permitted CCCs in HDR-DDR Mode....31
2.9. Hot-Join and In-Band Interrupt (IBI) Handling....38
2.10. Controller to Controller Handoff....40
2.11. Mixed Bus Mode....40
3. IP Parameter Description....41
3.1. General....41
3.2. Secondary Controller....43
3.3. IP Parameter Settings for Example Use Cases....44
4. Signal Description....46
5. Register Description....48
5.1. System Clock Divider Register 0x01....49
5.2. Controller Configuration 0 Register 0x02....49
5.3. Open Drain Timer Register 0x03....51

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

5.4. I2C Clock Divider Register 0x04 51
5.5. Pull Up Resistor Disable Register 0x05 51
5.6. HDR-DDR Configuration 0 Register 0x06 52
5.7. Soft Reset Register 0x08 53
5.8. Transmit Start Register 0x11 53
5.9. Secondary Controller Status Info Register 0x15 53
5.10. Set Device Role Register 0x16 54
5.11. Secondary Controller Timeout Register 0x17–0x19 54
5.12. Dynamic Address ACKed Counter Register 0x1C 55
5.13. IBI Read Count Register 0x1D 55
5.14. IBI Response Register 0x1E 55
5.15. IBI Address Register 0x1F 56
5.16. Interrupt Status 0 Register 0x20 56
5.17. Interrupt Set 0 Register 0x21 56
5.18. Interrupt Enable 0 Register 0x22 57
5.19. Interrupt Status 1 Register 0x24 57
5.20. Interrupt Set 1 Register 0x25 58
5.21. Interrupt Enable 1 Register 0x26 58
5.22. Bus Condition Debug Register 0x28 58
5.23. Last NAK Address Debug Register 0x29 59
5.24. Last ACK Address Debug Register 0x2A 59
5.25. Interrupt Status 2 Register 0x2C 60
5.26. Interrupt Set 2 Register 0x2D 60
5.27. Interrupt Enable 2 Register 0x2E 61
5.28. Tx FIFO Register 0x30 61
5.29. Rx FIFO Register 0x40 61
6. Example Design 62
6.1. Example Design Supported Configuration 62
6.2. Overview of the Example Design and Features 63
6.3. Design Components Example 64
6.4. Generating the Example Design 65
6.5. Using the Prebuilt Example Design for CertusPro-NX Device 68
6.6. Hardware Testing 68
6.6.1. Hardware Testing Setup 68
6.6.2. Expected Output 68
7. Designing with the IP 69
7.1. Generating and Instantiating the IP 69
7.1.1. Generated Files and File Structure 71
7.2. Design Implementation 72
7.3. Timing Constraints 72
7.4. Physical Constraints 72
7.5. Running Functional Simulation 72
8. Software Driver API 76
Appendix A. Resource Utilization 79
References 81
Technical Support Assistance 82
Revision History 83

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Figures

Figure 2.1. I3C Controller IP Core Functional Diagram (Controller Only) 13
Figure 2.2. I3C Controller IP Core Functional Diagram (with Secondary Controller Capability) 14
Figure 2.3. I3C Controller IP Clock Domain Block Diagram (Internal Clock Divider Enabled) 14
Figure 2.4. I3C Controller IP Clock Domain Block Diagram (Internal Clock Divider Disabled) 15
Figure 2.5. User Packet Frame Structure 16
Figure 2.6. I3C Private Write 18
Figure 2.7. I3C Private Read 18
Figure 2.8. I3C Consecutive Private Write 19
Figure 2.9. I3C Consecutive Private Write and Private Read 20
Figure 2.10. Enter Dynamic Address Assignment 21
Figure 2.11. Reset Dynamic Address 22
Figure 2.12. Generic Broadcast CCC Format 22
Figure 2.13. Generic Direct CCC Format 23
Figure 2.14. I2C Private Write 24
Figure 2.15. I2C Private Read 25
Figure 2.16. I2C Consecutive Private Write and Private Read 25
Figure 2.17. HDR-DDR Mode Frame 27
Figure 2.18. I3C Controller Handling of IBI/Hot-Join 38
Figure 2.19. Sample Sequence 39
Figure 6.1. I3C Controller IP in Propel SoC Project 63
Figure 6.2. Sample C-Code Test Routine 64
Figure 6.3. I3C Controller Example Design Block Diagram 64
Figure 6.4. Create SoC Project 65
Figure 6.5. Define Instance 66
Figure 6.6. Build SoC Project Result 66
Figure 6.7. Lattice C/C++ Design Project 67
Figure 6.8. Build C/C++ Project Result 67
Figure 6.9. Sample ENTDAA sequence sent by I3C Controller 68
Figure 7.1. Module/IP Block Wizard 69
Figure 7.2. IP Configuration 70
Figure 7.3. Check Generated Result 71
Figure 7.4. Simulation Wizard 73
Figure 7.5. Add and Reorder Source 74
Figure 7.6. Parse HDL Files for Simulation 74
Figure 7.7. Summary 75
Figure 7.8. Simulation Waveform 75

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Tables

Table 1.1. Summary of the I3C Controller IP 10
Table 1.2. I3C Controller IP Support Readiness 10
Table 2.1. User Interfaces and Supported Protocols 16
Table 2.2. User Packet Frame Structure 17
Table 2.3. Write 4 Bytes to I3C Target with Address 7'h10 Example 18
Table 2.4. Read 4 Bytes from I3C Target with Address 7'h10 Example 18
Table 2.5. Write 4 Bytes to I3C Target with Address 7'h10 then Write 8 Bytes to I3C Target with Address 7'h20 Example 19
Table 2.6. Write 4 Bytes to I3C Target with Address 7'h10 then Read 4 Bytes from I3C Target with Address 7'h10 Example 20
Table 2.7. Example 1: Assign DA to 1 I3C Target (DA = 0x10) 21
Table 2.8. Example 2: Assign DA to 3 I3C Targets (DA_0 = 0x10, DA_1 = 0x11, DA_2 = 0x12) 21
Table 2.9. Reset Dynamic Address (RSTDAA) CCC 22
Table 2.10. Generic Broadcast CCC Format 22
Table 2.11. Generic Direct CCC Format 23
Table 2.12. I3C Controller IP Core Supported CCCs 23
Table 2.13. Write 4 Bytes to I2C Target with Address 7'h20 Example 25
Table 2.14. Read 4 Bytes from I2C Target with Address 7'h20 Example 25
Table 2.15. Write 4 Bytes to I2C Target with Address 7'h10 then Read 4 Bytes from I2C Target with Address 7'h10 Example 26
Table 2.16. Write 4 Bytes to I3C Target with Address 7'h10 Using HDR-DDR Mode Example 28
Table 2.17. Read 4 Bytes from I3C Target with Address 7'h10 Using HDR-DDR Mode Example 29
Table 2.18. HDR-DDR Consecutive Write and Read 30
Table 2.19. Allowed CCCs During HDR-DDR Mode and Limitations 31
Table 2.20. Generic Broadcast CCC Format 32
Table 2.21. Broadcast ENTAS0 CCC in HDR-DDR Mode Example 33
Table 2.22. Broadcast SET MWL CCC in HDR-DDR Mode Example 33
Table 2.23. Generic Direct Set CCC Format 34
Table 2.24. Direct ENTAS0 CCC in HDR-DDR Mode Example 35
Table 2.25. Direct SET MWL CCC in HDR-DDR Mode Example 35
Table 2.26. Generic Direct Get CCC Format 36
Table 2.27. Direct GET MWL CCC in HDR-DDR Mode Example 37
Table 3.1. General Attributes 41
Table 3.2. Secondary Controller (Available only if Secondary Controller Capable is Enabled) 43
Table 3.3. IP Parameter Settings for Example Test Cases 44
Table 4.1. Ports Description 46
Table 5.1. Register Access Types 48
Table 5.2. I3C Controller IP Register Summary 48
Table 5.3. System Clock Divider Register 0x01 49
Table 5.4. Controller Configuration 0 Register 0x02 49
Table 5.5. Open Drain Timer Register 0x03 51
Table 5.6. I2C Clock Divider Register 0x04 51
Table 5.7. Pull Up Resistor Disable Register 0x05 51
Table 5.8. HDR-DDR Configuration 0 Register 0x06 52
Table 5.9. Soft Reset Register 0x08 53
Table 5.10. Transmit Start Register 0x11 53
Table 5.11. Secondary Controller Status Info Register 0x15 53
Table 5.12. Set Device Role Register 0x16 54
Table 5.13. Secondary Controller Timeout Register 0x17–0x19 55
Table 5.14. Dynamic Address ACKed Counter Register 0x1C 55
Table 5.15. IBI Read Count Register 0x1D 55
Table 5.16. IBI Response Register 0x1E 55
Table 5.17. IBI Address Register 0x1F 56
Table 5.18. Interrupt Status 0 Register 0x20 56

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Table 5.19. Interrupt Set 0 Register 0x21 56
Table 5.20. Interrupt Enable 0 Register 0x22 57
Table 5.21. Interrupt Status 1 Register 0x24 57
Table 5.22. Interrupt Set 1 Register 0x25 58
Table 5.23. Interrupt Enable 1 Register 0x26 58
Table 5.24. Bus Condition Debug Register 0x28 58
Table 5.25. Last NAK Address Debug Register 0x29 59
Table 5.26. Last ACK Address Debug Register 0x2A 59
Table 5.27. Interrupt Status 2 Register 0x2C 60
Table 5.28. Interrupt Set 2 Register 0x2D 60
Table 5.29. Interrupt Enable 2 Register 0x2E 61
Table 5.30. Tx FIFO Register 0x30 61
Table 5.31. Rx FIFO Register 0x40 61
Table 6.1. I3C Controller IP Configuration Supported by the Example Design 62
Table 7.1. Generated File List 71
Table A.1. LFCPNX-100-7ASG256C Device Resource Utilization 79
Table A.2. LIFCL-40-7BG256I Device Resource Utilization 79
Table A.3. LAV-AT-E70-1LFG676I Device Resource Utilization 80
Table A.4. LN2-CT-20-1CBG484I Device Resource Utilization 80

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Abbreviations in This Document

A list of abbreviations used in this document.

Abbreviation	Definition
ACK	Acknowledgement
AHB	Advanced High-performance Bus
AMBA	Advanced Microcontroller Bus Architecture
APB	Advanced Peripheral Bus
API	Application Programming Interface
BCR	Bus Characteristics Register
CCC	Common Command Code
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CSR	Control and Status Registers
D2D	Device-to-Device
DA	Dynamic Address
DAA	Dynamic Address Assignment
DCR	Device Characteristics Register
DDR	Double Data Rate
DSP	Digital Signal Processor
DWORD	Double Word
EBR	Embedded Block RAM
FIFO	First In, First Out
FPGA	Field Programmable Gate Array
GPIO	General Purpose Input/Output
GUI	Graphical User Interface
HDL	Hardware Description Language
HDR	High Data Rate
I/O	Input/Output
I2C	Inter-Integrated Circuit
I3C	Improved Inter-Integrated Circuit
IBI	In-Band Interrupt
IP	Intellectual Property
LMMI	Lattice Memory Mapped Interface
LSE	Lattice Synthesis Engine
LUT	Look-Up Table
MC	Microcontroller
MDB	Mandatory Data Byte
MIPI	Mobile Industry Processor Interface
N/A	Not Applicable
NAK	Negative Acknowledgement
PIC	Programmable Interrupt Controller
PLL	Phase-Locked Loop
RAM	Random Access Memory
RISC-V	Reduced Instruction Set Computer Five
RO	Read-Only access
RTL	Register Transfer Level
RW	Read-Write access

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Abbreviation	Definition
RW1C	Read and Write 1 to Clear
Rx	Receiver
SCL	Serial Clock
SDA	Serial Data
SDK	Software Development Kit
SDR	Single Data Rate
SoC	System on Chip
SRAM	Static Random Access Memory
Tx	Transmitter
UART	Universal Asynchronous Receiver/Transmitter
WO	Write-Only access

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

1. Introduction

1.1. Overview of the IP

The Lattice I3C IP is designed to comply with the MIPI I3C® specification.

The MIPI I3C interface eases sensor system design architectures in mobile wireless products by providing a fast, low-cost, low-power, and two-wire digital interface for sensors. I3C protocol is a single scalable, cost effective, and a power efficient protocol. Implementing the I3C specification increases the implementation flexibility for an ever-expanding sensor subsystem as efficiently and at the lowest cost possible.

The I3C protocol is backward compatible with many legacy I2C devices. The I3C protocol offers greater than 10× speed improvements, more efficient bus power management, new communication modes, and new device roles which includes an ability to change device roles over time. For example, the initial Controller can cooperatively pass the Controller role to another I3C device on the Bus, if the requesting I3C device supports Secondary Controller feature.

1.2. Quick Facts

Table 1.1. Summary of the I3C Controller IP

IP Requirements	Supported Devices	iCE40 UltraPlus™, MachXO3D™, CrossLink™-NX, Certus™-NX, Certus-NX-RT, CertusPro™-NX, CertusPro-NX-RT, Mach™-NX, MachXO5™-NX, Lattice Avant™, Certus-N2, and Mach-N2.
	IP Changes[1]	For a list of changes to the IP, refer to the I3C Controller IP Release Notes (FPGA-RN-02017).
Resource Utilization	Supported User Interface	Lattice Memory Mapped Interface (LMMI), Advanced Peripheral Bus (APB), and Advanced High-Performance Bus-Lite (AHB-Lite)
	Resources	Refer to Appendix A. Resource Utilization
Design Tool Support	Lattice Implementation	IP Core v3.7.2 – Lattice Radiant™ Software 2026.1.1 and Lattice Propel™ Builder Software 2026.1.1
	Synthesis	Lattice Synthesis Engine (LSE) Synopsys Synplify Pro® for Lattice
	Simulation	For the list of supported simulators, see the Lattice Radiant Software User Guide.

Note:

1. In some instances, the IP may be updated without changes to the user guide. This user guide may reflect an earlier IP version but remains fully compatible with the later IP version. Refer to the IP Release Notes for the latest updates.

1.3. IP Support Summary

Table 1.2. I3C Controller IP Support Readiness

Device Family	System Clock Frequency (MHz)[1]	Supported Feature	Radiant Timing Model
iCE40 UltraPlus Crosslink-NX Certus-NX Certus-NX-RT CertusPro-NX CertusPro-NX-RT MachXO5-NX	25	Dynamic Address Assignment	Final
		IBI	Final
		Hot-Join	Final
		HDR-DDR	Final
		Secondary Controller	Final
		I3C Write/Read	Final
		I2C Write/Read	Final
Avant Certus-N2 Mach-N2	25	Dynamic Address Assignment	Preliminary
		IBI	Preliminary
		Hot-Join	Preliminary

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Device Family	System Clock Frequency (MHz)[1]	Supported Feature	Radiant Timing Model
		HDR-DDR	Preliminary
		Secondary Controller	Preliminary
		I3C Write/Read	Preliminary
		I2C Write/Read	Preliminary

Note:

1. This is the system clock frequency used during hardware validation. For the actual frequency supported by the IP, refer to Appendix A. Resource Utilization.

The example design for the I3C Controller IP allows for simulation and deployment to development boards for testing. Refer to the Example Design section for more details on how to run the example design on hardware and simulation.

1.4. Features

An I3C bus requires exactly one I3C device at a time functioning as an I3C Controller device. In I3C terms, this I3C Controller device is the active Controller at that time. In typical applications, the active Controller is the I3C device on the bus that sends majority of the I3C commands, addressing either all targets (Broadcast CCCs) or specific individual target (Directed CCCs). The active Controller is also the only device on the I3C bus allowed to send I2C Messages.

The Lattice I3C Controller supports the following features:

- Compatible with MIPI I3C Specification v1.1.1
- Two-wire serial interface up to 12.5 MHz using Push-Pull
- Legacy I2C device coexist on the same bus (with some limitations)
- Dynamic Addressing while supporting Static Addressing for legacy I2C devices
- I2C-like SDR messaging
- HDR-DDR mode support
- In-Band Interrupt support
- Hot-Join support
- Primary Controller
- Secondary Controller
- Configurable receive and transmit FIFO implementation and size

The Lattice I3C Controller IP does not support the following features:

- HDR-TSL for Ternary Symbol Legacy-inclusive-Bus (I2C Devices allowed)
- HDR-TSP for Ternary Symbol for Pure Bus (no I2C Devices allowed)
- HDR-BT for Bulk Transport
- Synchronous Timing and Asynchronous Time Stamping (Expect Mode 0)
- Monitoring Device Early Termination
- D2D Tunneling
- Multi-Lane Data Transfer

1.5. Licensing and Ordering Information

The I3C Controller IP is provided at no additional cost with the Lattice Radiant software.

1.6. Minimum Device Requirements

There is no limitation in device speed grade for I3C Controller IP. See Appendix A. Resource Utilization for minimum required resources to instantiate this IP and maximum clock frequency supported.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

1.7. Naming Conventions

1.7.1. Nomenclature

The nomenclature used in this document is based on Verilog HDL.

1.7.2. Signal Names

- _*n* are active low (asserted when value is logic 0)
- _*i* are input signals
- _*o* are output signals
- _*oe* are output enable signals
- _*io* are bidirectional signals

1.7.3. Attribute Names

Attribute names in this document are formatted in title case and italicized (*Attribute Name*).

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

2. Functional Description

2.1. IP Architecture Overview

I3C Controller IP supports several communication formats, all sharing a two-wire interface: SDA bidirectional data line and SCL bidirectional clock.

The Lattice I3C Controller supports the following modes:

- SDR mode
- HDR-DDR mode

SDR Mode is the default mode of the I3C bus and is primarily used for private messaging from the Current Controller device to Secondary devices. SDR Mode is also used to enter other modes and for built-in features, such as Common Command Codes (CCCs), In-Band Interrupts, and transition from I2C to I3C by assignment of a Dynamic Address.

I3C SDR Mode is significantly like the I2C protocol in terms of procedures and conditions. As a result, I3C devices and many legacy I2C Secondary devices (but not I2C Controller devices) can coexist on the same I3C bus.

Both SDR and HDR-DDR Mode use SCL as a clock. However, in HDR-DDR Mode, SDA is changed on both SCL edges (when High and Low), effectively doubling the data rate. By contrast, in SDR Mode, SDA is changed only when SCL is Low. SDR Mode defines it as a START or a STOP when SDA changes while SCL remains High.

Throughout this document, *HDR-DDR-Capable* indicates that the IP is capable to support both SDR Mode and HDR-DDR Mode.

HDR-DDR moves data by Words. Each Word contains the following bits:

- 16 payload bits as two bytes in a byte stream + 2 parity bits, which gives an 18-bit Word;
- The HDR-DDR Protocol precedes the 18-bit Word with a 2-bit Preamble, which gives a 20-bit Word.

There are four HDR-DDR Word Types which are defined as follows:

- Command Word
- Data Word
- CRC Word
- Reserved Word

The I3C Controller accepts commands from an LMMI interface or APB/AHB-Lite when you select the optional standard interface. These commands are translated into I3C signals and forwarded to I3C Target Device.

The I3C Controller can operate in interrupt or polling mode. This means that you can choose to poll the I3C Controller for a change in status at periodic intervals or wait to be interrupted by the I3C Controller when data needs to be read or written.

I3C Controller IP Core functional diagram is shown in Figure 2.1.

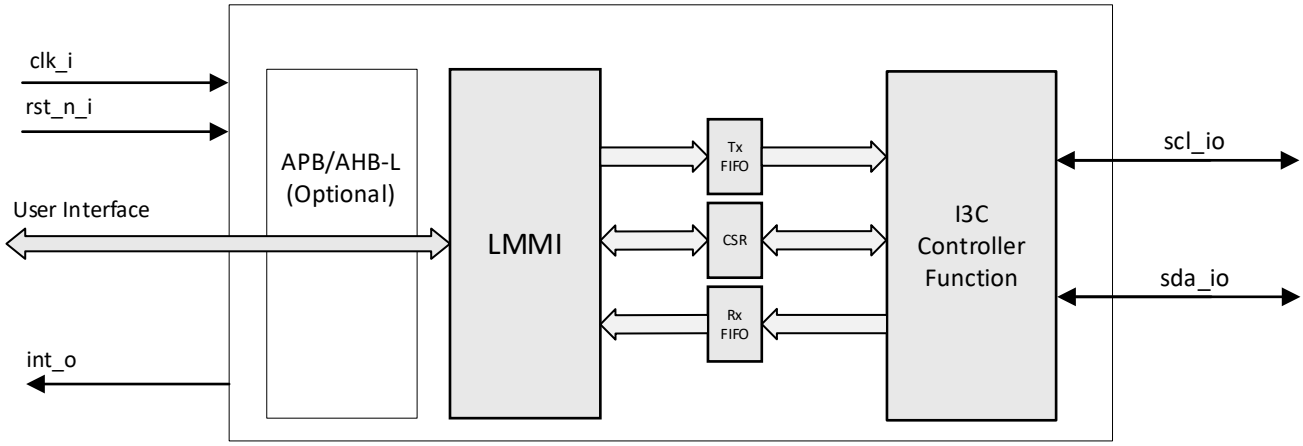


Figure 2.1. I3C Controller IP Core Functional Diagram (Controller Only)

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

The I3C Controller IP Core has an option to act as a Target when Secondary Controller Capable attribute is checked in the IP configuration GUI.

Figure 2.2 shows the functional diagram of the IP Core with Secondary Controller Capability.

The device role can switch from Controller to Target or Target to Controller using the controller handoff procedure defined in the MIPI I3C Specification.

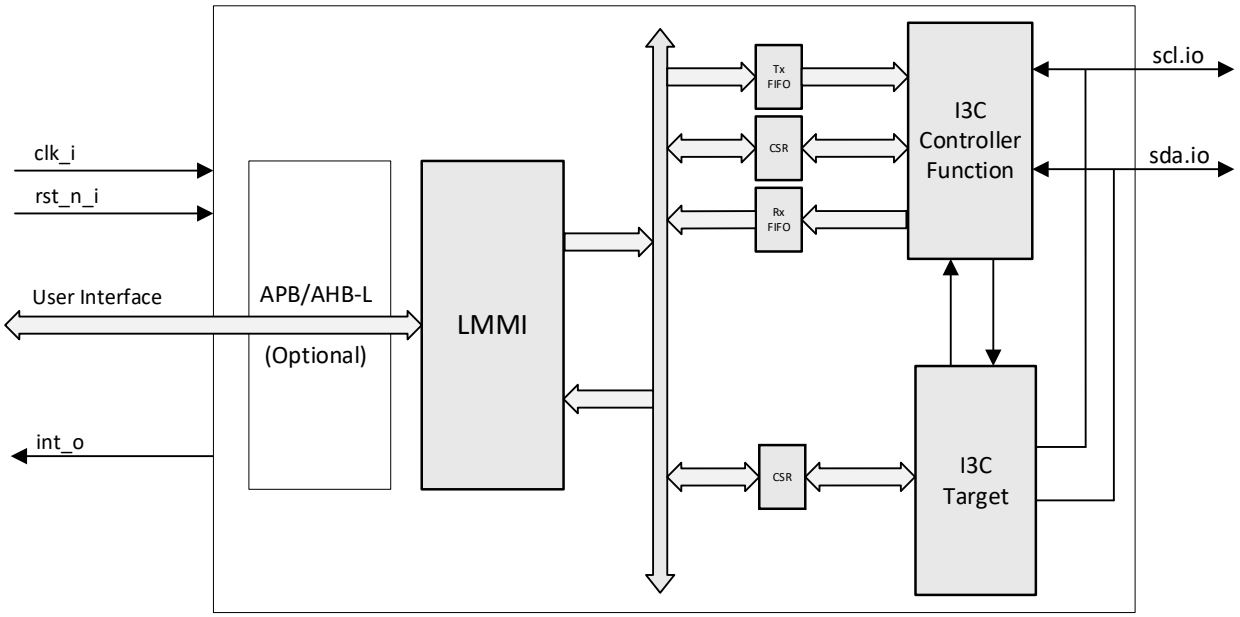


Figure 2.2. I3C Controller IP Core Functional Diagram (with Secondary Controller Capability)

2.2. Clocking

Figure 2.3 shows the clock diagram when Use Internal Clock Divider attribute is checked in the IP configuration GUI. There is one input clock clk_i and one output clock scl_io. Each clock is discussed in detail in the Clocking Overview section.

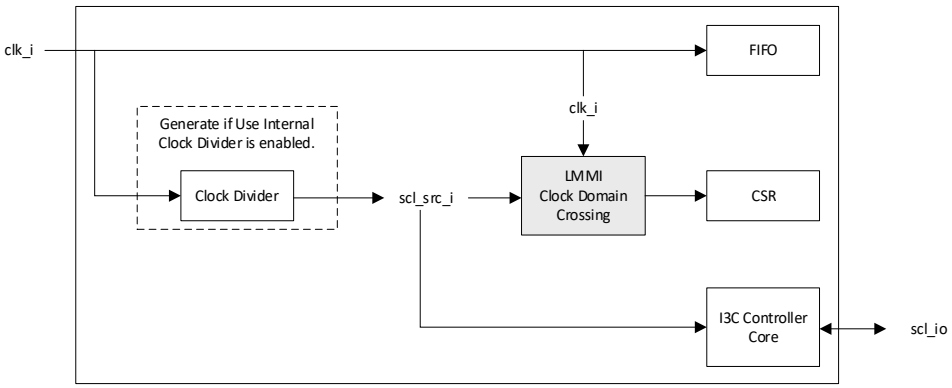


Figure 2.3. I3C Controller IP Clock Domain Block Diagram (Internal Clock Divider Enabled)

Figure 2.4 shows the clock diagram when Use Internal Clock Divider attribute is unchecked in the IP configuration GUI. There are two input clocks, (1) clk_i and (2) src_clk_scl_i, and one output clock scl_io. Each clock is discussed in detail in the Clocking Overview section.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

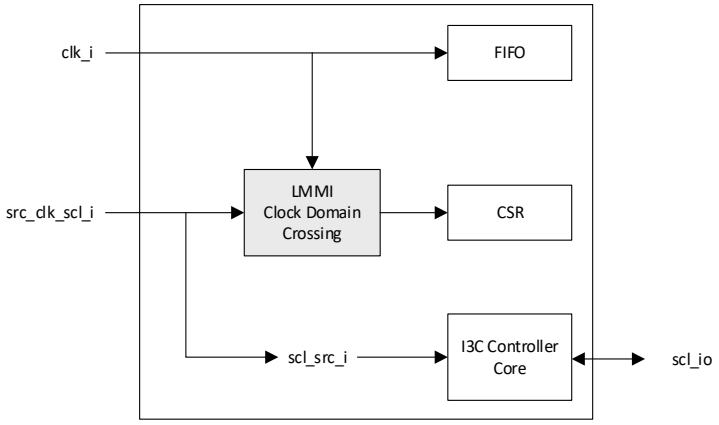


Figure 2.4. I3C Controller IP Clock Domain Block Diagram (Internal Clock Divider Disabled)

2.2.1. Clocking Overview

When internal clock divider is enabled, the I3C Controller IP has the following clocks:

- clk_i
  System clock – Can be set in the range of 0.8 MHz – 50 MHz.
- scl_src_i
  Core Clock – This is a generated clock. When the system clock frequency is set to greater than 25 MHz, the core clock frequency is equal to half of the system clock frequency. Otherwise, the core clock frequency is equal to the system clock frequency.
- scl_io
  Serial clock – Maximum frequency is equal to half of internal clock frequency. Initial frequency depends on the SCL Pulse width setting in IP configuration GUI and can be reprogrammed by writing to sys_clk_div register.

When internal clock divider is disabled, the I3C Controller IP has the following clocks:

- clk_i
  System clock – Can be set in the range of 0.8 MHz – 50 MHz.
- src_clk_scl_i
  Core Clock – Can be set in the range of 0.02 MHz – 25 MHz. Equal to internal scl_src_i.
- scl_io
  Serial clock – Maximum frequency is equal to half of core clock frequency. Initial frequency is equal to half of core clock frequency and can be reprogrammed by writing to sys_clk_div register.

2.3. Reset

This IP has one asynchronous active low reset rst_n_i.

To ensure that reset has been properly propagated inside the IP, wait for at least 20 system clock (clk_i) cycles after system reset (rst_n_i) de-assertion before doing any IP operation.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

2.4. User Interfaces

Table 2.1 shows the user interfaces and supported protocols. The memory-mapped interface of I3C Controller IP Core is selected by the Interface attribute. It can be LMMI interface, AHB-Lite interface, or APB interface.

Table 2.1. User Interfaces and Supported Protocols

User Interface	Supported Protocols	Description
Selectable Memory-Mapped Interface	LMMI	For LMMI interface, refer to the Lattice Memory Mapped Interface and Lattice Interrupt Interface (FPGA-UG-02039) document for information and timing diagram of the LMMI.
	AHB-Lite	For AHB-Lite interface, refer to the AMBA 3 AHB-Lite Protocol v1.0 Specification for information and timing diagram of the APB interface.
	APB	For APB interface, refer to the AMBA 3 APB Protocol v1.0 Specification for information and timing diagram of the APB interface.
Device Receiver/Transmitter Interface	I3C	The I3C Interface can complete the communication between Lattice I3C Controller IP core and external I3C Target devices. Refer to the MIPI I3C Specification for more information on the I3C protocol.

2.5. User Packet Frame Structure

The user command for sending CCC or private write and read access to I3C target shall follow the defined user packet frame structure. These user packets will be translated by the I3C controller to I3C signaling format defined in the MIPI I3C Specification. The user packet will be written to Tx FIFO through LMMI or other user selected bus interface. Read data response from target device will be stored in Rx FIFO and can be read by though LMMI or other user selected bus interface.

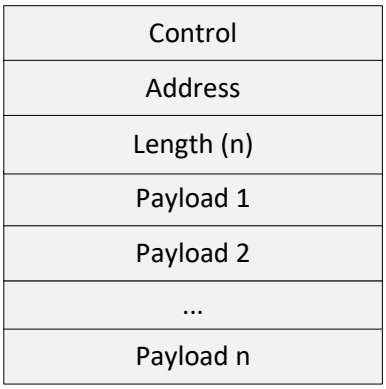


Figure 2.5. User Packet Frame Structure

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Table 2.2. User Packet Frame Structure

Name	Bits	Description
Control	[0]	Set to 0 when sending Private Write or Private Read access. Set to 1 when sending CCC.
	[1]	Set to 0 when the frame will start with Start symbol (S). Set to 1 when the frame is a continuation of the previous packet and will start with Repeated Start symbol (Sr).
	[2]	Set to 0 when the next frame will continue with Repeated Start (Sr). Set to 1 when the frame will end with Stop symbol (P).
	[3]	Set to 0 when the frame is not a start of CCC. Set to 1 when the frame is a CCC and will start with Start symbol (S).
	[4]	Set to 0 when the frame is to be sent using I3C protocol. Set to 1 when the frame is to be sent using I2C protocol. Note: If the packet is to be sent in I2C mode, you must also set the configuration register (i2c_mode_allowed) to enable I2C frames.
	[5]	0 – no wait 1 – wait for next packet This is used to make sure that the succeeding I3C frames will continue with Repeated Start. There are times when you cannot give the next packet immediately, but it is expected to continue with Repeated Start. Normally, when Tx FIFO becomes empty, the I3C controller will end the transaction with Stop. However, if this control bit is set, the I3C controller will pause the SCL and wait for the next user packet. In HDR-DDR mode, it is recommended to set this to 1 when there are multiple I3C frames in one packet to ensure that HDR-DDR mode will not be exited in case of the following: • The preceding frame is NAK-ed, or • User read/write command is terminated early, or • HDR-DDR framing error is detected. If this is set to 0, reset the IP core and cleanup the contents of FIFO when the mentioned conditions occur to avoid incorrect packet framing and an unknown internal state the of IP core.
	[6]	Unused – set to 0.
	[7]	Set to 0 when sending frames in SDR mode. Set to 1 when sending frames in HDR-DDR mode.
Address	[0]	This indicates the access type, either Write (1'b0) or Read (1'b1).
	[7:1]	This is the 7-bit address of the target device or the I3C broadcast address 0x7E.
Length	[7:0]	0 – no payload 1 – 1 byte 2 – 2 bytes … 255 – 255 bytes This indicates the number of data bytes that follow. In write request, this determines the number of bytes of payload that follows. For read request, this is the number of bytes expected to be returned by the target device.
Payload 1…n	[7:0]	The data bytes are sent for write access.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

2.6. Performing Various I3C Transfers (SDR)

The following examples show how to create the packet and send it to LMMI or a selected bus to generate I3C transactions.

2.6.1. I3C Private Write



Figure 2.6. I3C Private Write

Table 2.3. Write 4 Bytes to I3C Target with Address 7'h10 Example

LMMI Address	LMMI Write Data	Comment
0x30	0x04	// Control
		// [0] = 0 -> Private Write
		// [1] = 0 -> not repeated start
		// [2] = 1 -> terminate frame with Stop (P)
		// [7:3] = 0
0x30	{7'h10,1'b0}	// 7b Target Address, Write
0x30	0x04	// Length = 4 bytes
0x30	0x55	// Payload 1
0x30	0xAA	// Payload 2
0x30	0xCC	// Payload 3
0x30	0x33	// Payload 4
0x11	0x01	// start

2.6.2. I3C Private Read



Figure 2.7. I3C Private Read

Table 2.4. Read 4 Bytes from I3C Target with Address 7'h10 Example

LMMI Address	LMMI Write Data	Comment
0x30	0x04	// Control
		// [0] = 0 -> Private Read
		// [1] = 0 -> not repeated start
		// [2] = 1 -> terminate frame with Stop (P)
		// [7:3] = 0
0x30	{7'h10,1'b1}	// 7b Target Address, Read
0x30	0x04	// Length = 4 bytes
0x11	0x01	// start

When Read data is available, interrupt (int_o) will assert (if interrupt status rx_fifo_not_empty is enabled) and you can get the read data by sending LMMI read request at address offset 0x40.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

2.6.3. I3C Consecutive Private Write

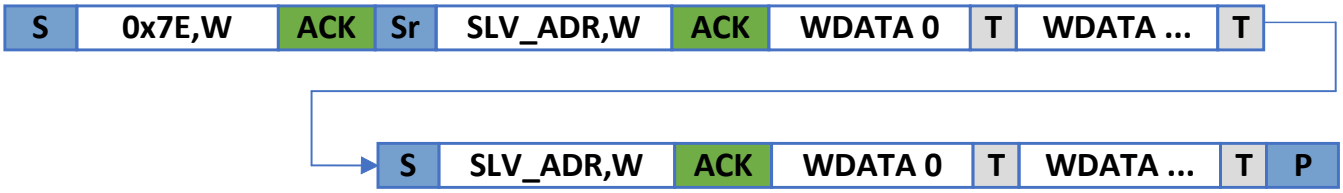


Figure 2.8. I3C Consecutive Private Write

Table 2.5. Write 4 Bytes to I3C Target with Address 7'h10 then Write 8 Bytes to I3C Target with Address 7'h20 Example

LMMI Address	LMMI Write Data	Comment
0x30	0x00	// Control
		// [0] = 0 -> Private Write
		// [1] = 0 -> not repeated start
		// [2] = 0 -> do not terminate frame with Stop (P)
		// [7:3] = 0
0x30	{7'h10,1'b0}	// 7b Target Address, Write
0x30	0x04	// Length = 4 bytes
0x30	0x01	// Payload 1
0x30	0x23	// Payload 2
0x30	0x45	// Payload 3
0x30	0x67	// Payload 4
0x30	0x06	// Control
		// [0] = 0 -> Private Write
		// [1] = 1 -> repeated start
		// [2] = 1 -> terminate frame with Stop (P)
		// [7:3] = 0
0x30	{7'h20,1'b0}	// 7b Target Address, Write
0x30	0x08	// Length = 8 bytes
0x30	0x89	// Payload 1
0x30	0xAB	// Payload 2
0x30	0xCD	// Payload 3
0x30	0xEF	// Payload 4
0x30	0x00	// Payload 5
0x30	0x11	// Payload 6
0x30	0x22	// Payload 7
0x30	0x33	// Payload 8
0x11	0x01	// start

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

2.6.4. I3C Consecutive Private Write and Private Read

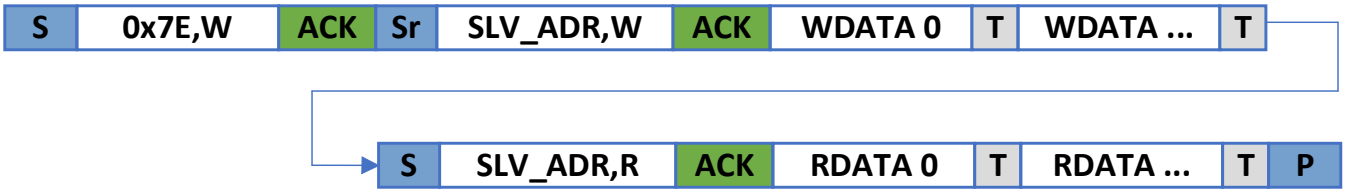


Figure 2.9. I3C Consecutive Private Write and Private Read

Table 2.6. Write 4 Bytes to I3C Target with Address 7'h10 then Read 4 Bytes from I3C Target with Address 7'h10 Example

LMMI Address	LMMI Write Data	Comment
0x30	0x00	// Control
		// [0] = 0 -> Private Write
		// [1] = 0 -> not repeated start
		// [2] = 0 -> do not terminate frame with Stop (P)
		// [7:3] = 0
0x30	{7'h10,1'b0}	// 7b Target Address, Write
0x30	0x04	// Length = 4 bytes
0x30	0x01	// Payload 1
0x30	0x23	// Payload 2
0x30	0x45	// Payload 3
0x30	0x67	// Payload 4
0x30	0x06	// Control
		// [0] = 0 -> Private Read
		// [1] = 1 -> repeated start
		// [2] = 1 -> terminate frame with Stop (P)
		// [7:3] = 0
0x30	{7'h10,1'b1}	// 7b Target Address, Read
0x30	0x04	// Length = 4 bytes
0x11	0x01	// start

When Read data is available, interrupt (int_o) will assert (if interrupt status rx_fifo_not_empty is enabled) and you can get the read data by sending LMMI read request at address offset 0×40.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

2.6.5. Enter Dynamic Address Assignment (DAA) CCC

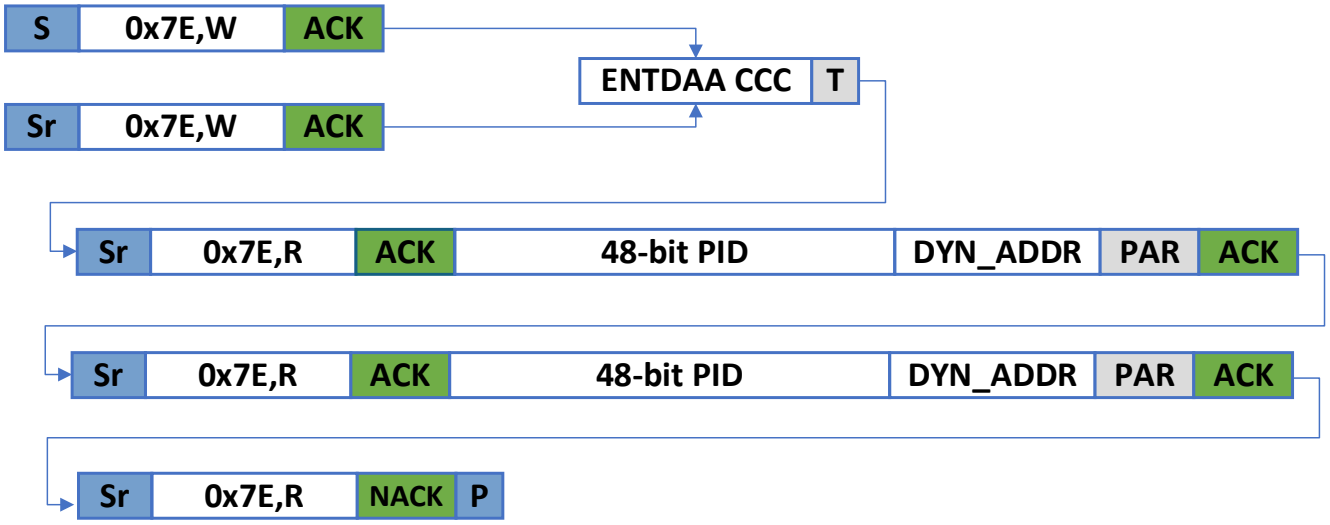


Figure 2.10. Enter Dynamic Address Assignment

For more details on the ENTDAA flow, refer to the MIPI I3C Specification.

Table 2.7. Example 1: Assign DA to 1 I3C Target (DA = 0x10)

LMMI Address	LMMI Write Data	Comment
0x30	0x0D	// Control
		// [0] = 1 -> CCC
		// [1] = 0 -> not repeated start
		// [2] = 1 -> terminate frame with Stop (P)
		// [3] = 1 -> CCC start
		// [7:4] = 0
0x30	{7'h7E,1'b0}	// Reserved Address 0x7E, Write
0x30	0x02	// Length = 1 + number of target address (1)
0x30	0x07	// ENTDAA CCC = 0x07
0x30	{7'h10,1'b0}	// {DA, Odd Parity of DA} // Parity will be auto generated by I3C controller
0x11	0x01	// start

Table 2.8. Example 2: Assign DA to 3 I3C Targets (DA_0 = 0x10, DA_1 = 0x11, DA_2 = 0x12)

LMMI Address	LMMI Write Data	Comment
0x30	0x0D	// Control
		// [0] = 1 -> CCC
		// [1] = 0 -> not repeated start
		// [2] = 1 -> terminate frame with Stop (P)
		// [3] = 1 -> CCC start
		// [7:4] = 0
0x30	{7'h7E,1'b0}	// Reserved Address 0x7E, Write
0x30	0x04	// Length = 1 + number of target address (3)

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

LMMI Address	LMMI Write Data	Comment
0x30	0x07	// ENTDAA CCC = 0x07
0x30	{7'h10,1'b0}	// {DA, Odd Parity of DA} // Parity will be auto generated by I3C controller
0x30	{7'h11,1'b1}	// {DA, Odd Parity of DA} // Parity will be auto generated by I3C controller
0x30	{7'h12,1'b1}	// {DA, Odd Parity of DA} // Parity will be auto generated by I3C controller
0x11	0x01	// start

2.6.6. Reset Dynamic Address (RSTDAA) CCC



Figure 2.11. Reset Dynamic Address

Table 2.9. Reset Dynamic Address (RSTDAA) CCC

LMMI Address	LMMI Write Data	Comment
0x30	0x0D	// Control
		// [0] = 1 -> CCC
		// [1] = 0 -> not repeated start
		// [2] = 1 -> terminate frame with Stop (P)
		// [3] = 1 -> CCC start
		// [7:4] = 0
0x30	{7'h7E,1'b0}	// Reserved Address 0x7E, Write
0x30	0x01	// Length = 1
0x30	0x06	// RSTDAA CCC = 0x06
0x11	0x01	// start

2.6.7. Generic Broadcast CCC Format



Figure 2.12. Generic Broadcast CCC Format

Table 2.10. Generic Broadcast CCC Format

LMMI Address	LMMI Write Data	Comment
0x30	0x0D	// Control
		// [0] = 1 -> CCC
		// [1] = 0 -> not repeated start
		// [2] = 1 -> terminate frame with Stop (P)
		// [3] = 1 -> CCC start
		// [7:4] = 0
0x30	{7'h7E,1'b0}	// Reserved Address 0x7E, Write
0x30	0x01+0xNN	// Length = 1+ number of optional bytes
0x30	<CCC>, …, <n bytes>	// CCC and Optional bytes

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

LMMI Address	LMMI Write Data	Comment
0x11	0x01	// start

2.6.8. Generic Direct CCC Format

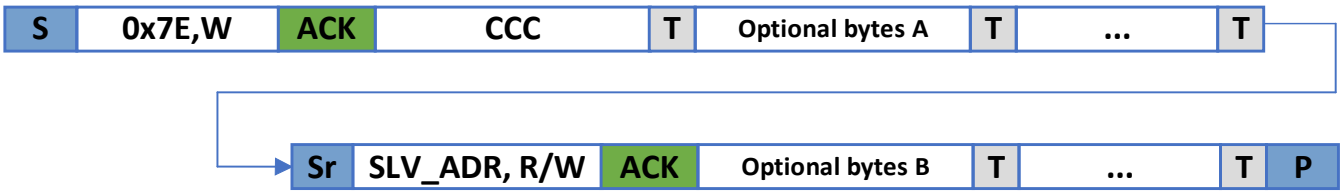


Figure 2.13. Generic Direct CCC Format

Table 2.11. Generic Direct CCC Format

LMMI Address	LMMI Write Data	Comment
0x30	0x09	// Control
		// [0] = 1 -> CCC
		// [1] = 0 -> not repeated start
		// [2] = 0 -> do not terminate frame with Stop (P)
		// [3] = 1 -> CCC start
		// [7:4] = 0
0x30	{7'h7E,1'b0}	// Reserved Address 0x7E, Write
0x30	0x01+0xNN	// Length = 1+ number of optional bytes (A)
0x30	<CCC>, …, <n bytes>	// CCC and Optional bytes
0x30	0x07	// Control
		// [0] = 1 -> CCC
		// [1] = 1 -> repeated start
		// [2] = 1 -> terminate frame with Stop (P)
		// [3] = 0 -> not CCC start
		// [7:4] = 0
0x30	{<TGT_ADR>, <R/W>}	// 7b Target Address, Read/Write
0x30	0xMM	// Length = number of optional bytes (B)
0x30	<m bytes>	// Optional bytes (B)
0x11	0x01	// start

2.6.9. Common Command Codes

Table 2.12 lists the supported CCCs of this IP.

Table 2.12. I3C Controller IP Core Supported CCCs

CCC	Type	Required [1]	Command Name
0x00	Broadcast	R	ENEC
0x01	Broadcast	R	DISEC
0x02	Broadcast	C	ENTAS0
0x03	Broadcast	O	ENTAS1
0x04	Broadcast	O	ENTAS2
0x05	Broadcast	O	ENTAS3

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

CCC	Type	Required [1]	Command Name
0x06	Broadcast	R	RSTDAA
0x07	Broadcast	R	ENTDAA
0x08	Broadcast	C	DEFTGTS
0x09	Broadcast	R	SETMWL
0x0A	Broadcast	R	SETMRL
0x12	Broadcast	C	ENDXFER
0x20	Broadcast	C	ENTHDR0
0x28	Broadcast	C	SETXTIME
0x29	Broadcast	O	SETAASA
0x2A	Broadcast	R	RSTACT
0x80	Direct	R	ENEC
0x81	Direct	R	DISEC
0x82	Direct	C	ENTAS0
0x83	Direct	O	ENTAS1
0x84	Direct	O	ENTAS2
0x85	Direct	O	ENTAS3
0x87	Direct	O	SETDASA
0x88	Direct	C	SETNEWDA
0x89	Direct	R	SETMWL
0x8A	Direct	R	SETMRL
0x8B	Direct	R	GETMWL
0x8C	Direct	R	GETMRL
0x8D	Direct	C	GETPID
0x8E	Direct	C	GETBCR
0x8F	Direct	C	GETDCR
0x90	Direct	R	GETSTATUS
0x91	Direct	C	GETACCCR
0x92	Direct	C	ENDXFER
0x94	Direct	C	GETMXDS
0x95	Direct	C	GETCAPS
0x98	Direct	C	SETXTIME
0x99	Direct	C	GETXTIME
0x9A	Direct	R	RSTACT

Note:

1. R – Required, O – Optional, and C – Conditional.

2.7. Performing Various I2C Transfers

To send I2C frames, the configuration register must be set to enable I2C transfer. This is done by setting the register i2c_mode_allowed to 1.

2.7.1. I2C Private Write



Figure 2.14. I2C Private Write

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Table 2.13. Write 4 Bytes to I2C Target with Address 7'h20 Example

LMMI Address	LMMI Write Data	Comment
0x30	0x14	// Control
		// [0] = 0 -> Private Write
		// [1] = 0 -> not repeated start
		// [2] = 1 -> terminate frame with Stop (P)
		// [3] = 0
		// [4] = 1 -> send frame in I2C mode
		// [7:5] = 0
0x30	{7'h20,1'b0}	// 7b Target Address, Write
0x30	0x04	// Length = 4 bytes
0x30	0x11	// Payload 1
0x30	0x22	// Payload 2
0x30	0x33	// Payload 3
0x30	0x44	// Payload 4
0x11	0x01	// start

2.7.2. I2C Private Read



Figure 2.15. I2C Private Read

Table 2.14. Read 4 Bytes from I2C Target with Address 7'h20 Example

LMMI Address	LMMI Write Data	Comment
0x30	0x14	// Control
		// [0] = 0 -> Private Read
		// [1] = 0 -> not repeated start
		// [2] = 1 -> terminate frame with Stop (P)
		// [3] = 0
		// [4] = 1 -> send frame in I2C mode
		// [7:5] = 0
0x30	{7'h20,1'b1}	// 7b Target Address, Read
0x30	0x04	// Length = 4 bytes
0x11	0x01	// start

2.7.3. I2C Consecutive Private Write and Private Read

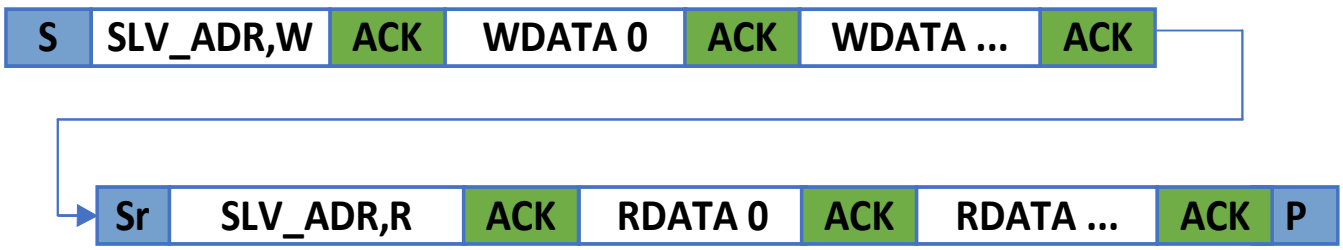


Figure 2.16. I2C Consecutive Private Write and Private Read

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Table 2.15. Write 4 Bytes to I2C Target with Address 7'h10 then Read 4 Bytes from I2C Target with Address 7'h10 Example

LMMI Address	LMMI Write Data	Comment
0x30	0x10	// Control
		// [0] = 0 -> Private Write
		// [1] = 0 -> not repeated start
		// [2] = 0 -> do not terminate frame with Stop (P)
		// [3] = 0
		// [4] = 1 -> sent frame in I2C mode
		// [7:5] = 0
0x30	{7'h10,1'b0}	// 7b Target Address, Write
0x30	0x04	// Length = 4 bytes
0x30	0x01	// Payload 0
0x30	0x23	// Payload 1
0x30	0x45	// Payload 2
0x30	0x67	// Payload 3
0x30	0x16	// Control
		// [0] = 0 -> Private Read
		// [1] = 1 -> repeated start
		// [2] = 1 -> terminate frame with Stop (P)
		// [3] = 0
		// [4] = 1 -> sent frame in I2C mode
		// [7:5] = 0
0x30	{7'h10,1'b1}	// 7b Target Address, Read
0x30	0x04	// Length = 4 bytes
0x11	0x01	// start

When Read data is available, interrupt (int_o) will assert (if interrupt status rx_fifo_not_empty is enabled) and you can get the read data by sending LMMI read request at address offset 0x40.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

2.8. Performing HDR-DDR Mode Transfers

This is only applicable if the Data Rate Mode attribute is set to HDR-DDR-capable in the IP configuration GUI.

Figure 2.17 illustrates a typical HDR-DDR Mode Frame with two HDR Commands and their associated data:

- I3C START or Repeated START
- I3C CCC to Enter HDR-DDR Mode
- After entering HDR-DDR Mode, there is one HDR-DDR Command Word followed by one or more HDR-DDR Data Words, and then an HDR-DDR CRC Word.
- Then, a HDR Restart Pattern
- Then, another HDR-DDR Command Word, HDR-DDR Data Word, and HDR-DDR CRC Word.
- Finally, HDR-DDR Mode is ended by the HDR Exit Pattern followed by I3C STOP.

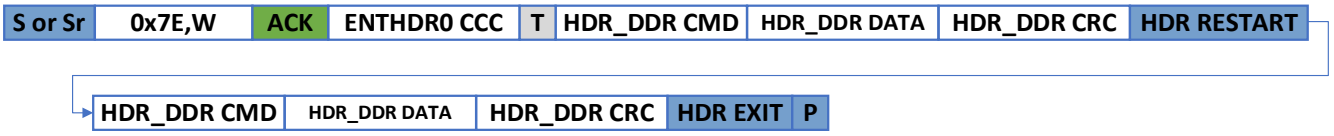


Figure 2.17. HDR-DDR Mode Frame

For more details on the HDR-DDR mode framing, refer to the MIPI I3C Specification.

The HDR frame is encapsulated inside the SDR frame. The framing is like generic Direct CCC format.

However, be aware that in HDR-DDR mode, the data is transmitted or received in words (2 bytes).

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

2.8.1. HDR-DDR Write

Table 2.16. Write 4 Bytes to I3C Target with Address 7'h10 Using HDR-DDR Mode Example

LMMI Address	LMMI Write Data	Comment
0x30	0x09	// Control
		// [0] = 1 -> CCC
		// [1] = 0 -> not repeated start
		// [2] = 0 -> do not terminate frame with Stop (P)
		// [3] = 1 -> CCC start
		// [6:4] = 0
		// [7] = 0 -> SDR mode
0x30	{7'h7E,1'b0}	// Reserved Address 0x7E, Write
0x30	0x01	// Length = 1
0x30	0x20	// ENTHDR0 CCC = 0x20
0x30	0xA6	// Control
		// [0] = 0 -> User command
		// [1] = 1 -> repeated start
		// [2] = 1 -> terminate frame with Stop (P)
		// [3] = 0 -> not CCC start
		// [4] = 0 -> I3C
		// [5] = 1 -> Wait next packet
		// [6] = 0 -> unused
		// [7] = 1 -> HDR-DDR mode
0x30	{7'h10,1'b0}	// 7b Target Address, Write
0x30	0x05	// Length = 1 byte command code + 4 bytes write data // Write data is always multiple of 2 bytes
0x30	{1'b0,<7b user_code>}	// {Write, 7b User command code}
0x30	0xA1	// Payload 1
0x30	0xB2	// Payload 2
0x30	0xC3	// Payload 3
0x30	0xD4	// Payload 4
0x11	0x01	// start

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

2.8.2. HDR-DDR Read

Table 2.17. Read 4 Bytes from I3C Target with Address 7'h10 Using HDR-DDR Mode Example

LMMI Address	LMMI Write Data	Comment
0x30	0x09	// Control
		// [0] = 1 -> CCC
		// [1] = 0 -> not repeated start
		// [2] = 0 -> do not terminate frame with Stop (P)
		// [3] = 1 -> CCC start
		// [6:4] = 0
		// [7] = 0 -> SDR mode
0x30	{7'h7E,1'b0}	// Reserved Address 0x7E, Write
0x30	0x01	// Length = 1
0x30	0x20	// ENTHDR0 CCC = 0x20
0x30	0xA2	// Control
		// [0] = 0 -> User command
		// [1] = 1 -> repeated start
		// [2] = 0 -> do not terminate frame with Stop (P)
		// [3] = 0 -> not CCC start
		// [4] = 0 -> I3C
		// [5] = 1 -> Wait next packet
		// [6] = 0 -> unused
		// [7] = 1 -> HDR-DDR mode
0x30	{7'h10,1'b0}	// 7b Target Address, Write
0x30	0x01	// Length = 1-byte command code
0x30	{1'b1,<7b user_code>}	// {Read, 7b User command code}
0x30	0xA6	// Control
		// [0] = 0 -> User command
		// [1] = 1 -> repeated start
		// [2] = 1 -> terminate frame with Stop (P)
		// [3] = 0 -> not CCC start
		// [4] = 0 -> I3C
		// [5] = 1 -> Wait next packet
		// [6] = 0 -> unused
		// [7] = 1 -> HDR-DDR mode
0x30	{7'h10,1'b1}	// 7b Target Address, Read
0x30	0x04	// Length = 4 bytes read data // Read data is always multiple of 2 bytes
0x11	0x01	// start

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

2.8.3. HDR-DDR Consecutive Write and Read

Example of writing 4 bytes to I3C target with address 7'h10 followed by read using HDR-DDR mode.

Table 2.18. HDR-DDR Consecutive Write and Read

LMMI Address	LMMI Write Data	Comment
0x30	0x09	// Control
		// [0] = 1 -> CCC
		// [1] = 0 -> not repeated start
		// [2] = 0 -> do not terminate frame with Stop (P)
		// [3] = 1 -> CCC start
		// [6:4] = 0
		// [7] = 0 -> HDR-DDR mode
0x30	{7'h7E,1'b0}	// Reserved Address 0x7E, Write
0x30	0x01	// Length = 1
0x30	0x20	// ENTHDR0 CCC = 0x20
0x30	0xA2	// Control
		// [0] = 0 -> User command
		// [1] = 1 -> repeated start
		// [2] = 0 -> do not terminate frame with Stop (P)
		// [3] = 0 -> not CCC start
		// [4] = 0 -> I3C
		// [5] = 1 -> Wait next packet
		// [6] = 0 -> unused
		// [7] = 1 -> HDR-DDR mode
0x30	{7'h10,1'b0}	// 7b Target Address, Write
0x30	0x04	// Length = 4 bytes
0x30	0xA1	// Payload 1
0x30	0xB2	// Payload 2
0x30	0xC3	// Payload 3
0x30	0xD4	// Payload 4
0x30	0xA2	// Control
		// [0] = 0 -> User command
		// [1] = 1 -> repeated start
		// [2] = 0 -> do not terminate frame with Stop (P)
		// [3] = 0 -> not CCC start
		// [4] = 0 -> I3C
		// [5] = 1 -> Wait next packet
		// [6] = 0 -> unused
		// [7] = 1 -> HDR-DDR mode
0x30	{7'h10,1'b0}	// 7b Target Address, Write
0x30	0x01	// Length = 1-byte command code
0x30	{1'b1,<7b user_code>}	// {Read, 7b user command code}
0x30	0xA6	// Control
		// [0] = 0 -> User command
		// [1] = 1 -> repeated start
		// [2] = 1 -> terminate frame with Stop (P)

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

LMMI Address	LMMI Write Data	Comment
		// [3] = 0 -> not CCC start
		// [4] = 0 -> I3C
		// [5] = 1 -> Wait next packet
		// [6] = 0 -> unused
		// [7] = 1 -> HDR-DDR mode
0x30	{7'h10,1'b1}	// 7b Target Address, Read
0x30	0x04	// Length = 4 bytes
0x11	0x01	// start

2.8.4. Permitted CCCs in HDR-DDR Mode

There are several CCCs that are allowed during HDR-DDR mode. Table 2.19 shows the allowed CCCs and those with limitations.

Table 2.19. Allowed CCCs During HDR-DDR Mode and Limitations

CCC	Type	Command Name	Notes and Limitations
0x00	Broadcast	ENEC	May not generally be useful in HDR-DDR Mode, as the Controller would need to exit HDR and return to SDR Mode for the Target to be able to raise any of the request types that are affected by these CCCs; See Section 5.2.1.2.4 of MIPI I3C Specification.
0x01	Broadcast	DISEC	
0x02	Broadcast	ENTAS0	See Section 5.2.1.2.4 of MIPI I3C Specification.
0x03	Broadcast	ENTAS1	
0x04	Broadcast	ENTAS2	
0x05	Broadcast	ENTAS3	
0x06	Broadcast	RSTDAA	Not permitted in HDR-DDR mode.
0x07	Broadcast	ENTDAA	Not permitted in HDR-DDR mode.
0x08	Broadcast	DEFTGTS	Not recommended for use in HDR-DDR Mode.
0x09	Broadcast	SETMWL	See Section 5.2.1.2.4 of MIPI I3C Specification.
0x0A	Broadcast	SETMRL	
0x12	Broadcast	ENDXFER	Not recommended for use in HDR-DDR Mode.
0x20	Broadcast	ENTHDR0	Not permitted in HDR-DDR mode.
0x28	Broadcast	SETXTIME	No limitations.
0x29	Broadcast	SETAASA	Not permitted in HDR-DDR mode.
0x2A	Broadcast	RSTACT	Not recommended for use in HDR-DDR Mode.
0x80	Direct	ENEC	May not generally be useful in HDR-DDR Mode, as the Controller would need to exit HDR and return to SDR Mode for the Target to be able to raise any of the request types that are affected by these CCCs; See Section 5.2.1.2.4 of MIPI I3C Specification.
0x81	Direct	DISEC	
0x82	Direct	ENTAS0	See Section 5.2.1.2.4 of MIPI I3C Specification.
0x83	Direct	ENTAS1	
0x84	Direct	ENTAS2	
0x85	Direct	ENTAS3	
0x87	Direct	SETDASA	Not permitted in HDR-DDR mode.
0x88	Direct	SETNEWDA	Not permitted in HDR-DDR mode.
0x89	Direct	SETMWL	See Section 5.2.1.2.4 of MIPI I3C Specification.
0x8A	Direct	SETMRL	

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

CCC	Type	Command Name	Notes and Limitations
0x8B	Direct	GETMWL	No limitations.
0x8C	Direct	GETMRL	
0x8D	Direct	GETPID	Not permitted in HDR-DDR mode.
0x8E	Direct	GETBCR	Not recommended for use in HDR-DDR Mode.
0x8F	Direct	GETDCR	
0x90	Direct	GETSTATUS	No limitations.
0x91	Direct	GETACCCR	Not permitted in HDR-DDR mode.
0x92	Direct	ENDXFER	Not recommended for use in HDR-DDR Mode.
0x94	Direct	GETMXDS	No limitations.
0x95	Direct	GETCAPS	No limitations.
0x98	Direct	SETXTIME	No limitations.
0x99	Direct	GETXTIME	
0x9A	Direct	RSTACT	Not recommended for use in HDR-DDR Mode.

2.8.4.1. Generic Broadcast CCC Format

Table 2.20. Generic Broadcast CCC Format

LMMI Address	LMMI Write Data	Comment
0x30	0x09	// Control
		// [0] = 1 -> CCC
		// [1] = 0 -> not repeated start
		// [2] = 0 -> do not terminate frame with Stop (P)
		// [3] = 1 -> CCC start
		// [6:4] = 0
		// [7] = 0 -> SDR mode
0x30	{7'h7E,1'b0}	// Reserved Address 0x7E, Write
0x30	0x01	// Length = 1
0x30	0x20	// ENTHDR0 CCC = 0x20
0x30	0xAF	// Control
		// [0] = 1 -> CCC
		// [1] = 1 -> repeated start
		// [2] = 1 -> terminate frame with Stop (P)
		// [3] = 1 -> CCC start
		// [4] = 0 -> I3C
		// [5] = 1 -> Wait next packet
		// [6] = 0 -> unused
		// [7] = 1 -> HDR-DDR mode
0x30	{7'h7E,1'b0}	// Reserved Address 0x7E, Write
0x30	0x03+0xNN	// Length = 1+2+ number of optional bytes // If the optional bytes are not multiple of 2 bytes, then you must provide the necessary padding.
0x30	0x00, <CCC, 0x00>, …, <n bytes>	// 0x00, <CCC, 0x00>, and Optional bytes
0x11	0x01	// start

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Table 2.21. Broadcast ENTAS0 CCC in HDR-DDR Mode Example

LMMI Address	LMMI Write Data	Comment
0x30	0x09	// Control
0x30	{7'h7E,1'b0}	// Reserved Address 0x7E, Write
0x30	0x01	// Length = 1
0x30	0x20	// ENTHDR0 CCC = 0x20
0x30	0xAF	// Control
0x30	{7'h7E,1'b0}	// Reserved Address 0x7E, Write
0x30	0x03	// Length
0x30	0x00	// {Write, user command code[6:0]==0}
0x30	0x02	// ENTAS0 CCC = 0x02
0x30	0x00	// 0 Padding (since no defining byte)
0x11	0x01	// start

Table 2.22. Broadcast SET MWL CCC in HDR-DDR Mode Example

LMMI Address	LMMI Write Data	Comment
0x30	0x09	// Control
0x30	{7'h7E,1'b0}	// Reserved Address 0x7E, Write
0x30	0x01	// Length = 1
0x30	0x20	// ENTHDR0 CCC = 0x20
0x30	0xAF	// Control
0x30	{7'h7E,1'b0}	// Reserved Address 0x7E, Write
0x30	0x05	// Length
0x30	0x00	// {Write, user command code[6:0]==0}
0x30	0x09	// SETMWL CCC = 0x09
0x30	0x00	// 0 Padding (since no defining byte)
0x30	<mwl msb>	// MWL MSB
0x30	<mwl lsb >	// MWL LSB
0x11	0x01	// start

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

2.8.4.2. Generic Direct Set CCC Format

Table 2.23. Generic Direct Set CCC Format

LMMI Address	LMMI Write Data	Comment
0x30	0x09	// Control
		// [0] = 1 -> CCC
		// [1] = 0 -> not repeated start
		// [2] = 0 -> do not terminate frame with Stop (P)
		// [3] = 1 -> CCC start
		// [6:4] = 0
		// [7] = 0 -> SDR mode
0x30	{7'h7E,1'b0}	// Reserved Address 0x7E, Write
0x30	0x01	// Length = 1
0x30	0x20	// ENTHDR0 CCC = 0x20
0x30	0xAB	// Control
		// [0] = 1 -> CCC
		// [1] = 1 -> repeated start
		// [2] = 0 -> do not terminate frame with Stop (P)
		// [3] = 1 -> CCC start
		// [4] = 0 -> I3C
		// [5] = 1 -> Wait next packet
		// [6] = 0 -> unused
		// [7] = 1 -> HDR-DDR mode
0x30	{7'h7E,1'b0}	// Reserved Address 0x7E, Write
0x30	0x03+0xNN	// Length = 1+2+ number of optional bytes // If the optional bytes are not multiple of 2 bytes, then you must provide the necessary padding.
0x30	0x00, <CCC, 0x00>, …, <n bytes>	// 0x00, <CCC, 0x00>, and Optional bytes
0x30	0xA7	// Control
		// [0] = 1 -> CCC
		// [1] = 1 -> repeated start
		// [2] = 1 -> terminate frame with Stop (P)
		// [3] = 0 -> CCC start
		// [4] = 0 -> I3C
		// [5] = 1 -> Wait next packet
		// [6] = 0 -> unused
		// [7] = 1 -> HDR-DDR mode
0x30	{<TGT_ADR>, <W>}	// 7b Target Address, Write
0x30	0x01+0xMM	// Length = 1+number of optional bytes (B) // If the optional bytes are not multiple of 2 bytes, then you must provide the necessary padding.
0x30	<m bytes>	// Optional bytes (B)
0x11	0x01	// start

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Table 2.24. Direct ENTAS0 CCC in HDR-DDR Mode Example

LMMI Address	LMMI Write Data	Comment
0x30	0x09	// Control
0x30	{7'h7E,1'b0}	// Reserved Address 0x7E, Write
0x30	0x01	// Length = 1
0x30	0x20	// ENTHDR0 CCC = 0x20
0x30	0xAB	// Control
0x30	{7'h7E,1'b0}	// Reserved Address 0x7E, Write
0x30	0x03	// Length
0x30	0x00	// {Write, user command code[6:0]==0}
0x30	0x02	// ENTAS0 CCC = 0x02
0x30	0x00	// 0 Padding (since no defining byte)
0x30	0xA7	// Control
0x30	{<TGT_ADR>, 1'b0}	// 7b Target Address, Write
0x30	0x03	// Length
0x30	0x00	// {Write, user command code[6:0]==0}
0x30	0x00	// dummy bytes since ENTASx have no data
0x30	0x00	// dummy bytes since ENTASx have no data
0x11	0x01	// start

Table 2.25. Direct SET MWL CCC in HDR-DDR Mode Example

LMMI Address	LMMI Write Data	Comment
0x30	0x09	// Control
0x30	{7'h7E,1'b0}	// Reserved Address 0x7E, Write
0x30	0x01	// Length = 1
0x30	0x20	// ENTHDR0 CCC = 0x20
0x30	0xAF	// Control
0x30	{7'h7E,1'b0}	// Reserved Address 0x7E, Write
0x30	0x03	// Length
0x30	0x00	// {Write, user command code[6:0]==0}
0x30	0x89	// SETMWL CCC = 0x89
0x30	0x00	// 0 Padding (since no defining byte)
0x30	0xA7	// Control
0x30	{<TGT_ADR>, 1'b0}	// 7b Target Address, Write
0x30	0x03	// Length
0x30	0x00	// {Write, user command code[6:0]==0}
0x30	<mwl msb>	// MWL MSB
0x30	< mwl lsb >	// MWL LSB
0x11	0x01	// start

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

2.8.4.3. Generic Direct Get CCC Format

Table 2.26. Generic Direct Get CCC Format

LMMI Address	LMMI Write Data	Comment
0x30	0x09	// Control
		// [0] = 1 -> CCC
		// [1] = 0 -> not repeated start
		// [2] = 0 -> do not terminate frame with Stop (P)
		// [3] = 1 -> CCC start
		// [6:4] = 0
		// [7] = 0 -> SDR mode
0x30	{7'h7E,1'b0}	// Reserved Address 0x7E, Write
0x30	0x01	// Length = 1
0x30	0x20	// ENTHDR0 CCC = 0x20
0x30	0xAB	// Control
		// [0] = 1 -> CCC
		// [1] = 1 -> repeated start
		// [2] = 0 -> do not terminate frame with Stop (P)
		// [3] = 1 -> CCC start
		// [4] = 0 -> I3C
		// [5] = 1 -> Wait next packet
		// [6] = 0 -> unused
		// [7] = 1 -> HDR-DDR mode
0x30	{7'h7E,1'b0}	// Reserved Address 0x7E, Write
0x30	0x03+0xNN	// Length = 1+2+ number of optional bytes // If the optional bytes are not multiple of 2 bytes, then you must provide the necessary padding.
0x30	0x00, <CCC, 0x00>, …, <n bytes>	// 0x00, <CCC, 0x00>, and Optional bytes
0x30	0xA3	// Control
		// [0] = 1 -> CCC
		// [1] = 1 -> repeated start
		// [2] = 0 -> do not terminate frame with Stop (P)
		// [3] = 0 -> CCC start
		// [4] = 0 -> I3C
		// [5] = 1 -> Wait next packet
		// [6] = 0 -> unused
		// [7] = 1 -> HDR-DDR mode
0x30	{<TGT_ADR>, <W>}	// 7b Target Address, Write
0x30	0x01	// Length = 1
0x30	0x80	// 0x80 – read command
0x30	0xA7	// Control
		// [0] = 1 -> CCC
		// [1] = 1 -> repeated start
		// [2] = 1 -> terminate frame with Stop (P)

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

LMMI Address	LMMI Write Data	Comment
		// [3] = 0 -> CCC start
		// [4] = 0 -> I3C
		// [5] = 1 -> Wait next packet
		// [6] = 0 -> unused
		// [7] = 1 -> HDR-DDR mode
0x30	{<TGT_ADR>, <R>}	// 7b Target Address, Read
0x30	0xMM	// Length = number of read bytes
0x11	0x01	// start

Table 2.27. Direct GET MWL CCC in HDR-DDR Mode Example

LMMI Address	LMMI Write Data	Comment
0x30	0x09	// Control
0x30	{7'h7E,1'b0}	// Reserved Address 0x7E, Write
0x30	0x01	// Length = 1
0x30	0x20	// ENTHDR0 CCC = 0x20
0x30	0xAB	// Control
0x30	{7'h7E,1'b0}	// Reserved Address 0x7E, Write
0x30	0x03	// Length
0x30	0x00	// {Write, user command code[6:0]==0}
0x30	0x8B	// GETMWL CCC = 0x8B
0x30	0x00	// 0 Padding (since no defining byte)
0x30	0xA3	// Control
0x30	{<TGT_ADR>, 1'b0}	// 7b Target Address, Write
0x30	0x01	// Length = 1
0x30	0x80	// {Read, user command code[6:0]==0}
0x30	0xA7	// Control
0x30	{<TGT_ADR>, 1'b1}	// 7b Target Address, Read
0x30	0x02	// Length
0x11	0x01	// start

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

2.9. Hot-Join and In-Band Interrupt (IBI) Handling

I3C Controller provides an interrupt status for Hot-Join and IBI. If the corresponding interrupt bits are enabled, then the reception of Hot-Join or IBI would cause the assertion of interrupt signal int_o. The target address of interrupt requester can be retrieved by reading IBI address register (0x1F). Figure 2.18 illustrates the process flow of IBI and Hot-Join handling of the I3C Controller.

The response of the I3C controller will depend on the configuration setting ibi_auto_resp. The ibi_auto_resp register can be set to high to prioritize the completion of ongoing transactions. The I3C controller will always NAK the IBI or Hot-Join while the Tx FIFO is not empty. When the Tx FIFO is empty, waiting_ibi_resp interrupt will be asserted. You should program or set the IBI read count register (0x1D), and the IBI response register (0x1E).

When ibi_auto_resp register is set to low or when Tx FIFO is empty, the I3C controller will pause the SCL (hold low) during ACK phase and wait for your response. You should program or set the IBI read count register (0x1D) and the IBI response register (0x1E) when waiting_ibi_resp is asserted. For Hot-Join interrupt, IBI read count (0x1D) should be set to 0. Write access on IBI response register (0x1E) will trigger the I3C controller to resume the SCL clock and forward the indicated response (ACK or NAK). For IBI, the I3C controller will read the mandatory byte and optional bytes depending on the setting of IBI read count (0x1D) and will be stored in the Rx FIFO.

If the ibi_auto_resp register is set to low and there is ongoing transaction (Tx FIFO is not empty), I3C controller will stop the ongoing transaction to prioritize the interrupt. Once the handling of interrupt is done, the I3C controller will retry the previous transaction.

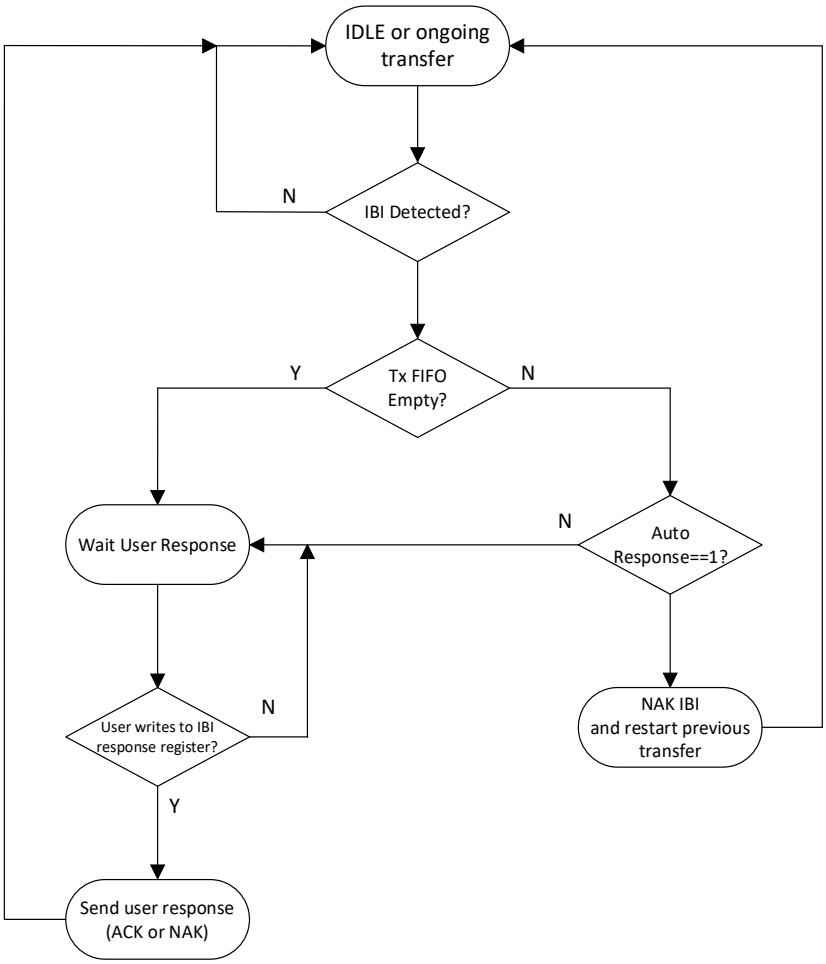


Figure 2.18. I3C Controller Handling of IBI/Hot-Join

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

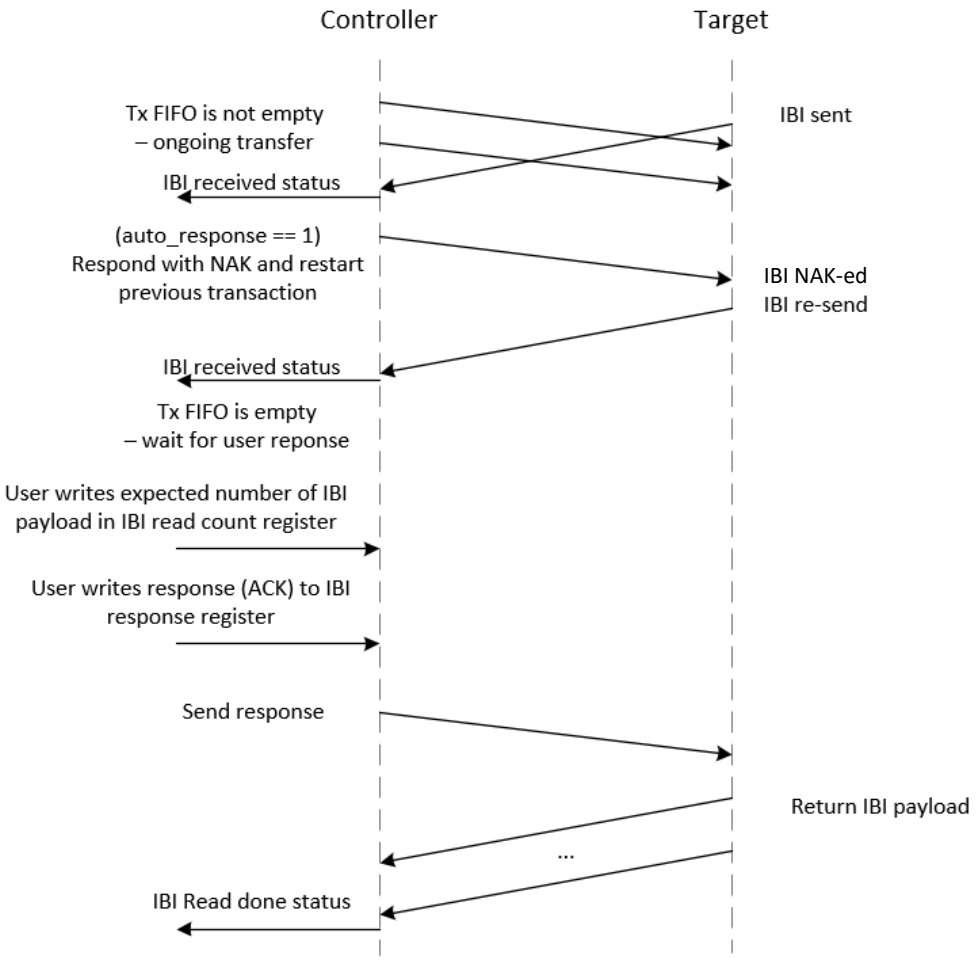


Figure 2.19. Sample Sequence

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

2.10. Controller to Controller Handoff

This is only applicable if the Secondary Controller Capable attribute is checked in the IP configuration GUI.

The Active controller role can be transferred to another controller as described in the MIPI I3C Specification.

- After the Active Controller has prepared for handoff, the Active Controller shall then issue a GETACCCR CCC (section 5.1.9.3.16) followed by a STOP if the handoff was successful, whereupon it shall release control of the SCL line, and therefore also releasing control of the I3C bus (that is, passing the Controller Role) to the selected Secondary Controller.
- Following the STOP and Bus Available Condition, the selected Secondary Controller assumes the role of the Active Controller and takes control of the I3C bus. The former Active Controller should monitor the I3C bus to make sure that the new Active Controller asserts its Controller Role, according to the flow described in section 5.1.10.2.4 regarding Error Type CE3.

Before sending the GETACCCR CCC, you must configure the Secondary Controller Timeout register (0x19-0x17) with the appropriate timeout value. This timeout is used when checking the new Active Controller, if it asserts its role. If the timeout expires before the new Active Controller assert its role (failing to drive SCL after SDA goes low), then the handoff has failed, and the former Active Controller remains as the Active Controller.

You may read the interrupt status 1 register (0x24) to get the result of the handoff process. Additional information is also available in the Secondary Controller Status Info register.

2.11. Mixed Bus Mode

When it is expected that there are both I3C and I2C devices on the same bus, you can configure the Lattice I3C Controller IP in two different ways:

- Set the SCL pulse width in the system clock divider register (0x01) to produce the SCL frequency that can support the slowest I2C device.
- Enable the dynamic I3C-I2C clock switching in IP configuration GUI and set the appropriate I2C SCL pulse width based on the slowest device.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

3. IP Parameter Description

The configurable attributes of the I3C Controller IP are shown in the following tables. You can configure the IP by setting the attributes accordingly in the IP Catalog's Module/IP wizard of the Lattice Radiant software.

Wherever applicable, default values are in **bold**.

3.1. General

Table 3.1. General Attributes

Attribute	Selectable Values	Description
Bus Characteristics		
Device Role	**Primary Controller**, Secondary Controller	Select the device role. Secondary Controller option is available only when Secondary Controller Capability is enabled.
Secondary Controller Capable	Checked, **Unchecked**	Determines whether IP can support Secondary Controller Capability.
IBI Capable	Checked, **Unchecked**	Determines whether IP can support IBI Capability. Automatically checked when Secondary Controller Capability is enabled.
Hot-Join Capable	Checked, **Unchecked**	Determines whether IP can support Hot-Join Capability. Automatically checked when Secondary Controller Capability is enabled.
Data Rate Mode	**SDR only**, HDR-DDR-capable	Determines whether IP can support HDR-DDR mode or SDR only.
Register Interface		
Interface	LMMI, **APB**, AHBL	Select the bus interface for accessing registers and FIFO.
Register Offset	Address offset in Bytes, **Address offset in DWORD**	Select the addressing mode of the registers. Selectable only for APB and AHB-Lite interfaces. When connecting to a RISC-V processor, use *Address offset in DWORD.* If the interface is LMMI, this value is fixed to *Address offset in Bytes*.
FIFO		
FIFO Implementation	**EBR**, LUT, HARD_IP	Selects the FPGA resource that are used to implement the FIFO.
FIFO Depth	64, 128, 256, **512**, 1024	Specifies the number of FIFO levels. Accepts only power of two values.
Optional Interface		
Enable Direct FIFO Interface	Checked, **Unchecked**	Select to provide a separate interface for (FIFO) data path.
Tx Data Width	—	Display only when Direct FIFO Interface is enabled. Default value is 8.
Rx Data Width	—	Display only when Direct FIFO Interface is enabled. Default value is 8.
SCL Clocking		
Bus Clock Domain	—	Display only. Default value is ASYNC.
Use Internal Clock Divider	Checked, **Unchecked**	Select to enable the use of internal clock divider.
System Clock Frequency (MHz) (clk_i)	0.8–50	Set the system clock (clk_i) frequency. The recommended and default value is 25. Refer to the Clocking Overview section for more details.
Core Clock Frequency (MHz)	0.02–25	Set the core clock (src_clk_scl_i) frequency. Selectable only when Use Internal Clock Divider is Unchecked. Default value is 25. Refer to Clocking Overview section for more details.
SCL Pulse Width	1–256	Set the SCL pulse width in terms of number of cycles of system clock (clk_i). Applicable only if the Use Internal Clock Divider attribute is enabled. Default value is 1. Refer to Clocking Overview section for more details.
SCL Clock Frequency (MHz)	—	Display only. Default value is 12.5. Refer to the Clocking Overview section for more details.
SCL Clock Period (ns)	—	Display only. Default value is 80.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Attribute	Selectable Values	Description
Open Drain Pulse Width (Number of SCL cycles)	0–16	Set the open drain clock pulse width in terms of number of cycles of the normal SCL. Default value is 3. The I3C specification requires a minimum open-drain pulse width of 200 ns. In open-drain mode, the clock pulse is extended. For example, if the SCL clock period is 80 ns, the open-drain pulse width must be at least 3 SCL clock cycles to produce 240 ns, which satisfies the 200 ns minimum: • If *Use Internal Clock Divider* is unchecked, the acceptable range is 1 to 16. • If *Use Internal Clock Divider* is checked, the acceptable range is dynamic and depends on the SCL clock period. If the SCL clock period is at least twice the minimum open-drain pulse width (≥400 ns), then this parameter can be set to 0. Otherwise, the minimum acceptable value is the number of SCL clock cycles required to produce a pulse width greater than 200 ns.
Open Drain Clock Frequency (MHz)	—	Display only. Equal to *1 / (Open Drain Pulse width * 2)*. Default value is 2.0833.
Open Drain Pulse width (ns)	—	Display only. *Equal to SCL Clock Period * Open Drain Pulse Width.* Default value is 240.
Enable Dynamic I3C – I2C clock switching	**Checked**, Unchecked	Select to enable dynamic clock switching between I3C and I2C. This is used when you want to use a different SCL clock frequency for I2C.
I2C SCL Pulse width	1–256	Set the SCL pulse width in terms of number of cycles of the system clock (clk_i). Applicable only if dynamic I3C – I2C clock switching is enabled. Default value is 13.
I2C SCL Clock Frequency (MHz)	—	Display only when Dynamic I3C – I2C clock switching is enabled. Default value is 0.961538.
I2C SCL Clock Period (ns)	—	Display only when Dynamic I3C – I2C clock switching is enabled. Default value is 520.
I/O Primitive		
Include IO Primitive	**Checked**, Unchecked	Option to include or remove the I/O primitive instance inside the IP. This means that SDA and SCL are seen as bidirectional I/O ports at the top level. This option is enabled by default. Some FPGA devices might not have an I/O that supports I3C. In that case, you must disable this option and provide a custom I/O. When this option is disabled, the SDA and SCL I/O control signals (including pull up resistor controls) are exposed at the top level as ports.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

3.2. Secondary Controller

Refer to the I3C Target IP User Guide (FPGA-IPUG-02227) for more information on the attributes below.

Table 3.2. Secondary Controller (Available only if Secondary Controller Capable is Enabled)

Attribute	Selectable Values	Description
IBI Payload Size (including MDB)	0–255	Payload size (including MDB) following an IBI. If set to 0, no payload follows the IBI. Default value is 1. Applicable only when IBI Capable = True.
Maximum Data Speed Limitation	**Checked**, Unchecked	Sets Bit 0 of BCR. When set to 1, Target device supports GETMXDS CCC to relay timing information to the Controller.
DCR (HEX)	00–FF	Device Characteristics Register. It is recommended to assign a value to this register according to the list of the devices, which can be found at I3C Device Characteristics Register. Default value is 00.
Manufacturer ID	0–32767	Defines Provisional ID[47:33] bits. Default value is 414.
Part ID	0–65535	Defines Provisional ID[31:16] bits. Default value is 1.
Instance ID	0–15	Defines Provisional ID[15:12] bits. Default value is 1.
Additional ID	0–4095	Defines Provisional ID[11:0] bits. Default value is 0.
Static Address Enable	**Checked**, Un- checked	Indicates that the Target device may be assigned a Static Address.
Static Address (HEX)[1]	00–7F	Sets device static address. Default value is 08. Applicable only when Static Address enable = True.
Dynamic Address (HEX)[1]	00–7F	Sets initial device dynamic address. Default value is 08. Applicable only when Secondary Controller Capable = True and Device Role = Primary Controller.
Timing Characteristics		
Write Maximum Data Rate (MHz)	—	Display only when Maximum Data Speed Limitation = True. Indicates maximum sustained data rate for non-CCC Messages sent by Controller Device to Target Device. Default value is 12.5.
Clock-to-data Turnaround Delay (ns) (t_{SCO})	—	Display only when Maximum Data Speed Limitation = True. Clock-to-Data Turnaround Time. Default value is ≤8.
Read Maximum Data Rate (MHz)	—	Display only when Maximum Data Speed Limitation = True. Indicates maximum sustained data rate for non-CCC Messages sent by Target Device to Controller Device. Default value is 12.5.
Maximum Read Turnaround Time (µs)	—	Display only when Maximum Data Speed Limitation = True. Indicates how long the Controller needs to wait before reading the data it requested. Default value is 0.

Note:
1. Static and dynamic address must not use I2C reserved and I3C invalid addresses.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

3.3. IP Parameter Settings for Example Use Cases

Table 3.3 shows the parameter settings for example test cases. Wherever applicable, parameters equal to "—" are automatically set in IP configuration GUI.

Table 3.3. IP Parameter Settings for Example Test Cases

Target Use Case	I3C Controller Only	Secondary Controller Capable	Secondary Controller Capable (External clock source for SCL)
Bus Characteristics			
Device Role	Primary Controller	• Primary Controller (initialize as Controller) • Secondary Controller (initialize as Target)	• Primary Controller (initialize as Controller) • Secondary Controller (initialize as Target)
Secondary Controller Capable	Unchecked	Checked	Checked
IBI Capable	Checked	—	—
Hot-Join Capable	Checked	—	—
Data Rate Mode	SDR only OR HDR-DDR-capable	SDR only OR HDR-DDR-capable	SDR only OR HDR-DDR-capable
Register Interface			
Interface	LMMI or APB or AHBL	LMMI or APB or AHBL	LMMI or APB or AHBL
Register Offset	Address offset in DWORD	Address offset in DWORD	Address offset in DWORD
Optional Interface			
Enable Direct FIFO Interface	Unchecked	Unchecked	Unchecked
Tx Data Width	—	—	—
Rx Data Width	—	—	—
SCL Clocking			
Bus Clock Domain	—	—	—
Use Internal Clock Divider	Checked	Checked	Unchecked
System Clock Frequency (MHz)	25	25	50
Internal Clock Frequency (MHz)	—	—	N/A
Core Clock Frequency	N/A	N/A	25
SCL Pulse width	1	1	N/A
SCL Clock Frequency (MHz)	—	—	—
SCL Clock Period (ns)	—	—	N/A
Open Drain Pulse Width (Number of SCL cycles)	3	3	3
Open Drain Clock Frequency (MHz)	—	—	—
Open Drain Pulse width (ns)	—	—	—
Enable Dynamic I3C – I2C clock switching	Checked	Checked	Checked
I2C SCL Pulse width	13	13	13
I2C SCL Clock Frequency (MHz)	—	—	—
I2C SCL Clock Period (ns)	—	—	—
I/O Primitive			
Include IO Primitive	Checked	Checked	Checked
Attribute			
IBI Payload Size (including MDB)	N/A	2	2
Maximum Data Speed Limitation	N/A	Checked	Checked
DCR (HEX)	N/A	0	0
Manufacturer ID	N/A	414	414

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Target Use Case	I3C Controller Only	Secondary Controller Capable	Secondary Controller Capable (External clock source for SCL)
Part ID	N/A	1	1
Instance ID	N/A	1	1
Additional ID	N/A	0	0
Static Address Enable	N/A	Checked	Checked
Static Address (HEX)[1]	N/A	08	08
Dynamic Address (HEX)[1]	N/A	08	08

Note:

1. Static and dynamic address must not use reserved I2C addresses.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

4. Signal Description

Table 4.1 lists the input and output signals for I3C Controller IP along with their descriptions.

Table 4.1. Ports Description

Port	Type	Description
System Clock and Reset		
clk_i	Input	System clock. Refer to the Clocking Overview section for more details.
src_clk_scl_i	Input	Core clock. Applicable only when Use Internal Clock Divider is disabled. Refer to the Clocking Overview section for more details.
rst_n_i	Input	Asynchronous active low reset. Refer to the Reset section for more details.
sc_rst_o	Output	Applicable only when Secondary Controller capability is enabled. Active High flag for Full Chip Reset by Target Reset escalation. For more details about target reset action see Section 5.1.11 of the MIPI I3C Basic v1.1.1 Specification.
I3C Interface (Internal I/O Primitive)[1]		
scl_io	Input/Output	Bidirectional I3C Serial Clock. Refer to the Clocking Overview section for more details.
sda_io	Input/Output	Bidirectional I3C Serial Data
I3C Interface (External I/O Primitive)[2]		
ext_io_scl_i	Input	I3C Serial Clock Input
ext_io_scl_o	Output	I3C Serial Clock Output
ext_io_scl_oe	Output	Active High I3C Serial Clock Output Enable
ext_io_scl_spu_n	Output	Active Low I3C Serial Clock Strong Pull-Up Enable
ext_io_scl_wpu_n	Output	Active Low I3C Serial Clock Weak Pull-Up Enable
ext_io_sda_i	Input	I3C Serial Data Input
ext_io_sda_o	Output	I3C Serial Data Output
ext_io_sda_oe	Output	Active High I3C Serial Data Output Enable
ext_io_sda_spu_n	Output	Active Low I3C Serial Data Strong Pull-Up Enable
ext_io_sda_wpu_n	Output	Active Low I3C Serial Data Weak Pull-Up Enable
LMMI Interface[3]		
lmmi_request_i	Input	Start transaction
lmmi_wr_rdn_i	Input	Write = HIGH, Read = LOW
lmmi_offset_i[7:0]	Input	Register byte offset
lmmi_wdata_i[7:0]	Input	Write data
lmmi_rdata_o[7:0]	Output	Read data
lmmi_rdata_valid_o	Output	The read transaction is complete and lmmi_rdata_o contains valid data.
lmmi_ready_o	Output	The secondary is ready to start a new transaction. Secondary can insert wait states by holding this signal low.
Interrupt Interface		
int_o	Output	Level sensitive active high interrupt signal. This signal will assert when any of the enabled interrupt status is asserted.
AHB-Lite Interface[4]		
ahbl_hsel_i	Input	AHB-Lite Select signal
ahbl_hready_i	Input	AHB-Lite Ready Input signal
ahbl_haddr_i[31:0]	Input	AHB-Lite Address signal. Refer to the Register Description section for usage depending on the Register Offset parameter.
ahbl_hburst_i[2:0]	Input	AHB-Lite Burst Type signal. Only burst type 0 is supported.
ahbl_hsize_i[2:0]	Input	AHB-Lite Transfer Size signal. 1-byte RW is supported (SIZE = 3'd0).
ahbl_htrans_i[1:0]	Input	AHB-Lite Transfer Type signal

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Port	Type	Description
ahbl_hwrite_i	Input	AHB-Lite Direction signal. Write = High, Read = Low
ahbl_hwdata_i[31:0]	Input	AHB-Lite Write Data signal. Tie Bits[31:24] to 0. Refer to the Register Description section for more details.
ahbl_hreadyout_o	Output	AHB-Lite Ready Output signal
ahbl_hrdata_o[31:0]	Output	AHB-Lite Read Data signal. Bits[31:24] are tied to 0. Refer to the Register Description section for more details.
ahbl_hresp_o	Output	AHB-Lite Transfer Response signal
ahbl_hmastlock_i	Input	Not supported. Tie to 0.
ahbl_hprot_i[3:0]	Input	Not supported. Tie to 0.
APB Interface[5]		
apb_paddr_i[31:0]	Input	APB Address signal. Refer to the Register Description section for usage depending on the Register Offset parameter.
apb_psel_i	Input	APB Select signal
apb_penable_i	Input	APB Enable signal
apb_pwrite_i	Input	APB Direction signal
apb_pwdata_i[31:0]	Input	APB Write Data signal. Tie Bits[31:24] to 0. Refer to the Register Description section for more details.
apb_pready_o	Output	APB Ready signal
apb_prdata_o[31:0]	Output	APB Read Data signal. Bits[31:24] are tied to 0. Refer to the Register Description section for more details.
apb_pslverr_o	Output	APB Error signal
Direct FIFO Interface[6]		
tx_valid_i	Input	Tx data valid signal. Set this to high when sending data (tx_data_i) to Tx FIFO.
tx_ready_o	Output	Tx FIFO ready signal. When high, it means Tx FIFO can accept incoming data. Tx_data_i will be written to Tx FIFO when tx_valid_i and tx_ready_o == 1.
tx_data_i [7:0]	Input	Tx Data
rx_valid_o	Output	Rx data valid signal. When high, it means Rx FIFO is not empty and rx_data_o contains valid data.
rx_ready_i	Input	Rx ready signal. Set this to high when it is ready to accept the Rx data. Rx_data_o will take the next Rx FIFO entry when rx_valid_o and rx_ready_i == 1.
rx_data_o[7:0]	Output	Rx data

Notes:

1. Bidirectional I3C interface is only available when internal I/O primitives is checked in the IP configuration GUI.
2. External I/O I3C interface is only available when internal I/O primitives is unchecked in IP configuration GUI.
3. LMMI Interface is only available when selected from the Register Interface in IP configuration GUI.
4. AHB-Lite Interface is only available when selected from the Register Interface in IP configuration GUI. Refer to the AMBA 3 AHB-Lite Protocol v1.0 Specification for details of the protocol.
5. APB Interface is only available when selected from the Register Interface in IP configuration GUI. Refer to the AMBA 3 APB Protocol v1.0 Specification for details of the protocol.
6. Direct FIFO interface is only available when enabled in IP configuration GUI.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

5. Register Description

This section defines the configuration, control, and status registers of the I3C Controller IP. The total address space of the IP is 1 KiB.

Table 5.1 lists the register access types. Table 5.2 shows the mapping of registers and register groups to their addresses. These registers are accessible through the user interface (LMMI, APB, or AHB-Lite).

The address map uses DWORD addressing by default, which is available only when the user interface is APB or AHB-Lite. There is an option to convert the addressing to byte addressing. When byte addressing is selected, the register address shifts right by 2.

Example: APB/AHB-Lite DWORD offset = 10'h004 or {8'h01, 2'b00} → LMMI byte offset {8'h01}

If the selected interface is APB or AHB-Lite, both the data and address ports are 32 bits wide. For input data (ahbl_hwdata_i and apb_pwdata_i), the upper 24 bits are unused. For output data (ahbl_hrdata_o and apb_prdata_o), the upper 24 bits are set to 0. For address inputs (ahbl_haddr_i and apb_paddr_i), the upper 24 bits (in byte addressing) or upper 22 bits (in DWORD addressing) are unused.

When secondary controller capability is enabled, the I3C Controller IP includes I3C target functionality.

Registers related to the I3C target are defined in the I3C Target IP User Guide (FPGA-IPUG-02227). These registers are mapped with an address offset of 0x80 (add 0x80 to the register address).

Table 5.1. Register Access Types

Access Type	Abbreviation	Behavior on Read Access	Behavior on Write Access
Read only	RO	Returns register value.	Ignores write access.
Write only	WO	Returns 0.	Updates register value.
Read and write	RW	Returns register value.	Updates register value.
Read and write 1 to clear	RW1C	Returns register value.	Writing 1'b1 on register bit clears the bit to 1'b0. Writing 1'b0 on register bit is ignored.

Table 5.2. I3C Controller IP Register Summary

Byte Address Offset[1]	DWORD Address Offset[1]	Register Name	Description
0x01	0x004	System Clock Divider	System clock divider for SCL clock generation.
0x02	0x008	Controller Configuration 0	Controller configuration.
0x03	0x00C	Open Drain Timer	Number of SCL clock cycles to generate open-drain clock pulse width.
0x04	0x010	I2C Clock Divider	System clock divider for SCL clock generation in I2C mode.
0x05	0x014	Pull Up Resistor Disable	SDA and SCL pull-up resistor configuration.
0x06	0x018	HDR-DDR Configuration 0	Controller configuration for HDR-DDR mode.
0x08	0x020	Soft Reset	Soft reset.
0x11	0x044	Transmit Start	Transmit start signal to initiate processing of Tx FIFO contents.
0x15	0x054	Secondary Controller Status Info	Secondary controller status information.
0x16	0x058	Set Device Role	Manually set the device role.
0x17 0x18 0x19	0x05C 0x060 0x064	Secondary Controller Timeout	Controller role handoff timeout value.
0x1C	0x070	Dynamic Address ACKed Counter	Number of dynamic addresses that acknowledged during the Enter DAA command.
0x1D	0x074	IBI Read Count	Number of bytes to read from the target during IBI.
0x1E	0x078	IBI Response	Response to IBI or Hot-Join request.
0x1F	0x07C	IBI Address	Address of the I3C target or secondary controller that requested the IBI.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Byte Address Offset[1]	DWORD Address Offset[1]	Register Name	Description
0x20	0x080	Interrupt Status 0	Interrupt status 0.
0x21	0x084	Interrupt Set 0	Interrupt set 0.
0x22	0x088	Interrupt Enable 0	Interrupt enable 0.
0x24	0x090	Interrupt Status 1	Interrupt status 1.
0x25	0x094	Interrupt Set 1	Interrupt set 1.
0x26	0x098	Interrupt Enable 1	Interrupt enable 1.
0x28	0x0A0	Bus Condition Debug	Bus condition information for debug.
0x29	0x0A4	Last NAK Address Debug	Last NAK target address.
0x2A	0x0A8	Last ACK Address Debug	Last ACK target address.
0x2C	0x0B0	Interrupt Status 2	Interrupt status 2.
0x2D	0x0B4	Interrupt Set 2	Interrupt set 2.
0x2E	0x0B8	Interrupt Enable 2	Interrupt enable 2.
0x30	0x0C0	Tx FIFO	Transmit FIFO.
0x40	0x100	Rx FIFO	Receive FIFO.
0x41 – 0x7F	0x104 – 0x3FC	Reserved	Reserved.

Note:

1. Skipped addresses in the total address space are reserved. The access type is read-only.

5.1. System Clock Divider Register 0x01

System clock divider for SCL clock generation. This register is applicable only when the *Use Internal Clock Divider* attribute is checked in the IP configuration GUI.

Table 5.3. System Clock Divider Register 0x01

Field	Name	Access	Description	Reset
[7:0]	sys_clk_div	RW	Used for SCL clock generation. This setting indicates the clock pulse width of the generated SCL signal in terms of the number of system clock (clk_i) cycles: • 0 – 1 clk_i clock cycle • 1 – 2 clk_i clock cycles • … • 255 – 256 clk_i clock cycles **Example:** If the clk_i frequency is 25 MHz and sys_clk_div == 0 (Actual Divider = sys_clk_div + 1), then the SCL frequency is 12.5 MHz, since the pulse width of the generated SCL is 1 cycle of clk_i.	Takes the *SCL Pulse Width* parameter value – 1.

5.2. Controller Configuration 0 Register 0x02

Controller configuration.

Table 5.4. Controller Configuration 0 Register 0x02

Field	Name	Access	Description	Reset
[7]	en_ack_handoff	RW	Applicable only when In-Band interrupt capability is enabled. Configuration options: • 1 – Controller releases SDA at the SCL rising edge during the ACK phase of IBI.	0x0

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Field	Name	Access	Description	Reset
			• 0 – Controller releases SDA at the next SCL falling edge during the ACK phase of IBI.	
[6]	auto_assert_role	RW	Applicable only when secondary controller capability is enabled. Configuration options: • 1 – Enables auto-assert role. The device automatically asserts its role when taking over as the new active controller after the GETACCCR CCC. The device sends the broadcast address {0x7E, W} to assert its role. • 0 – Disables auto-assert role. The device waits until another device pulls the SDA low, or until data is written to the Tx FIFO and 1 is written to the tx_start register.	0x0
[5]	ibi_auto_resp	RW	Applicable only when In-Band interrupt capability is enabled. Configuration options: • 1 – Always NAK the IBI request and continue the current transfer until the Tx FIFO is empty. • 0 – Pause the SCL on low during the ACK phase and wait for you to program the appropriate response. For more details, refer to the *ibi_rcnt* and *ibi_resp* registers.	0x1
[4]	ignore_cmd_done	RW	Configuration options: • 1 – The I3C controller continues orocessing the contents of the Tx FIFO until it is empty. • 0 – The I3C controller stops after transferring one packet. The start register must be set/program again to resume the transfer.	0x0 0x1 – if the optional FIFO interface is enabled.
[3]	i2c_mode_allowed	RW	Configuration options: • 1 – Allows transfer of I2C frames. • 0 – Only I3C frames can be transferred.	0x0
[2]	ignore_rcvd_nak	RW	Configuration options: • 1 – The IP will continue to process the contents of the Tx FIFO (until empty) even if a NAK was received. • 0 – The I3C controller will stop after receiving a NAK and you will have to set/program the start register again to resume the transfer.	0x0 0x1 – if the optional FIFO interface is enabled.
[1]	en_daa_uid_in_rxfifo	RW	During dynamic address assignment, I3C targets broadcast their unique IDs (48 bits), including BCR and DCR, as part of the arbitration process. Configuration options: • 1 – The unique IDs, BCR, and DCR are stored in the Rx FIFO, which you can read. • 0 – The information is discarded.	0x0
[0]	i3c_priv_rw_no_7e	RW	Configuration options: • 1 – Starts private write/read access using the dynamic address instead of the I3C broadcast address 0x7E. • 0 – Starts private access with the 0x7E address, followed by a repeated start (Sr), then the target's dynamic address.	0x0

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

5.3. Open Drain Timer Register 0x03

Number of SCL clock cycles to generate open-drain clock pulse width.

Table 5.5. Open Drain Timer Register 0x03

Field	Name	Access	Description	Reset
[7:4]	Reserved	RO	Reserved.	0x0
[3:0]	od_timer	RW	Used for SCL clock generation in open-drain mode of I3C transaction. This setting indicates the clock pulse width of the open-drain SCL clock in terms of the number of normal SCL clock cycles, as generated using the sys_clk_div setting: • 0 – 1 SCL clock cycle. Uses the normal SCL clock generated by the system clock divider. • 1 – 2 SCL clock cycles • … • 15 – 16 SCL clock cycles **Example:** If the normal SCL frequency is 12.5 MHz and od_timer == 3, then the open-drain SCL frequency is 2.083 MHz, since the pulse width of the generated SCL is 3 cycles of the normal SCL.	Takes the *Open Drain Pulse Width* parameter value.

5.4. I2C Clock Divider Register 0x04

System clock divider for SCL clock generation in I2C mode. This register is applicable only when the *Use Internal Clock Divider* attribute is checked in the IP configuration GUI.

Table 5.6. I2C Clock Divider Register 0x04

Field	Name	Access	Description	Reset
[7:0]	i2c_clkdiv	RW	Used for SCL clock generation in I2C mode. This setting indicates the clock pulse width of the SCL signal in terms of the number of system clock (clk_i) cycles: • 0 – 1 clk_i clock cycle • 1 – 2 clk_i clock cycles • … • 255 – 256 clk_i clock cycles **Example:** If the clk_i frequency is 50 MHz and i2c_clkdiv == 24 (Actual Divider = i2c_clkdiv + 1), then the SCL frequency in I2C mode is 1 MHz, since the pulse width of the generated SCL is 25 cycles of clk_i.	Takes the *I2C SCL Pulse Width* parameter value – 1.

5.5. Pull Up Resistor Disable Register 0x05

SDA and SCL pull-up resistor configuration. This register is used to configure the SDA and SCL pull-up resistors.

Table 5.7. Pull Up Resistor Disable Register 0x05

Field	Name	Access	Description	Reset
[7:5]	Reserved	RO	Reserved.	0x0
[3]	scl_spu_n	RW	When set to high, disables the strong pull-up resistor on SCL.	0x1

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Field	Name	Access	Description	Reset
[2]	scl_wpu_n	RW	When set to high, disables the weak pull-up resistor on SCL.	0x0 0x1 – if secondary-capable but not the active controller.
[1]	sda_spu_n	RW	When set to high, disables the strong pull-up resistor on SDA.	0x0 0x1 – if secondary-capable but not the active controller.
[0]	sda_wpu_n	RW	When set to high, disables the weak pull-up resistor on SDA.	0x1

5.6. HDR-DDR Configuration 0 Register 0x06

Controller configuration for HDR-DDR mode. This register is applicable only when the *Data Rate Mode* attribute is set to HDR-DDR-capable in the IP configuration GUI.

Table 5.8. HDR-DDR Configuration 0 Register 0x06

Field	Name	Access	Description	Reset
[7:5]	Reserved	RO	Reserved.	0x0
[4]	ddr_ignore_rcvd_nak	RW	Applicable only when HDR-DDR capability is enabled. Configuration options: • 1 – The IP continues to process the contents of the Tx FIFO (until empty), even if a NAK is received. • 0 – The I3C controller stops after receiving a NAK. The tx_start register must be set/program again to resume the transfer.	0x0
[3]	ddr_ignore_cmd_done	RW	Applicable only when HDR-DDR capability is enabled. Configuration options: • 1 – The I3C controller continues to process the contents of the Tx FIFO until it is empty. • 0 – The I3C controller stops after transferring one packet. The tx_start register must be set/program again to resume the transfer.	0x1
[2]	no_crc_after_term	RW	Applicable only when HDR-DDR capability is enabled. Configures the data transfer ending procedure: • 1 – No CRC word follows the early termination request. • 0 – A CRC word follows the early termination request.	0x1
[1]	en_wr_early_term	RW	Applicable only when HDR-DDR capability is enabled. Configures the data transfer ending procedure: • 1 – Enables the write early termination request. • 0 – Disables the write early termination request.	0x1
[0]	en_wrcmd_acknak_cap	RW	Applicable only when HDR-DDR capability is enabled. Configures the data transfer ending procedure: • 1 – Enables the ACK/NAK capability for write command. • 0 – Disables the ACK/NAK capability for write command.	0x1

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

5.7. Soft Reset Register 0x08

Soft reset.

Table 5.9. Soft Reset Register 0x08

Field	Name	Access	Description	Reset
[7:5]	Reserved	RO	Reserved.	0x0
[4]	ip_csr_rst	RW	When set to high, resets the IP registers only. Auto-clear.	0x0
[3]	ip_core_rst	RW	When set to high, resets the I3C Controller only. Auto-clear.	0x0
[2]	tx_fifo_rst	RW	When set to high, resets the Tx FIFO only. Auto-clear.	0x0
[1]	rx_fifo_rst	RW	When set to high, resets the Rx FIFO only. Auto-clear.	0x0
[0]	ip_main_rst	RW	When set to high, resets the entire IP, including registers and FIFO. Auto-clear.	0x0

5.8. Transmit Start Register 0x11

Transmit start signal to initiate processing of Tx FIFO contents

Table 5.10. Transmit Start Register 0x11

Field	Name	Access	Description	Reset
[7:1]	Reserved	RO	Reserved.	0x0
[0]	tx_start	RW	When set to high, the IP starts processing the contents of the Tx FIFO. This register auto-clears when a command is done or a NAK is received, unless bypassed by the ignore_cmd_done or ignore_rcvd_nak register settings in the Controller Configuration 0 register. To continue the transfer, set this field to 1 again when tx_start is deasserted due to command completion or NAK reception.	0x0 0x1 – if the optional FIFO interface is enabled.

5.9. Secondary Controller Status Info Register 0x15

Secondary controller status information. This register is applicable only when the *Secondary Controller Capable* attribute is checked in the IP configuration GUI.

Table 5.11. Secondary Controller Status Info Register 0x15

Field	Name	Access	Description	Reset
[7:3]	Reserved	RO	Reserved.	0x0
[2]	get_acccr_result	RO	This status is the result of the GETACCCR command. The information is valid only when the if the get_acccr_done bit in the interrupt status register is asserted. The status is cleared on write of 1 to the get_acccr_done bit in the interrupt status register. Status values: • 0 – Controller handoff is successful • 1 – Controller handoff failed	0x0
[1]	get_acccr_started	RO	Indicates that the device has started or sent the GETACCCR CCC. This status is cleared on write of 1 to the get_acccr_done bit in the interrupt status register.	0x0

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Field	Name	Access	Description	Reset
			Status values: • 1 – Started GETACCCR CCC • 0 – No GETACCCR CCC	
[0]	active_controller	RO	Indicates the current role of this device. The default value of this register reflects the *Device Role* GUI parameter setting. If primary controller is selected, the default value is 1 (active controller); otherwise, the device acts as a 0 (target). The value of this register may change if a controller handoff is successful or if you force it by programming the Set Device Role register. Status values: • 1 – Active Controller • 0 – Target	0x1 – Takes the *Device Role* parameter value.

5.10. Set Device Role Register 0x16

Manually set the device role. This register is applicable only when the *Secondary Controller Capable* attribute is checked in the IP configuration GUI.

Table 5.12. Set Device Role Register 0x16

Field	Name	Access	Description	Reset
[7:1]	Reserved	RO	Reserved.	0x0
[0]	set_device_role	RW	Writing to this register changes the setting of the active_controller register. You can write to this register to reset the device role, which is useful when the system loses information on the device role. Configuration options: • 0 – Sets the active controller register to 0. Enables the i3c_target block, disables the i3c_controller block. • 1 – Sets the active controller register to 1. Enables the i3c_controller block, disables the i3c_target block. When the device is acting as controller and you want to change its role to target, write 0 to the set_device_role register. This action initiates the controller handoff procedure. If the system loses information on the device role, write 1 to this register and write 0 to the device_role register (on the target side) to reset the device role on both the controller and target. For more details, refer to the Set Device Role register (0x48) in the I3C Target IP User Guide (FPGA-IPUG-02227).	0x1 – Takes the *Device Role* parameter value.

5.11. Secondary Controller Timeout Register 0x17–0x19

Controller role handoff timeout value. This register is applicable only when the *Secondary Controller Capable* attribute is checked in the IP configuration GUI.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Table 5.13. Secondary Controller Timeout Register 0x17–0x19

Field	Name	Access	Description	Reset
[7:4]	Reserved	RO	Reserved.	0x0
[3:0]	crh_timeout [19:16]	RW	Controller role handoff timeout value. This field defines the number of system clock (clk_i) cycles. The default value is equivalent to 100 µs (40 ns × 2500). This timeout value is used during the controller handoff procedure, when the former active controller confirms whether the new active controller assert its controller role. Register addresses: • Address 0x17 – crh_timeout[7:0] • Address 0x18 – crh_timeout[15:8] • Address 0x19 – crh_timeout[19:16]	0x009C4 – Calculated based on the *clk_i* period.
[7:0]	crh_timeout [15:8]	RW		
[7:0]	crh_timeout [7:0]	RW		

5.12. Dynamic Address ACKed Counter Register 0x1C

Number of dynamic addresses that acknowledged during the Enter DAA command.

Table 5.14. Dynamic Address ACKed Counter Register 0x1C

Field	Name	Access	Description	Reset
[7:0]	num_da_acked	RW	Running counter that indicates the number of dynamic addresses acknowledged during the Enter DAA command. This counter resets to 0 when all bits are written as 1.	0x0

5.13. IBI Read Count Register 0x1D

Number of bytes to read from the target during IBI. This register is applicable only when the *IBI Capable* attribute is checked in the IP configuration GUI.

Table 5.15. IBI Read Count Register 0x1D

Field	Name	Access	Description	Reset
[7:0]	ibi_rcnt	RW	Indicates the number of bytes to read from the target during IBI. Set this register based on the target's configuration. Set this register to 0 when responding to Hot-Join or Secondary Controller interrupt.	0x0

5.14. IBI Response Register 0x1E

Response to IBI or Hot-Join request. This register is applicable only when the *IBI Capable* attribute is checked in the IP configuration GUI.

Table 5.16. IBI Response Register 0x1E

Field	Name	Access	Description	Reset
[7:1]	Reserved	RO	Reserved.	0x0
[0]	ibi_resp	RW	This field defines the I3C controller's response to an IBI or Hot-Join request.	0x1

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Field	Name	Access	Description	Reset
			Writing to this register triggers the IP to send the indicated response. Configuration options: • 0 – ACK the IBI or Hot-Join • 1 – NAK the IBI or Hot-Join	

5.15. IBI Address Register 0x1F

Address of the I3C target or secondary controller that requested the IBI. This register is applicable only when the *IBI Capable* attribute is checked in the IP configuration GUI.

Table 5.17. IBI Address Register 0x1F

Field	Name	Access	Description	Reset
[7:1]	ibi_addr	RO	Reserved.	0x0
[0]	Reserved	RO	This field displays the address of the I3C target or secondary controller that requested the IBI. If the interrupt is a Hot-Join, this field displays the reserved target address 7'h02.	0x0

5.16. Interrupt Status 0 Register 0x20

Interrupt status. When high, this register indicates an event has occurred. If the corresponding interrupt enable bit is enabled, it causes the assertion of the interrupt port (int_o).

Table 5.18. Interrupt Status 0 Register 0x20

Field	Name	Access	Description	Reset
[7]	rcvd_slv_nak	RW1C	When set to high, this field indicates that a NAK is received.	0x0
[6]	command_done	RW1C	When set to high, this field indicates that a command has been sent (Private Write/Read or CCC).	0x0
[5]	rcvd_sec_ibi	RW1C	When set, this field indicates that a Secondary Controller interrupt is received.	0x0
[4]	rcvd_ibi	RW1C	When set, this field indicates that an In-band interrupt is received.	0x0
[3]	rcvd_hot_join	RW1C	When set to high, this field indicates that a Hot-Join interrupt is received.	0x0
[2]	tx_fifo_full	RW1C	When set to high, this field indicates that the Tx FIFO is full.	0x0
[1]	rx_fifo_not_empty	RW1C	When set to high, this field indicates that the Rx FIFO is not empty.	0x0
[0]	rd_cmd_done	RW1C	When set to high, this field indicates that a private read is done, and all requested data specified by length are available in the Rx FIFO.	0x0

5.17. Interrupt Set 0 Register 0x21

Interrupt set. This is a dummy register used to test the assertion of the interrupt status. When set to high, this register triggers the corresponding interrupt status 0 signal.

Table 5.19. Interrupt Set 0 Register 0x21

Field	Name	Access	Description	Reset
[7]	rcvd_slv_nak_set	WO	Manually sets the corresponding interrupt status to high.	0x0
[6]	command_done_set	WO		0x0
[5]	rcvd_sec_ibi_set	WO		0x0

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Field	Name	Access	Description	Reset
[4]	rcvd_ibi_set	WO		0x0
[3]	rcvd_hot_join_set	WO		0x0
[2]	tx_fifo_full_set	WO		0x0
[1]	rx_fifo_not_empty_set	WO		0x0
[0]	rd_cmd_done_set	WO		0x0

5.18. Interrupt Enable 0 Register 0x22

Interrupt enable.

Table 5.20. Interrupt Enable 0 Register 0x22

Field	Name	Access	Description	Reset
[7]	rcvd_slv_nak_en	RW	When set to high, this field enables the corresponding interrupt status 0 signal to cause the assertion of the interrupt port signal (int_o).	0x0
[6]	command_done_en	RW		0x0
[5]	rcvd_sec_ibi_en	RW		0x0
[4]	rcvd_ibi_en	RW		0x0
[3]	rcvd_hot_join_en	RW		0x0
[2]	tx_fifo_full_en	RW		0x0
[1]	rx_fifo_not_empty_en	RW		0x0
[0]	rd_cmd_done_en	RW		0x0

5.19. Interrupt Status 1 Register 0x24

Interrupt status. When high, this register indicates an event has occurred. If the corresponding interrupt enable bit is enabled, it causes the assertion of the interrupt port (int_o).

Table 5.21. Interrupt Status 1 Register 0x24

Field	Name	Access	Description	Reset
[7]	Reserved	RO	Reserved.	0x0
[6]	waiting_ibi_resp	RW1C	When set to high, this field indicates that there is a pending IBI that the controller must respond to. You must write to the ibi_resp register to specify whether to ACK or NAK the IBI.	0x0
[5]	rx_fifo_full	RW1C	When set to high, this field indicates that the Rx FIFO is full.	0x0
[4]	crh_timeout_expired	RW1C	Applicable only when Secondary Controller Capability is enabled. When set to high, this field indicates that the controller handoff has failed due to the new controller failing to assert its role within the specified timeout value in Secondary Controller Timeout register. This means the device remains as the active controller.	0x0
[3]	get_acccr_done	RW1C	Applicable only when Secondary Controller Capability is enabled. When set to high, this field indicates that the GETACCCR CCC has completed, and the result is available in the Secondary Controller Status Info register.	0x0
[2]	ibi_rd_done	RW1C	When set to high, this field indicates that the IBI payload has been received and the IBI read transaction has been terminated.	0x0

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Field	Name	Access	Description	Reset
[1]	wr_cmd_early_term	RW1C	When set to high, this field indicates that the private write data is not fully transferred due to the target terminating the transaction with NAK.	0x0
[0]	rd_cmd_early_term	RW1C	When set to high, this field indicates that the private read is done due to the target terminating the transaction with T bit==0, and the read data received is less than the requested value.	0x0

5.20. Interrupt Set 1 Register 0x25

Interrupt set.

Table 5.22. Interrupt Set 1 Register 0x25

Field	Name	Access	Description	Reset
[7]	Reserved	RO	Reserved.	0x0
[6]	waiting_ibi_resp_set	WO	This is a dummy register used to test the assertion of the interrupt status. When set to high, this register triggers the corresponding interrupt status 1 signal.	0x0
[5]	rx_fifo_full_set	WO		0x0
[4]	crh_timeout_expired_set	WO		0x0
[3]	get_acccr_done_set	WO		0x0
[2]	ibi_rd_done_set	WO		0x0
[1]	wr_cmd_early_term_set	WO		0x0
[0]	rd_cmd_early_term_set	WO		0x0

5.21. Interrupt Enable 1 Register 0x26

Interrupt enable.

Table 5.23. Interrupt Enable 1 Register 0x26

Field	Name	Access	Description	Reset
[7]	Reserved	RO	When set to high, this field enables the corresponding interrupt status 1 signal to cause the assertion of the interrupt port signal (int_o).	0x0
[6]	waiting_ibi_resp_en	RW		0x0
[5]	rx_fifo_full_en	RW		0x0
[4]	crh_timeout_expired_en	RW		0x0
[3]	get_acccr_done_en	RW		0x0
[2]	ibi_rd_done_en	RW		0x0
[1]	wr_cmd_early_term_en	RW		0x0
[0]	rd_cmd_early_term_en	RW		0x0

5.22. Bus Condition Debug Register 0x28

Bus condition information for debug.

Table 5.24. Bus Condition Debug Register 0x28

Field	Name	Access	Description	Reset
[7]	hdr_ddr_mode	RO	When high, this field indicates that the bus is currently in HDR-DDR mode. Otherwise, the bus operates in SDR mode.	0x0
[6]	i3c_line_idle	RO	When high, this field indicates that the bus state is in idle.	0x1

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Field	Name	Access	Description	Reset
			This also implies that the bus is free (both SCL and SDA are high).	
[5]	scl_oe	RO	Shows the current value of the SCL output enable: • 1 – SCL output is driven • 0 – SCL output is not driven	0x0
[4]	scl_o	RO	Shows the current value of the SCL output generated by the IP: • 1 – SCL output is driven • 0 – SCL output is not driven	0x1 0x0 – if Secondary-Capable but not Active Controller
[3]	sda_wpu	RO	Shows the current value of the SDA weak pull-up control signal: • 1 – weak pull up resistor is disabled • 0 – weak pull up resistor is enabled	0x1
[2]	sda_spu	RO	Shows the current value of the SDA strong pull-up control signal: • 1 – strong pull up resistor is disabled • 0 – strong pull up resistor is enabled	0x0 0x1 – if Secondary-Capable but not Active Controller
[1]	sda_oe	RO	Shows the current value of the SDA output enable: • 1 – SDA output is driven • 0 – SDA output is not driven	0x0
[0]	sda_o	RO	Shows the current value of the SDA output generated by the IP: • 1 – SDA output is driven • 0 – SDA output is not driven	0x1

5.23. Last NAK Address Debug Register 0x29

Last NAK target address.

Table 5.25. Last NAK Address Debug Register 0x29

Field	Name	Access	Description	Reset
[7:1]	tgt_addr	RW	Shows the latest target address driven by the I3C controller that receives a NAK response. Writing to this register resets its value to 0.	0x0
[0]	read1_write0	RW	Indicates the transaction type: read (1) or write (0). Writing to this register resets its value to 0.	0x0

5.24. Last ACK Address Debug Register 0x2A

Last ACK target address.

Table 5.26. Last ACK Address Debug Register 0x2A

Field	Name	Access	Description	Reset
[7:1]	tgt_addr	RW	Shows the latest target address driven by the I3C controller that receives an ACK response. Writing to this register resets its value to 0.	0x0
[0]	read1_write0	RW	Indicates the transaction type: read (1) or write (0). Writing to this register resets its value to 0.	0x0

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

5.25. Interrupt Status 2 Register 0x2C

Interrupt status. When high, this register indicates an event has occurred. If the corresponding interrupt enable bit is enabled, it causes the assertion of the interrupt port (int_o).

Table 5.27. Interrupt Status 2 Register 0x2C

Field	Name	Access	Description	Reset
[7]	hdr_rcvd_nak	RW1C	Applicable only when HDR-DDR capability is enabled. When set to high, this field indicates that a NAK was received.	0x0
[6]	hdr_command_done	RW1C	Applicable only when HDR-DDR capability is enabled. When set to high, this field indicates that a command has been sent (HDR Write/Read or CCC).	0x0
[5]	hdr_frm_error	RW1C	Applicable only when HDR-DDR capability is enabled. When set, this field indicates that a framing error (incorrect preamble bits) is detected.	0x0
[4]	hdr_par_error	RW1C	Applicable only when HDR-DDR capability is enabled. When set, this field indicates that a parity error is detected.	0x0
[3]	hdr_crc_error	RW1C	Applicable only when HDR-DDR capability is enabled. When set, this field indicates that a CRC error is detected.	0x0
[2]	hdr_rd_cmd_early_term	RW1C	Applicable only when HDR-DDR capability is enabled. When set to high, this field indicates that the HDR read is done due to the target terminating the transfer and the read data received is less than the requested value.	0x0
[1]	hdr_wr_cmd_early_term	RW1C	Applicable only when HDR-DDR capability is enabled. When set to high, this field indicates that the HDR write data is not fully transferred due to the target terminating the transfer.	0x0
[0]	hdr_rd_cmd_done	RW1C	Applicable only when HDR-DDR capability is enabled. When set to high, this field indicates that the HDR read command is done and all requested data, as specified by the length, are available in the Rx FIFO.	0x0

5.26. Interrupt Set 2 Register 0x2D

Interrupt Set.

Table 5.28. Interrupt Set 2 Register 0x2D

Field	Name	Access	Description	Reset
[7]	hdr_rcvd_nak_set	WO	This is a dummy register used to test the assertion of the interrupt status. When set to high, this register triggers the corresponding interrupt status 2 signal.	0x0
[6]	hdr_command_done_set	WO		0x0
[5]	hdr_frm_error_set	WO		0x0
[4]	hdr_par_error_set	WO		0x0
[3]	hdr_crc_error_set	WO		0x0
[2]	hdr_rd_cmd_early_term_set	WO		0x0
[1]	hdr_wr_cmd_early_term_set	WO		0x0
[0]	hdr_rd_cmd_done_set	WO		0x0

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

5.27. Interrupt Enable 2 Register 0x2E

Interrupt Enable.

Table 5.29. Interrupt Enable 2 Register 0x2E

Field	Name	Access	Description	Reset
[7]	hdr_rcvd_nak_en	RW	When set to high, this field enables the corresponding interrupt status 0 signal to cause the assertion of the interrupt port signal (int_o).	0x0
[6]	hdr_command_done_en	RW		0x0
[5]	hdr_frm_error_en	RW		0x0
[4]	hdr_par_error_en	RW		0x0
[3]	hdr_crc_error_en	RW		0x0
[2]	hdr_rd_cmd_early_term_en	RW		0x0
[1]	hdr_wr_cmd_early_term_en	RW		0x0
[0]	hdr_rd_cmd_done_en	RW		0x0

5.28. Tx FIFO Register 0x30

Tx FIFO access.

Table 5.30. Tx FIFO Register 0x30

Field	Name	Access	Description	Reset
[7:0]	tx_fifo_data	RW	This is a dummy register used to fill data into the Tx FIFO. Use this register to send packet frames to the Tx FIFO through repeated write accesses. Reading this register returns the status of the Tx FIFO not empty flag.	0x0

5.29. Rx FIFO Register 0x40

Rx FIFO access.

Table 5.31. Rx FIFO Register 0x40

Field	Name	Access	Description	Reset
[7:0]	rx_fifo_data	RO	This is a dummy register used to retrieve read data from the Rx FIFO. The read data from a read request is stored in the Rx FIFO. Use this register to access the received data from the Rx FIFO through repeated read operations. It is recommended to check the interrupt status 0 register (bit[1]) first to determine whether the Rx FIFO is not empty.	0x0

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

6. Example Design

The I3C Controller example design allows you to compile, simulate, and test the I3C Controller IP on the following Lattice evaluation boards:

- CrossLink-NX PCIe Bridge Board
- CertusPro-NX Evaluation Board
- Avant-E Evaluation Board
- CertusPro-NX Sensor to Ethernet Bridge Board

6.1. Example Design Supported Configuration

Table 6.1. I3C Controller IP Configuration Supported by the Example Design

Attribute	Primary Controller	Secondary Controller (Target)
Bus Characteristics		
Device Role	Primary Controller	Secondary Controller
Secondary Controller Capable	Checked	Checked
IBI Capable	Checked	Checked
Hot-Join Capable	Checked	Checked
Data Rate Mode	HDR-DDR-capable	HDR-DDR-capable
Register Interface		
Interface	APB	APB
Register Offset	Address offset in DWORD	Address offset in DWORD
Optional Interface		
Enable Direct FIFO Interface	—	—
SCL Clocking		
Bus Clock Domain	ASYNC *(Display only)*	ASYNC *(Display only)*
Use Internal Clock Divider	Checked	Checked
System Clock Frequency (MHz)	25	25
Internal Clock Frequency (MHz)	25 *(Display only)*	25 *(Display only)*
SCL Pulse width	1	1
SCL Clock Frequency (MHz)	12.5 *(Display only)*	12.5 *(Display only)*
SCL Clock Period (ns)	80 *(Display only)*	80 *(Display only)*
Open Drain Pulse Width (Number of SCL cycles)	3	3
Open Drain Clock Frequency (MHz)	2.083 *(Display only)*	2.083 *(Display only)*
Open Drain Pulse width (ns)	239.99 *(Display only)*	239.99 *(Display only)*
Enable Dynamic I3C – I2C clock switching	Checked	Checked
I2C SCL Pulse width	13	13
I2C SCL Clock Frequency (MHz)	0.96 *(Display only)*	0.96 *(Display only)*
I2C SCL Clock Period (ns)	520 *(Display only)*	520 *(Display only)*
Include IO Primitive	Checked	Checked
Secondary Controller Tab		
Bus Characteristics		
IBI Payload Size (including MDB)	1	1
Maximum Data Speed Limitation	Checked	Checked
Device Characteristics		
DCR (HEX)	00	00
Manufacturer ID	414	414
Part ID	1	1
Instance ID	1	2

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Attribute	Primary Controller	Secondary Controller (Target)
Additional ID	0	0
Static Address Enable	Checked	Checked
Static Address (HEX) 1	08	09
Dynamic Address (HEX) 1	08	—
Timing Characteristics		
Write Maximum Data Rate (MHz)	12.5 *(Display only)*	12.5 *(Display only)*
Clock-to-data Turnaround Delay (ns) (t_{SCO})	≤ 8 *(Display only)*	≤ 8 *(Display only)*
Read Maximum Data Rate (MHz)	12.5 *(Display only)*	12.5 *(Display only)*
Maximum Read Turnaround Time (µs)	0 *(Display only)*	0 *(Display only)*

6.2. Overview of the Example Design and Features

The example design discussed in this section is created using the *RISC-V MC SoC Project* template in the Lattice Propel Design Environment. The generated project includes the following components:

- Processor – RISC-V (MC w/ PIC/T)
- GPIO
- Asynchronous SRAM
- UART – Serial port
- PLL
- Glue Logic

I3C Controller and I3C Target (Secondary Controller) are instantiated and connected in the project as shown in Figure 6.1. In this example, I3C Controller and I3C Target (Secondary Controller) are instantiated in the same system.

To establish communication between the I3C Controller and external I3C/I2C Target or Secondary Controller devices, use flywire connections. This involves manually wiring signal lines such as SDA, SCL, and any required control signals between the controller and the external device.

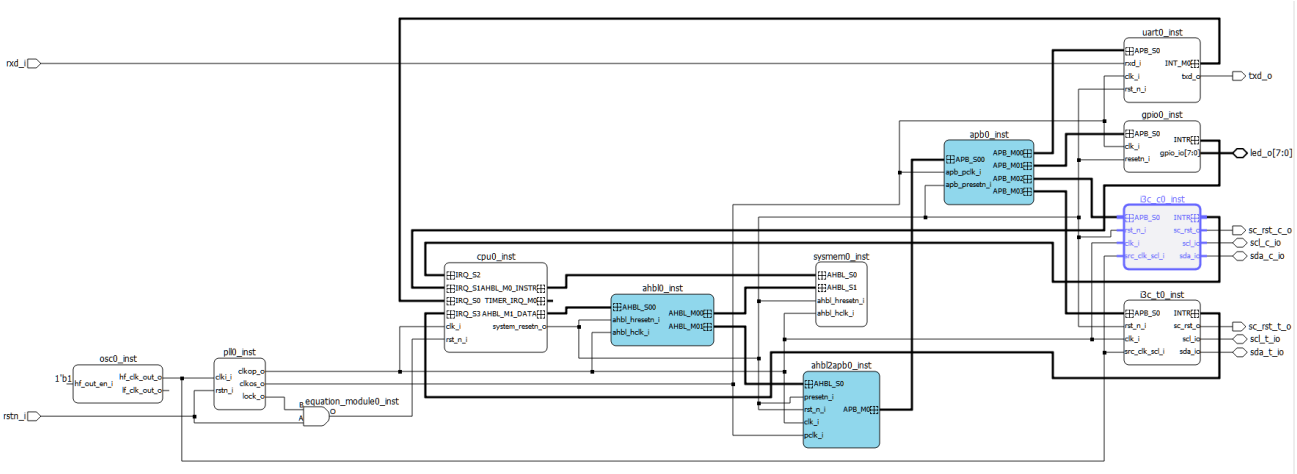


Figure 6.1. I3C Controller IP in Propel SoC Project

An Embedded C/C++ Project is also created in the Propel software to enable developing and debugging application code for different IP features. I3C Controller features can be tested by sending I3C Commands from the I3C Controller to the I3C Target. Runtime configuration of IP and feature testing can be done through C-Code Test Routine. Figure 6.2 shows an example routine to send ENTDAA CCC to multiple I3C Targets.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

```c
/*
 * @brief i3c_controller_entdaa_ntargets function for i3c controller.
 * @param : handle : handle for the controller
 * @return : status for entdaa_config
 */
unsigned int i3c_controller_entdaa_multitargets(i3c_controller_handle_t *handle){
    unsigned int wr_length;
    unsigned int control;
    unsigned int status;
    unsigned int index;
    if((handle->base_addr != ZERO)&& (handle->target_dyn_addr != ZERO) && (handle->target_nums != ZERO))
    {
        i3c_cntler_reg_t *i3c_contrlr=(i3c_cntler_reg_t *)(handle->base_addr);
        wr_length=SET+handle->target_nums;
        control = CCC_CMD1_WR_RD0|STOP_IN_FRAME|CCC_COMMAND_WITH_START;
        i3c_contrlr->i3c_controller_tx_fifo= control;              //control
        i3c_contrlr->i3c_controller_tx_fifo= RESERVED_ADDRESS<<SET;      //broadcast
        i3c_contrlr->i3c_controller_tx_fifo= wr_length;           //write length
        i3c_contrlr->i3c_controller_tx_fifo=  ENTDAA;             //ENTDAA CCC
        for(index=ZERO;index<wr_length-SET;index++)
        {
            i3c_contrlr->i3c_controller_tx_fifo=handle->target_dyn_addr_multi[index] <<SET;
        }
        i3c_contrlr->i3c_controller_start_reg = START;
        status=SUCCESS;
    }
    else
    {
        status=FAILURE;
    }
    return status;
}
```

Figure 6.2. Sample C-Code Test Routine

6.3. Design Components Example

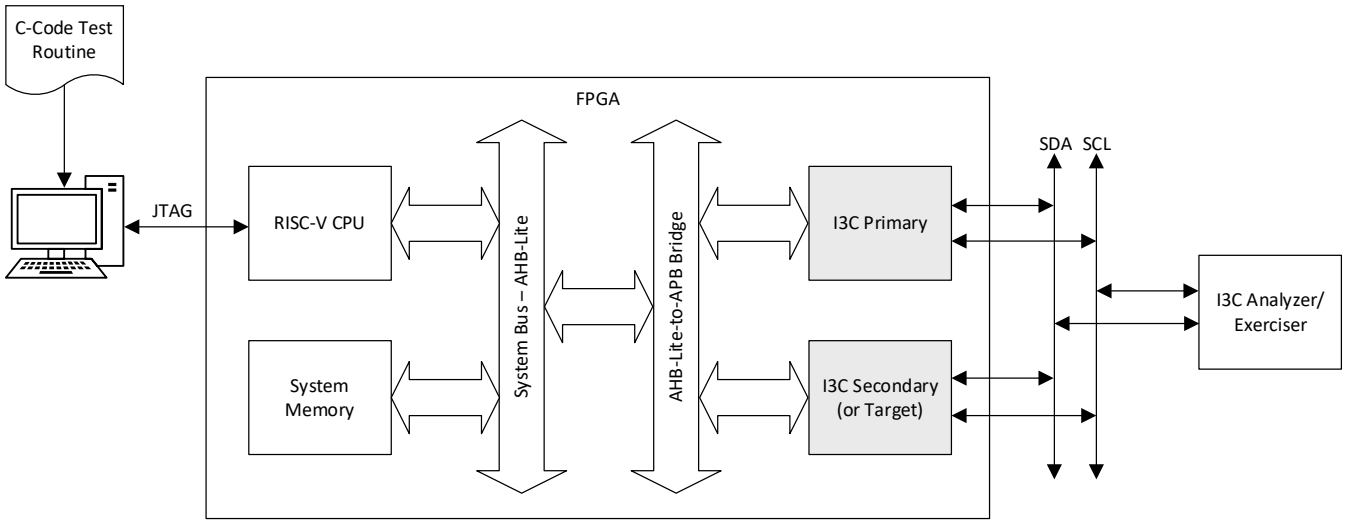


Figure 6.3. I3C Controller Example Design Block Diagram

The I3C Controller example design includes the following blocks:

- RISC-V CPU – Passes the C-Code Test Routine from system memory to system bus. Handles interrupts.
- Memory – Contains commands for testing.
- System Bus – AHB-Lite systems bus for transfers between memory and IP
- I3C Controller IP – IP instance connected to I3C bus (SCL and SDA)
- I3C Target Device(s)

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

6.4. Generating the Example Design

Use the Lattice Propel software to generate an I3C Controller SoC project. The steps below describe how to generate the I3C Controller SoC project. For more details, refer to the Lattice Propel SDK User Guide.

1. Launch Lattice Propel software and set your workspace directory.
2. In the Propel software, create a new Lattice SoC Design Project by navigating to **File** > **New** > **Lattice SoC Design Project.**
3. The **Create SoC Project** window will open.
   - In **Device Select** section, indicate the correct details of the device or board that you are using. In Figure 6.4, device is set to LFCPNX-100-8LFG672C since CertusPro-NX Evaluation Board is used in the hardware testing.
   - In **Template Design** section, choose **RISC-V MC SoC Project**. Click **Finish**.

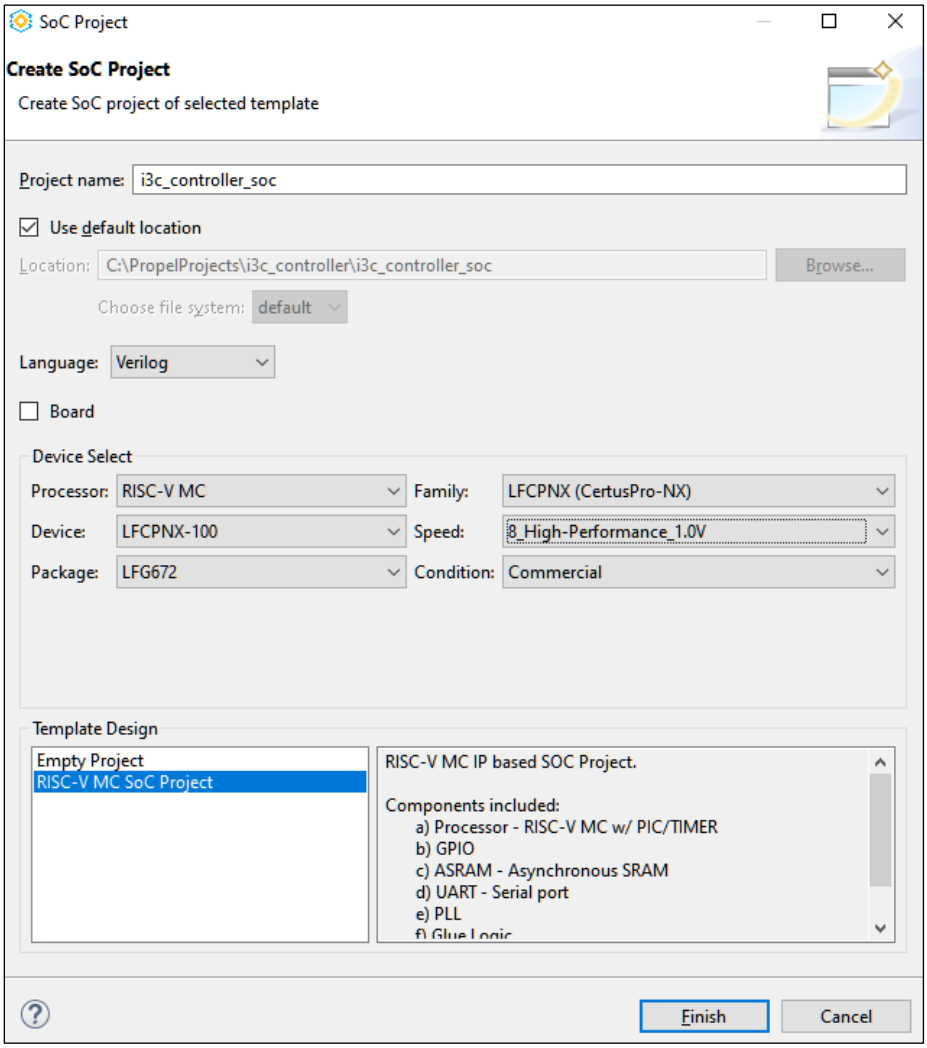


Figure 6.4. Create SoC Project

4. Run Propel Builder by clicking the icon or navigate to **LatticeTools** > **Open Design** in Propel Builder**.** The Propel Builder will open and load the design template.
5. In the **IP Catalog** tab, instantiate the I3C Controller IP. Refer to the Generating and Instantiating the IP section for more details. In this example, there are two instances of the I3C Controller IP:
   - Primary Controller
   - Secondary Controller

   See the Example Design Supported Configuration section for the corresponding parameter settings.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

6. After generating the IP, the **Define Instance** window will open. Modify the instance name if needed, then click **OK.**

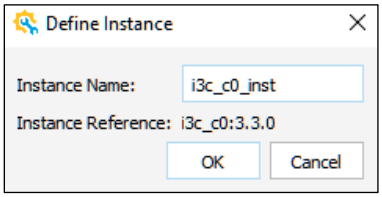


Figure 6.5. Define Instance

7. Connect the instantiated IPs to the system. Refer to Figure 6.1 for the connections used in this IP. You will need to update other components of the system for clock and reset sources, interrupt, and bus interface.
8. Click the icon or navigate to **Design** > **Run Radiant** to launch the Lattice Radiant Software.
9. Update your constraints file accordingly and generate the programming file.
10. In the Lattice Radiant software, set the PULLMODE of SDA and SCL I/O to I3C in the **Device Constraint Editor**, or copy the following code to the .pdc file.

```
ldc_set_port -iobuf {SLEWRATE=FAST PULLMODE=I3C} [get_ports scl_c_io]
ldc_set_port -iobuf {SLEWRATE=FAST PULLMODE=I3C} [get_ports sda_c_io]
ldc_set_port -iobuf {SLEWRATE=FAST PULLMODE=I3C} [get_ports scl_t_io]
ldc_set_port -iobuf {SLEWRATE=FAST PULLMODE=I3C} [get_ports sda_t_io]
```

11. Click the icon (Run All) located on the toolbar to perform the Lattice Radiant software full design compilation, which generates the example design bitstream file for the hardware test.
12. Download the generated bitstream to the evaluation board through the Lattice Radiant Programmer.
13. In the Lattice Propel software, build your SoC project to generate the system environment needed for the embedded C/C++ project. Select your SoC project then navigate to **Project** > **Build Project**.
14. Check the build result from the **Console** view.

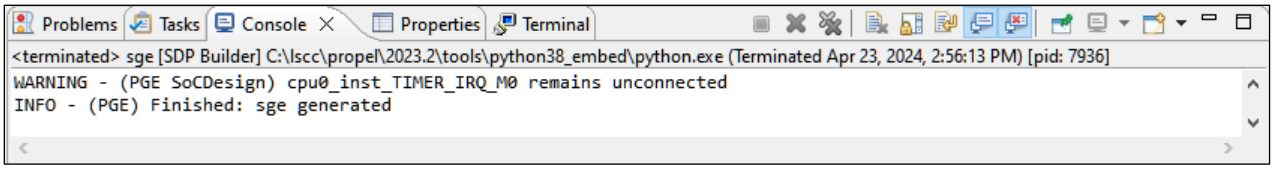


Figure 6.6. Build SoC Project Result

15. Generate a new Lattice C/C++ project by navigating to **File** > **New** > **Lattice C/C++ Project**. Update your **Project name**, click **Next**, and then click **Finish**.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

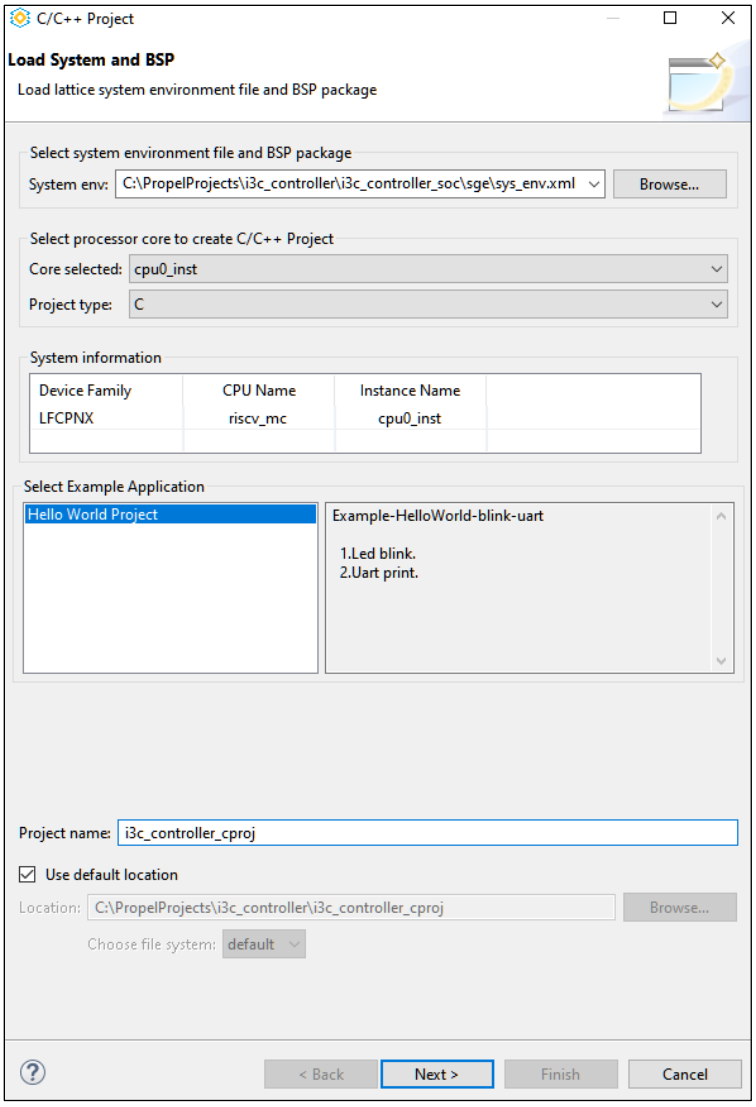


Figure 6.7. Lattice C/C++ Design Project

16. In **Project Explorer**, locate your Lattice C/C++ Project and find *main.c* within the *src* directory.
    You may copy the sample code generated by the IP, located at *eval/sw/main.c*, to perform basic I3C transactions. Refer to the Software Driver API section for more details.
17. Select your C/C++ project then select **Project** > **Build**.
18. Check the build result from the **Console** view.

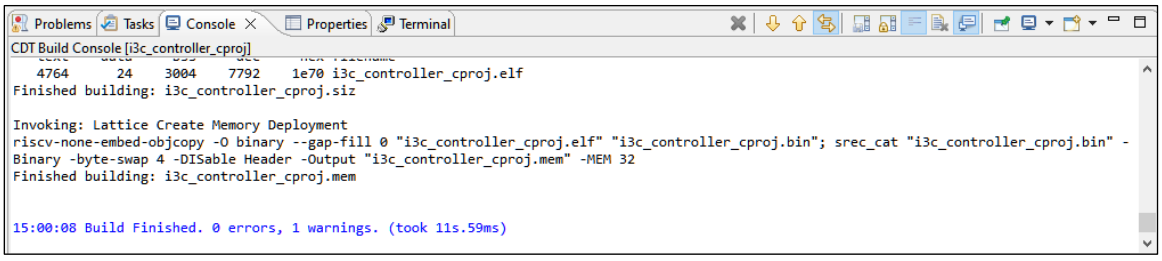


Figure 6.8. Build C/C++ Project Result

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

19. This environment is now ready for running your tests on the device. Refer to the *Running Demo on MachXO3D Breakout Board – Hello World* section of the Lattice Propel SDK User Guide for step-by-step guide.

6.5. Using the Prebuilt Example Design for CertusPro-NX Device

You can download the prebuilt example design from this Lattice Knowledge Base article.

6.6. Hardware Testing

6.6.1. Hardware Testing Setup

Download the generated bitstream file from the Generating the Example Design section to the CertusPro-NX Evaluation Board through the Lattice Radiant Programmer.

Before running tests, ensure the following connections are properly configured:

- Connect the I3C Controller to external I3C/I2C Target or Secondary Controller devices using flywire.
- Verify correct pin mapping between the controller and the external device to prevent communication errors or hardware damage.
- Keep wire lengths short to maintain signal integrity and minimize noise or crosstalk.
- Add pull-up resistors to the SDA and SCL lines if required by your I3C configuration.

6.6.2. Expected Output

Below is a sample waveform captured with Reveal Inserter and Reveal Analyzer tools. Refer to the relevant sections in the Lattice Radiant Software User Guide for more information on how to use the Reveal Inserter and Reveal Analyzer tools.



Figure 6.9. Sample ENTDAA sequence sent by I3C Controller

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

7. Designing with the IP

This section provides information on how to generate the IP Core using the Lattice Radiant software and how to run simulation and synthesis. For more details on the Lattice Radiant software, refer to the Lattice Radiant Software User Guide.

Note: The screenshots provided are for reference only. Details may vary depending on the version of the IP or software being used. If there have been no significant changes to the GUI, a screenshot may reflect an earlier version of the IP.

7.1. Generating and Instantiating the IP

You can use the Lattice Radiant software to generate IP modules and integrate them into the device architecture. The steps below describe how to generate the I3C Controller IP in the Lattice Radiant software.

To generate the I3C Controller IP:

1. Create a new Lattice Radiant software project or open an existing project.
2. In the **IP Catalog** tab, double-click **I3C Controller** under **IP** > **Processors_Controllers_and_Peripherals** category. The **Module/IP Block Wizard** opens as shown in Figure 7.1. Enter values in the **Component name** and the **Create in** fields and click **Next**.

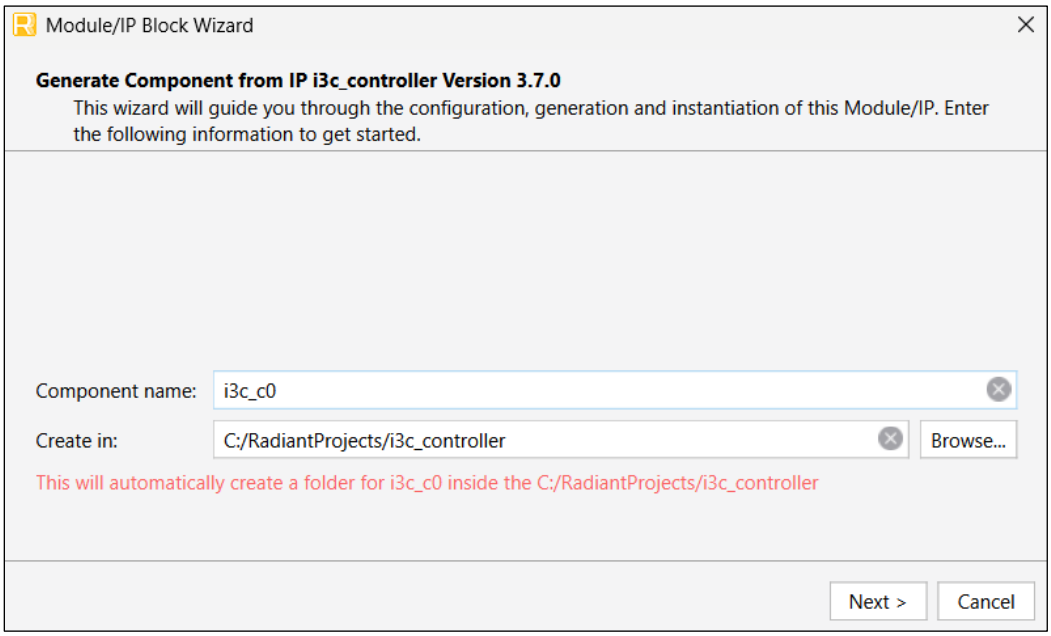


Figure 7.1. Module/IP Block Wizard

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

3. In the next **Module/IP Block Wizard** window, customize the selected I3C Controller IP using drop-down lists and check boxes. Figure 7.2 shows an example configuration of the I3C Controller IP. For details on the configuration options, refer to the IP Parameter Description section.

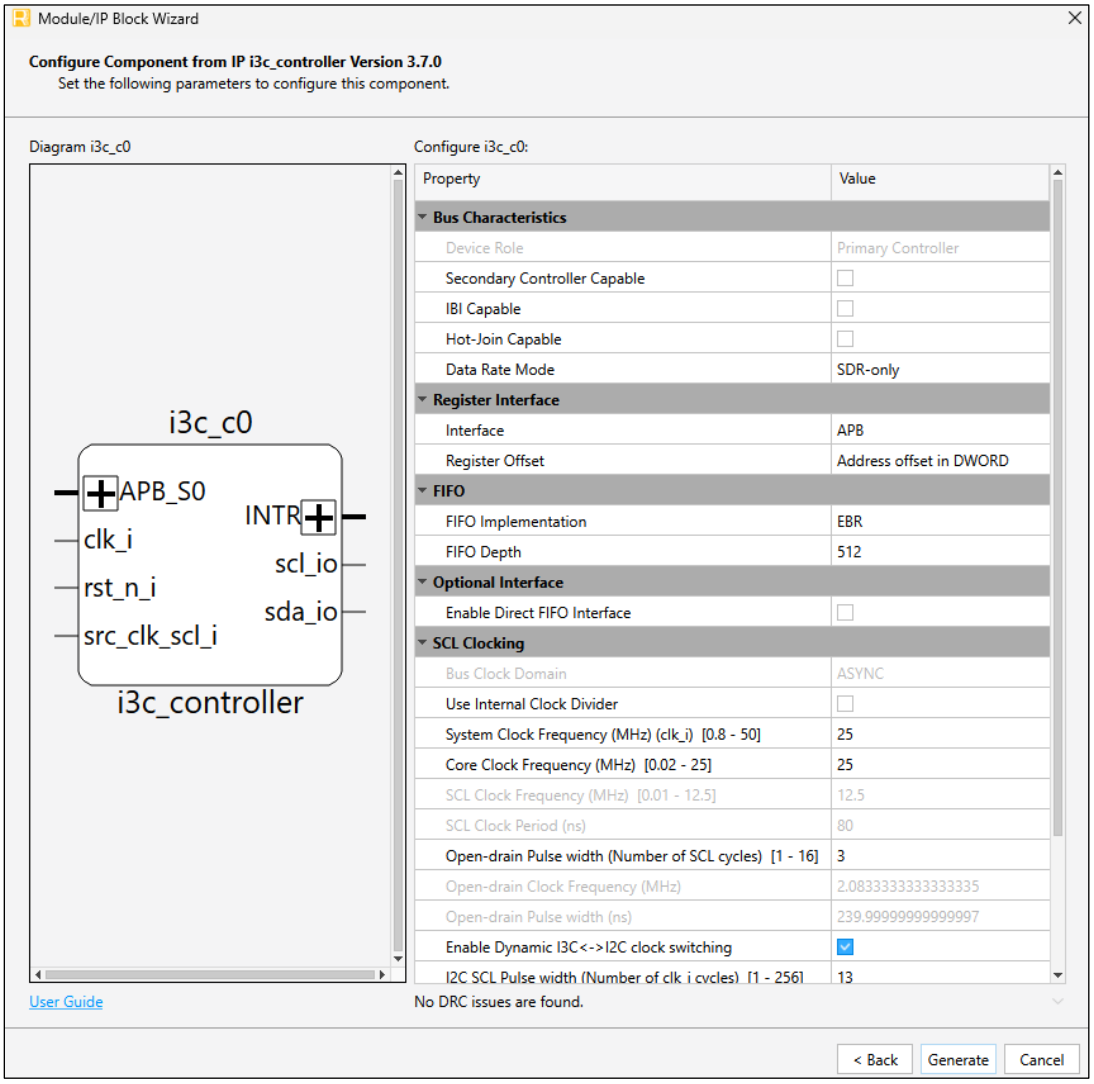


Figure 7.2. IP Configuration

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

4. Click **Generate**. The **Check Generated Result** dialog box opens, showing design block messages and results as shown in Figure 7.3.

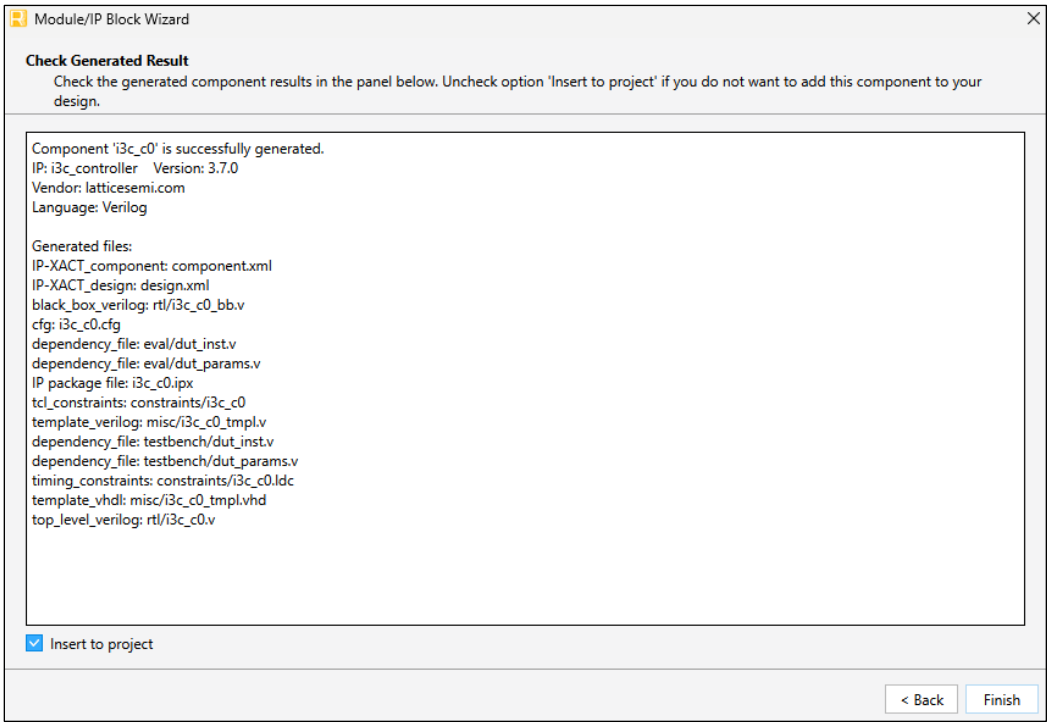


Figure 7.3. Check Generated Result

5. Click **Finish**. All the generated files are placed under the directory paths in the **Create in** and the **Component name** fields shown in Figure 7.1.

7.1.1. Generated Files and File Structure

The generated I3C Controller module package includes the closed-box (<Component name>_bb.v) and instance templates (<Component name>_tmpl.v/vhd) that can be used to instantiate the core in a top-level design. An example RTL top-level reference source file (<Component name>.v) that can be used as an instantiation template for the module is also provided. You may also use this top-level reference as the starting template for the top-level for their complete design. The generated files are listed in Table 7.1.

Table 7.1. Generated File List

Attribute	Description
<Component name>.ipx	This file contains the information on the files associated to the generated IP.
<Component name>.cfg	This file contains the parameter values used in IP configuration.
component.xml	Contains the ipxact: component information of the IP.
design.xml	Documents the configuration parameters of the IP in IP-XACT 2014 format.
rtl/<Component name>.v	This file provides an example RTL top file that instantiates the module.
rtl/<Component name>_bb.v	This file provides the synthesis closed-box.
misc/<Component name>_tmpl.v misc /<Component name>_tmpl.vhd	These files provide instance templates for the module.
eval/constraint.pdc	This file provides information on how to constrain this IP in your design. Refer to Timing Constraints section on how to use this file.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

7.2. Design Implementation

Completing your design includes additional steps to specify analog properties, pin assignments, and timing and physical constraints. You can add and edit the constraints using the Device Constraint Editor or by manually creating a PDC File.

Post-Synthesis constraint files (.pdc) contain both timing and non-timing constraint.pdc source files for storing logical timing/physical constraints. Constraints that are added using the Device Constraint Editor are saved to the active .pdc file. The active post-synthesis design constraint file is then used as input for post-synthesis processes.

Refer to the relevant sections in the Lattice Radiant Software User Guide for more information on how to create or edit constraints and how to use the Device Constraint Editor.

7.3. Timing Constraints

You need to provide proper timing and physical design constraints to ensure that your design meets the desired performance goals on the FPGA. Add the content of the following IP constraint file to your design constraints: *<IP_Instance_Path>/<IP_Instance_Name>/eval/constraint.pdc.*

The constraint file has been verified during IP evaluation with the IP instantiated directly in the top-level module. You can modify the constraints in this file with thorough understanding of the effect of each constraint.

To use this constraint file, copy the content of *constraint.pdc* to the top-level design constraint for post-synthesis.

Refer to Lattice Radiant Timing Constraints Methodology (FPGA-AN-02059) for details on how to constraint your design.

7.4. Physical Constraints

When Enable I/O primitive option is checked, ensure that PULLMODE of scl_io and sda_io are set to I3C. You can check this in **Tools** > **Device Constraint Editor**. If not yet set, you can change the PULLMODE to I3C in the *Device Constraint Editor* or you can add the following constraints to your constraint file:

```
ldc_set_port -iobuf {PULLMODE=I3C} [get_ports scl_io]
ldc_set_port -iobuf {PULLMODE=I3C} [get_ports sda_io]
```

7.5. Running Functional Simulation

You can run functional simulation after the IP is generated.

To run functional simulation:

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

1. Click the  button located on the **Toolbar** to initiate the **Simulation Wizard** shown in Figure 7.4.

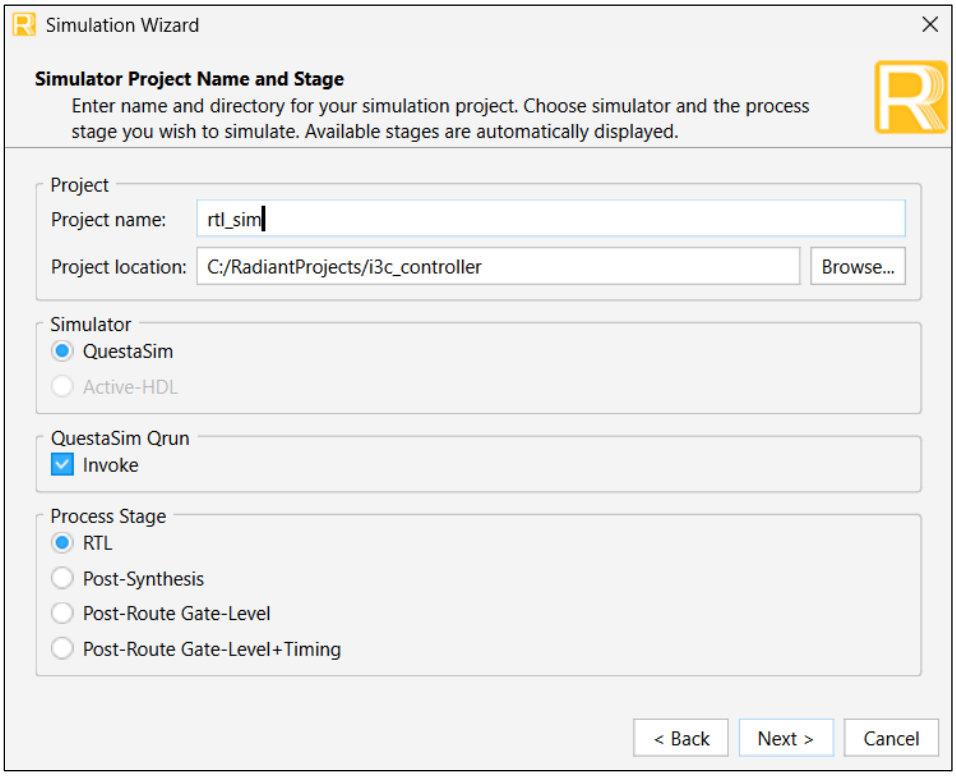


Figure 7.4. Simulation Wizard

2. Click **Next** to open the **Add and Reorder Source** window as shown in Figure 7.5. Ensure that the generated *rtl/<component name>.v* and *testbench/tb_top.v* are the only **Source Files**. Remove other files if there are any.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

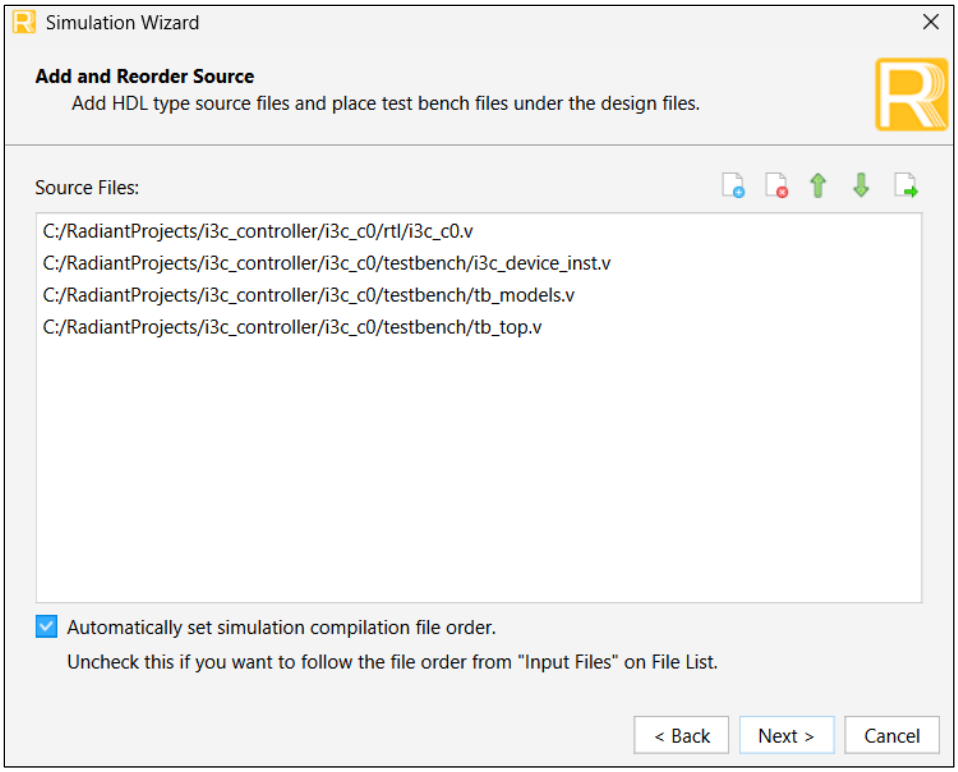


Figure 7.5. Add and Reorder Source

3. Click **Next**. The **Parse HDL files for simulation** window is shown. Confirm that Simulation Top Module is *tb_top.*

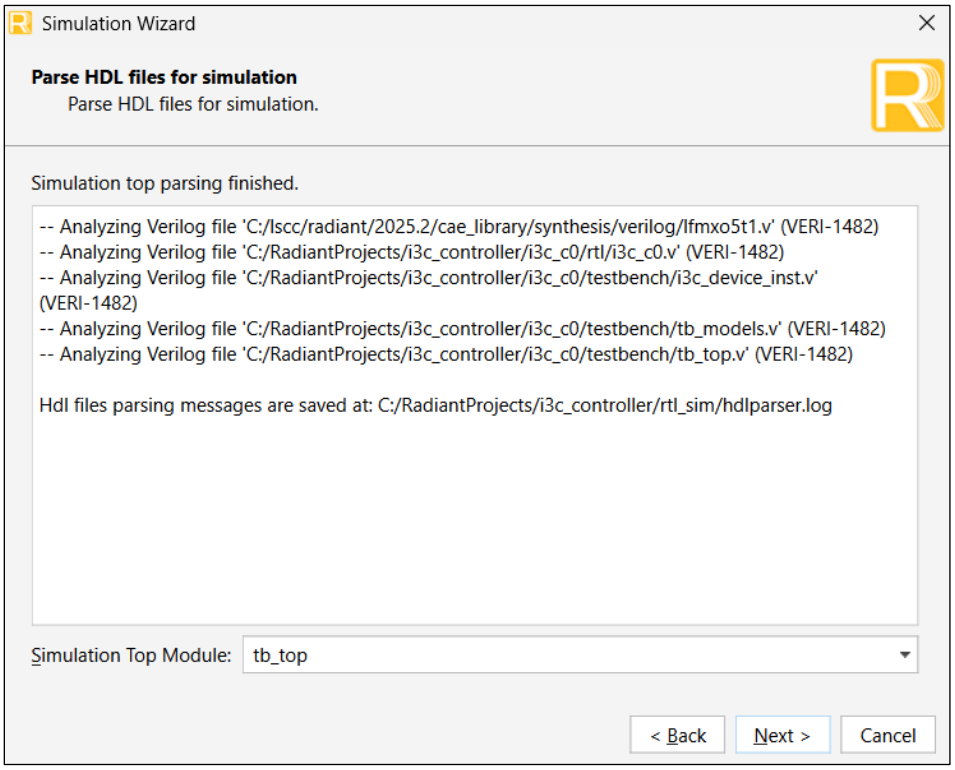


Figure 7.6. Parse HDL Files for Simulation

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

4. Click **Next**. The **Summary** window is shown.

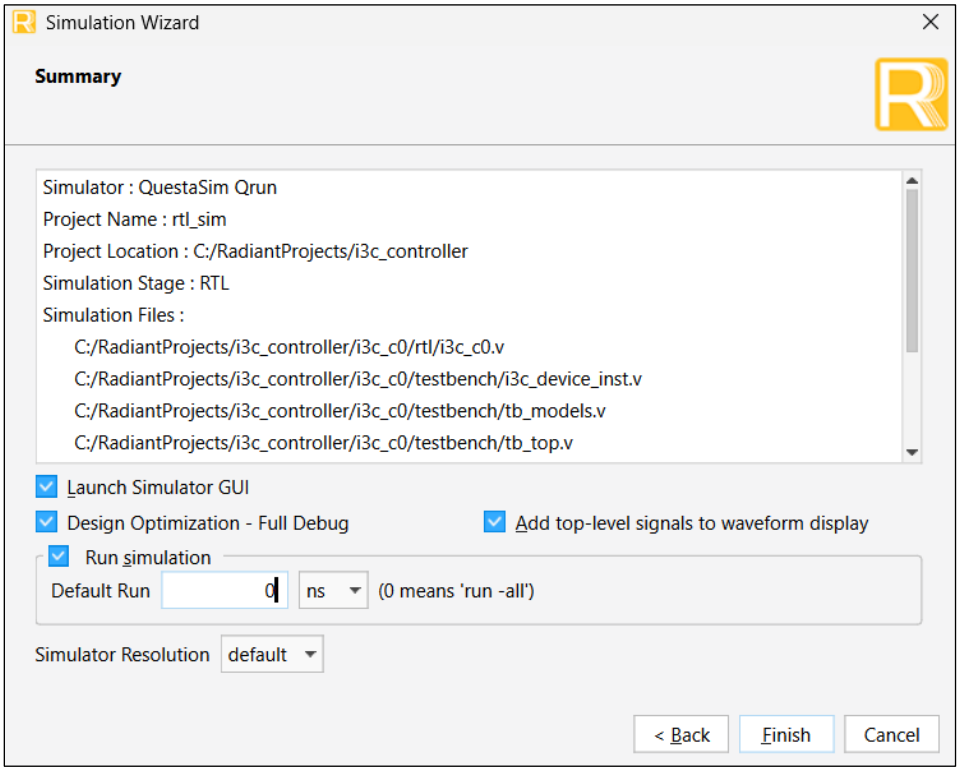


Figure 7.7. Summary

5. Click **Finish** to run the simulation. The waveform in Figure 7.8 shows an example simulation result.

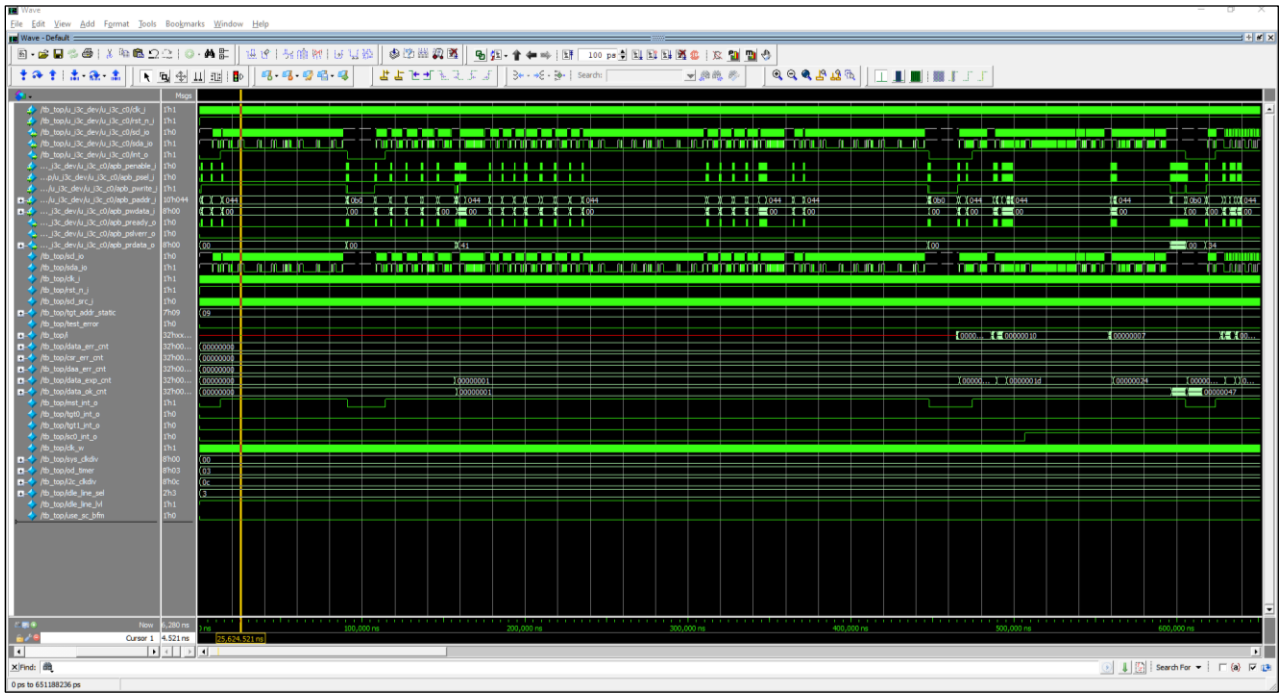


Figure 7.8. Simulation Waveform

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

8. Software Driver API

The following is an example of software application code located in eval/main.c:

```
#include "i3c_controller.h"
#include "uart.h"
#include "utils.h"
#include <stdio.h>

#define I3C_SLAVE_NUMS 1
#define DATA_LENGTH 32
#define SLAVE_ADDR 0x20
#define STATIC_ADDR 0x0B
#define STATIC_ADDR_WR 0x16
#define STATIC_ADDR_RD 0x17

struct uart_instance uart_core_uart;
struct i3c_master_instance i3cm;
struct i3c_dev_info i3c_devinfo;

const uint8_t wr_buf[DATA_LENGTH] = {0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A,
0x0B, 0x0C, 0x0D, 0x0E, 0x0F, 0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16, 0x17, 0x18, 0x19, 0x1A, 0x1B,
0x1C, 0x1D, 0x1E, 0x1F};
uint8_t rd_buf[DATA_LENGTH];

int main(void) {
  uint8_t ret = 0;
  int i;
  uint8_t dyn_address;
  unsigned char *reg_value;

  i3c_devinfo.pid = 0;

  uart_init(&uart_core_uart, UART0_INST_BASE_ADDR, UART0_INST_SYS_CLK, UART0_INST_BAUD_RATE, 1, 8);

  #ifdef LSCC_STDIO_UART_APB
  extern struct uart_instance *g_stdio_uart;
  g_stdio_uart = &uart_core_uart;
  #endif

  if (i3c_master_init(&i3cm, U_I3C_M0_BASE_ADDR, 0, U_I3C_M0_GUI_I2C_SCL_PULSE_WIDTH-1, I3C_SLAVE_NUMS) ==
0)
  {
    printf("I3C master initialization is success\n");
  } else {
    printf("Fail to initialize I3C master\n");
    return -1;
  }

  printf("STARTING DAA TEST...\n");

  ret = i3c_master_daa(&i3cm, &i3c_devinfo, &dyn_address, I3C_SLAVE_NUMS, SLAVE_ADDR);

  if (ret) {
    printf("Fail I3C master DAA: error = %d\n", ret);
  } else {
    printf("I3C master DAA is success\n");
#ifdef ENABLE_DAA_UID
```

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

```
    printf("bcr = %x\n", i3c_devinfo.bcr);
    printf("dcr = %x\n", i3c_devinfo.dcr);
    printf("pid = %016llx \n", i3c_devinfo.pid);
    printf("dyn addr = %x\n", i3c_devinfo.init_dyn_addr);
#endif
  }

  printf("STARTING I3C TESTS: Private Write ...\n");
  printf("Writing data from I3C controller to the target\n");
  i3c_master_private_i3c_write(&i3cm, (uint8_t *) wr_buf, DATA_LENGTH, SLAVE_ADDR);

  printf("STARTING I3C TESTS: Private Read ...\n");
  i3c_master_private_i3c_read(&i3cm, (uint8_t *) rd_buf, DATA_LENGTH-2, SLAVE_ADDR);
  for (i = 0; i < DATA_LENGTH-2; i++)
  {
    printf("Read back data from I3C slave at count %d: %x\n", i, rd_buf[i]);

    if (rd_buf[i] == wr_buf[i]) {
      printf("Read data matches write\n");
    } else {
      printf("Data mismatch %x != %x\n", rd_buf[i], wr_buf[i]);
    }
  }

  i3c_master_private_i3c_read(&i3cm, (uint8_t *) rd_buf, 2, SLAVE_ADDR);
  printf("Read back data from I3C target: %x\n", rd_buf[0]);
  printf("Read back data from I3C target: %x\n", rd_buf[1]);

  printf("TEST FINISHED\n");

  return 0;
}
```

The application in main.c performs the following operations:

- I3C Initialization – Sets up register settings such as clock period and other relevant parameters.

```
if (i3c_master_init(&i3cm, U_I3C_M0_BASE_ADDR, 0, U_I3C_M0_GUI_I2C_SCL_PULSE_WIDTH-1, I3C_SLAVE_NUMS)
== 0)
Unknown macro: { printf("I3C master initialization is successn"); }
else
Unknown macro: { printf("Fail to initialize I3C mastern"); return -1; }
```

- Dynamic Address Assignment (DAA) – Assigns a dynamic address to the target device.

```
printf("STARTING DAA TEST...\n");
ret = i3c_master_daa(&i3cm, &i3c_devinfo, &dyn_address, I3C_SLAVE_NUMS, SLAVE_ADDR);
if (ret)
Unknown macro: { printf("Fail I3C master DAA}
else
Unknown macro: { printf("I3C master DAA is successn"); #ifdef ENABLE_DAA_UID printf("bcr = %xn",
i3c_devinfo.bcr); printf("dcr = %xn", i3c_devinfo.dcr); printf("pid = %016llx n", i3c_devinfo.pid);
printf("dyn addr = %xn", i3c_devinfo.init_dyn_addr); #endif }
```

- Transaction Verification – Compares write and read transactions between the I3C Controller and the Target.

```
printf("STARTING I3C TESTS: Private Write ...\n");
printf("Writing data from I3C controller to the target\n");
i3c_master_private_i3c_write(&i3cm, (uint8_t *) wr_buf, DATA_LENGTH, SLAVE_ADDR);
printf("STARTING I3C TESTS: Private Read ...\n");
i3c_master_private_i3c_read(&i3cm, (uint8_t *) rd_buf, DATA_LENGTH-2, SLAVE_ADDR);
```

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

```
for (i = 0; i < DATA_LENGTH-2; i++)
{
printf("Read back data from I3C slave at count %d: %x\n", i, rd_buf[i]);
if (rd_buf[i] == wr_buf[i])
Unknown macro: { printf("Read data matches writen"); }
else
Unknown macro: { printf("Data mismatch %x != %xn", rd_buf[i], wr_buf[i]); }
}
i3c_master_private_i3c_read(&i3cm, (uint8_t *) rd_buf, 2, SLAVE_ADDR);
printf("Read back data from I3C target: %x\n", rd_buf[0]);
printf("Read back data from I3C target: %x\n", rd_buf[1]);
```

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Appendix A. Resource Utilization

Table A.1 shows the resource utilization of the I3C Controller IP using the LFCPNX-100-7ASG256C device using Synplify Pro of Lattice Radiant Software 2023.2. Default configuration is used, and some attributes are changed from default value to show the effect on the resource utilization.

Table A.1. LFCPNX-100-7ASG256C Device Resource Utilization

Configuration	clk Fmax (MHz)	Registers	LUTs	EBRs	DSPs
Default	75.2	468	1035	2	0
IBI Capable = True Hot-Join Capable = True	73.0	529	1220	2	0
Secondary Capable = True IBI Capable = True Hot-Join Capable = True	71.2	1360	3312	2	0
Secondary Capable = True IBI Capable = True Hot-Join Capable = True Data Rate = HDR-DDR capable	73.6	1657	4430	2	0

Notes:

1. Fmax is generated when the FPGA design contains only the I3C Controller module, and the target frequency is 25 MHz. These values may be reduced when user logic is added to the FPGA design.
2. The distributed RAM utilization is accounted for in the total LUT4s utilization. The actual LUT4 utilization is distributed among logic, distributed RAM, and ripple logic.

Table A.2 shows the resource utilization of the I3C controller IP using the LIFCL-40-7BG256I device using Synplify Pro of Lattice Radiant Software 2023.2. Default configuration is used, and some attributes are changed from default value to show the effect on the resource utilization.

Table A.2. LIFCL-40-7BG256I Device Resource Utilization

Configuration	clk Fmax (MHz)	Registers	LUTs	EBRs	DSPs
Default	69.6	468	1032	2	0
IBI Capable = True Hot-Join Capable = True	73.2	529	1220	2	0
Secondary Capable = True IBI Capable = True Hot-Join Capable = True	67.4	1360	3312	2	0
Secondary Capable = True IBI Capable = True Hot-Join Capable = True Data Rate = HDR-DDR capable	70.1	1657	4430	2	0

Notes:

1. Fmax is generated when the FPGA design contains only the I3C Controller module, and the target frequency is 25 MHz. These values may be reduced when user logic is added to the FPGA design.
2. The distributed RAM utilization is accounted for in the total LUT4s utilization. The actual LUT4 utilization is distributed among logic, distributed RAM, and ripple logic.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Table A.3 shows the resource utilization of the I3C controller IP using the LAV-AT-E70-1LFG676I device using Synplify Pro of Lattice Radiant Software 2023.2. Default configuration is used, and some attributes are changed from default value to show the effect on the resource utilization.

Table A.3. LAV-AT-E70-1LFG676I Device Resource Utilization

Configuration	clk Fmax (MHz)	Registers	LUTs	EBRs	DSPs
Default	133.00	462	1017	2	0
IBI Capable = True Hot-Join Capable = True	138.70	523	1205	2	0
Secondary Capable = True IBI Capable = True Hot-Join Capable = True	60.05	1382	3256	2	0
Secondary Capable = True IBI Capable = True Hot-Join Capable = True Data Rate = HDR-DDR capable	48.00	1656	4565	2	0

Notes:

1. Fmax is generated when the FPGA design contains only the I3C Controller module, and the target frequency is 25 MHz. These values may be reduced when user logic is added to the FPGA design.
2. The distributed RAM utilization is accounted for in the total LUT4s utilization. The actual LUT4 utilization is distributed among logic, distributed RAM, and ripple logic.

Table A.4 shows the resource utilization of the I3C controller IP using the LN2-CT-20-1CBG484I device using Synplify Pro of Lattice Radiant Software 2024.2. Default configuration is used, and some attributes are changed from default value to show the effect on the resource utilization.

Table A.4. LN2-CT-20-1CBG484I Device Resource Utilization

Configuration	clk Fmax (MHz)	Registers	LUTs	EBRs	DSPs
Default	201.207	478	1154	2	0
IBI Capable = True Hot-Join Capable = True	211.327	531	1213	2	0
Secondary Capable = True IBI Capable = True Hot-Join Capable = True	181.851	1397	3365	2	0
Secondary Capable = True IBI Capable = True Hot-Join Capable = True Data Rate = HDR-DDR capable	160.411	1673	4371	2	0

Notes:

1. Fmax is generated when the FPGA design contains only the I3C Controller module, and the target frequency is 25 MHz. These values may be reduced when user logic is added to the FPGA design.
2. The distributed RAM utilization is accounted for in the total LUT4s utilization. The actual LUT4 utilization is distributed among logic, distributed RAM, and ripple logic.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

References

- Avant-E web page
- Avant-G web page
- Avant-X web page
- Certus-N2 web page
- Certus-NX web page
- CertusPro-NX web page
- CrossLink-NX web page
- iCE40 UltraPlus web page
- Mach-N2 web page
- Mach-NX web page
- MachXO3D web page
- MachXO5-NX web page
- I3C Controller IP Core web page
- Lattice Radiant Software web page
- Lattice Solutions IP Cores web page
- Lattice Solutions Reference Designs web page
- Lattice Propel Design Environment web page
- MIPI I3C Specification web page
- I3C Device Characteristics Register web page
- CrossLink-NX PCIe Bridge Board web page
- Avant-E Evaluation Board web page
- CertusPro-NX Evaluation Board web page
- CertusPro-NX Sensor to Ethernet Bridge Board web page
- AMBA 3 AHB-Lite Protocol v1.0 Specification
- AMBA 3 APB Protocol v1.0 Specification
- Lattice Memory Mapped Interface and Lattice Interrupt Interface (FPGA-UG-02039)
- Lattice Radiant Timing Constraints Methodology (FPGA-AN-02059)
- I3C Target IP User Guide (FPGA-IPUG-02227)
- I3C Controller IP Release Notes (FPGA-RN-02017)
- I3C Controller IP Prebuilt Example Design for the CertusPro-NX Device
- Lattice Insights web page for Lattice Semiconductor training courses and learning plans

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

For frequently asked questions, refer to the Lattice Answer Database at www.latticesemi.com/Support/AnswerDatabase.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Revision History

Note: In some instances, the IP may be updated without changes to the user guide. The user guide may reflect an earlier IP version but remains fully compatible with the later IP version. Refer to the IP Release Notes for the latest updates.

Revision 1.7, IP v3.7.2, October 2026

Section	Change Summary
All	• Updated the IP version on the cover page. • Made editorial fixes.
Abbreviations in This Document	• Added *DSP*, *LUT*, and *N/A*. • Updated the *MDB* definition from *Maximum Data Speed* to *Mandatory Data Byte*. • Removed *HJ*.

Introduction	• In Table 1.1. Summary of the I3C Controller IP:
	  • Added Mach-N2 to *Supported Devices*.
	  • Updated *Lattice Implementation*.
	• Updated the IP Support Summary section.
	• Remove the *Hardware Support* section.

Functional Description	• Updated the bit [2] description for LMMI Write Data 0x09 in Table 2.11. Generic Direct CCC Format. • Updated the bit [2] description for LMMI Write Data 0xAB in Table 2.23. Generic Direct Set CCC Format and Table 2.26. Generic Direct Get CCC Format.
Register Description	Updated the descriptions for *rcvd_sec_ibi* and *rcvd_ibi* in Table 5.18. Interrupt Status 0 Register 0x20.
References	Added the Mach-N2 and I3C Controller IP Core web pages.

Revision 1.6, IP v3.7.0, December 2025

Section	Change Summary
All	• Updated the IP version on the cover page. • Added a note on the IP version in the *Quick Facts* and *Revision History* sections. • Made editorial fixes.
Abbreviations in This Document	Added *AMBA*, *API*, *CPU*, *DWORD*, *GPIO*, *RO*, *RTL*, *RW*, *RW1C*, *Rx*, *SRAM*, *Tx*, *UART*, and *WO*.

Introduction	• Updated *Lattice Implementation* in Table 1.1. Summary of the I3C Controller IP.
	• In Table 1.2. I3C Controller IP Support Readiness:
	  • Added a table note.
	  • Updated hardware validation information for the MachXO5-NX device family.
	• Updated the receive and transmit FIFO feature in the Features section.
	• Updated the Licensing and Ordering Information section and removed the *Ordering Part Number* section.
	• Added *_oe* to the Signal Names section.
	• Added the Attribute Names section.

Functional Description	• Updated Table 2.1. User Interfaces and Supported Protocols. • Added figure captions to Figure 2.6. I3C Private Write – Figure 2.9. I3C Consecutive Private Write and Private Read and Figure 2.11. Reset Dynamic Address – Figure 2.16. I2C Consecutive Private Write and Private Read. • Updated Figure 2.10. Enter Dynamic Address Assignment and Figure 2.17. HDR-DDR Mode Frame. • Added references to the MIPI I3C Specification in the Enter Dynamic Address Assignment (DAA) CCC and Performing HDR-DDR Mode Transfers sections.

IP Parameter Description	• In Table 3.1. General Attributes:
	  • Updated the descriptions for *Device Role*, *Register Offset*, and *Open Drain Pulse Width (Number of SCL cycles)*.
	  • Added attributes for *FIFO*.
	  • Updated the *System Clock Frequency (MHz) (clk_i)* attribute name.
	• Updated the table note for Table 3.2. Secondary Controller (Available only if Secondary Controller

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Section	Change Summary
	Capable is Enabled).
Signal Description	In Table 4.1. Ports Description: • Updated the description for *scl_io*. • Updated the *rx_data_o[7:0]* port name.
Register Description	Updated this section.

Example Design	• Added the *CertusPro-NX Sensor to Ethernet Bridge Board*.
	• Replaced ✓ with *Checked* in Table 1.1. Summary of the I3C Controller IP.
	• Updated the descriptions in the following sections:
	  • Overview of the Example Design and Features
	  • Generating the Example Design
	  • Hardware Testing Setup
	• Added the Using the Prebuilt Example Design for CertusPro-NX Device section.

Designing with the IP	• Added a note on screenshots in this section. • Updated all figures in this section.
Software Driver API	Added this section.
References	Added the *Mach-NX*, *MachXO3D*, *I3C Controller IP Prebuilt Example Design for the CertusPro-NX Device*, *CertusPro-NX Sensor to Ethernet Bridge Board*, and *Lattice Solutions Reference Designs* web pages.

Revision 1.5, IP v3.6.0, July 2025

Section	Change Summary
All	Updated the IP version information on the cover page.

Introduction	• In Table 1.1. Summary of the I3C Controller IP:
	  • Updated *Supported FPGA Family* to *Supported Devices* and rearranged devices order.
	  • Added *Mach-NX* and *MachXO3D* device families to *Supported Devices*.
	  • Removed *Targeted Devices*.
	  • Updated *Lattice Implementation*.
	• In Table 1.3. Ordering Part Number:
	  • Updated *Single Machine Annual* to *Single Seat Annual*.
	  • Updated *Multi-Site Perpetual* to *Single Seat Perpetual*.

Revision 1.4, IP v3.5.0, December 2024

Section	Change Summary
All	Added the IP version information on the cover page.

Introduction	• Updated Table 1.1. Summary of the I3C Controller IP:
	  • Added the *Certus-NX-RT*, *CertusPro-NX-RT*, and *Certus-N2* device families to *Supported FPGA Family*.
	  • Removed *IP Version*.
	  • Added *IP Changes* and *Resources*.
	  • Updated *Targeted Devices* and *Lattice Implementation*.
	• Replaced the *IP Validation Summary* section with the IP Support Summary section.
	• Added the Certus-N2 OPNs to Table 1.3. Ordering Part Number.
	• Added the Hardware Support section.
	• Made editorial fixes.

Example Design	Added an introductory paragraph to list the evaluation boards used for the example design.
Designing with the IP	Updated Figure 7.1. Module/IP Block Wizard, Figure 7.2. IP Configuration, and Figure 7.3. Check Generated Result.
Resource Utilization	• Added the *clk Fmax* unit in this section. • Added resource utilizations for the Lattice Radiant software version 2024.2. • Made editorial fixes.
References	• Added the *Certus-N2* web page, *CrossLink-NX PCIe Bridge Board* web page, *Avant-E Evaluation Board*

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Section	Change Summary
	web page, *CertusPro-NX Evaluation Board* web page, and *I3C Controller IP Release Notes (FPGA-RN-02017)*. • Removed *Lattice Radiant Software 2023.2 User Guide*.

Revision 1.3, June 2024

Section	Change Summary
All	• Removed *Core* from the document title. • Made editorial fixes. • Replaced *HDR mode* with *HDR-DDR* mode. • Updated the term *CCC command* or *CCC code* to *CCC*.

Abbreviations in This Document	• Replaced *acronyms* with *abbreviations* in this section.
	• Added the following abbreviations:
	  • *Acknowledgement (ACK)*
	  • *Bus Characteristics Register (BCR)*
	  • *Cyclic Redundancy Check (CRC)*
	  • *Control and Status Registers (CSR)*
	  • *Device-to-Device (D2D)*
	  • *Dynamic Address (DA)*
	  • *Device Characteristics Register (DCR)*
	  • *Hot-Join (HJ)*
	  • *Input/Output (I/O)*
	  • *Microcontroller (MC)*
	  • *Negative Acknowledgement (NAK)*
	  • *Programmable Interrupt Controller (PIC)*
	  • *Phase-Locked Loop (PLL)*
	  • *Reduced Instruction Set Computer Five (RISC-V)*
	  • *Single Data Rate (SDR)*
	  • *Software Development Kit (SDK)*
	  • *System on Chip (SoC)*

Introduction	• Moved introductory paragraph in the 1. Introduction section to the newly added 1.1. Overview of the IP section and updated the heading numbers of remaining sections accordingly. • Updated Table 1.1. Summary of the I3C Controller IP. • Updated the information on features not supported by the I3C Controller in the 1.3. Features. • Moved the content from previous *3.1. Licensing the IP* and *3.5. IP Evaluation* sections to the newly added 1.4. Licensing and Ordering Information section and updated its content. • Moved the content from previous *4. Ordering Part Number* section to the 1.4.1. Ordering Part Number section and updated its content. • Moved the content from previous *3.6. Hardware Validation* section to the newly added 1.5. IP Validation Summary section and updated its content. • Added the 1.6. Minimum Device Requirements section and updated the heading numbers of remaining sections accordingly. • Renamed the previous *1.3. Conventions* section to *1.7. Naming Conventions* and removed the *Attribute* information.
Functional Description	• Renamed the previous *2.1 Overview* section to *2.1. IP Architecture Overview* and updated its content. • Updated Figure 2.1. I3C Controller IP Core Functional Diagram (Controller Only) and Figure 2.2. I3C Controller IP Core Functional Diagram (with Secondary Controller Capability). • Added the 2.2. Clocking section and updated the heading numbers of remaining sections accordingly. • Renamed the previous *2.2 Reset Propagation* section to *2.3. Reset* and updated its content. • Added the 2.4. User Interfaces section and updated the heading numbers of remaining sections accordingly.
IP Parameter	• Moved the content from previous *2.11. Attributes Summary* section to this newly added section.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Section	Change Summary

Description	• Updated the following tables:
	  • Table 3.1. General Attributes
	  • Table 3.2. Secondary Controller (Available only if Secondary Controller Capable is Enabled
	  • Table 3.3. IP Parameter Settings for Example Test Cases

Signal Description	• Moved the content from previous *2.10. Signal Description* section to this section.
	• Updated the names or/and descriptions of the following signals in Table 4.1. Ports Description:
	  • *src_clk_scl_i*
	  • *sc_rst_o*
	  • *ahbl_hsize_i[2:0]*
	  • *ahbl_hprot_i[3:0]*
	  • *tx_ready_o*
	  • *tx_data_i [7:0]*
	  • *rx_valid_o*
	  • *rx_ready_i*
	  • *rx_data_o*
	  • *lmmi_ready_o*
	  • *ahbl_haddr_i[31:0]*
	  • *ahbl_hwdata_i[31:0]*
	  • *ahbl_hrdata_o[31:0]*
	  • *apb_paddr_i[31:0]*
	  • *apb_pwdata_i[31:0]*
	  • *apb_prdata_o[31:0]*
	• Updated the Notes for Table 4.1. Ports Description.

Register Description	• Moved the content from previous *2.12. Register Description* section to this section. • Added a paragraph about APB or AHB-Lite interface. • Updated the *Note* section.
Example Design	• Added this section and updated the heading numbers of remaining sections accordingly.
Designing with the IP	• Moved the content from previous *3.2. Generation and Synthesis* section to the 7.1. Generating and Instantiating the IP section and updated its content. • Added the 7.2. Design Implementation section and updated the heading numbers of remaining sections accordingly. • Moved the content from previous *3.4. Constraining the IP* section to the 7.3. Timing Constraints section and updated its content. • Added the 7.4. Physical Constraints section and updated the heading numbers of remaining sections accordingly. • Updated the contents including all figures in the 7.5. Running Functional Simulation section.
References	Added the following references: • *Lattice Solutions for IP Cores web page* • *Lattice Propel Design Environment web page* • *MIPI I3C Specification web page* • *I3C Device Characteristics Register web page* • *Lattice Radiant Software 2023.2 User Guide* • *AMBA 3 AHB-Lite Protocol v1.0 Specification* • *AMBA 3 APB Protocol v1.0 Specification* • *Lattice Memory Mapped Interface and Lattice Interrupt Interface (FPGA-UG-02039)* • *Lattice Radiant Timing Constraints Methodology (FPGA-AN-02059)* • *I3C Target IP User Guide (FPGA-IPUG-02227)*

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Revision 1.2, December 2023

Section	Change Summary
All	• Changed the document name from *I3C Controller IP Core - Lattice Radiant Software* to *I3C Controller IP Core*. • Formatted fonts for input signals, output signals, configuration settings, and name of registers to italic. • Updated the content to better suit second person point of view. • Minor adjustments to ensure the document is consistent with Lattice Semiconductor's inclusive language policy.
Disclaimers	Updated boilerplate.
Functional Description	• Changed the heading name from *Permitted Cs* in HDR Mode to Permitted CCCs in HDR Mode. • Updated paragraph of Hot Join and In-band Interrupt (IBI) Handling section by adding the sentence, *When the Tx FIFO is empty, waiting_ibi_resp interrupt will be asserted. You should program or set the IBI read count register (0x1D), and the IBI response register (0x1E).* • Added register *(0x24)* to the paragraph in the Controller to Controller Handoff section. • Updated *auto_assert_role* paragraph in the Controller Configuration 0 Register 0x02 section. • Added new paragraphs on how to initiate the controllership handoff procedure, and how to reset the device role on both the Controller and Target sides to the Set Device Role Register 0x16 section. • Updated the field and width of Interrupt Status *reserved*, and added new Interrupt Status *waiting_ibi_resp* in the Interrupt Status 1 Register 0x24 section. • Updated the field and width of Interrupt Set *reserved* and added new Interrupt Set *waiting_ibi_resp_set* in the Interrupt Set 1 Register 0x25 section. • Updated the field and width of Interrupt Enable *reserved* and added new Interrupt Enable *waiting_ibi_resp_en* in the Interrupt Enable 1 Register 0x26 section. • Changed from *not_empty flag* to *not empty flag* in the Tx FIFO Register 0x30 section.
IP Core Generation, Simulation, and Validation	• Added the sentence You can evaluate the IP core through functional simulation and implementation (synthesis, map, place, and route) without an IP license string to the Licensing the IP section. • Updated Figure 3.1 and Figure 3.3 in the Generation and Synthesis section. • Added Figure 3.2 in the Generation and Synthesis section. • Added new attribute *eval/constraint.pdc* to Table 3.1 in the Generation and Synthesis section. • Added Constraining the IP section. • Updated the header numbers for IP Evaluation and Hardware Validation sections.
Ordering Part Number	Updated Table 4.1 to add new part numbers, removed obsolete part numbers, and updated the license types.
Resource Utilization	• Added the sentence *Table A.1 shows the resource utilization of I3C Controller IP using LFCPNX-100-7ASG256C device using Synplify Pro of Lattice Radiant Software 2023.2*, and *Table A.2 shows the resource utilization of I3C controller IP using LIFCL-40-7BG256I device using Synplify Pro of Lattice Radiant Software 2023.2* to the Appendix A. Resource Utilization section. • Renamed Table A.1 to *Resource Utilization for LFCPNX-100-7ASG256C*, and Table A.2 to *Resource Utilization for LIFCL-40-7BG256I*. • Added new configurations *Secondary Capable = True*, and *Data Rate = HDR-DDR capable* to Table A.1, and Table A.2. • Removed configurations *SCL Pulse Width*, and *Open-Drain Pulse Width* from Table A.1, and Table A.2. • Updated the *clk Fmax*, *Registers*, and *LUTs* values for all configurations in Table A.1, and Table A.2. • Updated the table notes to change from *SDR module* to *I3C Controller module*, and the target frequency from *200 Mhz* to *25 Mhz* for Table A.1, and Table A.2. • Added Table A.3 to the Appendix A. Resource Utilization section.
References	Added links to Lattice Avant-G, Lattice Avant-X, and Lattice Insights web pages.

Revision 1.1, July 2023

Section	Change Summary
Introduction	• Deleted paragraph *I3C is a two-wire bi-directional serial bus, optimized for multiple sensor Secondary devices and controlled by only one I3C Controller device at a time. I3C is backward compatible with many legacy I2C devices, but I3C devices also support significantly higher speeds, new*

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Section	Change Summary
	communication modes, and new device roles, including an ability to change device roles over time. For example, the initial Controller can cooperatively pass the Controllership to another I3C device on the bus, if the requesting I3C device supports a Secondary Controller feature in Introduction section. • Updated Table 1.1. Quick Facts for below: • Added Resources row. • Added *Lattice Avant*™ device to Supported FPGA Families. • Replaced Radiant Software version from *2022* to *2022.1*.
Functional Description	• Added Reset Propagation section. • Added description of Control 5 Bit *In HDR mode, it is recommended to set this to 1 when there are multiple I3C frames in one packet to ensure that HDR mode will not be exited in case that (1) the preceding frame is NAK-ed, (2) User read/write command is terminated early, or (3) HDR-DDR framing error is detected. If this is set to 0, User shall reset the IP core and cleanup the contents of FIFO when the mentioned conditions occur to avoid incorrect packet framing and unknown internal state of IP core* in Table 2.1. User Packet Frame Structure Definition. • Added comment *// [4] = 0 -> I3C // [5] = 1 -> Wait next packet // [6] = 0 -> unused* in below figures: • Table 2.16. Example: Read 4 Bytes from I3C Target with Address 7'h10 using HDR-DDR Mode • Table 2.17. HDR-DDR Consecutive Write and Read • Table 2.19. Generic Broadcast CCC Format • Table 2.22. Generic Direct Set CCC Format • Table 2.25. Generic Direct Get CCC Format. • Updated Table 2.27. Ports Description for below: • Added port names *ext_io_scl_i, ext_io_scl_o, ext_io_scl_oe, ext_io_scl_spu_n, ext_io_scl_wpu_n, ext_io_sda_i, ext_io_sda_o, ext_io_sda_oe, ext_io_sda_spu_n, ext_io_sda_wpu_n*. • Added footnotes *Bidirectional I3C interface is only available when internal IO primitives are enabled in IP generation GUI* and *External IO I3C interface is only available when internal IO primitives are disabled in IP generation GUI*. • Replaced Clear *when interrupt status get get_acccr_done is cleared* with *Cleared on write of 1 to interrupt status get_acccr_done* in Secondary Controller Status Info Register 0x15 section. • Updated Table 2.28. Attributes Table for below changes: • Added attributes *Internal Clock Frequency (MHz)* and *Dynamic Address (HEX)*[1] • Replaced Default values from *LMMI* and *Address Offset in Byte* with *APB* and *Address Offset in DWORD.* • Replaced register name from *reserved* to *en_ack_handoff* in Table 2.29. Controller Configuration 0 Register 0x02. • Added *en_ack_handoff* information in Controller Configuration 0 Register 0x02 section. • Added *ddr_ignore_rcvd_nak* and *ddr_ignore_cmd_done* information and [4] and [5] fields in HDR-DDR Configuration 0 Register 0x06 section. • Added *Address 0x17 – crh_timeout[7:0], Address 0x18 – crh_timeout[15:8], Address 0x19 – crh_timeout[19:16]* in Secondary Controller Timeout Register 0x19 – 0x17 section. • Added *hdr_frm_error* information in Interrupt Status 2 Register 0x2C section.
IP Core Generation, Simulation, and Validation	• Added below steps in Running Functional Simulation section: • *In the File List view, right click on Input Files > Add > Existing File.* • *Select testbench/tb_top.v* • Added Figure 3.5. Adding Existing Input File and Figure 3.6. Adding tb_top.v and updated Figure 3.7. Simulation Wizard and Figure 3.8. Adding and Reordering Source. • Added Constraining the IP and Hardware Validation sections.
Ordering Part Numbers	Added CrossLink-NX, Certus-NX, CertusPro-NX, MachXO5-NX, iCE-40 UltraPlus, Avant-E part numbers in Table 4.1. Ordering Part Numbers.
References	Added links for CrossLink-NX, Certus-NX, CertusPro-NX, MachXO5-NX, iCE-40 UltraPlus, Avant-E, and Lattice Radiant web pages.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Revision 1.0, April 2023

Section	Change Summary
All	Initial release.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.



www.latticesemi.com