I3C Target IP

IP Version: v3.7.0

User Guide

FPGA-IPUG-02227-1.6

December 2025

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.

Contents

Figures

Tables

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
BCR	Bus Characteristics Register
CCC	Common Command Code
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
D2D	Device to Device
DA	Dynamic Address
DAA	Dynamic Address Assignment
DCR	Device Characteristics Register
DDR	Double Data Rate
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
HJ	Hot-Join
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
MC	Microcontroller
MIPI	Mobile Industry Processor Interface
MDB	Mandatory Data Byte
MWL	Maximum Write Length
NACK/NAK	Negative Acknowledgement
PIC	Programmable Interrupt Controller
PID	Provisioned Identification
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
RW1C	Read and Write 1 to Clear
Rx	Receiver

Abbreviation	Definition
SA	Static Address
SCL	Serial Clock
SDA	Serial Data
SDR	Single Data Rate
SoC	System on Chip
SRAM	Static Random Access Memory
Tx	Transmitter
UART	Universal Asynchronous Receiver/Transmitter
WO	Write-Only access

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, two-wire digital interface for sensors. I3C protocol is a single scalable, cost effective, and a power efficient protocol. Implementing the I3C specification greatly increases the implementation flexibility for an ever-expanding sensor subsystem as efficiently and at as low cost as 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 Target 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™, and Certus-N2.
	IP Changes[1]	For a list of changes to the IP, refer to the I3C Target IP Release Notes (FPGA-RN-02018).
Resource Utilization	Supported User Interface	Lattice Memory Mapped Interface (LMMI), Advanced Peripheral Bus (APB), Advanced High-Performance Bus-Lite (AHB-Lite)
	Resources	Refer to Appendix A. Resource Utilization
Design Tool Support	Lattice Implementation	IP Core v3.7.0 – Lattice Radiant™ Software 2025.2 and Lattice Propel™ Builder Software 2025.2
	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 Target IP Support Readiness

Device Family	System Clock Frequency (MHz)[1]	Supported Feature	Radiant Timing Model	Hardware Validated
Crosslink	25	Dynamic Address Assignment	Final	Yes
		IBI	Final	Yes
		Hot-Join	Final	Yes
		HDR-DDR	Final	No
		I3C Write/Read	Final	Yes
		I2C Write/Read	Final	Yes
CertusPro-NX	25	Dynamic Address Assignment	Final	Yes
		IBI	Final	Yes
		Hot-Join	Final	Yes
		HDR-DDR	Final	Yes

Device Family	System Clock Frequency (MHz)[1]	Supported Feature	Radiant Timing Model	Hardware Validated
		I3C Write/Read	Final	Yes
		I2C Write/Read	Final	Yes
MachXO5-NX	25	Dynamic Address Assignment	Final	Yes
		IBI	Final	Yes
		Hot-Join	Final	Yes
		HDR-DDR	Final	Yes
		I3C Write/Read	Final	Yes
		I2C Write/Read	Final	Yes
Avant	25	Dynamic Address Assignment	Preliminary	Yes
		IBI	Preliminary	Yes
		Hot-Join	Preliminary	Yes
		HDR-DDR	Preliminary	Yes
		I3C Write/Read	Preliminary	Yes
		I2C Write/Read	Preliminary	Yes

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.

1.4. Features

The maximum number of devices that an I3C bus supports depends on trace length, capacitive load per device, and the types of devices (I2C versus I3C) present on the bus, because these factors affect clock frequency requirements. The Lattice I3C Target IP 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 co-existence on the same bus (with some limitations)
- Dynamic Addressing with optional Static Addressing for I3C Target acting as I2C Target
- I2C -like SDR messaging
- HDR-DDR mode support
- In-Band Interrupt support
- Hot-Join support
- Asynchronous Time Stamping (Mode 0)
- Target Reset without additional wires
- Configurable receive and transmit FIFO depth and implementation

The Lattice I3C Target 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
- Virtual target
- Synchronous Timing and Asynchronous Time Stamping (Except Mode 0)
- Group Addressing
- Monitoring Device Early Termination
- D2D Tunneling
- Multi-Lane Data Transfer

1.5. Licensing and Ordering Information

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

1.6. Hardware Support

Refer to the Example Design section for more information on the boards used.

1.7. Minimum Device Requirements

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

1.8. Naming Conventions

1.8.1. Nomenclature

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

1.8.2. 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

1.8.3. Attribute Names

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

2. Functional Description

2.1. IP Architecture Overview

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

The Lattice I3C Target IP supports the following modes:

- SDR mode
- HDR-DDR mode

The I3C Target IP monitors the I3C bus for relevant I3C commands sent by the I3C Controller and responds accordingly. This includes commands that address all targets devices (Broadcast CCCs) and commands addressed specifically to that I3C Target device (Directed CCCs), provided these commands are supported by the I3C Target device.

Optionally, the I3C Target can perform the following operations:

- Request In-Band Interrupts
- Generate Hot-Join events

The I3C Target IP accepts commands from LMMI or from the optional APB/AHB-Lite interface. These commands are decoded into the following:

- Configurations for the I3C Target that may be requested by the Controller through CCC
- I3C signals that the Target device may transmit to the I3C bus

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

Figure 2.1 shows the functional diagram of the IP Core with Secondary Controller Capability.

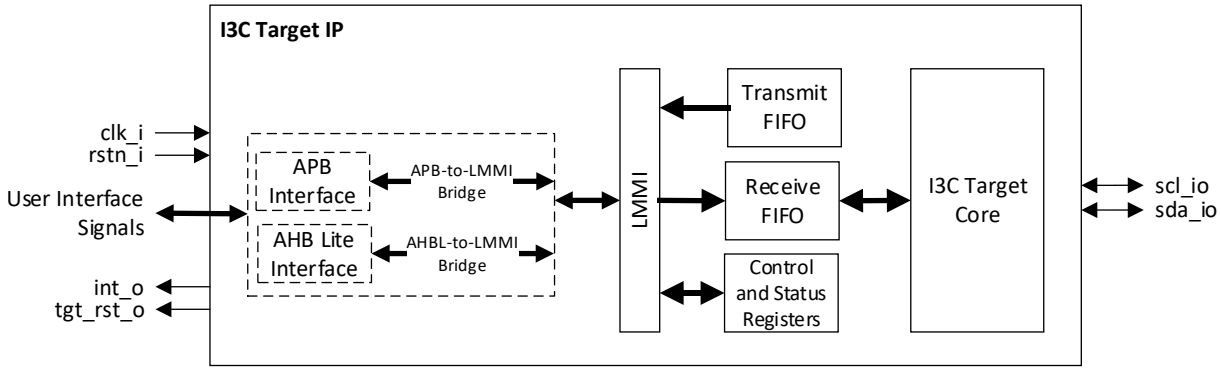

Figure 2.1. I3C Target IP Core Block Diagram

2.2. Clocking

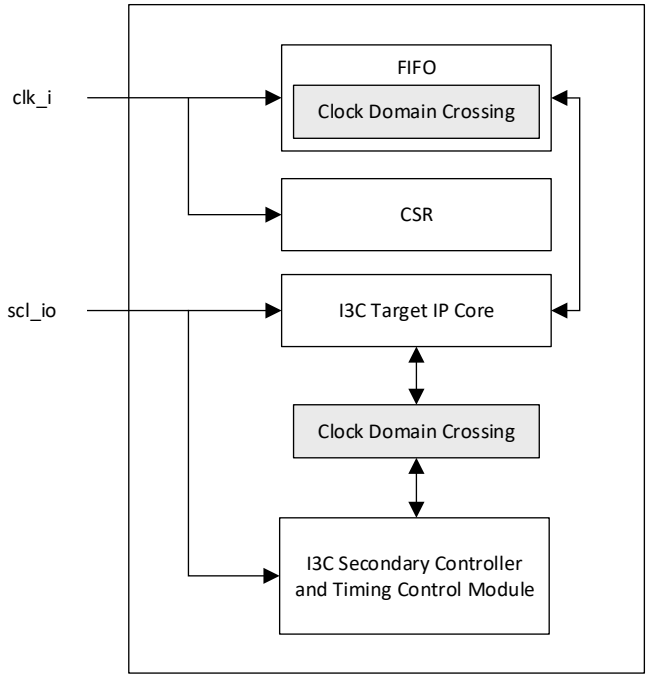

Figure 2.2. I3C Target IP Clock Domain Block Diagram

2.2.1. Clocking Overview

The I3C Target IP has the following clocks:

- clk_i
  System clock – Can be set in the range of 0.8 MHz – 50 MHz. Used to clock internal FIFO, register access, and I3C Timing Control and Secondary Controller support logic.
- scl_io
  Serial clock – Can support up to 12.5 MHz (based on I3C specification). Clock for I3C Target IP core.

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.

2.4. User Interfaces

Table 2.1 shows the user interfaces and supported protocols. The memory-mapped interface of I3C Target 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 the Lattice I3C Target IP core and external I3C controller and target devices. Refer to the MIPI I3C Specification for more information on the I3C protocol.

2.5. I3C Transfers in SDR Mode

The following section describes the I3C Target response for different I3C transactions from the Controller in SDR mode.

2.5.1. Broadcast CCC

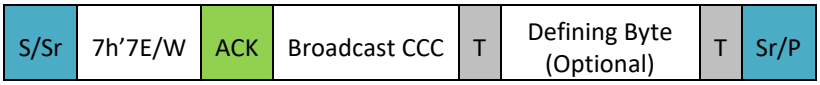

Figure 2.3. Broadcast CCC

When I3C Controller sends a broadcast CCC using the above format, the Target will ACK the I3C Broadcast Address 7'h7E/W. Depending on the received CCC and optional defining byte, the Target will process the command and respond accordingly.

2.5.2. Direct CCC

Figure 2.4. Direct CCC

When I3C Controller sends a direct CCC to the Target using the above format, the Target will ACK the I3C Broadcast Address 7'h7E/W. When the Target Address is transmitted in the bus at the address header, it will either:

- ACK if the CCC and the optional defining byte are both supported and the CCC T-bit is correct.
- NACK if CCC is not supported or CCC T-bit is incorrect.

Depending on the received CCC, the Target will process the command and respond accordingly.

2.5.3. Private Write

Figure 2.5. I3C Private Write Initiated with START Condition

Figure 2.6. I3C Private Write Initiated with Repeated START Condition

When I3C Controller initiates a Private Write to Target, the Target will ACK if it receives the I3C Broadcast Address 7'h7E/W or its own Address followed by a Write bit.

Data written by the Controller will be stored in the Rx FIFO of Target. If enabled in Target, interrupt rxfifo_not_empty will be asserted to notify that Rx FIFO has data for reading.

2.5.4. Private Read

Write data to the Tx FIFO of Target first before the I3C Controller initiates a Private Read.

Figure 2.7. I3C Private Read Initiated with START Condition

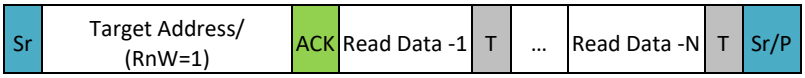

Figure 2.8. I3C Private Read Initiated with Repeated START Condition

When I3C Controller initiates a Private Read from Target, the Target will ACK if it receives the I3C Broadcast Address 7'h7E/W then its own Address followed by a Read bit. It will then proceed to transmit data that is stored in the Tx FIFO. Target will continue sending the Read data until Read is aborted by one of the following scenarios:

- Controller stops the Read by pulling SDA Low in the T-bit of data
- All content of Tx FIFO has been transmitted. Target will signal end of Message by pulling SDA Low while SCL is Low in the T-bit of the last data then releasing SDA when it sees SCL positive edge. The Controller shall then take over SDA and generate a STOP or a Repeated START.

If Tx FIFO is empty when Controller initiates Private Read, Target will respond depending on txfifo_empty_rd_nak register:

- If txfifo_empty_rd_nak is set to 0 (Default), Target will ACK its Address and return 0xFF Read Data then pull SDA Low at T-bit to signal end-of-message by default. If enabled, *read_txfifo_empty* interrupt will be set to notify the Private Read attempt by the Controller.
- If txfifo_empty_rd_nak is set to 1, Target will NACK its Address.

2.6. I2C Mode

When Target is not yet assigned a Dynamic Address, it may act as an I2C Target given that it is assigned a Static Address. To assign a Static Address to the I3C Target IP, check the Static Address Enable attribute and input a valid Static Address (must not use reserved I3C addresses).

If Target is already assigned a Dynamic Address, it will no longer ACK when its Static Address is transmitted in the I3C bus.

2.6.1. I2C Write

In I2C Write, Target will ACK when its Static Address is transmitted in the I3C bus. The Controller then proceeds to send the 8-bit write data, then Target will pull down ACK at 9th bit to accept data.

Figure 2.9. I2C Write

2.6.2. I2C Read

In I2C Read, Target will ACK when its Static Address is transmitted in the I3C bus. It will then proceed to transmit read data in Open Drain mode then Controller will either: (1) ACK and continue reading the data or (2) NACK to end reading of data.

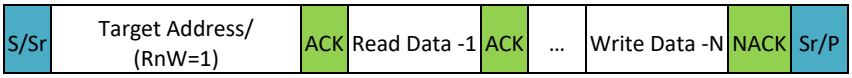

Figure 2.10. I2C Read

If Tx FIFO is empty when Controller initiates I2C Read, Target will respond depending on the txfifo_empty_rd_nak register.

- If txfifo_empty_rd_nak is set to 0 (Default), Target will ACK its Address and return 0xFF Read Data then wait for Controller to end I2C read by sending NACK. If enabled, *read_txfifo_empty* interrupt will be set to notify the Private Read attempt by the Controller.
- If txfifo_empty_rd_nak is set to 1, Target will NACK its Address.

2.7. Common Command Codes

Table 2.2 lists the supported CCC of this IP.

Table 2.2. I3C Target IP Supported CCC

CCC	Type	Required[1]	Command
0x00	Broadcast	R	ENEC
0x01	Broadcast	R	DISEC
0x06	Broadcast	R	RSTDAA
0x07	Broadcast	R	ENTDAA
0x09	Broadcast	R	SETMWL
0x0A	Broadcast	R	SETMRL
0x2A	Broadcast	R	RSTACT
0x80	Direct	R	ENEC
0x81	Direct	R	DISEC
0x89	Direct	R	SETMWL
0x8A	Direct	R	SETMRL
0x8B	Direct	R	GETMWL
0x8C	Direct	R	GETMRL
0x90	Direct	R	GETSTATUS
0x9A	Direct	R	RSTACT
0x02	Broadcast	C	ENTAS0
0x03	Broadcast	O	ENTAS1
0x04	Broadcast	O	ENTAS2
0x05	Broadcast	O	ENTAS3
0x28	Broadcast	C	SETXTIME[2]
0x29	Broadcast	O	SETAASA
0x82	Direct	C	ENTAS0
0x83	Direct	O	ENTAS1
0x84	Direct	O	ENTAS2
0x85	Direct	O	ENTAS3
0x87	Direct	O	SETDASA
0x88	Direct	C	SETNEWDA
0x8D	Direct	C	GETPID
0x8E	Direct	C	GETBCR
0x8F	Direct	C	GETDCR
0x94	Direct	C	GETMXDS
0x95	Direct	C	GETCAPS
0x98	Direct	C	SETXTIME[2]
0x99	Direct	C	GETXTIME

Notes:

1. R – Required, O – Optional, and C – Conditional.
2. Supported Defining Bytes for SETXTIME CCC: 0xDF and 0xFF.

2.8. I3C Transfers in HDR-DDR Mode

An HDR-DDR Mode period in the I3C bus involves five steps:

1. The Controller sends a Broadcast Enter HDR-DDR Mode Broadcast CCC indicating which HDR-DDR Mode to enter.
2. The I3C bus switches from SDR Mode to the requested HDR-DDR Mode.
3. The Controller issues the first structured protocol per the HDR-DDR Mode framing, typically a Command or Header followed by optional Data sent by the Controller or the Target.
4. The Controller sends an HDR Restart Pattern or Exit Pattern.
5. If an HDR Restart Pattern is sent, then the Controller issues another structured protocol element for the New HDR-DDR Mode transfer. The Controller may repeat this process to remain in HDR-DDR Mode or send an HDR Exit Pattern to exit the HDR-DDR Mode.
6. If the Controller sends an HDR Exit Pattern, then it is always followed by an I3C STOP, which ends in the Bus Free Condition.

2.8.1. Typical HDR-DDR Mode Frame

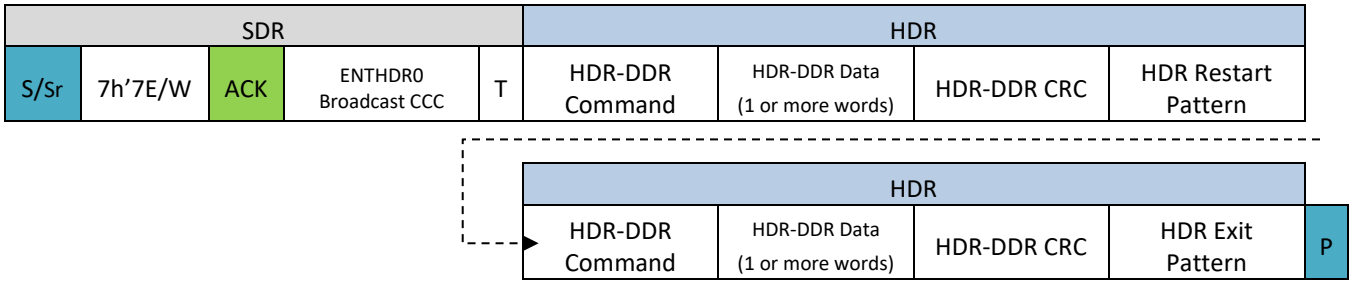

Figure 2.11. Typical HDR-DDR Mode Frame

2.8.2. HDR-DDR Write

	HDR-DDR Command			HDR-DDR Write Data (N)		HDR-DDR CRC	
Enter HDR or HDR Restart	[15] **1'b0** (Write) [14:8] Reserved	[7:1] Target Address [0] Parity Adj. (P0 = 1'b1)	ACK	[15:8] 1st Data Byte	[7:0] 2nd Data Byte	CRC5 for all Data	HDR Restart or Exit Pattern

Figure 2.12. HDR-DDR Write

2.8.3. HDR-DDR Read

	HDR-DDR Command			HDR-DDR Read Data (N)		HDR-DDR CRC	
Enter HDR or HDR Restart	[15] **1'b1** (Read) [14:8] Reserved	[7:1] Target Address [0] Parity Adj. (P0 = 1'b1)	ACK	[15:8] 1st Data Byte	[7:0] 2nd Data Byte	CRC5 for all Data	HDR Restart or Exit Pattern

Figure 2.13. HDR-DDR Read

2.8.4. HDR-DDR Broadcast CCC

	HDR-DDR CCC Indicator			HDR-DDR CCC Command	
Enter HDR or HDR Restart	[15] **1'b0** (Write) [14:8] Reserved	[7:1] Broadcast Address [0] Parity Adj. (P0 = 1'b1)	ACK	[15:8] Broadcast CCC	[7:0] Defining Byte, 0x0 if unused

HDR-DDR CCC Data (Optional)		HDR-DDR CCC CRC	HDR-DDR CCC End Procedure
[15:8] 1st Data Byte	[7:0] 2nd Data Byte	CRC for all Data	

Figure 2.14. HDR-DDR Broadcast CCC

2.8.5. HDR-DDR Direct Set CCC

	HDR-DDR CCC Indicator			HDR-DDR CCC Command		HDR-DDR CCC CRC
Enter HDR or HDR Restart	[15] **1'b0** (Write) [14:8] Reserved	[7:1] Broadcast Address [0] Parity Adj. (P0 = 1'b1)	ACK	[15:8] Direct CCC	[7:0] Defining Byte, 0x0 if unused	CRC for CCC Command

	HDR-DDR CCC Selector			HDR-DDR CCC Set Data (N times)		HDR-DDR CCC CRC
HDR Restart	[15] **1'b0** (Write) [14:8] Reserved	[7:1] Target Address [0] Parity Adj. (P0 = 1'b1)	ACK	[15:8] 1st Data Byte	[7:0] 2nd Data Byte	CRC for all Data

HDR-DDR CCC End Procedure

Figure 2.15. HDR-DDR Direct Set CCC

2.8.6. HDR-DDR Direct Get CCC

	HDR-DDR CCC Indicator			HDR-DDR CCC Command		HDR-DDR CCC CRC
Enter HDR or HDR Restart	[15] **1'b0** (Write) [14:8] Reserved	[7:1] Broadcast Address [0] Parity Adj. (P0 = 1'b1)	ACK	[15:8] Direct CCC	[7:0] Defining Byte, 0x0 if unused	CRC for CCC Command

	HDR-DDR CCC Selector			HDR-DDR CCC Get Data (N times)		HDR-DDR CCC CRC
HDR Restart	[15] **1'b1** (Read) [14:8] Reserved	[7:1] Target Address [0] Parity Adj. (P0 = 1'b1)	ACK	[15:8] 1st Data Byte	[7:0] 2nd Data Byte	CRC for all Data

Target to Controller Handoff	HDR-DDR CCC End Procedure

Figure 2.16. HDR-DDR Direct Get CCC

2.9. Hot-Join Mechanism

The I3C Target Device may issue a Hot-Join request to join the I3C bus after the bus is already configured. Target initiates Hot-Join Request by sending an IBI using the reserved address 7'h02 with Write bit after a START condition.

The Target may initiate Hot-Join by one of the following methods:

- Passive – I3C Target may wait for a START condition transmitted in the I3C bus. It may then proceed to transmit the reserved address 7’h02 in the arbitrable address header.
  Note: I3C Target needs to know first if it is in an I3C bus before initiating Hot-Join passively. An I3C bus is determined by an SDR Frame with START followed by the Broadcast Address. See Errata 01 for MIPI I3C Basic Specification, Specification Version 1.1.1.
- Active – If the I3C Target detects Bus Idle condition, it may generate a START by pulling the SDA line Low and waiting for the I3C Controller to complete the start condition by pulling SCL low.

The Target will then proceed to send the reserved address 7’h02 in the arbitrable address header after the START condition.

2.9.1. Generating a Hot-Join Request

The I3C Target IP supports Hot-Join when the Hot-Join Capable attribute is checked.

To initiate a Hot-Join request, confirm that Hot-Join is allowed by the Controller by reading the hj_en_ec register. If allowed (hj_en_ec is set to 1’b1), set the hj_req register to 1’b1. I3C Target will attempt to transmit the Hot-Join request in the I3C bus by either active or passive method as described above. If the Target loses address arbitration, it will stop transmitting the Hot-Join address and wait for the next valid opportunity to retransmit the Hot-Join request depending on retry settings. When the Hot-Join request has been generated, hj_req_gen interrupt will be asserted.

If I3C Controller ACKs the request, I3C Target has successfully joined the I3C bus and hj_done interrupt will be asserted to inform that Hot-Join was successful. The Target will then wait for Dynamic Address Assignment.

If the Hot-Join request is NACKed by I3C Controller, hj_acnack interrupt will be asserted and I3C Target will then wait for the next valid condition to retry transmitting the Hot-Join request until the I3C Controller ACKs the request. If the Hot-Join request has been generated and NACKed by the Controller for hj_ibi_retry times, hj_done and hj_acknack interrupts will be asserted to inform that Hot-Join was not successful.

2.10. In-Band Interrupt

In-Band Interrupts may be issued by I3C Targets to signal pending action from the I3C Controller.

The I3C Target issues IBI by sending its own address in the Arbitrated Address Header with a Read bit after a START condition.

The Target may initiate IBI by one of the following methods:

- Passive – I3C Target may wait for a START condition transmitted in the I3C bus. It may then proceed to transmit its own dynamic address in the arbitrable address header.
- Active – If I3C bus Available condition is detected, I3C Target may generate a START by pulling the SDA line Low and waiting for the I3C Controller to complete the start condition by pulling SCL low.

The Target will then proceed to transmit its address in the arbitrable address header after the START condition.

2.10.1. Generating an In-band Interrupt

The I3C Target IP supports IBI when the IBI Capable attribute is checked.

To initiate an In-Band Interrupt, confirm that IBI is allowed by the Controller by reading the ibi_en_ec register. If allowed (ibi_en_ec is set to 1’b1), set the ibi_req register to 1’b1. I3C Target will attempt to transmit its own Dynamic Address in the arbitrable Address Header after a START condition. If the Target loses address arbitration, it will stop sending the IBI and wait for the next valid opportunity to retry sending the IBI. If IBI has been successfully generated, ibi_req_gen interrupt will be asserted.

If IBI is ACKed by the Controller, I3C Controller shall read the payload following IBI depending on BCR[2] of I3C Target.

- If BCR[2] is set to 0: there is no data byte following the IBI. ibi_done interrupt will be asserted.
- If BCR[2] is set to 1: the IBI has the following Mandatory Data Byte that shall be read by the Controller. The Target may also optionally send additional IBI data bytes (up to IBI Payload Size Limit by the Controller via the SETMRL CCC command). ibi_done interrupt will be asserted when reading of the IBI payload is stopped either by the Target or the Controller.

If IBI is NACKed by the I3C Controller, ibi_acknack interrupt will be asserted and I3C Target will then wait for the next valid condition to send the IBI until the I3C Controller ACKs the request. If IBI has been generated and NACKed by the Controller for hj_ibi_retry times, ibi_done and ibi_acknack interrupts will also be set to 1 to inform that IBI was not successful.

2.10.2. Sending the IBI Payload

To send an IBI with mandatory data byte and optional additional payload (up to maximum IBI payload size), write the data bytes to Tx FIFO with MDB as the first data and the additional IBI payload as the succeeding data. When the IBI is generated by Target and ACKed by the Controller, the Target will proceed to send the MDB and the additional data bytes until the maximum IBI payload size.

When all IBI payload is transmitted, that is, maximum IBI payload size has been reached or Tx FIFO is empty, Target will end the transfer by pulling SDA Low at T-bit. ibi_done interrupt will be asserted to notify that IBI is done, and all payload has been transmitted.

If the Controller chooses to end reading the additional IBI payload by pulling SDA Low during T-bit, ibi_done and ibi_payld_terminated interrupts will be set to 1 to notify that IBI transfer is done but payload transfer is incomplete. When this interrupt is received, you may choose to reset the Tx FIFO by setting the txfifo_rst register.

2.10.3. Pending Read Notification

Pending Read Notification (MDB[7:5] = 3'b101) is supported by this IP.

If the I3C Controller accepts an IBI and reads the Pending Read Notification MDB from the Target, the Target considers Pending Read Notification as Active. Read Data associated with the MDB shall be available at the next Private Read.

For pending read notification, write the data bytes to the Tx FIFO in this order:

1. Pending Read Notification MDB
2. IBI Payload (equal to maximum IBI payload size)
3. Associated read data

Only one Pending Read Notification may be active at a time while the Target is waiting for the Controller to read data. This means that when a Pending Read Notification is active, the Target cannot send another IBI with MDB for Pending Read Notification.

2.10.4. Limitation for IBI and Private Read

IBI Payload and read data are both stored in the Tx FIFO. If IBI Payload or Data for Private Read are written continuously to the Tx FIFO without being transmitted to the Controller (via IBI or Private Read), the intended data may not be sent correctly by the Target.

Ensure that all data corresponding to an intended IBI or Private Read has been transferred before initiating another IBI or Private Read. You may read the status registers ibi_payld_terminated (for IBI) and read_aborted (for Private Read) to confirm if all contents of Tx FIFO have been transmitted. If the status registers are set, transfer of Tx FIFO contents is not completed. When this occurs, you have the option to reset the Tx FIFO before writing new data for the next IBI or Private Read to ensure correctness of data.

2.11. Target Reset

Target Reset Action is configured by the Controller via RSTACT CCC. This will be the action of Target when it receives the Target Reset Pattern following an RSTACT CCC in a single frame. Following are the supported Defining Bytes of this IP:

Table 2.3. Target Reset

Defining Byte	Action	Description
0x0	No Action	No action
0x1	Reset I3C Peripheral only	Reset I3C states and FIFO. Equivalent to ip_core_rst soft reset
0x2	Reset Whole Target	Reset whole Target including DAA and previously configured settings via CCCs and register access. Equivalent to ip_main_rst soft reset.

When the IP receives RSTACT CCC, it informs you by asserting rstact_ccc_rcvd interrupt. The configured reset action is stored in tgt_rst_act_set register. Read this register to determine what action to do when the Target Reset Pattern is received. When Target detects the following Target Reset Pattern, it then asserts tgt_rst_ptrn_rcvd interrupt. When you see this interruption, you may then perform the configured reset action or inaction. You may use the soft reset registers, or the system reset (for whole Target reset) to perform the configured reset action.

The behavior is different when Target receives Target Reset Pattern without RSTACT. When Target receives Target Reset Pattern without RSTACT for the first time, it automatically resets the I3C Peripheral. If this occurs a second time, tgt_rst_o output is asserted. This informs you to reset Whole Chip (Target Reset Escalation).

2.12. Secondary Controller Support

The I3C Target IP is instantiated when Secondary Controller feature is enabled in the I3C Controller IP. All the capabilities of the I3C Target will then be available to the Secondary Controller Capable device.

2.12.1. Generating the Controller Role Request

To initiate a Controller Role request, confirm that Controller Role request is allowed by the Controller by reading the cr_en_ec register. If allowed (cr_en_ec is set to 1'b1), set the cr_req register to 1'b1. I3C Target will attempt to transmit its own Dynamic Address in the arbitrable Address Header after a START condition. If Target loses address arbitration, it will stop sending the Controller Role request and wait for the next valid opportunity to retry sending the request. If the Controller Role request has been successfully generated, cr_req_gen interrupt will be asserted.

If the request is ACKed by the Controller, cr_req_done interrupt will be asserted.

If the request is NACKed by the I3C Controller, I3C Target will then wait for the next valid condition to send the request until the I3C Controller ACKs. If the Controller Role request has been generated and NACKed by the Controller for hj_ibi_retry times, cr_req_done and cr_req_acknack interrupts will also be set to 1 to inform that Controller Role request was not successful.

2.12.2. CCC for Secondary Controller Support

Table 2.4 lists the additional CCCs supported when Secondary Controller feature is enabled.

Table 2.4. CCCs for Secondary Controller Support

CCC	Type	Command Name
0x08	Broadcast	DEFTGTS
0x91	Direct	GETACCCR

2.13. HDR-DDR Support

Table 2.5 lists CCCs supported by this IP in SDR mode and permitted in HDR-DDR mode, with notes and limitations.

Table 2.5. CCCs Supported in HDR-DDR Mode

CCC	Type	Command Name	Notes and Limitations	Supported in this IP
0x00	Broadcast	ENEC	May not generally be useful in HDR-DDR Modes, 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 Basic v1.1.1 specification.	Yes
0x01	Broadcast	DISEC		Yes
0x02	Broadcast	ENTAS0	See Section 5.2.1.2.4 of MIPI I3C Basic v1.1.1 specification	Yes
0x03	Broadcast	ENTAS1	See Section 5.2.1.2.4 of MIPI I3C Basic v1.1.1 specification	Yes
0x04	Broadcast	ENTAS2	See Section 5.2.1.2.4 of MIPI I3C Basic v1.1.1 specification	Yes
0x05	Broadcast	ENTAS3	See Section 5.2.1.2.4 of MIPI I3C Basic v1.1.1 specification	Yes
0x08	Broadcast	DEFTGTS	Not recommended for use in HDR-DDR Modes; generally only used for Secondary Controllers, to announce lists of known Targets, following changes to Target Dynamic Addresses or Hot-Join events, neither of which are permitted within HDR-DDR Modes.	No
0x09	Broadcast	SETMWL	See Section 5.2.1.2.4 of MIPI I3C Basic v1.1.1 specification	Yes
0x0A	Broadcast	SETMRL	See Section 5.2.1.2.4 of MIPI I3C Basic v1.1.1 specification	Yes
0x12	Broadcast	ENDXFER	Not recommended for use in HDR-DDR Modes	Yes
0x28	Broadcast	SETXTIME	No limitations	Yes
0x2A	Broadcast	RSTACT	Not recommended for use in HDR-DDR Modes	No
0x80	Direct	ENEC	May not generally be useful in HDR-DDR Modes, 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 Basic v1.1.1 specification.	Yes
0x81	Direct	DISEC		Yes
0x82	Direct	ENTAS0	See Section 5.2.1.2.4 of MIPI I3C Basic v1.1.1 specification	Yes
0x83	Direct	ENTAS1	See Section 5.2.1.2.4 of MIPI I3C Basic v1.1.1 specification	Yes
0x84	Direct	ENTAS2	See Section 5.2.1.2.4 of MIPI I3C Basic v1.1.1 specification	Yes
0x85	Direct	ENTAS3	See Section 5.2.1.2.4 of MIPI I3C Basic v1.1.1 specification	Yes
0x89	Direct	SETMWL	See Section 5.2.1.2.4 of MIPI I3C Basic v1.1.1 specification	Yes
0x8A	Direct	SETMRL	See Section 5.2.1.2.4 of MIPI I3C Basic v1.1.1 specification	Yes
0x8B	Direct	GETMWL	No limitations	Yes
0x8C	Direct	GETMRL	No limitations	Yes
0x8E	Direct	GETBCR	Not recommended for use in HDR-DDR Modes	Yes
0x8F	Direct	GETDCR	Not recommended for use in HDR-DDR Modes	Yes
0x90	Direct	GETSTATUS	No limitations	Yes
0x92	Direct	ENDXFER	Not recommended for use in HDR-DDR Modes	Yes
0x94	Direct	GETMXDS	No limitations	Yes
0x95	Direct	GETCAPS	No limitations	Yes
0x98	Direct	SETXTIME	No limitations	Yes
0x99	Direct	GETXTIME	No limitations	Yes
0x9A	Direct	RSTACT	Not recommended for use in HDR-DDR Modes	No

3. IP Parameter Description

The configurable attributes of the I3C Target 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
General		
User Interface	LMMI, **APB**, AHBL	Selects memory-mapped interface from the list for register access by the host.
Address Offset	Address Offset in Bytes, **Address Offset in DWORD**	Selects addressing mode for registers. Selectable only if the interface is APB or AHB-Lite. When connecting to a RISC-V processor, use *Address Offset in DWORD*. If Interface is LMMI, this is fixed to Address offset in Bytes.
FIFO		
FIFO Implementation	**EBR**, LUT, HARD_IP	Selects the FPGA resource 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.
I/O Primitive Enable		
Enable internal 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.
Bus Characteristics		
Bus Type	**SDR only**, HDR-capable	Indicates whether IP supports SDR mode only or both SDR and HDR-DDR modes.
HDR Mode	—	Display only when Bus Type = HDR-capable. Indicates which HDR mode is supported. Default value is HDR-DDR.
IBI Capable	**Checked**, Unchecked	Determines whether IP can support IBI Capability.
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.
Hot-Join Capable	**Checked**, Unchecked	Determines whether IP can support Hot-Join Capability.
Maximum Data Speed Limitation	**Checked**, Unchecked	Sets Bit 0 of BCR. When set to 1, Target device supports GETMXDS CCC to relay timing information of its speed limitation to the Controller.
Device Characteristics		
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 the MIPI 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.

Attribute	Selectable Values	Description
Additional ID	0–4095	Defines Provisional ID[11:0] bits. Default value is 0.
Static Address Enable	**Checked**, Unchecked	Indicates that 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
Timing Characteristics		
System Clock Frequency (MHz)	0.8–50	Sets the system clock (clk_i) frequency. Default value is 25. Refer to Clocking Overview section for more details.
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. Indicates 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 addresses must not use I2C reserved or I3C invalid addresses.

3.2. IP Parameter Settings for Example Use Cases

Table 3.2 shows the parameter settings for example test cases. Wherever applicable, parameters equal to "—" are not editable or automatically set in IP configuration GUI.

Table 3.2. IP Parameter Settings for Example Use Cases

Target Use Case	SDR-only, I3C device only (No static address)	HDR-DDR, Bidirectional Buffer not instantiated inside IP
General		
CPU Interface		
User Interface	APB	APB
Address Offset	Address Offset in DWORD	Address Offset in DWORD
Optional Interface		
Enable Direct FIFO Interface	Unchecked	Unchecked
Tx Data Width	—	—
Rx Data Width	—	—
I/O Primitive Enable		
Enable internal IO primitive	Checked	Unchecked
Bus Characteristics		
Bus Type	SDR only	HDR-capable
HDR Mode	N/A	—
IBI Capable	Checked	Checked
IBI Payload Size (including MDB)	1	1
Hot-Join Capable	Checked	Checked
Maximum Data Speed Limitation	Checked	Checked
Device Characteristics		
DCR (HEX)	00	00
Manufacturer ID	414	414

Target Use Case	SDR-only, I3C device only (No static address)	HDR-DDR, Bidirectional Buffer not instantiated inside IP
Part ID	1	1
Instance ID	1	1
Additional ID	0	0
Static Address Enable	Unchecked	Checked
Static Address (HEX)	N/A	08
Timing Characteristics		
System Clock Frequency (MHz)	25	25
Write Maximum Data Rate (MHz)	—	—
Clock-to-data Turnaround Delay (ns) (t_{SCO})	—	—
Read Maximum Data Rate (MHz)	—	—
Maximum Read Turnaround Time (µs)	—	—

4. Signal Description

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

Table 4.1. Ports Description

Port	Type	Description
System Clock and Reset		
clk_i	Input	System Clock. Refer to Clocking Overview section for more details.
rst_n_i	Input	Asynchronous active low reset. Refer to the Reset section for more details.
tgt_rst_o	Output	Active High flag for Full Chip Reset by Target Reset escalation[1]
I3C Interface (Internal I/O Primitive)[2]		
scl_io	Input/Output	Bidirectional I3C Serial Clock. Refer to Clocking Overview section for more details.
sda_io	Input/Output	Bidirectional I3C Serial Data
I3C Interface (External I/O Primitive)[3]		
ext_scl_i	Input	I3C Serial Clock Input
ext_sda_i	Input	I3C Serial Data Input
ext_sda_o	Output	I3C Serial Data Output
ext_sda_oe	Output	Active High I3C Serial Data Output Enable
LMMI Interface[4]		
lmmi_request_i	Input	Start transaction
lmmi_wr_rdn_i	Input	Write = High, Read = Low.
lmmi_offset_i[7:0]	Input	Register offset within Target, starting at offset 0.
lmmi_wdata_i[7:0]	Input	Write data
lmmi_rdata_o[7:0]	Output	Read data
lmmi_rdata_valid_o	Output	Read transaction is complete and lmmi_rdata_o contains valid data
lmmi_ready_o	Output	Target is ready to start a new transaction
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[5]		
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
ahbl_hwrite_i	Input	AHB-Lite Direction signal. Write = High, Read = Low.
ahbl_hwdata_i[31:0]	Input	AHB-Lite Write Data signal. Tie Field[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. Field[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.

Port	Type	Description
APB Interface[6]		
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 Field[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. Field[31:24] are tied to 0. Refer to the Register Description section for more details.
apb_pslverr_o	Output	APB Completer Error signal
Direct FIFO Interface[7]		
tx_valid_i	Input	Tx data valid signal. Set this signal to high when sending data (tx_data_i) to the Tx FIFO.
tx_ready_o	Output	Tx FIFO ready signal. When the signal is high, it indicates that the Tx FIFO can accept incoming data. tx_data_i is written to the 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 the signal is high, it indicates that the Rx FIFO is not empty and rx_data_o contains valid data.
rx_ready_i	Input	Rx ready signal. Set this signal to high when it is ready to accept the Rx data. rx_data_o takes the next Rx FIFO entry when rx_valid_o and rx_ready_i == 1.
rx_data_o [7:0]	Output	Rx data

Notes:

1. For more details on target reset, see Section 5.1.11 of the MIPI I3C Basic v1.1.1 Specification.
2. Bidirectional I3C interface is only available when internal I/O primitives are enabled in IP configuration GUI.
3. External I/O I3C interface is only available when internal I/O primitives are disabled in IP configuration GUI.
4. LMMI Interface is only available when selected from the User Interface.
5. AHB-Lite Interface is only available when selected from the User Interface. Refer to the AMBA 3 AHB-Lite Protocol v1.0 Specification for details of the protocol.
6. APB Interface is only available when selected from the User Interface. Refer to the AMBA 3 APB Protocol v1.0 Specification for details of the protocol.
7. Direct FIFO interface is only available when enabled in IP configuration GUI.

5. Register Description

This section defines the configuration, control, and status registers of the I3C Target 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 address 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-L 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. 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.

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 Target IP Register Summary

Byte Address Offset[1]	DWORD Address Offset[1]	Register Name	Description
0x00	0x000	Bus Characteristics	Target Bus Characteristics register.
0x01	0x004	Device Characteristics	Target Device Characteristics register.
0x02	0x008	Dynamic Address	Assigned dynamic address and done flag.
0x03	0x00C	Events Command Enable	Hot-Join and IBI enable from the Controller through ENEC/DISEC CCC.
0x04	0x010	Events Command Device Configuration	Hot-Join and IBI capability configured using attributes.
0x05	0x014	Events Command Request	Hot-Join and IBI request.
0x06	0x018	Hot-Join/IBI Retry	Number of retries for Hot-Join and IBI.
0x07	0x01C	Maximum Write Length (MSB)	The most significant byte of the Maximum Write Length set by the Controller.
0x08	0x020	Maximum Write Length (LSB)	The least significant byte of the Maximum Write Length set by the Controller.
0x09	0x024	Maximum Read Length (MSB)	The most significant byte of the Maximum Read Length set by the Controller.
0x0A	0x028	Maximum Read Length (LSB)	The least significant byte of the Maximum Read Length set by the Controller.
0x0B	0x02C	Maximum IBI Payload Size	The maximum IBI payload size set by the Controller.
0x0C	0x030	Maximum Write Data Speed (MaxWr)	Applicable only when the Target is configured with Maximum Data Speed Limitation.
0x0D	0x034	Maximum Read Data Speed (MaxRd)	Applicable only when the Target is configured with Maximum Data Speed Limitation.
0x0E	0x038	Maximum Read Turnaround Time Byte2	Applicable only when the Target is configured with Maximum Data Speed Limitation.

Byte Address Offset[1]	DWORD Address Offset[1]	Register Name	Description
0x0F	0x03C	Maximum Read Turnaround Time Byte1	Applicable only when the Target is configured with Maximum Data Speed Limitation.
0x10	0x040	Maximum Read Turnaround Time Byte0	Applicable only when the Target is configured with Maximum Data Speed Limitation.
0x11	0x044	Device Provisioned ID Byte6	6th byte of the Target Device PID.
0x12	0x048	Device Provisioned ID Byte5	5th byte of the Target Device PID.
0x13	0x04C	Device Provisioned ID Byte4	4th byte of the Target Device PID.
0x14	0x050	Device Provisioned ID Byte3	3rd byte of the Target Device PID.
0x15	0x054	Device Provisioned ID Byte2	2nd byte of the Target Device PID.
0x16	0x058	Device Provisioned ID Byte1	1st byte of the Target Device PID.
0x17	0x05C	Static Address	The Target static address that is assigned.
0x18	0x060	Device Capabilities Byte1	Device capabilities that the Target returns when the Controller sends GETCAPS CCC.
0x19	0x064	Device Capabilities Byte2	Device capabilities that the Target returns when the Controller sends GETCAPS CCC.
0x1A	0x068	Device Capabilities Byte3	Device capabilities that the Target returns when the Controller sends GETCAPS CCC.
0x1C	0x070	Oscillator Inaccuracy	Inaccuracy of the Target's internal oscillator.
0x20	0x080	Receive FIFO	Receive FIFO.
0x22	0x088	Transmit FIFO	Transmit FIFO.
0x28	0x0A0	Soft Reset	Soft reset.
0x29	0x0A4	Target Response	Target response.
0x2A	0x0A8	Get Status MSB	The most significant byte that the Target returns when the Controller sends GETSTATUS CCC.
0x2B	0x0AC	Get Status LSB	The least significant byte that the Target returns when the Controller sends GETSTATUS CCC.
0x2C	0x0B0	Bus Activity State	The least significant byte that the Target returns when the Controller sends GETSTATUS CCC.
0x2D	0x0B4	Target Reset Action 1	Target reset action information 1.
0x2E	0x0B8	Target Reset Action 2	Target reset action information 2.
0x2F	0x0BC	Target Reset Action 3	Target reset action information 3.
0x30	0x0C0	Interrupt Status 1	Interrupt status 1.
0x31	0x0C4	Interrupt Enable 1	Interrupt enable 1.
0x32	0x0C8	Interrupt Set 1	Interrupt set 1.
0x33	0x0CC	Interrupt Status 2	Interrupt status 2.
0x34	0x0D0	Interrupt Enable 2	Interrupt enable 2.
0x35	0x0D4	Interrupt Set 2	Interrupt set 2.
0x36	0x0D8	Interrupt Status 3	Interrupt status 3.
0x37	0x0DC	Interrupt Enable 3	Interrupt enable 3.
0x38	0x0E0	Interrupt Set 3	Interrupt set 3.
0x3C	0x0F0	Interrupt Status 5	Interrupt status 5.
0x3D	0x0F4	Interrupt Enable 5	Interrupt enable 5.
0x3E	0x0F8	Interrupt Set 5	Interrupt set 5.
0x50	0x140	Bus Mode	I3C bus mode.
0x51	0x144	HDR-DDR Target Configuration	HDR-DDR target configuration for received HDR data.
0x54	0x150	HDR-DDR Abort Configuration	HDR-DDR abort configuration set by the Controller through ENDXFER CCC.
0x55 – 0xFF	0x154 – 0x3FC	Reserved	Reserved.

Note:

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

5.1. Bus Characteristics Register 0x00

Target Bus Characteristics register.

Table 5.3. Bus Characteristics Register 0x00

Field	Name	Access	Description	Reset
[7:6]	device_role	RO	Fixed to 2'b00. Configuration options: • 2'b00 – I3C Target. • 2'b01 – I3C Controller Capable (Not implemented in this IP). • Others - Reserved for future definition by the MIPI Alliance I3C WG (Not supported in this IP).	0x0 (Fixed)
[5]	advanced_caps	RO	Configuration options: • 0 – Does not support optional advanced capabilities. • 1 – Supports optional advanced capabilities. When the Target is configured with IBI capability and IBI payload, this is set to 1 for optional Pending Read MDB support.	0x1 if IBI Capable is checked and IBI Payload Size > 0; otherwise, 0x0.
[4]	virtual_tgt_support	RO	Fixed to 0. Configuration options: • 0 – Is not a Virtual Target and does not expose other downstream Device(s). • 1 – Is a Virtual Target, or exposes other downstream Device(s) (Not supported in this IP).	0x0 (Fixed)
[3]	offline_capable	RO	Fixed to 0. Configuration options: • 0 – Device always respond to I3C bus commands • 1 – Device does not always respond to I3C bus Commands (Not supported in this IP).	0x0 (Fixed)
[2]	ibi_payload	RO	From the *IBI Payload Size* attribute. Configuration options: • 0 – If the *IBI Payload Size* attribute is set to 0. No data bytes follow the IBI. • 1 – If the *IBI Payload Size* attribute is set to greater than or equal to 1. One data byte (MDB) follows the accepted IBI, and additional data bytes may follow.	0x1 if IBI Capable is checked and IBI Payload Size > 0; otherwise, 0x0.
[1]	ibi_capable	RO	From the *IBI Capable* attribute. Configuration options: • 0 – Unchecked. Not capable. • 1 – Checked. Capable.	0x1 if IBI Capable is checked; otherwise, 0x0.
[0]	max_d_speed_limit	RO	From the *Maximum Data Speed Limitation* attribute. Configuration options: • 0 – Unchecked. No maximum data speed limitation. • 1 – Checked. With maximum data speed limitation. The Controller uses GETMXDS CCC to get specific limitation from the Target.	0x1 if Maximum Data Speed Limitation is checked; otherwise, 0x0.

5.2. Device Characteristics Register 0x01

Target Device Characteristics register.

Table 5.4. Device Characteristics Register 0x01

Field	Name	Access	Description	Reset
[7:0]	dcr	RO	Device ID set through the *DCR* attribute. 255 available codes for describing the type of sensor or Device (such as accelerometer, gyroscope, composite devices). Default value is 8'b0 for generic Device.	Takes the *DCR* parameter value.

5.3. Dynamic Address Register 0x02

Assigned dynamic address and done flag.

Table 5.5. Dynamic Address Register 0x02

Field	Name	Access	Description	Reset
[7]	daa_done	RO	Configuration options: • 0 – dyn_addr is not yet assigned or is reset by RSTDAA CCC. • 1 – dyn_addr is assigned through ENTDAA, SETNEWDA, SETAASA, or SETDASA CCC.	0x0
[6:0]	dyn_addr	RO	Dynamic address assigned to the I3C Target.	0x0

5.4. Events Command Enable Register 0x03

Hot-Join and IBI enable from the Controller through ENEC/DISEC CCC.

Table 5.6. Events Command Enable Register 0x03

Field	Name	Access	Description	Reset
[7:4]	reserved	RO	Reserved.	0x0
[3]	hj_enec	RO	Hot-Join enable by the Controller. Configuration options: • 0 – Target-initiated Hot-Join is not allowed on the I3C bus. • 1 – Target-initiated Hot-Join is allowed on the I3C bus.	0x1 – If the Target is Hot-Join Capable. 0x0 – If the Target is not Hot-Join Capable.
[2:1]	reserved	RO	Reserved.	0x0
[0]	ibi_enec	RO	IBI enable by the Controller. Configuration options: • 0 – Target-initiated interrupts are not allowed on the I3C bus. • 1 – Target-initiated interrupts are allowed on the I3C bus.	0x1 – If Target is IBI Capable. 0x0 – If Target is not IBI Capable.

5.5. Events Command Device Configuration Register 0x04

Hot-Join and IBI capability configured using attributes.

Table 5.7. Events Command Device Configuration Register 0x04

Field	Name	Access	Description	Reset
[7:4]	reserved	RO	Reserved.	0x0
[3]	hj_cap	RO	Configuration options: • 0 – Target device is configured without Hot-Join capability. • 1 – Target device is configured with Hot-Join capability.	0x1 – If Target is Hot-Join Capable. 0x0 – If Target is not Hot-Join Capable.
[2:1]	reserved	RO	Reserved.	0x0
[0]	ibi_cap	RO	Configuration options: • 0 – Target device is configured without IBI capability. • 1 – Target device is configured with IBI capability.	0x1 – If Target is IBI Capable. 0x0 – If Target is not IBI Capable.

5.6. Events Command Request Register 0x05

Hot-Join and IBI requests.

Table 5.8. Events Command Request Register 0x05

Field	Name	Access	Description	Reset
[7:4]	reserved	RO	Reserved.	0x0
[3]	hj_req	RW	When set to high, the Target initiates a Hot-Join request at the next valid opportunity. Automatically resets to 0 when Hot-Join is done or disabled by the Controller.	0x0
[2:1]	reserved	RO	Reserved.	0x0
[0]	ibi_req	RW	When set to high, the Target initiates IBI at the next valid opportunity. Automatically resets to 0 when IBI is done or disabled by the Controller.	0x0

5.7. Hot-Join/IBI Retry Register 0x06

Number of retries for Hot-Join and IBI.

Table 5.9. Hot-Join/IBI Retry Register 0x06

Field	Name	Access	Description	Reset
[7:0]	hj_ibi_retry	RW	Indicates the number of times the Target retries Hot-Join or IBI when it is NACKed by the Controller. When set to 0, the number of tries is unlimited.	0x8

5.8. Maximum Write Length (MSB) Register 0x07

The most significant byte of maximum write length set by the Controller.

Table 5.10. Maximum Write Length (MSB) Register 0x07

Field	Name	Access	Description	Reset
[7:0]	mwl_msb	RO	The most significant byte of the maximum write	Takes the upper 8 bits of the *FIFO*

Field	Name	Access	Description	Reset
			length set by the I3C Controller through SETMWL CCC. If the value of SETMWL exceeds the FIFO depth, then the actual MWL is equal to the FIFO depth.	*Depth* parameter value.

5.9. Maximum Write Length (LSB) Register 0x08

The least significant byte of maximum write length set by the Controller.

Table 5.11. Maximum Write Length (LSB) Register 0x08

Field	Name	Access	Description	Reset
[7:0]	mwl_lsb	RO	The least significant byte of the maximum write length set by the I3C Controller through SETMWL CCC. If the value of SETMWL exceeds the FIFO depth, then the actual MWL is equal to the FIFO depth.	Takes the lower 8 bits of the *FIFO Depth* parameter value.

5.10. Maximum Read Length (MSB) Register 0x09

The most significant byte of maximum read length set by the Controller.

Table 5.12. Maximum Read Length (MSB) Register 0x09

Field	Name	Access	Description	Reset
[7:0]	mrl_msb	RO	The most significant byte of the maximum read length set by the I3C Controller through SETMRL CCC. If the value of SETMRL exceeds the FIFO depth, then the actual MRL is equal to the FIFO depth.	Takes the upper 8 bits of the *FIFO Depth* parameter value.

5.11. Maximum Read Length (LSB) Register 0x0A

The least significant byte of the maximum read length set by the Controller.

Table 5.13. Maximum Read Length (LSB) Register 0x0A

Field	Name	Access	Description	Reset
[7:0]	mrl_lsb	RO	The least significant byte of the maximum read length set by the I3C Controller through SETMRL CCC. If the value of SETMRL exceeds the FIFO depth, then the actual MRL is equal to the FIFO depth.	Takes the lower 8 bits of *FIFO Depth* parameter value.

5.12. Maximum IBI Payload Size Register 0x0B

The maximum IBI payload size set by the Controller. Applicable only when the Target is configured with IBI capability and IBI payload is set to greater than or equal to 1.

Table 5.14. Maximum IBI Payload Size Register 0x0B

Field	Name	Access	Description	Reset
[7:0]	max_ibi_payld	RO	The maximum IBI Payload set by the I3C Controller through SETMRL CCC. If the value exceeds either the FIFO depth or 0xFF, the actual maximum IBI	Takes the *IBI Payload Size* (including MDB) parameter value.

Field	Name	Access	Description	Reset
			payload size is equal to whichever is lower between the FIFO depth and 0xFF. Unlimited payload size is not supported in this IP.	

5.13. Maximum Write Data Speed (MaxWr) Register 0x0C

Applicable only when the Target is configured with Maximum Data Speed Limitation.

Table 5.15. Maximum Write Data Speed (MaxWr) Register 0x0C

Field	Name	Access	Description	Reset
[7:4]	reserved	RO	Reserved.	0x0
[3]	mxds2_w_defbyte	RO	Configuration options: • 0 – Target does not support defining byte for GETMXDS CCC. • 1 – Target supports defining byte for GETMXDS CCC. (Not supported in this IP)	0x0 (Fixed)
[2:0]	mxds2_w_rate	RW	Maximum Sustained Data Rate for non-CCC Messages sent by the Controller device to the Target device. Configuration options: • 0 – f_{SCL} Max (default value) • 1 – 8 MHz • 2 – 6 MHz • 3 – 4 MHz • 4 – 2 MHz • Others – Reserved for future use by the MIPI Alliance.	0x0

5.14. Maximum Read Data Speed (MaxRd) Register 0x0D

Applicable only when the Target is configured with Maximum Data Speed Limitation.

Table 5.16. Maximum Read Data Speed (MaxRd) Register 0x0D

Field	Name	Access	Description	Reset
[7]	reserved	RO	Reserved.	0x0
[6]	mxds2_r_wr2rd_stop	RO	If the maximum read turnaround time is not 0, then this field is used to tell the Controller whether the Target permits the Write-to-Read to be split by a STOP. Configuration options: • 0 – STOP would cancel the Read. • 1 – The Target permits the Write-to-Read to be split by a STOP (Not supported in this IP).	0x0 (Fixed)
[5:3]	mxds2_r_tsco	RW	Clock to Data Turnaround Time (tSCO). Configuration options: • 0 – ≤ 8 ns (default value) • 1 – ≤ 9 ns • 2 – ≤ 10 ns • 3 – ≤ 11 ns • 4 – ≤ 12 ns • 5 – 6 – Reserved for future use by the MIPI	0x0

Field	Name	Access	Description	Reset
			Alliance. • 7 – tSCO is > 12 ns, and is reported by private agreement	
[2:0]	mxds2_r_rate	RW	The maximum sustained data rate for non-CCC messages sent by the Target device to the Controller device. Configuration options: • 0 – f_{SCL} Max (default value) • 1 – 8 MHz • 2 – 6 MHz • 3 – 4 MHz • 4 – 2 MHz • Others – Reserved for future use by the MIPI Alliance.	0x0

5.15. Maximum Read Turnaround Time Byte2 Register 0x0E

Applicable only when the Target is configured with Maximum Data Speed Limitation.

Table 5.17. Maximum Read Turnaround Time Byte2 Register 0x0E

Field	Name	Access	Description	Reset
[7:0]	max_rdturn_b2	RW	The maximum read turnaround time in µs. The most significant byte of the 24-bit field that can encode turnaround times from 0.0 seconds to 16 seconds.	0x0

5.16. Maximum Read Turnaround Time Byte1 Register 0x0F

Applicable only when the Target is configured with Maximum Data Speed Limitation.

Table 5.18. Maximum Read Turnaround Time Byte1 Register 0x0F

Field	Name	Access	Description	Reset
[7:0]	max_rdturn_b1	RW	The maximum read turnaround time in µs. The middle byte of the 24-bit field that can encode turnaround times from 0.0 seconds to 16 seconds.	0x0

5.17. Maximum Read Turnaround Time Byte0 Register 0x10

Applicable only when the Target is configured with Maximum Data Speed Limitation.

Table 5.19. Maximum Read Turnaround Time Byte0 Register 0x10

Field	Name	Access	Description	Reset
[7:0]	max_rdturn_b0	RW	The maximum read turnaround time in µs. The least significant byte of the 24-bit field that can encode turnaround times from 0.0 seconds to 16 seconds.	0x0

5.18. Device Provisioned ID Byte6 Register 0x11

Byte 6 of the Target Device PID.

Table 5.20. Device Provisioned ID Byte6 Register 0x11

Field	Name	Access	Description	Reset
[7:0]	pid_manuf[14:7]	RW	Bits 14:7 of the 15-bit MIPI Manufacturer ID. Reset value may be configured through the *Manufacturer ID* attribute.	Takes the upper 8 bits of the *Manufacturer ID* parameter value.

5.19. Device Provisioned ID Byte5 Register 0x12

Byte 5 of the Target Device PID.

Table 5.21. Device Provisioned ID Byte5 Register 0x12

Field	Name	Access	Description	Reset
[7:1]	pid_manuf[6:0]	RW	Bits 6:0 of the 15-bit MIPI Manufacturer ID. Reset value may be configured through the *Manufacturer ID* attribute.	Takes the lower 7 bits of the *Manufacturer ID* parameter value.
[0]	pid_type_sel	RO	Provisioned ID type selector. Configuration options: • 0 – Vendor Fixed Value. • 1 – Random value generated by device (Not supported in this IP).	0x0 (Fixed)

5.20. Device Provisioned ID Byte4 Register 0x13

Byte 4 of the Target Device PID.

Table 5.22. Device Provisioned ID Byte4 Register 0x13

Field	Name	Access	Description	Reset
[7:0]	pid_part[15:8]	RW	Bits 15:8 of the 16-bit Part ID when pid_type_sel is 0. The meaning of this 16-bit field is left to the Device vendor to define. Reset value may be configured through the *Part ID* attribute.	Takes the upper 8 bits of the *Part ID* parameter value.

5.21. Device Provisioned ID Byte3 Register 0x14

Byte 3 of the Target Device PID.

Table 5.23. Device Provisioned ID Byte3 Register 0x14

Field	Name	Access	Description	Reset
[7:0]	pid_part[7:0]	RW	Bits 7:0 of the 16-bit Part ID when pid_type_sel is 0. The meaning of this 16-bit field is left to the Device vendor to define. Reset value may be configured through the *Part ID* attribute.	Takes the lower 8 bits of the *Part ID* parameter value

5.22. Device Provisioned ID Byte2 Register 0x15

Byte 2 of the Target Device PID.

Table 5.24. Device Provisioned ID Byte2 Register 0x15

Field	Name	Access	Description	Reset
[7:4]	pid_inst	RW	Instance ID field that identifies the individual device using a selected method selected (such as straps, fuses, non-volatile memory, or another appropriate method). Reset value may be configured through the *Instance ID* attribute.	Takes the *Instance ID* parameter value .
[3:0]	pid_add[11:8]	RW	Bits 11:8 of the 12-bit Additional ID for definitions with additional meaning (such as deeper device characteristics that may optionally include Device Characteristics Register values). Reset value may be configured through the *Additional ID* attribute.	Takes the upper 4 bits of the *Additional ID* parameter value.

5.23. Device Provisioned ID Byte1 Register 0x16

Byte 1 of the Target Device PID.

Table 5.25. Device Provisioned ID Byte1 Register 0x16

Field	Name	Access	Description	Reset
[7:0]	pid_add[7:0]	RW	Bits 7:0 of the 12-bit Additional ID for definitions with additional meaning (such as deeper device characteristics that may optionally include Device Characteristics Register values). Reset value may be configured through the *Additional ID* attribute.	Takes the lower 8 bits of the *Additional ID* parameter value.

5.24. Static Address Register 0x17

Applicable only when the *Static Address Enabled* attribute is checked.

Table 5.26. Static Address Register 0x17

Field	Name	Access	Description	Reset
[7]	reserved	RO	Reserved.	0x0
[6:0]	stat_addr	RW	Target Static Address that is assigned.	Takes the *Static Address* Attribute if static address is enabled; otherwise, 0x0.

5.25. Device Capabilities Byte1 Register 0x18

Device capabilities that the Target returns when the Controller sends GETCAPS CCC.

Table 5.27. Device Capabilities Byte1 Register 0x18

Field	Name	Access	Description	Reset
[7:4]	reserved	RO	Reserved.	0x0
[3]	hdr_bt_mode	RO	Set to 1 when the device is configured with HDR-BT capability. Not supported in this IP.	0x0 (Fixed)
[2]	hdr_tsl_mode	RO	Set to 1 when the device is configured with HDR-TSL capability. Not supported in this IP.	0x0 (Fixed)
[1]	hdr_tsp_mode	RO	Set to 1 when the device is configured with HDR-TSP capability. Not supported in this IP.	0x0 (Fixed)

Field	Name	Access	Description	Reset
[0]	hdr_ddr_mode	RO	Set to 1 when the device is configured with HDR-DDR capability. Only HDR-DDR mode is supported.	0x0 if the *Bus Type* parameter is SDR only. 0x1 if the *Bus Type* parameter is HDR-Capable and the *HDR Mode* parameter is HDR-DDR.

5.26. Device Capabilities Byte2 Register 0x19

Device capabilities that the Target returns when the Controller sends GETCAPS CCC.

Table 5.28. Device Capabilities Byte2 Register 0x19

Field	Name	Access	Description	Reset
[7]	hdr_ddr_abort_crc_caps	RO	I3C Target capability of emitting the CRC Word when a transaction in the HDR-DDR mode is aborted. Set to 1 when the device is configured with HDR capability. Configuration options: • 0: No • 1: Yes	0x0 – If only SDR capable. 0x1 – if HDR DDR capable.
[6]	hdr_ddr_wr_abort_caps	RO	I3C Target capability of issuing the Write Abort in the HDR-DDR mode. Set to 1 when the device is configured with HDR capability. Configuration options: • 0: No • 1: Yes	0x0 – If only SDR capable. 0x1 – if HDR DDR capable.
[5:4]	grp_adr_caps	RO	Indicates the Group Address function capabilities of this I3C Device. Fixed to 0. Configuration options: • 0 – Does not support Group Address function. • 1 – Can be assigned one Group Address (Not supported in this IP). • 2 – Can be assigned two Group Addresses (Not supported in this IP). • 3 – Can be assigned three or more Group Addresses (Not supported in this IP).	0x0 (Fixed)
[3:0]	i3c_spec_ver	RO	Indicates the minor version number of the MIPI I3C Specification with which this I3C v1.x Device complies (refers to the *x* in *I3C v1.x*). Setting to 0x0 is illegal.	0x1 (Fixed)

5.27. Device Capabilities Byte3 Register 0x1A

Device capabilities that the Target returns when the Controller sends GETCAPS CCC.

Table 5.29. Device Capabilities Byte3 Register 0x1A

Field	Name	Access	Description	Reset
[7]	Reserved	RO	Reserved.	0x0
[6]	pend_rd_mdb	RO	I3C Target support for IBI with Pending Read Notification MDB, which the Controller then follows with a Private Read request to fetch the data. Set to 1 when the device is configured with IBI with MDB capability.	0x1 – if IBI Capable and IBI Payload Size (including MDB) is greater than 1, 0 otherwise.

Field	Name	Access	Description	Reset
			Configuration options: • 0 – No • 1 – Yes	
[5]	hdr_bt_crc32	RO	I3C Target supports CRC-32 data integrity verification in HDR Bulk Transport mode. Configuration options: • 0 – No • 1 – Yes (Not supported in this IP)	0x0 (Fixed)
[4]	getstatus_defbyte	RO	I3C Target support for defining byte in GETSTATUS CCC. Configuration options: • 0 – No • 1 – Yes (Not supported in this IP)	0x0 (Fixed)
[3]	getcaps_defbyte	RO	I3C Target support for defining byte in GETCAPS CCC. Configuration options: • 0 – No • 1 – Yes (Not supported in this IP)	0x0 (Fixed)
[2]	d2dxfer_ibi	RO	I3C Target capability to initiate Device to Device (D2D) Transfer using IBI with MDB 0x37. Configuration options: • 0 – No • 1 – Yes (Not supported in this IP)	0x0 (Fixed)
[1]	d2dxfer	RO	I3C Target support for D2D transfers either as a Source or a Subscriber/Receiver. Configuration options: • 0 – No • 1 – Yes (Not supported in this IP)	0x0 (Fixed)
[0]	ml_data_xfer	RO	I3C Target support for multi-Lane data transfer. Configuration options: • 0 – No • 1 – Yes (Not supported in this IP)	0x0 (Fixed)

5.28. Oscillator Inaccuracy Register 0x1C

Inaccuracy of the Target's internal oscillator.

Table 5.30. Oscillator Inaccuracy Register 0x1C

Field	Name	Access	Description	Reset
[7:0]	osc_inaccuracy	RW	Describes the maximum variation of the Target's internal oscillator in 1/10th percent (0.1%) increments, up to 25.5%. **Example:** A value of 8'd25 represents a maximum frequency variation of 2.5%.	0x0

5.29. Receive FIFO Register 0x20

Receive FIFO.

Table 5.31. Receive FIFO Register 0x20

Field	Name	Access	Description	Reset
[7:0]	rx_fifo	RO	Data bytes received from the I3C bus are stored in this FIFO. Get the received data by reading from this address multiple times, depending on the number of data bytes to be read. Read rxfifo_not_empty interrupt status register (Interrupt Status 2 Bit 6) first to determine if there is data available to be read from rx_fifo.	0x0

5.30. Transmit FIFO Register 0x22

Transmit FIFO.

Table 5.32. Transmit FIFO Register 0x22

Field	Name	Access	Description	Reset
[7:0]	tx_fifo	RW	Data to be sent to the I3C bus is stored in this FIFO. You can do multiple writes to this address, depending on the number of data bytes to be written. Reading from this address indicates the *txfifo_empty* status (0x1 – Empty, 0x0 – Not Empty).	0x0

5.31. Soft Reset Register 0x28

Soft reset.

Table 5.33. Soft Reset Register 0x28

Field	Name	Access	Description	Reset
[7:5]	Reserved	RO	Reserved.	0x0
[4]	ip_csr_rst	RW	When set to high, this field resets the IP RW and WO registers only. Automatically cleared when the reset is successfully propagated.	0x0
[3]	ip_core_rst	RW	When set to high, this field resets only the internal state of the I3C core (including the Tx and Rx FIFO). Dynamic address and configurations set through CCCs are not reset. Automatically cleared when the reset is successfully propagated.	0x0
[2]	tx_fifo_rst	RW	When set to high, this field resets only the Tx FIFO. Automatically cleared when the reset is successfully propagated.	0x0
[1]	rx_fifo_rst	RW	When set to high, this field resets only the Rx FIFO. Automatically cleared when the reset is successfully propagated.	0x0
[0]	ip_main_rst	RW	When set to high, this field resets the entire IP, including registers, FIFO, dynamic address, and CCC configurations. Automatically cleared when the reset is successfully propagated.	0x0

5.32. Target Response Register 0x29

Target response.

Table 5.34. Target Response Register 0x29

Field	Name	Access	Description	Reset
[7:5]	Reserved	RO	Reserved.	0x0
[4]	fifo_loopback_en	RW	Configuration options: • 0 – Data written through Private Write is read from the Rx FIFO. Data written to the Tx FIFO is read through Private Read. • 1 – Data written through Private Write is stored in the Rx FIFO and transferred to the Tx FIFO to be read through Private Read. Data written to the Tx FIFO can still be read through Private Read.	0x0
[3:1]	Reserved	RO	Reserved.	0x0
[0]	txfifo_empty_rd_nak	RW	Target response when the Controller reads but the Tx FIFO is empty. Configuration options: • 0 – In SDR mode, the Target ACKs its address and returns 0xFF read data then End-of-Message by pulling SDA low at T-bit. If I3C bus is in the HDR-DDR mode and the Target is configured with HDR capability, the Target ACKs its address and returns 0x0 data then ends the Read. • 1 – The Target NACKs its address.	0x0

5.33. Get Status MSB Register 0x2A

The most significant byte that the Target returns when the Controller sends GETSTATUS CCC.

Table 5.35. Get Status MSB Register 0x2A

Field	Name	Access	Description	Reset
[7:0]	get_status_msb	RW	Reserved for vendor-specific meaning.	0x0

5.34. Get Status LSB Register 0x2B

The least significant byte that the Target returns when the Controller sends GETSTATUS CCC.

Table 5.36. Get Status LSB Register 0x2B

Field	Name	Access	Description	Reset
[7:6]	activity_mode	RW	Contains the 2-bit ID of the Target Device's current activity mode. For Target devices without Secondary Controller Capability, the meaning of this value depends on a private contract between the Target and the Controller.	0x0
[5:4]	Reserved	RO	Reserved.	0x0
[3:0]	pending_interrupt	RW	Contains the interrupt number of any pending interrupt, or 0 if no interrupts are pending.	0x0

5.35. Bus Activity State Register 0x2C

Bus activity state.

Table 5.37. Bus Activity State Register 0x2C

Field	Name	Access	Description	Reset
[7:2]	reserved	RO	Reserved.	0x0
[1:0]	bus_act	RO	When bus_act_rcvd is set to 1, this register indicates the received activity state. Status values: • 0 – 1 µs (Latency-free operation) • 1 – 100 µs • 2 – 2 ms • 3 – 50 ms (Lowest-activity operation)	0x0

5.36. Target Reset Action 1 Register 0x2D

Target reset action information 1.

Table 5.38. Target Reset Action 1 Register 0x2D

Field	Name	Access	Description	Reset
[7:0]	rst_act	RO	Target Reset Action is configured by the Controller through the Defining Byte of RSTACT CCC. When Target Reset Pattern is received after RSTACT CCC, the Target performs the configured reset action.	0x0

5.37. Target Reset Action 2 Register 0x2E

Target reset action information 2.

Table 5.39. Target Reset Action 2 Register 0x2E

Field	Name	Access	Description	Reset
[7:2]	reserved	RO	Reserved	0x0
[1]	rst_act_set0_get1	RO	Information on the Target Reset Action configured by the Controller. Configuration options: • 0 – Controller configured Target Reset Action by Direct SET RSTACT CCC. • 1 – Controller configured Target Reset Action by Direct GET RSTACT CCC.	0x0
[0]	rst_act_broad0_dir1	RO	Information on the Target Reset Action configured by the Controller. Configuration options: • 0 – Controller configured Target Reset Action by Broadcast RSTACT CCC. • 1 – Controller configured Target Reset Action by Direct Set/Get RSTACT CCC.	0x0

5.38. Target Reset Action 3 Register 0x2F

Target reset action information 3.

Table 5.40. Target Reset Action 3 Register 0x2F

Field	Name	Access	Description	Reset
[7:0]	rst_act_set	RO	Target Reset Action configured by the Controller through the Defining Byte of Direct SET RSTACT CCC. This is the value returned by the Target when the Controller sends Direct GET CCC.	0x80

5.39. Interrupt Status 1 Register 0x30

Interrupt status information 1.

Table 5.41. Interrupt Status 1 Register 0x30

Field	Name	Access	Description	Reset
[7]	hj_req_gen	RW1C	When set to 1, a Hot-Join request is generated by the Target.	0x0
[6]	hj_done	RW1C	When set to 1, the Hot-Join address with Write bit is transmitted by the Target and the Controller either ACKs or NACKs it for HJ_IBI_RETRY times.	0x0
[5]	hj_acknack	RW1C	When set to 1, the Hot-Join is NACKed by the Controller after HJ_IBI_RETRY times.	0x0
[4]	reserved	RO	Reserved.	0x0
[3]	ibi_req_gen	RW1C	When set to 1, an IBI request is generated by the Target.	0x0
[2]	ibi_done	RW1C	When set to 1, the Target Address with Read bit is transmitted by the Target and the Controller either ACKs or NACKs the request for HJ_IBI_RETRY times. If payload is present, the IBI payload is transmitted and finished either by End-of-Data (from the Target) or STOP (from the Controller).	0x0
[1]	ibi_acknack	RW1C	When set to 1, the IBI is NACKed by the Controller after HJ_IBI_RETRY times.	0x0
[0]	ibi_payld_terminated	RW1C	When set to 1, the Controller aborts reading the complete IBI payload.	0x0

5.40. Interrupt Enable 1 Register 0x31

Interrupt enable.

Table 5.42. Interrupt Enable 1 Register 0x31

Field	Name	Access	Description	Reset
[7]	hj_req_gen_en	RW	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]	hj_done_en	RW		0x0
[5]	hj_acknack_en	RW		0x0
[4]	reserved	RW		0x0
[3]	ibi_req_gen_en	RW		0x0
[2]	ibi_done_en	RW		0x0
[1]	ibi_acknack_en	RW		0x0
[0]	ibi_payld_terminated_en	RW		0x0

5.41. Interrupt Set 1 Register 0x32

Interrupt set.

Table 5.43. Interrupt Set 1 Register 0x32

Field	Name	Access	Description	Reset
[7]	hj_req_gen_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
[6]	hj_done_set	WO		0x0
[5]	hj_acknack_set	WO		0x0
[4]	reserved	RO		0x0
[3]	ibi_req_gen_set	WO		0x0
[2]	ibi_done_set	WO		0x0
[1]	ibi_acknack_set	WO		0x0
[0]	ibi_payld_terminated_set	WO		0x0

5.42. Interrupt Status 2 Register 0x33

Interrupt status 2.

Table 5.44. Interrupt Status 2 Register 0x33

Field	Name	Access	Description	Reset
[7]	txfifo_full	RW1C	When set to 1, this field indicates that the Tx FIFO is full.	0x0
[6]	rxfifo_not_empty	RW1C	When set to 1, this field indicates that the Rx FIFO is not empty.	0x0
[5]	rxfifo_full	RW1C	When set to 1, this field indicates that the Rx FIFO is full.	0x0
[4]	reserved	RO	Reserved.	0x0
[3]	read_txfifo_empty	RW1C	When set to 1, the Controller initiates a Read operation but the Tx FIFO is empty.	0x0
[2]	read_aborted	RW1C	When set to 1, the Read operation is aborted early by the Controller.	0x0
[1]	da_par_err	RW1C	When set to 1, a parity bit error occurs during dynamic address assignment.	0x0
[0]	tbit_err	RW1C	When set to 1, a data T-bit Error occurs.	0x0

5.43. Interrupt Enable 2 Register 0x34

Interrupt enable.

Table 5.45. Interrupt Enable 2 Register 0x34

Field	Name	Access	Description	Reset
[7]	txfifo_full_en	RW	When set to high, this field enables the corresponding interrupt status 2 signal to cause the assertion of the interrupt port signal (int_o).	0x0
[6]	rxfifo_not_empty_en	RW		0x0
[5]	rxfifo_full_en	RW		0x0
[4]	reserved	RO		0x0
[3]	read_txfifo_empty_en	RW		0x0
[2]	read_aborted_en	RW		0x0
[1]	da_par_err_en	RW		0x0
[0]	tbit_err_en	RW		0x0

5.44. Interrupt Set 2 Register 0x35

Interrupt set.

Table 5.46. Interrupt Set 2 Register 0x35

Field	Name	Access	Description	Reset
[7]	txfifo_full_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]	rxfifo_not_empty_set	WO		0x0
[5]	rxfifo_full_set	WO		0x0
[4]	reserved	RO		0x0
[3]	read_txfifo_empty_set	WO		0x0
[2]	read_aborted_set	WO		0x0
[1]	da_par_err_set	WO		0x0
[0]	tbit_err_set	WO		0x0

5.45. Interrupt Status 3 Register 0x36

Interrupt status 3.

Table 5.47. Interrupt Status 3 Register 0x36

Field	Name	Access	Description	Reset
[7]	enec_rcvd	RW1C	When set to 1, the Target receives Broadcast/Direct ENEC/DISEC CCC from the Controller.	0x0
[6]	tgt_rst_ptrn_rcvd	RW1C	When set to 1, the Target receives Target Reset Pattern from the Controller.	0x0
[5]	rstact_ccc_rcvd	RW1C	When set to 1, the Target receives RSTACT CCC from the Controller.	0x0
[4]	bus_act_rcvd	RW1C	When set to 1, the Target receives ENTASx CCC from the Controller.	0x0
[3]	setxtime_rcvd	RW1C	When set to 1, the Target receives SETXTIME with supported sub-command byte from the Controller.	0x0
[2]	reserved	RO	Reserved.	0x0
[1]	bus_aval	RW1C	When set to 1, the I3C bus is in the Bus Available condition.	0x0
[0]	bus_idle	RW1C	When set to 1, the I3C bus is in the Bus Idle condition.	0x0

5.46. Interrupt Enable 3 Register 0x37

Interrupt enable.

Table 5.48. Interrupt Enable 3 Register 0x37

Field	Name	Access	Description	Reset
[7]	enec_rcvd_en	RW	When set to high, this field enables the corresponding interrupt status 3 signal to cause the assertion of the interrupt port signal (int_o).	0x0
[6]	tgt_rst_ptrn_rcvd_en	RW		0x0
[5]	rstact_ccc_rcvd_en	RW		0x0
[4]	bus_act_rcvd_en	RW		0x0
[3]	setxtime_rcvd_en	RW		0x0
[2]	reserved	RO		0x0
[1]	bus_aval_en	RW		0x0
[0]	bus_idle_en	RW		0x0

5.47. Interrupt Set 3 Register 0x38

Interrupt set.

Table 5.49. Interrupt Set 3 Register 0x38

Field	Name	Access	Description	Reset
[7]	enec_rcvd_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 3 signal.	0x0
[6]	tgt_rst_ptrn_rcvd_set	WO		0x0
[5]	rstact_ccc_rcvd_set	WO		0x0
[4]	bus_act_rcvd_set	WO		0x0
[3]	setxtime_rcvd	WO		0x0
[2]	reserved	RO		0x0
[1]	bus_aval_set	WO		0x0
[0]	bus_idle_set	WO		0x0

5.48. Interrupt Status 5 Register 0x3C

Interrupt status 5.

Table 5.50. Interrupt Status 5 Register 0x3C

Field	Name	Access	Description	Reset
[7:4]	reserved	RO	Reserved.	0x0
[3]	hdr_ddr_frm_err	RW1C	When set to 1, an HDR-DDR framing error is detected.	0x0
[2]	hdr_ddr_par_err	RW1C	When set to 1, an HDR-DDR parity error is detected.	0x0
[1]	hdr_ddr_crc_err	RW1C	When set to 1, an HDR-DDR CRC error is detected.	0x0
[0]	hdr_ddr_mon_err	RW1C	When set to 1, the data transferred on the I3C bus differs from what the Target intends to send.	0x0

5.49. Interrupt Enable 5 Register 0x3D

Interrupt enable.

Table 5.51. Interrupt Enable 5 Register 0x3D

Field	Name	Access	Description	Reset
[7:4]	reserved	RO	Reserved.	0x0
[3]	hdr_ddr_frm_err_en	RW	When set to high, this field enables the corresponding interrupt status 5 signal to cause the assertion of the interrupt port signal (int_o).	0x0
[2]	hdr_ddr_par_err_en	RW		0x0
[1]	hdr_ddr_crc_err_en	RW		0x0
[0]	hdr_ddr_mon_err_en	RW		0x0

5.50. Interrupt Set 5 Register 0x3E

Interrupt set.

Table 5.52. Interrupt Set 5 Register 0x3E

Field	Name	Access	Description	Reset
[7:4]	reserved	RO	Reserved.	0x0
[3]	hdr_ddr_frm_err_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 5 signal.	0x0
[2]	hdr_ddr_par_err_set	WO		0x0
[1]	hdr_ddr_crc_err_set	WO		0x0

Field	Name	Access	Description	Reset
[0]	hdr_ddr_mon_err_set	WO		0x0

5.51. Bus Information Register 0x50

I3C bus mode.

Table 5.53. Bus Information Register 0x50

Field	Name	Access	Description	Reset
[7:2]	reserved	RO	Reserved.	0x0
[1]	hj_wait_start	RO	When the signal is high, a Hot-Join or IBI request is programmed through the corresponding Events Command Request register, but the IP is still waiting for valid condition to send the Hot-Join. Refer to the Hot-Join Mechanism section for more details on the operation sequence.	0x0
[0]	bus_hdr_mode	RO	When the signal is high, the I3C bus is in HDR-DDR mode. Otherwise, the bus is in SDR mode.	0x0

5.52. HDR-DDR Target Configuration Register 0x51

HDR-DDR target configuration for received HDR data.

Table 5.54. HDR-DDR Target Configuration Register 0x51

Field	Name	Access	Description	Reset
[7:2]	reserved	RO	Reserved.	0x0
[1]	hdr_ddr_wr_cmd_to_fifo	RW	Applicable only when the Target is configured with HDR capability. When set to high, both HDR-DDR Write Command and data are stored in the Rx FIFO. When set to low, the command is decoded to determine read or write and only the write data is stored in the Rx FIFO.	0x0
[0]	hdr_ddr_rd_cmd_to_fifo	RW	Applicable only when the Target is configured with HDR capability. When set to high, the HDR-DDR Read Command is stored in the Rx FIFO. When set to low, the command is decoded to determine read or write and no data is stored in the Rx FIFO.	0x0

5.53. HDR-DDR Abort Configuration Register 0x54

HDR-DDR abort configuration set by the Controller through ENDXFER CCC.

Table 5.55. HDR-DDR Abort Configuration Register 0x54

Field	Name	Access	Description	Reset
[7:6]	hdr_ddr_abort_crc	RO	Applicable only when the Target is configured with HDR capability. Configuration options: • 2'b11 – No CRC Word follows the Early Termination request. • 2'b01 – A CRC Word follows the Early Termination request. • Other – Reserved for future definition by the MIPI Alliance.	0x1
[5]	hdr_ddr_write_abort	RO	Applicable only when the Target is configured with HDR capability. When set to low, the Target can issue a Write Abort in HDR-DDR	0x0

Field	Name	Access	Description	Reset
			Mode. The IP issues a Write Abort in HDR-DDR Mode if the Rx FIFO is full. When set to high, the Target is not capable of issuing a Write Abort in HDR-DDR mode.	
[4]	hdr_ddr_write_nack	RO	Applicable only when the Target is configured with HDR capability. When set to low, the Target can issue a NACK to an HDR-DDR Write Command. The IP issues a NACK to an HDR-DDR Write Command if the Rx FIFO is full. When set to high, the Target is not capable of issuing a NACK to an HDR-DDR Write Command.	0x0
[3:0]	reserved	RO	Reserved.	0x0

5.54. Secondary Controller Registers

I3C Target registers are accessible when the I3C Controller IP is configured with Secondary Controller capability. In this configuration, the address space of the I3C Target is mapped into the address space of the I3C Controller IP, allowing access to the I3C Target registers through the Controller interface.

The mapping behavior depends on the device configuration:

- Target-only configuration (using I3C Target IP):
  The most significant bit of the register address is set to 0.
  Example:
  - 0x00 → Target-only
  - 0x01 → Target-only
- Secondary Controller configuration (using I3C Controller IP):
  The most significant bit of the register address is set to 1, indicating that the Target register space is now accessible through the Controller IP.
  Example:
  - 0x00 (Target-only) → 0x80 (Secondary Controller)
  - 0x01 (Target-only) → 0x81 (Secondary Controller)

Refer to Table 5.56 for a detailed mapping of I3C Target registers to their corresponding addresses in the I3C Controller IP. Registers marked with an asterisk (*) are I3C Target registers whose reset values and/or definitions change when used in Secondary Controller mode. These changes are discussed in the following section. For more details on this usage, refer to the I3C Controller IP User Guide (FPGA-IPUG-02228). Registers without an asterisk retain their definitions as when the device is configured as an I3C Target-only.

Table 5.56. Secondary I3C Controller Register Summary

Byte Address Offset[1]	DWORD Address Offset[1]	Register Name	Description
0x80	0x200	Bus Characteristics*	Target Bus Characteristics register.
0x81	0x204	Device Characteristics	Target Device Characteristics register.
0x82	0x208	Dynamic Address*	Assigned dynamic address and done flag.
0x83	0x20C	Events Command Enable*	Hot-Join and IBI enable from the Controller through ENEC/DISEC CCC.
0x84	0x210	Events Command Device Configuration*	Hot-Join and IBI capability configured through attributes.
0x85	0x214	Events Command Request*	Hot-Join and IBI request.
0x86	0x218	Hot-Join/IBI Retry	Number of retries for Hot-Join and IBI.
0x87	0x21C	Maximum Write Length (MSB)	The most significant byte of the Maximum Write Length set by the Controller.
0x88	0x220	Maximum Write Length (LSB)	The least significant byte of the Maximum Write Length set by

Byte Address Offset[1]	DWORD Address Offset[1]	Register Name	Description
			the Controller.
0x89	0x224	Maximum Read Length (MSB)	The most significant byte of the Maximum Read Length set by the Controller.
0x8A	0x228	Maximum Read Length (LSB)	The least significant byte of the Maximum Read Length set by the Controller.
0x8B	0x22C	Maximum IBI Payload	The maximum IBI payload size set by the Controller.
0x8C	0x230	Maximum Write Data Speed (MaxWr)*	Applicable only when the Target is configured with Maximum Data Speed Limitation.
0x8D	0x234	Maximum Read Data Speed (MaxRd)	Applicable only when the Target is configured with Maximum Data Speed Limitation.
0x8E	0x238	Maximum Read Turnaround Time Byte2	Applicable only when the Target is configured with Maximum Data Speed Limitation.
0x8F	0x23C	Maximum Read Turnaround Time Byte1	Applicable only when the Target is configured with Maximum Data Speed Limitation.
0x90	0x240	Maximum Read Turnaround Time Byte0	Applicable only when the Target is configured with Maximum Data Speed Limitation.
0x91	0x244	Device Provisioned ID Byte6	6th byte of the Target Device PID.
0x92	0x248	Device Provisioned ID Byte5	5th byte of the Target Device PID.
0x93	0x24C	Device Provisioned ID Byte4	4th byte of the Target Device PID.
0x94	0x250	Device Provisioned ID Byte3	3rd byte of the Target Device PID.
0x95	0x254	Device Provisioned ID Byte2	2nd byte of the Target Device PID.
0x96	0x258	Device Provisioned ID Byte1	1st byte of the Target Device PID.
0x97	0x25C	Static Address	The Target static address that is assigned.
0x98	0x260	Device Capabilities Byte1	Device capabilities that the Target returns when the Controller sends GETCAPS CCC.
0x99	0x264	Device Capabilities Byte2	Device capabilities that the Target returns when the Controller sends GETCAPS CCC.
0x9A	0x268	Device Capabilities Byte3*	Device capabilities that the Target returns when the Controller sends GETCAPS CCC.
0x9C	0x270	Oscillator Inaccuracy	Inaccuracy of the Target's internal oscillator.
0xA0	0x280	Receive FIFO	Receive FIFO.
0xA2	0x288	Transmit FIFO	Transmit FIFO.
0xA8	0x2A0	Soft Reset	Soft reset.
0xA9	0x2A4	Target Response	Target response.
0xAA	0x2A8	Get Status MSB	The most significant byte that the Target returns when the Controller sends GETSTATUS CCC.
0xAB	0x2AC	Get Status LSB	The least significant byte that the Target returns when the Controller sends GETSTATUS CCC.
0xAC	0x2B0	Bus Activity State	The least significant byte that the Target returns when the Controller sends GETSTATUS CCC.
0xAD	0x2B4	Target Reset Action 1	Target reset action information 1.
0xAE	0x2B8	Target Reset Action 2	Target reset action information 2.
0xAF	0x2BC	Target Reset Action 3	Target reset action information 3.
0xB0	0x2C0	Interrupt Status 1	Interrupt status 1.
0xB1	0x2C4	Interrupt Enable 1	Interrupt enable 1.
0xB2	0x2C8	Interrupt Set 1	Interrupt set 1.
0xB3	0x2CC	Interrupt Status 2	Interrupt status 2.
0xB4	0x2D0	Interrupt Enable 2	Interrupt enable 2.
0xB5	0x2D4	Interrupt Set 2	Interrupt set 2.

Byte Address Offset[1]	DWORD Address Offset[1]	Register Name	Description
0xB6	0x2D8	Interrupt Status 3	Interrupt status 3.
0xB7	0x2DC	Interrupt Enable 3	Interrupt enable 3.
0xB8	0x2E0	Interrupt Set 3	Interrupt set 3.
0xB9	0x2E4	Interrupt Status 4*	Interrupt status 4.
0xBA	0x2E8	Interrupt Enable 4*	Interrupt enable 4.
0xBB	0x2EC	Interrupt Set 4*	Interrupt set 4.
0xBC	0x2F0	Interrupt Status 5	Interrupt status 5.
0xBD	0x2F4	Interrupt Enable 5	Interrupt enable 5.
0xBE	0x2F8	Interrupt Set 5	Interrupt set 5.
0xC0	0x300	DEFTGTS Count*	Count information sent by the Active Controller through DEFTGTS CCC.
0xC1	0x304	DEFTGTS RxFIFO Start*	Pointer for the start address of DEFTGTS data in the Rx FIFO.
0xC2	0x308	DEFTGTS RxFIFO Count*	Number of DEFTGTS data bytes stored in the Rx FIFO.
0xC3	0x30C	Controller Role Handoff*	Target configuration for Controller Role Handoff.
0xC4	0x310	GETMXDS Controller Capable Device*	Return value for GETMXDS when the defining byte is 0x91.
0xC5	0x314	GETSTATUS Controller Capable Device*	Return value for GETSTATUS when the defining byte is 0x91.
0xC6	0x318	GETCAPS Controller Capable Device 1*	Return value for GETCAPS when the defining byte is 0x91.
0xC7	0x31C	GETCAPS Controller Capable Device 2*	Return value for GETCAPS when the defining byte is 0x91.
0xC8	0x320	Set Device Role*	Manually set the device role.
0xD0	0x340	Bus Mode	I3C bus mode.
0xD1	0x344	HDR-DDR Configuration	HDR-DDR target configuration for received HDR data.
0xD4	0x350	HDR-DDR Abort Configuration	HDR-DDR abort configuration set by the Controller through ENDXFER CCC.
0xD5 – 0xFF	0x354 – 0x3FC	Reserved	Reserved.

Note:

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

5.54.1. Bus Characteristics Register 0x80

Target Bus Characteristics register.

Table 5.57. Bus Characteristics Register 0x80

Field	Name	Access	Description	Reset
[7:6]	device_role	RO	Configuration options: • 2'b00 – I3C Target. • 2'b01 – I3C Controller Capable. • Others - Reserved for future definition by the MIPI Alliance I3C WG (Not supported in this IP).	0x0 – Device is Target only. 0x1 – Device is Controller Capable.
[5]	advanced_caps	RO	Fixed to 1 when the device is configured with Secondary Controller Capability. Configuration options: • 0 – Does not support optional advanced capabilities. • 1 – Supports optional advanced capabilities. When the Target is configured with IBI capability and IBI payload, this is set to 1 for	0x1 (Fixed)

Field	Name	Access	Description	Reset
			optional Pending Read MDB support.	
[4]	virtual_tgt_support	RO	Fixed to 0. Configuration options: • 0 – Is not a Virtual Target and does not expose other downstream Device(s). • 1 – Is a Virtual Target, or exposes other downstream Device(s) (Not supported in this IP).	0x0 (Fixed)
[3]	offline_capable	RO	Fixed to 0. Configuration options: • 0 – Device always respond to I3C bus commands. • 1 – Device does not always respond to I3C bus Commands (Not supported in this IP).	0x0 (Fixed)
[2]	ibi_payload	RO	From the *IBI Payload Size* attribute. Configuration options: • 0 – If the *IBI Payload Size* attribute is set to 0. No data bytes follow the IBI. • 1 – If the *IBI Payload Size* attribute is set to greater than or equal to 1. One data byte (MDB) follows the accepted IBI, and additional data bytes may follow.	0x1
[1]	ibi_capable	RO	From the *IBI Capable* attribute. Configuration options: • 0 – Unchecked. Not capable. • 1 – Checked. Capable.	0x1
[0]	max_d_speed_limit	RO	From the *Maximum Data Speed Limitation* attribute. Configuration options: • 0 – Unchecked. No limitation. • 1 – Checked. With limitation. The Controller uses GETMXDS CCC to get specific limitation from the Target.	0x0

5.54.2. Dynamic Address Register 0x82

Assigned dynamic address and done flag.

Table 5.58. Dynamic Address Register 0x82

Field	Name	Access	Description	Reset
[7]	daa_done	RO	Configuration options: • 0 – dyn_addr is not yet assigned or is reset by RSTDAA CCC. • 1 – dyn_addr is assigned through ENTDAA, SETNEWDA, SETAASA, or SETDASA CCC. This field is set to 1 if the Target is configured with Secondary Controller Capability.	0x1 – If Target is Secondary-Controller Capable 0x0 – If Target is not Secondary-Controller Capable
[6:0]	dyn_addr	RO	Dynamic address assigned to the I3C Target. If the Target is configured with Secondary Controller Capability and the IP is set to be the Primary Controller, this register indicates the Dynamic Address that the Controller uses when sending DEFTGTS CCC.	Takes Dynamic Address parameter value

5.54.3. Events Command Enable Register 0x83

Hot-Join and IBI enable from the Controller through ENEC/DISEC CCC.

Table 5.59. Events Command Enable Register 0x83

Field	Name	Access	Description	Reset
[7:4]	reserved	RO	Reserved.	0x0
[3]	hj_enec	RO	Hot-Join enablement by the Controller. Configuration options: • 0 – Target-initiated Hot-Join is not allowed on the I3C bus. • 1 – Target-initiated Hot-Join is allowed on the I3C bus.	0x1 – If Target is Hot-Join capable. 0x0 – If Target is not Hot-Join capable.
[2]	reserved	RO	Reserved.	0x0
[1]	cr_enec	RO	Controller role request enablement by the Controller. Configuration options: • 0 – Target-initiated Controller Role request is not allowed on the I3C bus. • 1 – Target-initiated Controller Role request is allowed on the I3C bus.	0x1
[0]	ibi_enec	RO	IBI enablement by the Controller. Configuration options: • 0 – Target-initiated interrupts are not allowed on the I3C bus. • 1 – Target-initiated interrupts are allowed on the I3C bus.	0x1 – If Target is IBI capable. 0x0 – If Target is not IBI capable.

5.54.4. Events Command Device Configuration Register 0x84

Hot-Join and IBI capability configured using attributes.

Table 5.60. Events Command Device Configuration Register 0x84

Field	Name	Access	Description	Reset
[7:4]	reserved	RO	Reserved.	0x0
[3]	hj_cap	RO	Configuration options: • 0 – Target device is configured without Hot-Join Capability. • 1 – Target device is configured with Hot-Join Capability.	0x1 – If Target is Hot-Join capable. 0x0 – If Target is not Hot-Join capable.
[2]	reserved	RO	Reserved.	0x0
[1]	cr_cap	RO	Configuration options: • 0 – Target device is configured without Secondary Controller Capability. • 1 – Target device is configured with Secondary Controller Capability.	0x1 – If Target is Secondary Controller Capable 0x0 – If Target is not Secondary Controller Capable
[0]	ibi_cap	RO	Configuration options: • 0 – Target device is configured without IBI Capability • 1 – Target device is configured with IBI Capability	0x1 – If Target is IBI capable. 0x0 – If Target is not IBI capable.

5.54.5. Events Command Request Register 0x85

Hot-Join and IBI capability configured using attributes.

Table 5.61. Events Command Request Register 0x85

Field	Name	Access	Description	Reset
[7:4]	reserved	RO	Reserved.	0x0
[3]	hj_req	RW	When set to high, the Target initiates a Hot-Join request at the next valid opportunity. Resets to 0 when the Hot-Join request is done or disabled by the Controller.	0x0
[2]	reserved	RO	Reserved.	0x0
[1]	cr_req	RW	When set to high, the Target initiates a Controller Role request at the next valid opportunity. Resets to 0 when the Controller Role request is done or disabled by the Controller.	0x0
[0]	ibi_req	RW	When set to high, the Target initiates an IBI request at the next valid opportunity. Reset to 0 when the IBI request is done or disabled by the Controller.	0x0

5.54.6. Maximum Write Data Speed (MaxWr) Register 0x8C

Applicable only when the Target is configured with Maximum Data Speed Limitation.

Table 5.62. Maximum Write Data Speed (MaxWr) Register 0x8C

Field	Name	Access	Description	Reset
[7:4]	reserved	RO	Reserved.	0x0
[3]	mxds2_w_defbyte	RO	Fixed to 1 when the device is configured with Secondary Controller Capability. Configuration options: • 0 – Target does not support defining byte for GETMXDS CCC. • 1 – Target supports defining byte for GETMXDS CCC.	0x1 (Fixed)
[2:0]	mxds2_w_rate	RW	Maximum Sustained Data Rate for non-CCC Messages sent by the Controller device to the Target device. Configuration options: • 0 – f_{SCL} Max (default value) • 1 – 8 MHz • 2 – 6 MHz • 3 – 4 MHz • 4 – 2 MHz • Others – Reserved for future use by the MIPI Alliance.	0x0

5.54.7. Device Capabilities Byte 3 Register 0x9A

Device capabilities that Target returns when Controller sends GETCAPS CCC.

Table 5.63. Device Capabilities Byte 3 Register 0x9A

Field	Name	Access	Description	Reset
[7]	reserved	RO	Reserved.	0x0
[6]	pend_rd_mdb	RO	I3C Target support for IBI with Pending Read Notification MDB, which the Controller then follows with a Private Read request to fetch the data. Set to 1 when the device is configured with IBI with MDB capability. Configuration options: • 0 – No • 1 – Yes	0x1 if Target is IBI capable and IBI payload size is greater than 0; otherwise, 0x0.

Field	Name	Access	Description	Reset
[5]	hdr_bt_crc32	RO	I3C Target support for CRC-32 data integrity verification in HDR Bulk Transport mode. Configuration options: • 0 – No • 1 – Yes (Not supported in this IP).	0x0 (Fixed)
[4]	getstatus_defbyte	RO	I3C Target support for defining byte in GETSTATUS CCC. Set to 1 when Secondary Controller Capable in enabled. Configuration options: • 0 – No • 1 – Yes	0x1 (Fixed)
[3]	getcaps_defbyte	RO	I3C Target support for defining byte in GETCAPS CCC. Set to 1 when Secondary Controller Capable is enabled. Configuration options: • 0 – No • 1 – Yes	0x1 (Fixed)
[2]	d2dxfer_ibi	RO	I3C Target capability to initiate a Device to Device (D2D) transfer using IBI with MDB 0x37. Configuration options: • 0 – No • 1 – Yes (Not supported in this IP).	0x1 (Fixed)
[1]	d2dxfer	RO	I3C Target support for D2D transfers, either as a Source or a Subscriber/Receiver. Configuration options: • 0 – No • 1 – Yes (Not supported in this IP).	0x0 (Fixed)
[0]	ml_data_xfer	RO	I3C Target support for Multi-Lane data transfer. Configuration options: • 0 – No • 1 – Yes (Not supported in this IP).	0x0 (Fixed)

5.54.8. Interrupt Status 4 Register 0xB9

Interrupt status 4.

Table 5.64. Interrupt Status 4 Register 0xB9

Field	Name	Access	Description	Reset
[7]	get_acccr_rcvd	RW1C	When set to 1, the Target receives GET_ACCCR CCC from the Controller.	0x0
[6]	get_acccr_acknak	RW1C	When set to 1, the Target NACKs GET_ACCCR CCC received from the Controller.	0x0
[5]	deftgts_rcvd	RW1C	When set to 1, the Target receives DEFTGTS CCC from the Controller (including data bytes describing each device in the bus).	0x0
[4]	reserved	RO	Reserved.	0x0
[3]	cr_req_gen	RW1C	When set to 1, the Controller Role request is generated by the Target.	0x0
[2]	cr_req_done	RW1C	When set to 1, the Target Address with Write bit is transmitted by the Target and the Controller either ACKs or NACKs the request for HJ_IBI_RETRY times.	0x0
[1]	cr_req_acknack	RW1C	When set to 1, the Controller Role request is NACKed by the Controller after HJ_IBI_RETRY times.	0x0
[0]	device_role_changed	RW1C	When set to 1, the device role is changed either from Controller to Target or Target to Controller. Read the actual role from the device_role register.	0x0

When the device_role_changed interrupt is asserted and device_role is 1'b0, the IP successfully sends the GETACCCR CCC and the controller role is transferred to the new controller. The IP then monitors whether the new controller asserts its role. If the new controller does not assert its role, the Controller Role Handoff is cancelled, and device_role is set to 1'b1 again.

When the device_role_changed interrupt is asserted and device_role is 1'b1, the IP successfully receives the GETACCCR CCC and controller role is transferred to the IP. The IP then asserts its role as the controller, depending on the auto_assert_role register in the I3C Controller IP.

5.54.9. Interrupt Enable 4 Register 0xBA

Interrupt enable.

Table 5.65. Interrupt Enable 4 Register 0xBA

Field	Name	Access	Description	Reset
[7]	get_acccr_rcvd_en	RW	When set to high, this field enables the corresponding interrupt status 4 signal to cause the assertion of the interrupt port signal (int_o).	0x0
[6]	get_acccr_acknak_en	RW		0x0
[5]	deftgts_rcvd_en	RW		0x0
[4]	reserved	RO		0x0
[3]	cr_req_gen_en	RW		0x0
[2]	cr_req_done_en	RW		0x0
[1]	cr_req_acknack_en	RW		0x0
[0]	device_role_changed_en	RW		0x0

5.54.10. Interrupt Set 4 Register 0xBB

Interrupt set.

Table 5.66. Interrupt Set 4 Register 0xBB

Field	Name	Access	Description	Reset
[7]	get_acccr_rcvd_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 4 signal.	0x0
[6]	get_acccr_acknak_set	WO		0x0
[5]	deftgts_rcvd_set	WO		0x0
[4]	reserved	RO		0x0
[3]	cr_req_gen_set	WO		0x0
[2]	cr_req_done_set	WO		0x0
[1]	cr_req_acknack_set	WO		0x0
[0]	device_role_changed_set	WO		0x0

5.54.11. DEFTGTS Count Register 0xC0

Count information sent by the Active Controller through the DEFTGTS CCC.

Table 5.67. DEFTGTS Count Register 0xC0

Field	Name	Access	Description	Reset
[7:0]	deftgts_count	RO	Describes the number of Targets and Groups present in the I3C bus.	0x0

5.54.12. DEFTGTS Rx FIFO Start Register 0xC1

Pointer for the start address of the DEFTGTS data in the Rx FIFO.

Table 5.68. DEFTGTS Rx FIFO Start Register 0xC1

Field	Name	Access	Description	Reset
[7:0]	rxfifo_deftgts_start	RO	Count N indicates the number of times you need to read from the Rx FIFO before the DEFTGTS data is read. When equal to 0, the next read data is the start of the DEFTGTS data.	0x0

5.54.13. DEFTGTS Rx FIFO Count Register 0xC2

Number of DEFTGTS data bytes stored in the Rx FIFO.

Table 5.69. DEFTGTS Rx FIFO Count Register 0xC2

Field	Name	Access	Description	Reset
[7:0]	rxfifo_deftgts_count	RO	Indicates the number of DEFTGTS data bytes stored in the Rx FIFO. When reading of DEFTGTS data starts (rxfifo_deftgts_start is equal to 8'h0), this count decreases by one with every Rx FIFO read. When the count reaches 0, all DEFTGTS data has been read from the Rx FIFO.	0x0

5.54.14. Controller Role Handoff Register 0xC3

Target configuration for the Controller Role Handoff.

Table 5.70. Controller Role Handoff Register 0xC3

Field	Name	Access	Description	Reset
[7]	get_acccr_auto_resp	RW	Target auto-response when receiving the GETACCCR CCC. The default response is NACK. Configuration options: • 0 – Target ACKs the GETACCCR CCC. • 1 – Target NACKs the GETACCCR CCC.	0x1
[6:0]	reserved	RO	Reserved.	0x0

5.54.15. GETMXDS Controller Capable Device Register 0xC4

Return value for GETMXDS when the defining byte is 0x91.

Table 5.71. GETMXDS Controller Capable Device Register 0xC4

Field	Name	Access	Description	Reset
[7:3]	reserved	RO	Reserved.	0x0
[2]	crh_set_act_state	RW	Indicates whether the Active Controller sets the bus to a certain Activity State before passing the Controller Role to the device. Configuration options: • 0 – Active Controller does not set the bus to any Activity State before passing the Controller Role to the Device. • 1 – Active Controller sets the bus to the Activity State set in bits 1:0 before passing the Controller Role to the Device.	0x0
[1:0]	crh_act_state	RW	When crh_set_act_state is set to 1, it indicates whether the device initially acts with a given Activity State after becoming the Active Controller on the bus. The indicated Activity State implies that the device may have a delayed response to bus activity. Therefore, the former Controller should wait the specified delay time for the indicated Activity State before testing this device, to	0x0

Field	Name	Access	Description	Reset
			confirm that it is controlling the bus before initiating the CE3 error recovery flow. Configuration options: • 0 – Acts according to Activity State 0. • 1 – Acts according to Activity State 1. • 2 – Acts according to Activity State 2. • 3 – Acts according to Activity State 3.	

5.54.16. GETSTATUS Controller Capable Device LSB Register 0xC5

Return value for GETSTATUS when the defining byte is 0x91. MSB of the return value is from the get_status_msb register at Addr 0x2A.

Table 5.72. GETSTATUS Controller Capable Device LSB Register 0xC5

Field	Name	Access	Description	Reset
[7:2]	reserved	RO	Reserved.	0x0
[1]	handoff_delay_nack	RW	Indicates whether the device is currently processing any DEFTGTS CCC broadcasts that may have been sent by the Active Controller. Status values: • 0 – The device is not currently processing broadcast data and can safely accept the Controller role. • 1 – The Device is currently processing DEFTGTS and may be updating its internal state. The Active Controller may wait before sending the GETACCCR CCC to this device, or should at least be aware that any attempt to send the GETACCCR CCC to this device will be met with a NACK response until this bit is read again later with a value of 1'b0.	0x0
[0]	deep_sleep_det	RW	Indicates whether the device enters a deep sleep state in which it may miss any DEFTGTS CCC sent by the Active controller. Consequently, this device's internal state of known Target devices should be considered outdated. Status values: • 0 – The device has not entered a deep sleep state. • 1 – The device has entered a deep sleep state. The Active Controller sends another DEFTGTS CCC to update the device's internal state before sending the GETACCCR CCC.	0x0

5.54.17. GETCAPS Controller Capable Device 1 Register 0xC6

Return value for GETCAPS when the defining byte is 0x91.

Table 5.73. GETCAPS Controller Capable Device 1 Register 0xC6

Field	Name	Access	Description	Reset
[7:3]	reserved	RO	Reserved.	0x0
[2]	multi_lane_support	RO	Configuration options: • 0 – The device does not use the MLANE CCC to change the ML configuration of other I3C Targets. • 1 – The device uses the MLANE CCC to change the ML configuration of other I3C Targets (Not supported in this IP).	0x0 (Fixed)
[1]	grp_mgmt_support	RO	Configuration options: • 0 – The device does not support Group Address capabilities. • 1 – The device supports Group Address handoff or management capabilities (Not supported in this IP).	0x0 (Fixed)

Field	Name	Access	Description	Reset
[0]	hot_join_support	RO	Configuration options: • 0 – The device does not support Hot-Join and NACKs any I3C Target Hot-Join requests (This setting is not supported in this IP when the device is configured with Secondary Controller Capability). • 1 – The device supports Hot-Join and ACKs an I3C Target Hot-Join request while it is the Active Controller on the bus.	0x1 (Fixed)

5.54.18. GETCAPS Controller Capable Device 2 Register 0xC7

GETCAPS controller capable device 2.

Table 5.74. GETCAPS Controller Capable Device 2 Register 0xC7

Field	Name	Access	Description	Reset
[7:4]	reserved	RO	Reserved.	0x0
[3]	dly_ctrl_handoff	RO	Tied to the get_acccr_auto_resp register. Configuration options: • 0 – The device does not need additional time to process Broadcast CCC data from the Active Controller. The Active Controller may expect it to ACK the GETACCCR CCC as part of the Controller Role Handoff procedure, even if the device did not initially send a Controller Role Request. • 1 – The device may need additional time to process data from the Controller. The Active Controller periodically checks the value returned by GETSTATUS with 0x91 to determine if the device is ready to accept the Controller Role.	0x1
[2]	deep_sleep_capable	RW	Configuration options: • 0 – The device remains active and continues to monitor the I3C bus to listen for Broadcast CCCs sent by the Active Controller. It does not enter a deep sleep state that requires resynchronization by the Active Controller before accepting the Controller role. • 1 – The device may enter a deep sleep state during which it may miss some Broadcast DEFTGTS sent by the Active Controller. Resynchronization is required upon re-entering a normal operating state before the device can accept the Controller role.	0x0
[1]	ctrl_pass_back	RW	Configuration options: • 0 – The device does not automatically pass the Controller role back to the former Active Controller and supports a Controller role request from any Controller-Capable device. • 1 – The device automatically passes the Controller role back to the former Active Controller from which it received the Controller role. This is done through the GETACCCR CCC after the device completes performing any tasks that required it to request and gain the Controller role. The device also supports a Controller role request from any Controller-Capable device.	0x0
[0]	ibi_support	RO	Configuration options: • 0 – The device does not support IBI and NACKs any I3C Target IBI (This setting is not supported in this IP when the device is configured with Secondary Controller Capability). • 1 – The device may support IBI and ACKs an I3C Target IBI while it is the Active Controller on the bus.	0x1 (Fixed)

5.54.19. Set Device Role Register 0xC8

Set the device role.

Table 5.75. Set Device Role Register 0xC8

Field	Name	Access	Description	Reset
[7:2]	reserved	RO	Reserved.	0x0
[1]	set_device_role	RW	This register allows resetting or changing the device role, which is particularly useful when the system loses track of the current role. Configuration values: • 0 – Configures the device as a Target. • 1 – Configures the device as a Controller. To switch from Target to Controller mode, write 1 to this register. This initiates the Controllership handoff process. If the system loses track of the device role, write 0 to this register to reset the Target role. Then write 1 to the corresponding set_device_role register on the Controller side to re-establish roles on both devices. Refer to the Set Device Role Register (0x16) in the I3C Controller IP User Guide (FPGA-IPUG-02228).	0x0
[0]	device_role	RO	Current device role. Status values: • 0 – Device is acting as the Target. • 1 – Device is acting as the Controller.	0x0

6. Example Design

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

- CrossLink-NX PCIe Bridge Board
- CertusPro-NX Evaluation Board
- Avant-E Evaluation Board

6.1. Example Design Supported Configuration

Table 6.1 shows the configuration of the I3C Target IP.

Table 6.2 shows the configuration of the I3C Controller IP version 3.3.0 since it is used as the Controller in this Example Design. Refer to the I3C Controller IP User Guide (FPGA-IPUG-02228) for more information.

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

Attribute	I3C Target
General	
User Interface	APB
Address Offset	Address Offset in DWORD
Optional Interface	
Enable Direct FIFO Interface	—
Tx Data Width	8 *(Display only)*
Rx Data Width	8 *(Display only)*
I/O Primitive Enable	
Enable internal I/O primitive	Checked
Bus Characteristics	
Bus Type	HDR-capable
HDR Mode	HDR-DDR *(Display only)*
IBI Capable	Checked
IBI Payload Size (including MDB)	1
Hot-Join Capable	Checked
Maximum Data Speed Limitation	Checked
Device Characteristics	
DCR (HEX)	00
Manufacturer ID	414
Part ID	1
Instance ID	1
Additional ID	0
Static Address Enable	Checked
Static Address (HEX) [1]	08
Timing Characteristics	
System Clock Frequency (MHz)	25
Write Maximum Data Rate (MHz)	12.5 *(Display only)*
Clock-to-data Turnaround Delay (ns) (t_{SCO})	0 *(Display only)*
Read Maximum Data Rate (MHz)	12.5 *(Display only)*
Maximum Read Turnaround Time (µs)	0 *(Display only)*

Table 6.2. I3C Controller IP Version 3.3.0 Configuration Supported by the Example Design

Attribute	Primary Controller
General	
Device Role	Primary Controller
Secondary Controller Capable	Checked
IBI Capable	Checked
Hot-Join Capable	Checked
Data Rate Mode	HDR-DDR-capable
Register Interface	
Interface	APB
Register Offset	Address offset in DWORD
Optional Interface	
Enable Direct FIFO Interface	—
SCL Clocking	
Bus Clock Domain	ASYNC *(Display only)*
Use internal clock divider	Checked
System Clock Frequency (MHz)	25
Internal Clock Frequency (MHz)	25 *(Display only)*
SCL Pulse width	1
SCL Clock Frequency (MHz)	12.5*(Display only)*
SCL Clock Period (ns)	80 *(Display only)*
Open Drain Pulse Width (Number of SCL cycles)	3
Open Drain Clock Frequency (MHz)	2.083 *(Display only)*
Open Drain Pulse width (ns)	239.99 *(Display only)*
Enable Dynamic I3C – I2C clock switching	Checked
I2C SCL Pulse width	13
I2C SCL Clock Frequency (MHz)	0.96 *(Display only)*
I2C SCL Clock Period (ns)	520 *(Display only)*
Include IO Primitive	Checked
Secondary Controller Tab	
Bus Characteristics	
IBI Payload Size (including MDB)	1
Maximum Data Speed Limitation	Checked
Device Characteristics	
DCR (HEX)	00
Manufacturer ID	414
Part ID	1
Instance ID	1
Additional ID	0
Static Address Enable	Checked
Static Address (HEX) 1	09
Dynamic Address (HEX) 1	09
Timing Characteristics	
Write Maximum Data Rate (MHz)	12.5 *(Display only)*
Clock-to-data Turnaround Delay (ns) (t_{SCO})	≤ 8 *(Display only)*
Read Maximum Data Rate (MHz)	12.5 *(Display only)*
Maximum Read Turnaround Time (µs)	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/TIMER
- GPIO
- Asynchronous SRAM
- UART – Serial port
- PLL
- Glue Logic

I3C Controller and I3C Target are instantiated and connected in the project as shown in Figure 6.1. In this example, I3C Controller and I3C Target are instantiated in the same system. Refer to the I3C Controller IP User Guide (FPGA-IPUG-02228) when generating the I3C Controller IP.

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

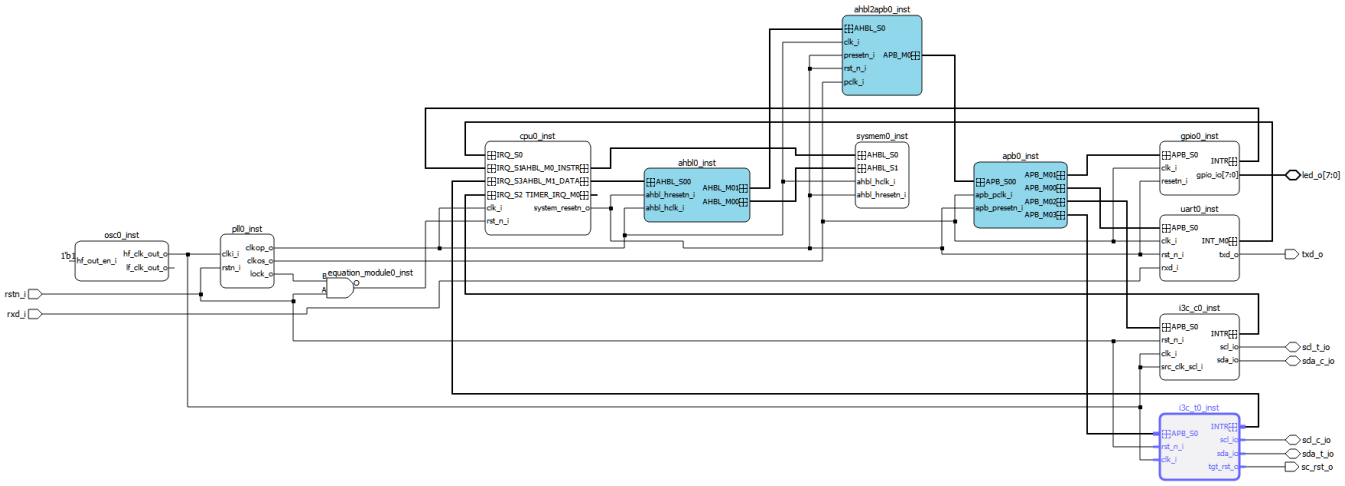

Figure 6.1. I3C Target IP in Propel SoC Project

An Embedded C/C++ Project is also created in the Propel software to enable developing and debugging application code for different IP features. I3C Target 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 for I3C Target Private Write.

```c
/**
 * @brief      This function is used for I3C  write.
 * @param[in]  Passing structure handle : base_addr, buf and len
 *             base_addr :  This parameter specify the I3C target IP base address.
 *             buf       :  This parameter specify the buffer where data is stored.
 *             len       :  This parameter specify the length of the buffer.
 * @return     Success or Failure
 * @exmaple    i3c_target_private_write(handle)
 */
unsigned int i3c_target_private_write(struct i3c_target_handle_t *handle)
{
    unsigned int status;
    unsigned int max_wr_len;
    unsigned int wr_len;
    unsigned char wr_len_msb;
    unsigned char wr_len_lsb;

    if((handle->base_addr != ZERO) && (handle->buf != ZERO) && (handle->len != ZERO))
    {
        i3c_tgt_reg_type_t *target = (i3c_tgt_reg_type_t *) (handle->base_addr);
        wr_len_msb =target->max_wr_legth_msb;
        wr_len_lsb=target->max_wr_legth_lsb;
        max_wr_len=(wr_len_msb<<EIGHT_BIT)|wr_len_lsb;
        wr_len=handle->len;
        if (wr_len>max_wr_len)
        {
            status = FAILURE;
        }
        else
        {
            for( int count=ZERO;count<handle->len;++count)         // loop wr_length times to send wr_length byte of data
            {
                target->tx_fifo=handle->buf[count];
            }
            target->interrupt_2=TX_FIFO_FULL_TGT;
            status = SUCCESS;
        }
    }
    else
    {
        status = FAILURE;
    }

    return status;
}
```

Figure 6.2. Sample C Code Test Routine

6.3. Example Design Components

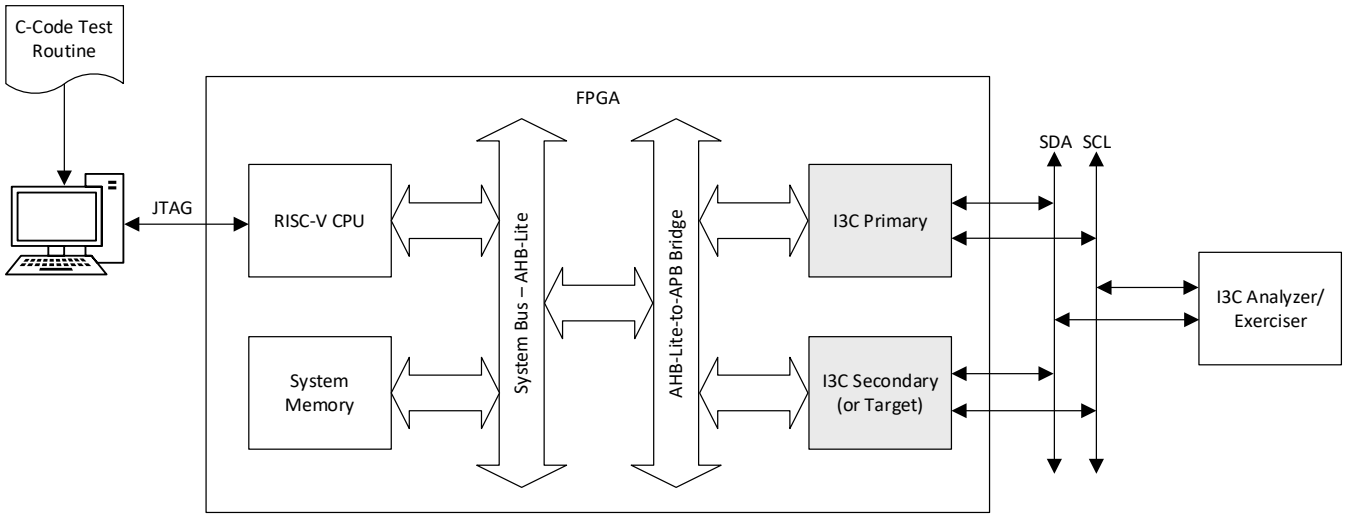

Figure 6.3. I3C Target Example Design Block Diagram

The I3C Target 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 to be done for testing.
- System Bus – AHB-Lite systems bus for transfers between memory and IP.
- I3C Target IP – IP instance connected to I3C bus (SCL and SDA).
- I3C Controller Device, and other I3C Target Device/s.

6.4. Generating the Example Design

Refer to the Lattice Propel SDK User Guide for more details in using Propel.

1. Launch Lattice Propel Software and set your workspace directory.
2. In Propel Software, create a new Lattice SOC Design Project. Click **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**.

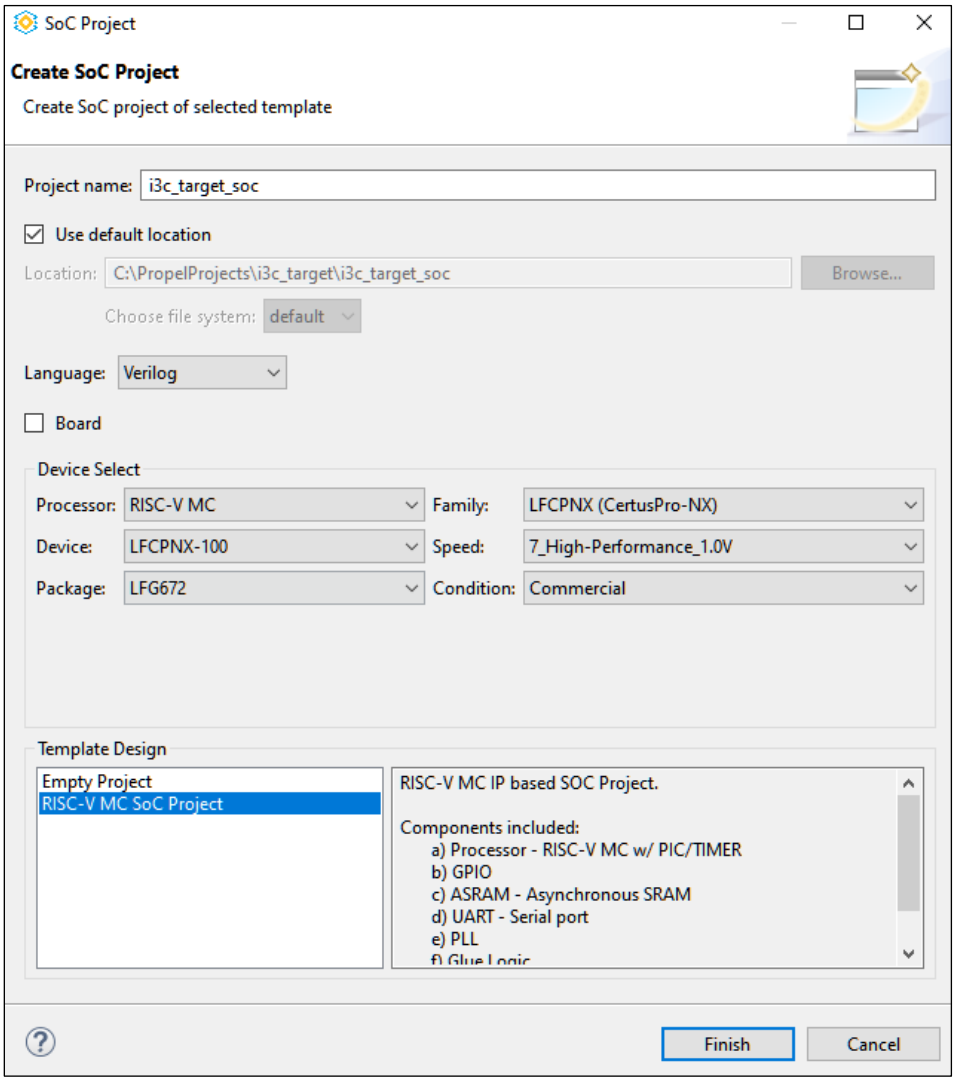

Figure 6.4. Create SoC Project

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 Target IP and the I3C Controller IP. Refer to the Generating and Instantiating the IP section for more details.

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

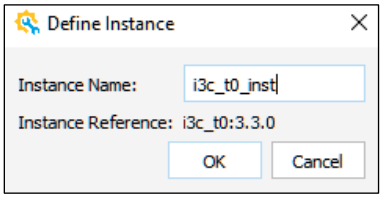

Figure 6.5. Define Instance

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/Os 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.

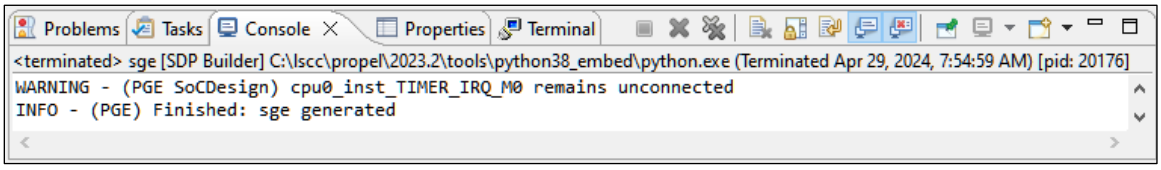

Figure 6.6. Build SOC Project Result

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**.

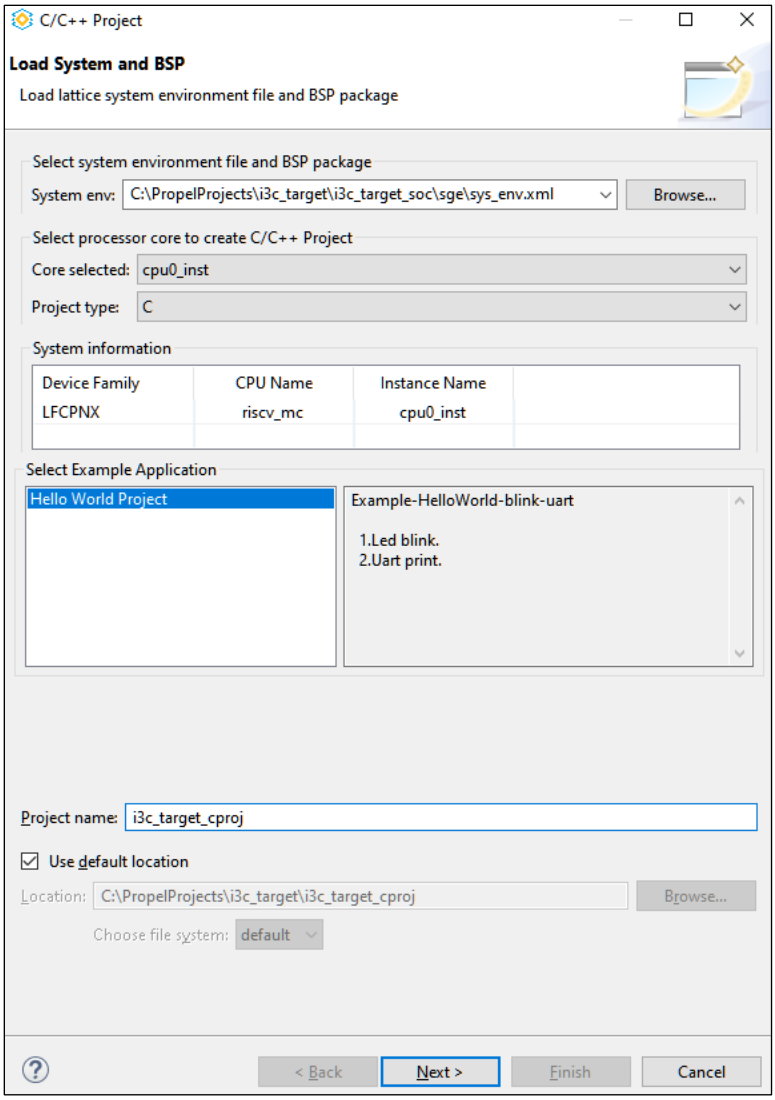

Figure 6.7. Lattice C/C++ Design Project

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.
17. Select your C/C++ project then click **Project** > **Build**.
18. Check the build result from the **Console** view.

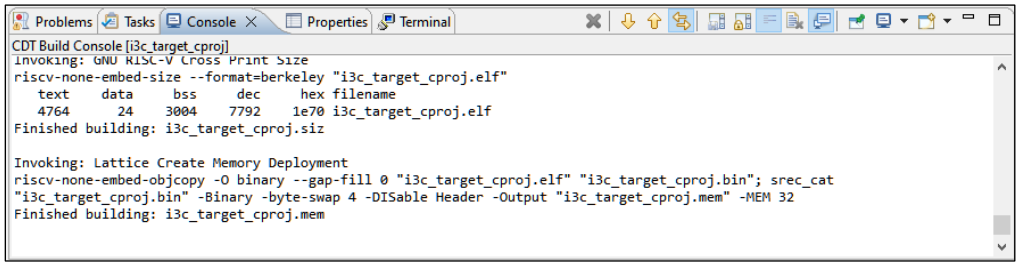

Figure 6.8. Build C/C++ Project Result

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

6.5. Hardware Testing

6.5.1. 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.

6.5.2. Expected Output

Below is a sample waveform captured via 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 Response by I3C Target

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 Target IP in the Lattice Radiant software.

To generate the I3C Target IP:

1. Create a new Lattice Radiant software project or open an existing project.
2. In the **IP Catalog** tab, double-click **I3C Target** 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**.

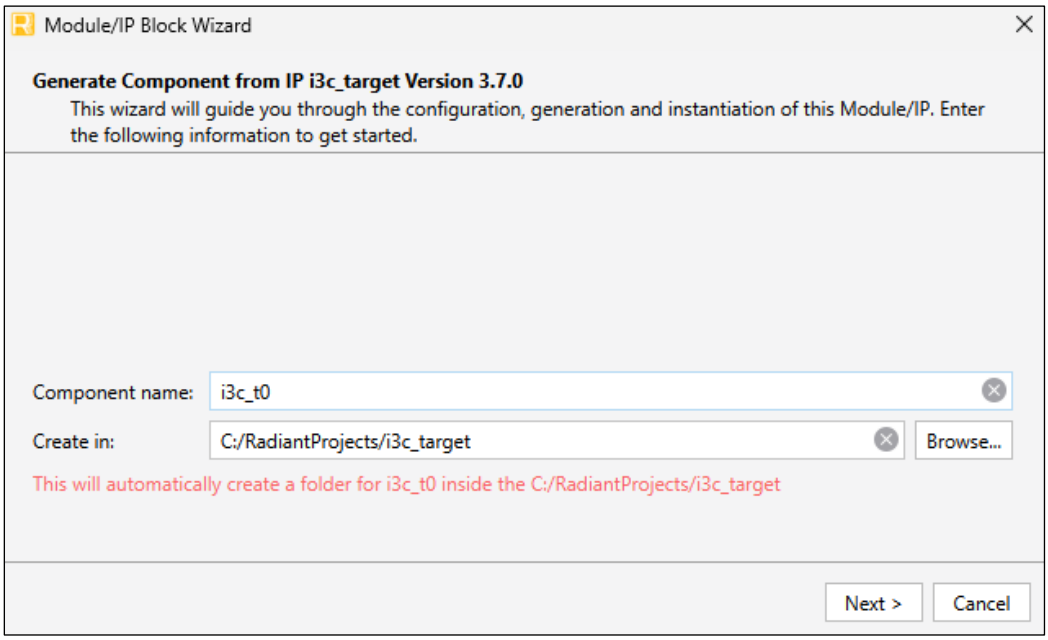

Figure 7.1. Module/IP Block Wizard

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

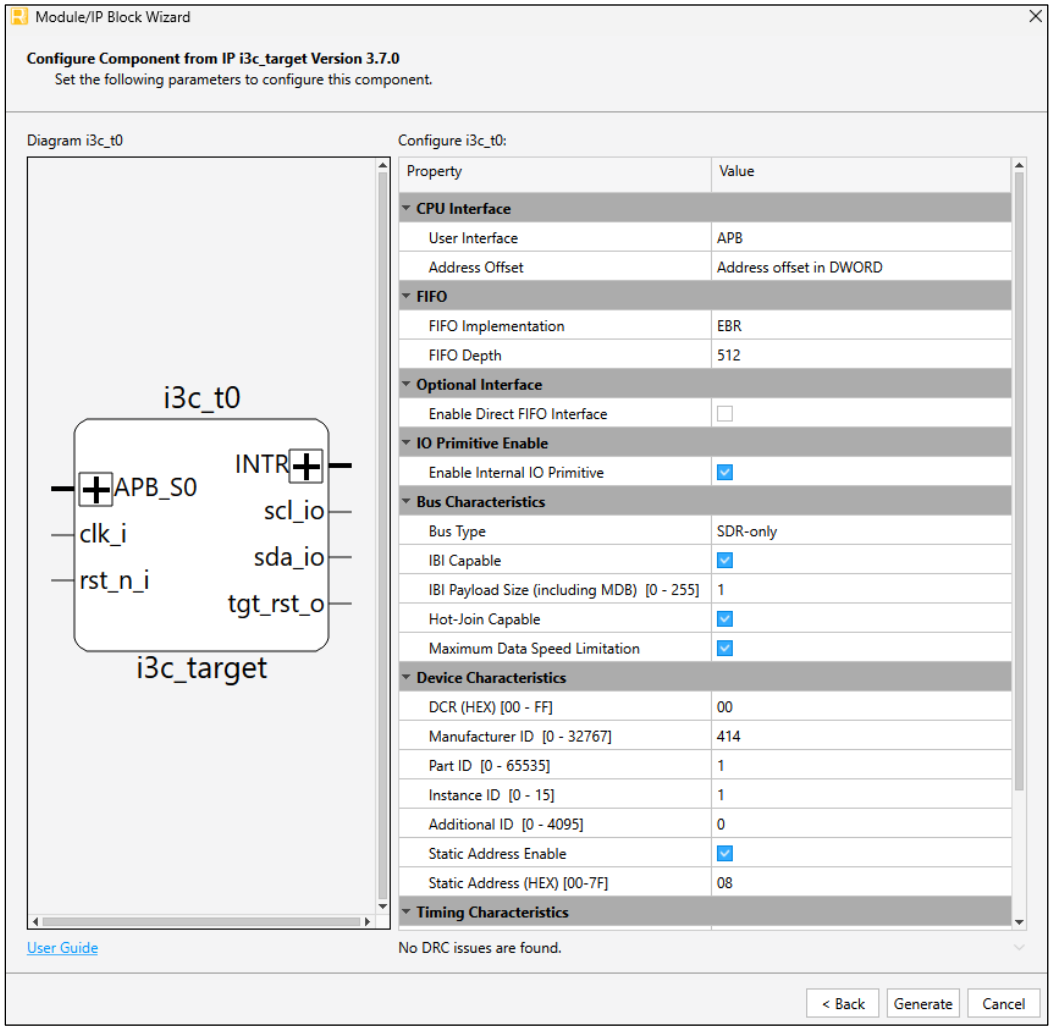

Figure 7.2. IP Configuration

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

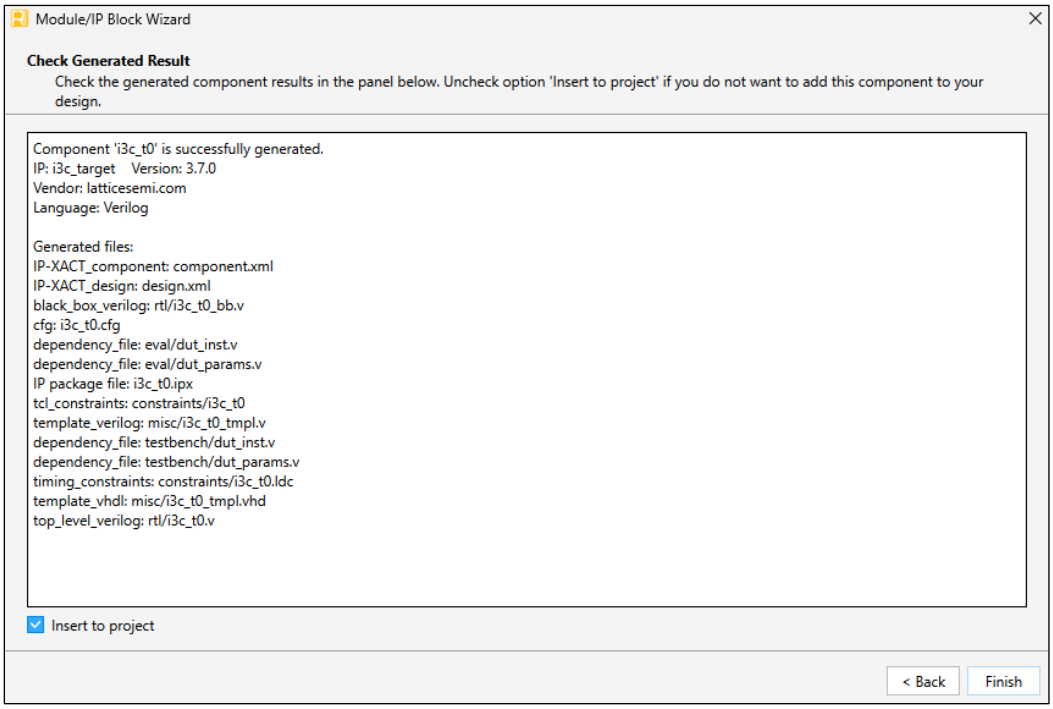

Figure 7.3. Check Generated Result

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 Target 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.

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]
```

You may encounter a PAR error if scl_io and sda_io are assigned to non-clock pins. In the IP, these signals function as clock signals:

- scl_io – serves as the clock on the I3C interface
- sda_io – serves as the clock for a small number of registers used for Start detection and Target Reset pattern

To resolve this error, remove the clock assignment restrictions on scl_io and sda_io by adding the following to your constraints file:

```
ldc_set_attribute {USE_PRIMARY=FALSE} [get_nets -hierarchical *i3c_target_inst/u_i3c_tgt/*sda_i*]
ldc_set_attribute {USE_PRIMARY=FALSE} [get_nets -hierarchical *i3c_target_inst/u_i3c_tgt/*scl_i*]
```

7.5. Running Functional Simulation

You can run functional simulation after the IP is generated.

To run functional simulation:

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

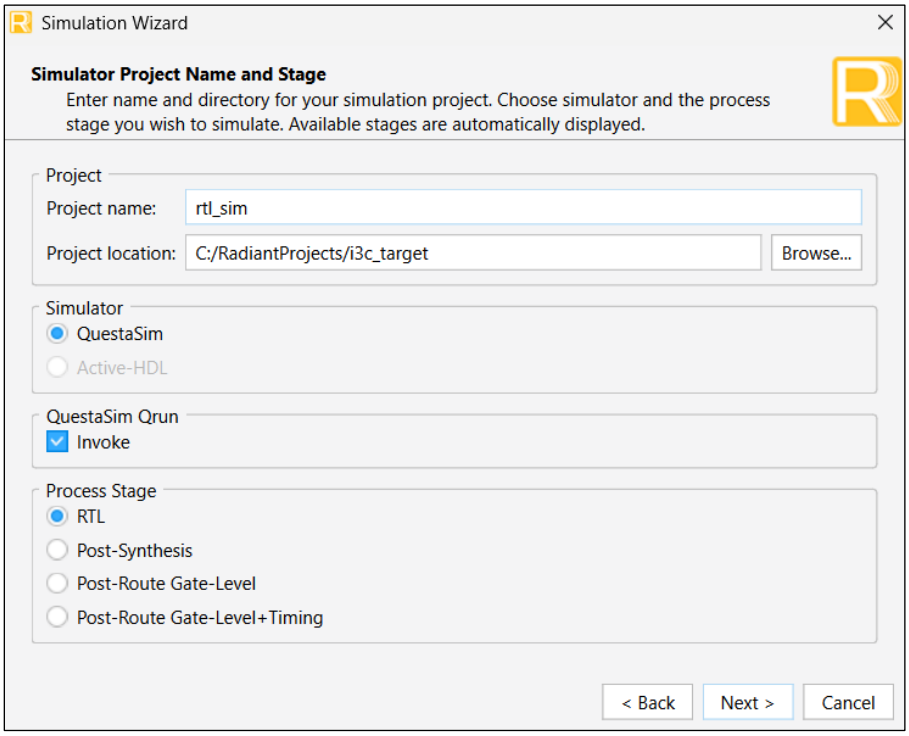

Figure 7.4. Simulation Wizard

2. Click **Next** to open the **Add and Reorder Source** window as shown in Figure 7.5.

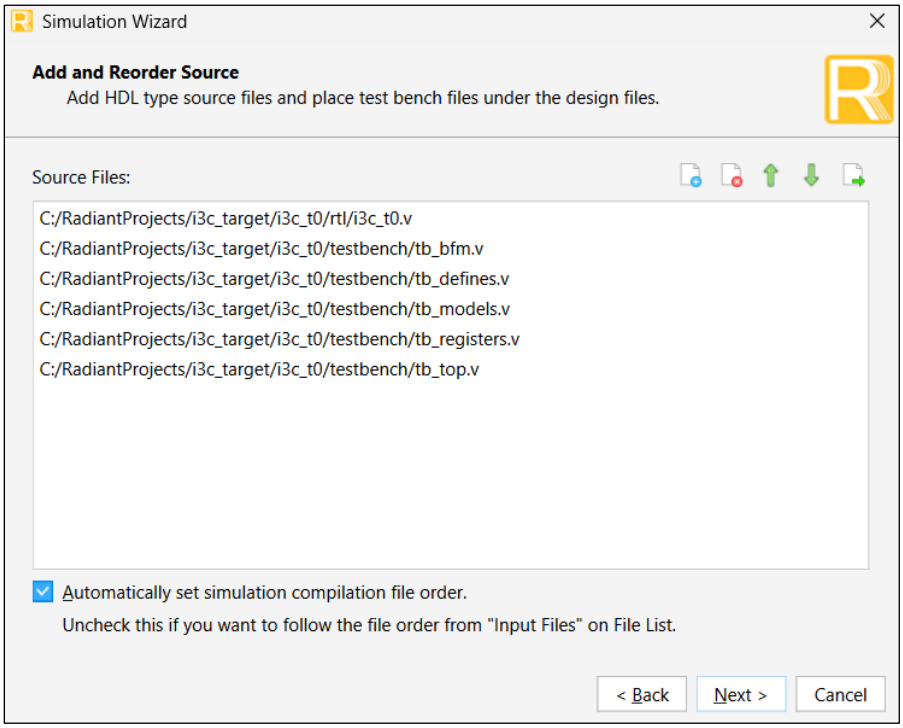

Figure 7.5. Add and Reorder Source

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

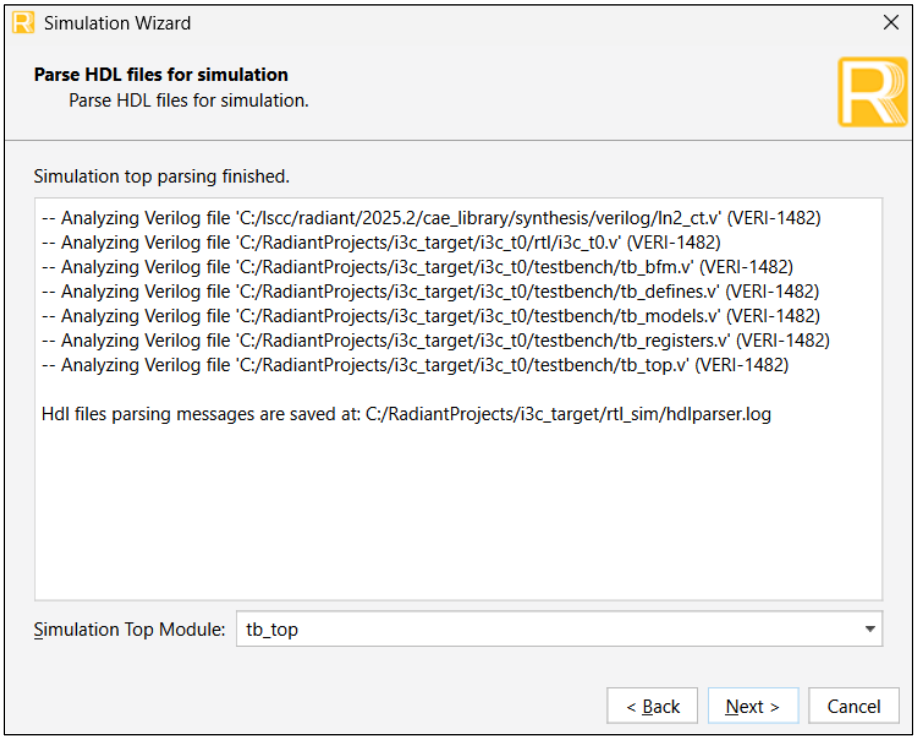

Figure 7.6. Parse HDL Files for Simulation

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

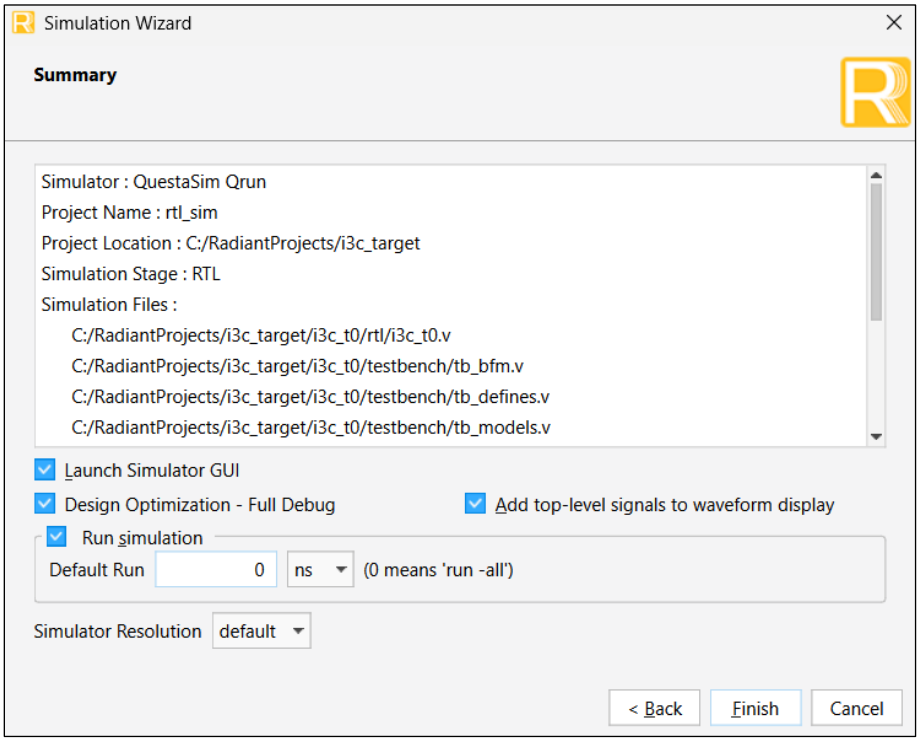

Figure 7.7. Summary

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

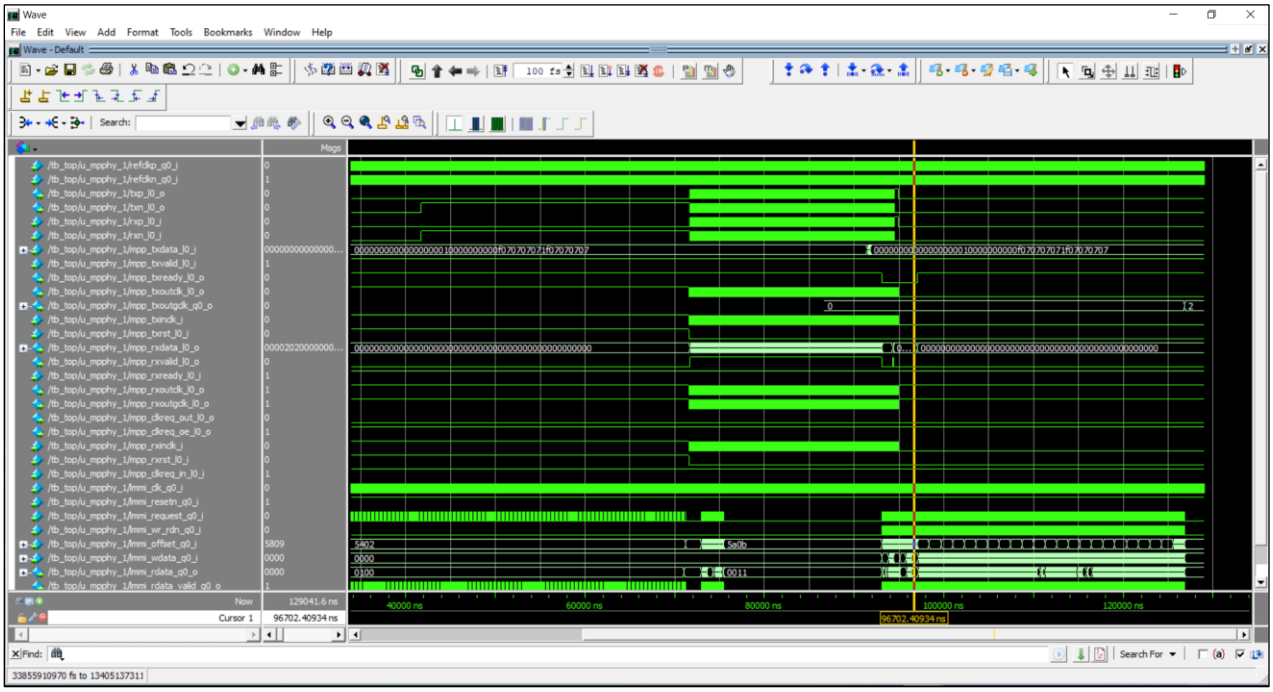

Figure 7.8. Simulation Waveform

Appendix A. Resource Utilization

Table A.1 shows the I3C Target resource utilization 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. Resource Utilization for LFCPNX-100-7ASG256C

Configuration	clk Fmax (MHz)	Registers	LUTs	EBRs	DSPs
Default	100.03	766	1829	2	0
IBI Capable = False, Hot-Join Capable = False, Others = Default	141.30	789	1927	2	0
Bus Type = HDR Capable, Others = Default	119.80	924	2489	2	0
Bus Type = HDR Capable, IBI Capable = False, Hot-Join Capable = False, Others = Default	144.70	829	1907	2	0

Notes:

1. Fmax is generated when the FPGA design only contains the I3C Target 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 distribution among logic, distributed RAM, and ripple logic.

Table A.2 shows the I3C Target resource utilization 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. Resource Utilization for LIFCL-40-7BG256I

Configuration	clk Fmax (MHz)	Registers	LUTs	EBRs	DSPs
Default	133.7	767	1817	2	0
IBI Capable = False, Hot-Join Capable = False, Others = Default	147.0	697	1470	2	0
Bus Type = HDR Capable, Others = Default	133.4	924	2489	2	0
Bus Type = HDR Capable, IBI Capable = False, Hot-Join Capable = False, Others = Default	156.5	829	1907	2	0

Notes:

1. Fmax is generated when the FPGA design only contains the I3C Target 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 distribution among logic, distributed RAM, and ripple logic.

Table A.3 shows the I3C Target utilization 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. Resource Utilization for LAV-AT-E70-1LFG676I

Configuration	clk Fmax (MHz)	Registers	LUTs	EBRs	DSPs
Default	249.5	793	1861	2	0
IBI Capable = False, Hot-Join Capable = False, Others = Default	215.5	696	1463	2	0
Bus Type = HDR Capable, Others = Default	162.9	939	2492	2	0
Bus Type = HDR Capable, IBI Capable = False, Hot-Join Capable = False, Others = Default	173.7	817	1856	2	0

Notes:

1. Fmax is generated when the FPGA design only contains the I3C Target 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 distribution among logic, distributed RAM, and ripple logic.

Table A.4 shows the I3C Target utilization 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. Resource Utilization for LN2-CT-20-1CBG484I

Configuration	clk Fmax (MHz)	Registers	LUTs	EBRs	DSPs
Default	200.00	937	1997	2	0
IBI Capable = False, Hot-Join Capable = False, Others = Default	183.79	839	1699	2	0
Bus Type = HDR Capable, Others = Default	192.35	1101	2669	2	0
Bus Type = HDR Capable, IBI Capable = False, Hot-Join Capable = False, Others = Default	172.38	1050	2349	2	0

Notes:

1. Fmax is generated when the FPGA design only contains the I3C Target 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 distribution among logic, distributed RAM, and ripple logic.

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-NX web page
- MachXO3D web page
- MachXO5-NX 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
- 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 Controller IP User Guide (FPGA-IPUG-02228)
- I3C Target IP Release Notes (FPGA-RN-02018)
- Lattice Insights web page for Lattice Semiconductor training courses and learning plans

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.

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 1.6, IP v.3.7.0, 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*, *CPU*, *DWORD*, *GPIO*, *MWL*, *PID*, *RO*, *RTL*, *RW*, *RW1C*, *Rx*, *SRAM*, *Tx*, *UART*, and *WO*.

Introduction	• Updated *Lattice Implementation* in Table 1.1. Summary of the I3C Target IP.
	• In Table 1.2. I3C Target IP Support Readiness:
	  • Added a table note.
	  • Updated hardware validation information for the MachXO5-NX device family.
	• Added the configurable 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	• Added *through CCC* to the *Configurations for the I3C Target that may be requested by the Controller through CCC* sentence in the IP Architecture Overview section. • Updated Table 2.1. User Interfaces and Supported Protocols. • Updated the table caption for Table 2.2. I3C Target IP Supported CCC.

IP Parameter Description	• In Table 3.1. General Attributes:
	  • Updated the description for *Address Offset*.
	  • Added attributes for *FIFO*.
	  • Updated *not checked* to *unchecked*.
	  • Updated the table note.
	• Removed the table note for Table 3.2. IP Parameter Settings for Example Use Cases.

Signal Description	Updated Table 4.1. Ports Description.
Register Description	Updated this section.

Example Design	• Replaced ✓ with *Checked* in Table 6.1. I3C Target IP Configuration Supported by the Example Design and Table 6.2. I3C Controller IP Version 3.3.0 Configuration Supported by the Example Design.
	• Updated the descriptions in the following sections:
	  • Overview of the Example Design and Features
	  • Generating the Example Design
	  • Hardware Testing

Designing with the IP	• Added a note on screenshots in this section. • Updated Figure 7.1. Module/IP Block Wizard – Figure 7.7. Summary. • Added the *PAR error* information in the Physical Constraints section.
References	Added the *Mach-NX*, *MachXO3D*, and *Lattice Solutions Reference Designs* web pages.

Revision 1.5, IP v.3.6.0, July 2025

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

Introduction	• In Table 1.1. Summary of the I3C Target 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*.

Section	Change Summary

	• 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 v.3.5.0, December 2024

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

Introduction	• Updated Table 1.1. Summary of the I3C Target IP:
	  • Added the *Certus-NX-RT*, *CertusPro-NX-RT*, *Certus-N2* device family 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 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* web page, *CertusPro-NX Evaluation Board* web page, and *I3C Target IP Release Notes (FPGA-RN-02018)*. • 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:
	  • *Bus Characteristics Register (BCR)*
	  • *Cyclic Redundancy Check (CRC)*
	  • *Device Characteristics Register (DCR)*
	  • *Input/Output (I/O)*
	  • *Microcontroller (MC)*
	  • *Mobile Industry Processor Interface (MIPI)*
	  • *Negative Acknowledgement (NACK)*
	  • *Programmable Interrupt Controller (PIC)*
	  • *Phase-Locked Loop (PLL)*
	  • *Reduced Instruction Set Computer Five (RISC-V)*
	  • *Single Data Rate (SDR)*
	  • *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.

Section	Change Summary
	• Updated Table 1.1. Summary of the I3C Target IP. • Updated the information in the 1.3. Features section. • Moved the content from previous *3.1. Licensing the IP* and *3.6. 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 *3.5. 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. • 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. • Removed the introductory paragraph and previous *CCCs Permitted in HDR Mode* subsection title in the 2.13. HDR-DDR Support section.

IP Parameter Description	• Moved the content from previous *2.13. Attributes Summary* section to this newly added section.
	• Updated the following tables:
	  • Table 3.1. General Attributes
	  • Table 3.2. IP Parameter Settings for Example Use Cases

Signal Description	• Moved the content from previous *2.10. Signal Description* section to this section.
	• Updated the descriptions of the following signals in Table 4.1. Ports Description:
	  • *clk_i*
	  • *rst_n_i*
	• 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.
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 Controller IP User Guide (FPGA-IPUG-02228)*

Revision 1.2, December 2023

Section	Change Summary
All	• Changed the document name from *I3C Target IP Core - Lattice Radiant Software* to *I3C Target IP Core*. • 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	• Added *additional* in paragraph of 2.10.2 CCC for Secondary Controller Support. • Corrected the spelling for *Target Reset Action* in 2.14.37 Target Reset Action 2 0x2E. • Updated the content for 2.14.53 HDR-DDR Abort Configuration 0x54. • Removed *device_role* in paragraph of 2.14.54.14 Controller Role Handoff 0x43, and in Table 2.73. • Changed the bit and width for reserved in Table 2.73. • Added 2.14.54.19 Set Device Role 0x48 section, and Table 2.78.
IP Core Generation, Simulation, and Validation	• Removed the sentence When the IP Core used in Lattice FPGA devices built on the Lattice Nexus™ platform from the 3.1 Licensing the IP section. • Updated Figure 3.1. • Added new attribute eval/constraint.pdc to Table 3.1. • Added 3.4 Constraining the IP section. • Updated the header number for 3.5 Hardware Validation, and 3.6 IP Evaluation sections.
Ordering Part Number	Updated Table 4.1 to add new part numbers, removed obsolete part numbers, and updated the license types.
Resource Utilization	• Specified the device name in captions of Table A.1, and Table A.2. • Added sentences *Table A.1 shows the I3C Target resource utilization using LFCPNX-100-7ASG256C device using Synplify Pro of Lattice Radiant Software 2023.2, and Table A.2 shows the I3C Target resource utilization using LIFCL-40-7BG256I device using Synplify Pro of Lattice Radiant Software 2023.2 to the Appendix A. Resource Utilization section.* • Updated the configurations and its corresponding *clk Fmax, Registers, and LUTs* values 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	• Updated the hyperlink to CrossLink-NX web page. • Added links to Lattice Avant-G, Lattice Avant-X, and Lattice Insights web pages.

Revision 1.1, July 2023

Section	Change Summary
Acronyms in This Document	Added *AHB*, *APB*, *GUI*, *HDL*, and *LSE* and their definitions.
Introduction	• Added sentence *The Lattice I3C IP Core is designed to comply with the MIPI I3C specification* in Introduction section. • Added Resource column in Table 1.1. Quick Facts. • Added bullet information *_io are bidirectional signals* in Signal Names section. • Added Attribute section.
Functional Description	• Updated Figure 2.1. I3C Target IP Core Functional Diagram. • Added Reset Propagation section and moved Common Command Codes section before I3C Transfers in SDR Mode section. • Replaced Broadcast with Target in Figure 2.14. HDR-DDR Direct Set CCC and Figure 2.15. HDR-DDR Direct Get CCC. • Updated Table 2.5 Ports Description for below: • Added signal names *tgt_rst_o, ext_scl_i, ext_sda_i, ext_sda_o, ext_sda_oe, Direct FIFO Interface8, tx_valid_i, tx_ready_o, tx_data_i [7:0], rx_valid_o, rx_ready_i*, and *rx_data_o*. • Deleted signal names *Immi_error_o* and *tgt_rst_o*.

Section	Change Summary

	• Added table notes For more details on target reset, see *Section 5.1.11 of the MIPI Specification for I3C. Bidirectional I3C interface is only available when internal I/O primitives are enabled in IP configuration GUI. External I/O I3C interface is only available when internal I/O primitives are disabled in IP configuration GUI. Direct FIFO interface is only available when enabled in IP configuration GUI*.
	• Updated Table 2.6 Attributes Summary for below:
	  • Replaced default values of User Interface and Address Offset from *LMMI and Bytes* to *APB and DWORD*.
	  • Added attributes *Optional Interface, Enable Direct FIFO Interface, Tx Data Width, Rx Data Width, I/O Primitive Enable*, and *Enable internal I/O primitive*.
	  • Replaced Selectable values of System Clock Frequency (MHz) from *125* to *50*.
	  • Deleted *Selectable values of Write Maximum Data Rate (MHz), Clock-to-data Turnaround Delay (ns) (tSCO), Read Maximum Data Rate (MHz), Maximum Read Turnaround Time (us)*.
	• Replaced Reset information of [3] Bits from 0x1 to 0x1 – *If Target is Hot-Join capable 0x0 – If Target is not Hot-Join capable and Reset information of [0] Bits from 0x1 to 0x1 – If Target is IBI capable 0x0 – If Target is not IBI capable*, and deleted table note in Table 2.10. Events Command Enable.
	• Replaced Reset information of [3] Bits from 0x1 to 0x1 – *If Target is Hot-Join capable 0x0 – If Target is not Hot-Join capable, and Reset information of [0] from 0x1 to 0x1 – If Target is IBI capable 0x0 – If Target is not IBI capable* in Table 2.11. Events Command Device Configuration.
	• Added Reset information of [0] Bits as – *SDR mode only 0x1 – SDR and HDR-DDR mode is supported* in Table 2.31. Device Capabilities Byte1.
	• Replaced Names from *hdr_ddr_abort_crc, hdr_ddr_wr_abort* to *hdr_ddr_abort_crc_caps, hdr_ddr_wr_abort_caps* and replaced Reset information with *0x0 – If only SDR capable, 0x1 – if HDR DDR capable and 0x0 – If only SDR capable, 0x1 – if HDR DDR capable*.
	Added
	• Interrupt Status 5 0x3C, Interrupt Status 5 Enable 0x3D, and Interrupt Status 5 Set 0x3E sections.
	• Updated the title of HDR-DDR Abort Configuration 0x54 section from *HDR-DDR Abort Configuration 0x54 set by Controller via ENDXFER CCC* and added sentence *HDR-DDR Abort Configuration set by Controller via ENDXFER CCC*.
	• Added sentences *I3C Target registers are accessible when I3C Controller IP is configured with Secondary-Controller Capability, and The following section describes I3C Target registers that have changes in reset value and/or definition when used for Secondary Controller. Registers not included in this section retain the definitions as when device is configured as I3C Target-only* in Secondary Controller Registers section.
	• Added Dynamic Address 0x02 and Maximum Write Data Speed (MaxWr) 0x0C section.
	• Replaced Reset information of device_role from *0x0 with 0x0 – Device is Target-only 0x1 – Device is Controller Capable* and Reset information of advanced_caps from *0x0* to *0x1 (Fixed)* in Table 2.61. Bus Characteristics Register.
	• Added sentence *Fixed to 1 when device is configured with Secondary-Controller capability* in Bus Characteristics Register 0x00 section.
	• Replaced Reset information of hj_enec from *0x1* with *0x1 – If Target is Hot-Join capable 0x0 – If Target is not Hot-Join capable* and Reset information of *advanced_caps from 0x1 to 0x1 – If Target is IBI capable 0x0 – If Target is not IBI capable* in Table 2.62. Events Command Enable.
	• Deleted sentences *Reset value is 1 if Target is configured with Hot-Join capability, 0 otherwise and Reset value is 1 if Target is configured with IBI capability, 0 otherwise* in Events Command Enable 0x03 section.
	• Deleted sentences *Reset value is 1 if Target is configured with Hot-Join capability, 0 otherwise. Reset value is 1 if Target is configured with Secondary-Controller Capability, 0 otherwise* and *Reset value is 1 if Target is configured with IBI capability, 0 otherwise* in Events Command Device Configuration 0x04 section.
	• Replaced Reset information of hj_cap, cr_cap, ibi_cap with *0x1 – If Target is Hot-Join capable 0x0 – If Target is not Hot-Join capable, 0x1 – If Target is Secondary Controller Capable 0x0 – If Target is not Secondary Controller Capable, 0x1 – If Target is IBI capable 0x0 – If Target is not IBI capable* in Table 2.63. Events Command Device Configuration.
	• Replaced name from *reserved* to *device_role_changed and access from read-only to read-write* in

Section	Change Summary
	Table 2.67. Interrupt Status 4 • Replaced name from *reserved* to *device_role_changed_en and access from read-only to read-write* in Table 2.68. Interrupt Status 4 Enable. • Added device_role_changed information in Interrupt Status 4 0x39 section. • Replaced Access from read-write to write-only in Table 2.69. Interrupt Status 4 Set. • Added *LSB* to title of GETSTATUS Controller-Capable Device LSB 0x45 section.
IP Core Generation, Simulation, and Validation	• Added Constraining the IP and IP Evaluation sections.
Ordering Part Number	Added Avant-E part numbers in Table 4.1. Ordering Part Numbers.
Reference	Added links for CrossLink-NX, Certus-NX, CertusPro-NX, MachXO5-NX, iCE-40 UltraPlus, Avant-E web pages.

Revision 1.0, April 2023

Section	Change Summary
All	Initial release.

www.latticesemi.com