



CrossLinkPlus I2C Hardened IP Usage Guide

Technical Note

FPGA-TN-02112-1.0

August 2019

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 and with all faults, and all risk associated with such information is entirely with Buyer. Buyer shall not rely on any data and performance specifications or parameters provided herein. 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. No Lattice products should be used in conjunction with mission- or safety-critical or any other application in which the failure of Lattice's product could create a situation where personal injury, death, severe property or environmental damage may occur. 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.

Contents

Acronyms in This Document	4
1. Introduction	5
2. I ² C IP Core Overview	5
3. Generating I ² C Module with Clarity Designer	6
3.1. Enabling Hardened IP	6
3.2. I ² C Settings	7
3.2.1. General Settings	8
3.2.2. Master Clock Rate	9
3.2.3. FIFO Mode	10
3.2.4. I ² C Addressing	10
3.2.5. PMU Wake Options	10
3.3. I ² C Interrupts	11
3.3.1. System Bus Interrupts	11
3.3.2. FIFO Interrupts	12
4. SB_I2C Hardened IP Macro Ports and Wrapper Connections	13
5. I ² C Usage Cases	15
References	16
Technical Support Assistance	17
Revision History	18

Figures

Figure 3.1. I ² C Module in Clarity Designer	6
Figure 3.2. Enabling Hard I ² C IP Blocks	7
Figure 3.3. I ² C Settings	8
Figure 3.4. I ² C I/O Buffer	9
Figure 3.5. I ² C I/O Connections	9
Figure 3.6. I ² C Interrupts	11
Figure 5.1. I ² C Circuit	15

Tables

Table 4.1. Pins for I ² C Hardened IP	13
--	----

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
ASIC	Application-Specific Integrated Circuit
FIFO	First In First Out
I ² C	Inter-Integrated Circuit
SCL	Serial Clock Line
SDA	Serial Data Line

1. Introduction

CrossLinkPlus™ from Lattice Semiconductor is a programmable video bridging device that supports a variety of protocols and interfaces for mobile image sensors and displays. The device is based on Lattice mobile FPGA technology. It combines the extreme flexibility of an FPGA with the low power, low cost, and small footprint of an ASIC.

CrossLinkPlus devices include integrated I²C blocks to interface with virtually all mobile sensors and application processors. CrossLinkPlus devices provide two I²C IP cores. The I2C0 core has dedicated I/O pins, called USER_SCL and USER_SDA, on the CrossLinkPlus device. This is in order to support the device sleep mode wakeup over I²C function. The SCL and SDA pins from the I2C1 core may be connected to any pin on the device.

This document provides guidance for the usage of the Lattice Semiconductor CrossLinkPlus I²C IP. It explains the steps and configuration settings required to generate the I²C module using the Lattice Diamond® Clarity Designer tool.

[Advanced CrossLinkPlus I2C Hardened IP Reference Guide \(FPGA-TN-02135\)](#) supplements this document.

2. I²C IP Core Overview

The I²C hardened IP provides industry standard two-pin communication interface that conforms to the I²C Bus Specification version 2.1. It may be configured in either Master or Slave mode.

In Master mode, the I²C hardened IP supports configurable data transfer rate and performs arbitration detection so as to allow the hardened IP to operate in multi-master systems. The hardened IP supports clock stretching in both Master and Slave modes with enable/disable capability.

The I²C hardened IP supports both seven bits and ten bits addressing in Slave mode with configurable Slave address. It supports general call address detection in both Master and Slave mode.

The I²C hardened IP provides interrupt logic for easy communication with host. It also provides configurable digital delay at SDA output for generating start/stop conditions reliably.

The key I²C IP core features of CrossLinkPlus devices:

- Configurable Master and Slave mode
- Support for 7-bit or 10-bit configurable Slave address
- Multi-master arbitration support
- Clock stretching to ensure data setup time
- Up to 1000 kHz data transfer speed, also supports 400 kHz, 100 kHz, and 50 kHz modes
- General call support
- Optional delaying input or output data, or both
- Optional FIFO mode
- Transmit FIFO size is 10 bits x 16 bytes, receive FIFO size is 10 bits x 32 bytes
- I2C0 core supports sleep mode wakeup over I²C. Refer to [Power Management and Calculation for CrossLinkPlus Devices \(FPGA-TN-02111\)](#) for more details.

3. Generating I²C Module with Clarity Designer

The Clarity Designer tool of Lattice Diamond is used to configure the I²C hardened IP functions and generate the I²C module.

3.1. Enabling Hardened IP

To enable the hardened user I²C blocks, perform the following steps:

1. On the Lattice Diamond top menu, click **Tools**, and then **Clarity Designer**. With a CrossLinkPlus device targeted for the Diamond project, the Clarity Designer window appears and the I²C module is listed under **Module > Architecture_Modules**, as shown in [Figure 3.1](#).

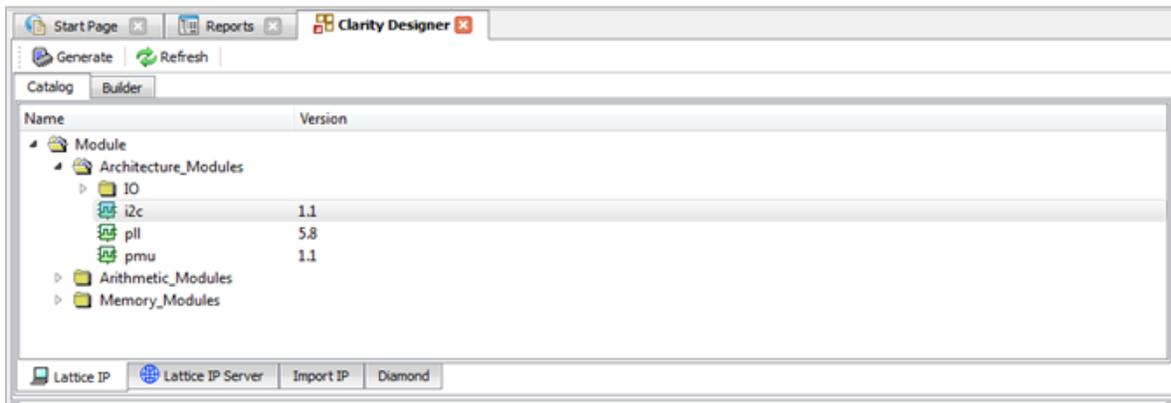


Figure 3.1. I²C Module in Clarity Designer

2. Click **i2c**. The Lattice FPGA module – I2C window appears, as shown in [Figure 3.2](#).
3. On the **Hard IP Enables** tab, select the desired I²C core (I2C0 or I2C1, or both), and enter your System Bus clock frequency. The **I2C Settings** and **I2C Interrupts** tabs are enabled when at least one of the Enable Hard Users I²C block is selected.

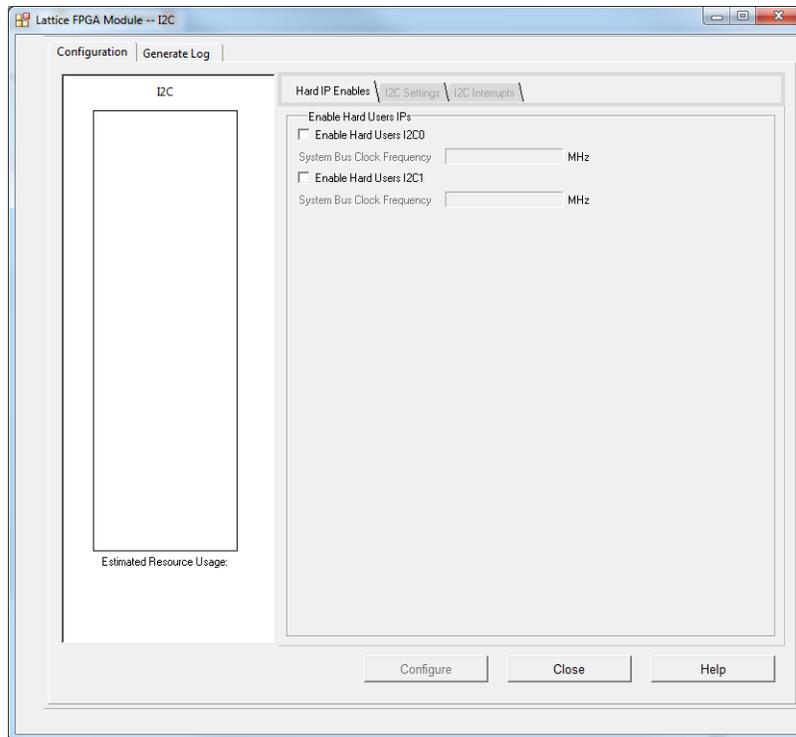


Figure 3.2. Enabling Hard I²C IP Blocks

3.2. I²C Settings

This section describes the I²C Settings tab of Lattice FPGA Module – I²C window, as shown in [Figure 3.3](#).

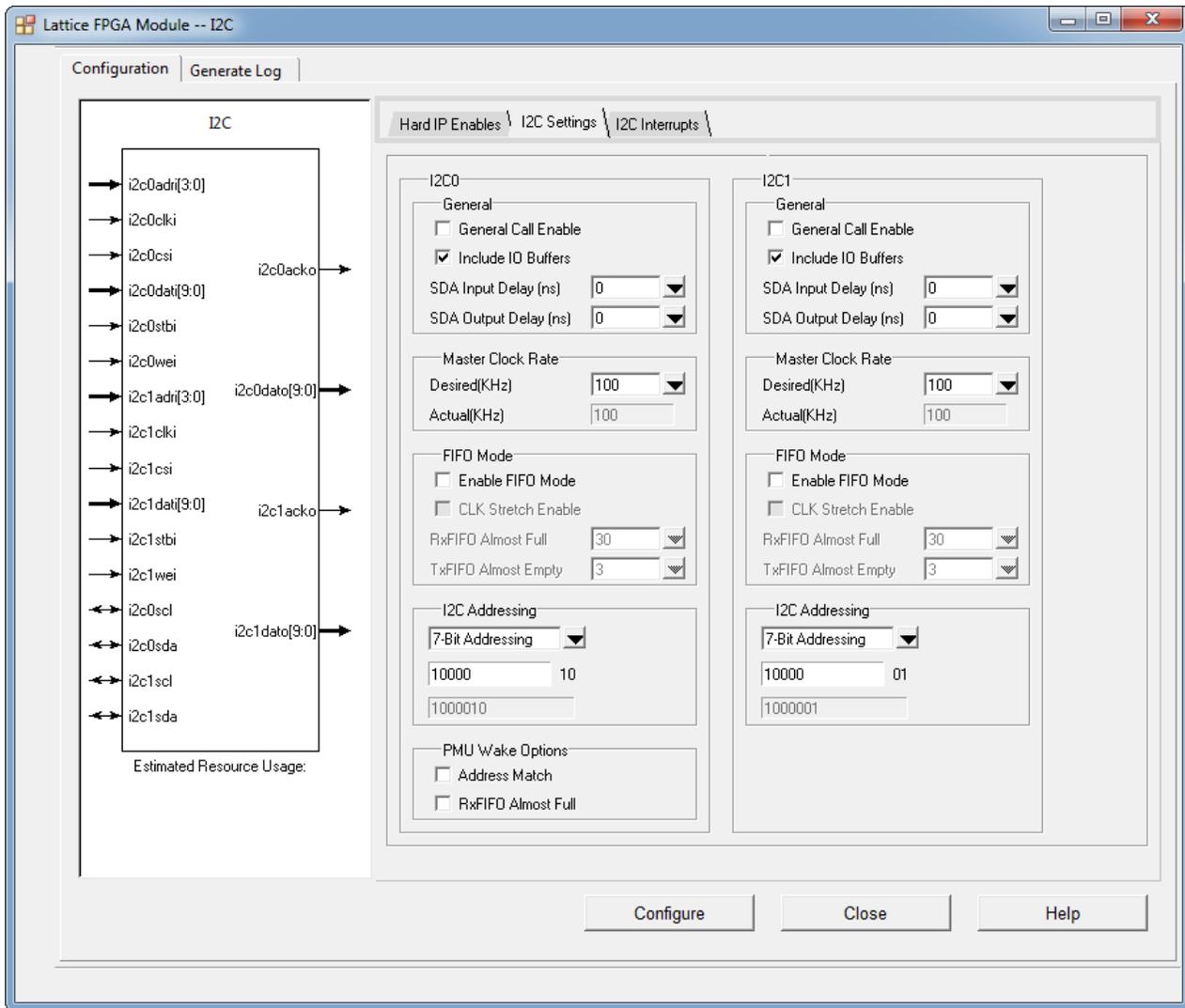


Figure 3.3. I²C Settings

3.2.1. General Settings

3.2.1.1. General Call Enable

Select this checkbox to enable I²C General Call response (addresses all devices on the bus using the I²C address 0) in Slave mode. This setting can be modified dynamically by enabling the GCEN bit in the I²C Control Register I2CCR1. Refer to [Advanced CrossLinkPlus I2C Hardened IP Reference Guide \(FPGA-TN-02135\)](#) for register descriptions.

3.2.1.2. Include I/O Buffers

Select this checkbox to include buffers for the I²C pins. Note that I2C0 pins are hard-routed through I/O buffer to the dedicated pins on the device. These buffers need to be included with I2C0.

Figure 3.4 and Figure 3.5 show the I²C I/O buffer and connections, respectively.

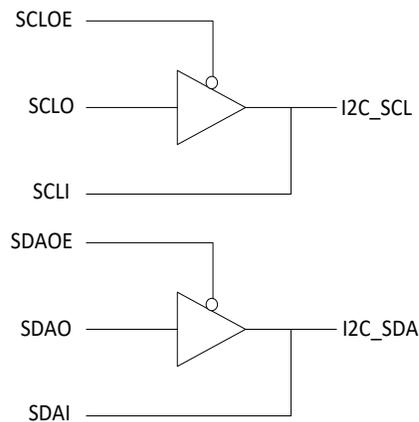


Figure 3.4. I2C I/O Buffer

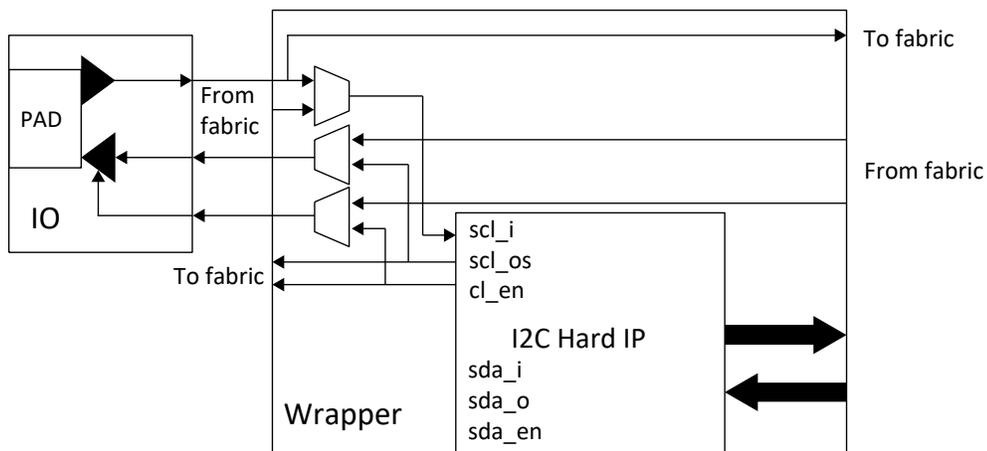


Figure 3.5. I2C I/O Connections

3.2.1.3. SDA Input Delay

Select SDA Input Delay between 0 ns and 50 ns.

3.2.1.4. SDA Output Delay

Select SDA Output Delay between 0 ns and 350 ns.

3.2.2. Master Clock Rate

3.2.2.1. Master Clock Desired

In the Master Clock Desired field, you can specify a desired Master clock frequency in kHz. A calculation is then made to determine a divider value to generate a clock close to this value from the input clock. The frequency of the input System Bus clock is specified on the **Hard IP Enables** tab. The divider value is rounded to the nearest integer after dividing the input System Bus clock by the value entered in this field.

3.2.2.2. Master Clock Actual

This is a read-only field. It is not always possible to divide the input System Bus clock to the exact value you requested, therefore the actual value in kHz is returned in this field. When both the desired I2C clock and System Bus clock fields contain valid data and either is updated, the Master Clock field returns the value (FREQ_SB/L2C_CLK_DIVIDER),

rounded to an integer, as shown in the example below. `FREQ_SB` is the System Bus clock frequency that has been entered on the **Hard IP Enables** tab. `I2C_CLK_DIVIDER` is a factor to be calculated as in the example below.

Master Clock calculation example:

- $\text{Divider} = \text{I2C_CLK_DIVIDER} = \text{FREQ_SB} / \text{I2C_CLK_FREQ}$
- $\text{Master Clock (Actual)} = \text{FREQ_SB} / \text{Divider Integer}$

For example, if `FREQ_SB` = 42.5 MHz and `I2C_BUS_PERF` = 400 kHz.

$\text{Divider} = 42500 / 400 = 106.25$

Therefore, `ROUND Divider Integer` = 106

$\text{Actual frequency} = 42500 / 106 = 400.9 \text{ kHz}$

The tool automatically calculates and displays the Actual I²C Master clock rate value based on the System Bus Frequency and Desired Master Clock rate selected.

In this example, if an `I2C_CLK_FREQ` of 400 kHz cannot be generated, you may use the actual value or change the System Bus clock frequency.

3.2.3. FIFO Mode

3.2.3.1. Enable FIFO Mode

Select this option to enable FIFO mode operation.

3.2.3.2. CLK Stretch Enable

Select this checkbox to enable Clock Stretching for FIFO mode. This setting may be used to ensure valid data is transmitted from the FIFO. If disabled, then overflow and underflow error flag must be monitored.

The Clock Stretch for Register mode is a separate control and can be enabled or disabled through the `CKSDIS` (bit2) inside `I2CCMDR` register. The `I2CCMDR` register can be directly accessed through the System Bus.

Refer to [Advanced CrossLinkPlus I2C Hardened IP Reference Guide \(FPGA-TN-02135\)](#) for more details.

3.2.3.3. RxFIFO Almost Full

Select the value from the drop down list to set the trigger for the RxFIFO Almost Full Flag. The maximum FIFO size of the RxFIFO is 32 bytes.

3.2.3.4. TxFIFO Almost Empty

Select the value from the drop down list to set the trigger for the TxFIFO Almost Empty Flag.

3.2.4. I²C Addressing

In the I²C Address box, you can choose between 7-bit or 10-bit addressing, and define the Hardened I²C address.

The lower two significant bits of the I²C address are fixed for each core. You can set the upper five significant bits in case of 7-bit addressing, or the upper eight significant bits in case of 10-bit addressing.

3.2.5. PMU Wake Options

This option is available for I2C0 only to enable the Wakeup port. The `WKUPEN` bit in the `I2CCR1` can be modified dynamically allowing the Wake Up function to be enabled or disabled.

3.2.5.1. Address Match

Select this checkbox to turn on the I²C wakeup on address match.

3.2.5.2. RxFIFO Almost Full

Select this checkbox to turn on the I²C wakeup on RxFIFO Almost Full Flag assertion.

3.3. I²C Interrupts

This section describes the I²C Interrupts tab of Lattice FPGA Module – I2C window, as shown in Figure 3.6.

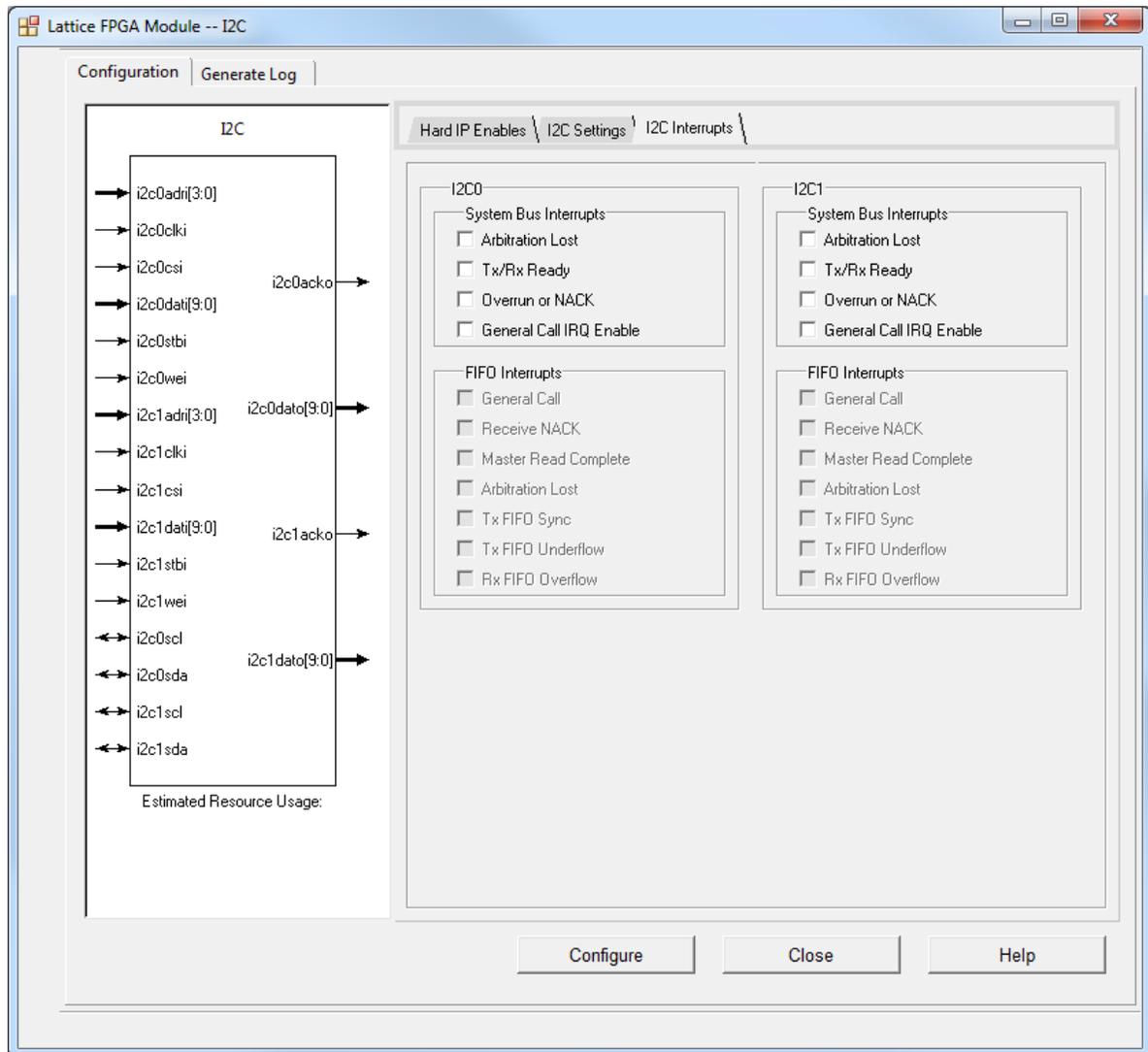


Figure 3.6. I²C Interrupts

3.3.1. System Bus Interrupts

System Bus interrupts enable the interrupt port when any of the interrupts are enabled. Refer to [Advanced CrossLinkPlus I2C Hardened IP Reference Guide \(FPGA-TN-02135\)](#) for more details on the interrupt registers described below.

3.3.1.1. Arbitration Lost

This interrupt indicates I²C lost arbitration. This interrupt is bit 3 (ARBL) of register I2CINTSR. When enabled, it indicates that ARBL is asserted. Writing 1 to this bit clears the interrupt.

This option can be changed dynamically by modifying bit ARBLEN in register I2CINTCR.

3.3.1.2. Tx/Rx Ready

This interrupt indicates that the I²C transmit data register (I2CTXDR) is empty or that the receive data register (I2CRXDR) is full. This interrupt is bit 2 (TRRDY) of register I2CINTSR. When enabled, it indicates that TRRDY is asserted. Writing 1 to this bit clears the interrupt.

This option can be changed dynamically by modifying bit TRRDYEN in register I2CINTCR.

3.3.1.3. Overrun or NACK

This interrupt indicates that the I2CRXDR received new data before the previous data. This interrupt is bit TROE of register I2CINTSR. When enabled, it indicates that TROE is asserted. Writing 1 to this bit clears the interrupt.

This option can be changed dynamically by modifying bit I2CINTSR in register I2CINTCR.

3.3.1.4. General Call IRQ Enable

This interrupt indicates that a general call has occurred. This interrupt is bit 0 (HGC) of register I2CINTSR. When enabled, it indicates that HGC is asserted. Writing 1 to this bit clears the interrupt.

This option can be changed dynamically by modifying bit HGCEN in register I2CINTCR.

3.3.2. FIFO Interrupts

3.3.2.1. General Call

This interrupt indicates that a general call has occurred. This interrupt is bit HGC of register I2CFIFOINTSR.

This bit can be changed by modifying bit HGCEN in register I2CFIFOINTCR.

3.3.2.2. Receive NACK

This interrupt indicates that a NACK has been received. This interrupt is bit RNACK of register I2CFIFOINTSR.

This bit can be changed by modifying bit RNACKEN in register I2CFIFOINTCR.

3.3.2.3. Master Read Complete

A transaction is considered complete when:

- the specified number of data bytes from the Slave have been received in the RX FIFO, or
- the Master terminates the read transaction before the specified number of data bytes received.

This is bit MRDCMPL of register I2CFIFOINTSR. This bit can be changed by modifying bit MRDCMPLEN in register I2CFIFOINTCR.

3.3.2.4. Arbitration Lost

This interrupt indicates I²C lost arbitration. This interrupt is bit ARBL of register I2CFIFOINTSR. When enabled, it indicates that ARBL is asserted. Writing 1 to this bit clears the interrupt.

This option can be changed dynamically by modifying bit ARBLEN in register I2CFIFOINTCR.

3.3.2.5. TX FIFO SYNC

This interrupt indicates I²C TXFIFO synchronization error. Synchronization error happens when there are back-to-back commands in the FIFO. The previous command is overwritten. This interrupt is bit TXSERR of register I2CFIFOINTSR.

This option can be changed dynamically by modifying bit TXSERREN in register I2CFIFOINTCR.

3.3.2.6. TX FIFO Underflow

This interrupt is mutually exclusive with clock stretching function. This interrupt is bit TXUNDERF of register I2CFIFOINTSR.

This option can be changed dynamically by modifying bit TXUNDERFEN in register I2CFIFOINTCR.

3.3.2.7. RX FIFO Overflow

This interrupt is mutually exclusive with clock stretching function. This interrupt is bit RXOVERF of register I2CFIFOINTSR.

This option can be changed dynamically by modifying bit RXOVERFEN in register I2CFIFOINTCR.

4. SB_I2C Hardened IP Macro Ports and Wrapper Connections

When the I²C Hardened IP is enabled, the necessary signals are included in the generated module. Table 4.1 lists the pins for the I²C hardened IP.

Table 4.1. Pins for I²C Hardened IP

Port Name	I ² C Wrapper Name	Pin Direction	Description
SBCSi	i2cXcsi*	Input	Chip select signal. Activates the IP to allow the System Bus or fabric interface to communicate with the IP.
SBCLKi	i2cXclki*	Input	System clock input.
SBWRI	i2cXwei*	Input	System read/write input. R=0, W=1
SBSTBi	i2cXstbi*	Input	System strobe signal. When asserted, indicates that the Slave component is selected.
SBADRI[3:0]	i2cXadri[3:0]*	Input	System bus control registers address.
SBDATI[9:0]	i2cXdati[9:0]*	Input	System data input [7:0] for register mode; [9:0] for FIFO mode.
SBDATO[9:0]	i2cXdato[9:0]*	Output	System data output [7:0] for register mode; [9:0] for FIFO mode.
SBACKo	i2cXacko*	Output	System acknowledgement
SBSRwo	i2cXsrdwr*	Output	Slave read/write signal. A 1 indicates a Slave transmitting (external Master receiving). A 0 means Slave receiving (external Master transmitting).
I2CPIRQ	i2cXi2cirq*	Output	Interrupt request output signal of the I ² C core – The intended use of this signal is for it to be connected to a Master controller (such as a microcontroller or state machine) and request an interrupt when a specific condition is met.
I2CPWKUP	i2cXi2cwkup*	Output	Wake-up signal – The signal is enabled only if the Wakeup Enable feature is set.
FIFO_RST	i2cXfiforst*	Input	Reset for the FIFO logic.
TXFIFO_AE	i2cXtxfifoe*	Output	TXFIFO almost empty status signal coming from TXFIFO, indicating user-defined almost empty threshold value is reached.
TXFIFO_E	i2cXtxfifoe*	Output	TXFIFO empty signal coming from TXFIFO.
TXFIFO_F	i2cXtxfifof*	Output	TXFIFO full signal.
RXFIFO_E	i2cXrxfifoe*	Output	RXFIFO is empty. It can be served as an active low DATA RDY signal.
RXFIFO_AF	i2cXrxfifof*	Output	RXFIFO almost full signal, indicating user-defined almost full threshold value is reached.
RXFIFO_F	i2cXrxfifof*	Output	RXFIFO full signal.
MRDCMPL	i2cXmrdcmpl*	Output	Master Read Complete – This is only valid for Master Read Mode. A transaction is considered complete when: <ul style="list-style-type: none"> The specified number of data bytes from the Slave has been received in the RX FIFO, or The Master terminates the read transaction before the specified number of data bytes received
I2CX_SCL*	i2cXscl*	Bi-directional	Open drain clock line of the I ² C core – The signal is an output if the I ² C core is performing a Master operation. The signal is an input for Slave operations. The <i>i2c1scl</i> signal (Right I ² C) uses dedicated I/O pin only. The <i>i2c0scl</i> signal (Left I ² C) can use any of the available GPIOs on the device.

Port Name	I ² C Wrapper Name	Pin Direction	Description
I2CX_SDA*	i2cXsda*	Bi-directional	Open drain data line of the I ² C core – The signal is an output when data is transmitted from the I ² C core. The signal is an input when data is received into the I ² C core. The <i>i2c1sda</i> signal (Right I ² C) uses dedicated I/O pin only. The <i>i2c0sda</i> signal (Left I ² C) can use any of the available GPIOs on the device.

***Note:** X indicates the I²C: X=0 for the hard I/O pin I²C; X=1 for the selectable I/O pin I²C.

5. I²C Usage Cases

The I²C usage cases are described below (Figure 5.1).

- The CrossLinkPlus device is used as I²C Master accessing Slave I²C devices.
 - A System Bus Master is implemented in the CrossLinkPlus logic.
 - I²C devices 1, 2, and 3 are all Slave devices.
 - The Master performs bus transactions to the I2C0 I²C controller to access external Slave I²C Device 1 on Bus A.
 - The Master performs bus transactions to the I2C1 I²C controller to access the external Slave I²C Device 2 or Device 3 on Bus B.
- External Master I²C device accessing Slave CrossLinkPlus I²C.
 - The I²C Devices 1, 2, and 3 are I²C Master devices.
 - The external Master I²C Device 1 on Bus A performs I²C memory cycles to access the I2C0 I²C controller using address yyyxxxx01.
 - The external Master I²C Device 2 or Device 3 on Bus B performs I²C memory cycles to access the I2C1 I²C user with the address yyyxxxx10.
 - The System Bus Master in the CrossLinkPlus fabric must manage data reception and transmission. The System Bus Master can use interrupts or polling techniques to manage data transfer, and to prevent data overrun conditions.

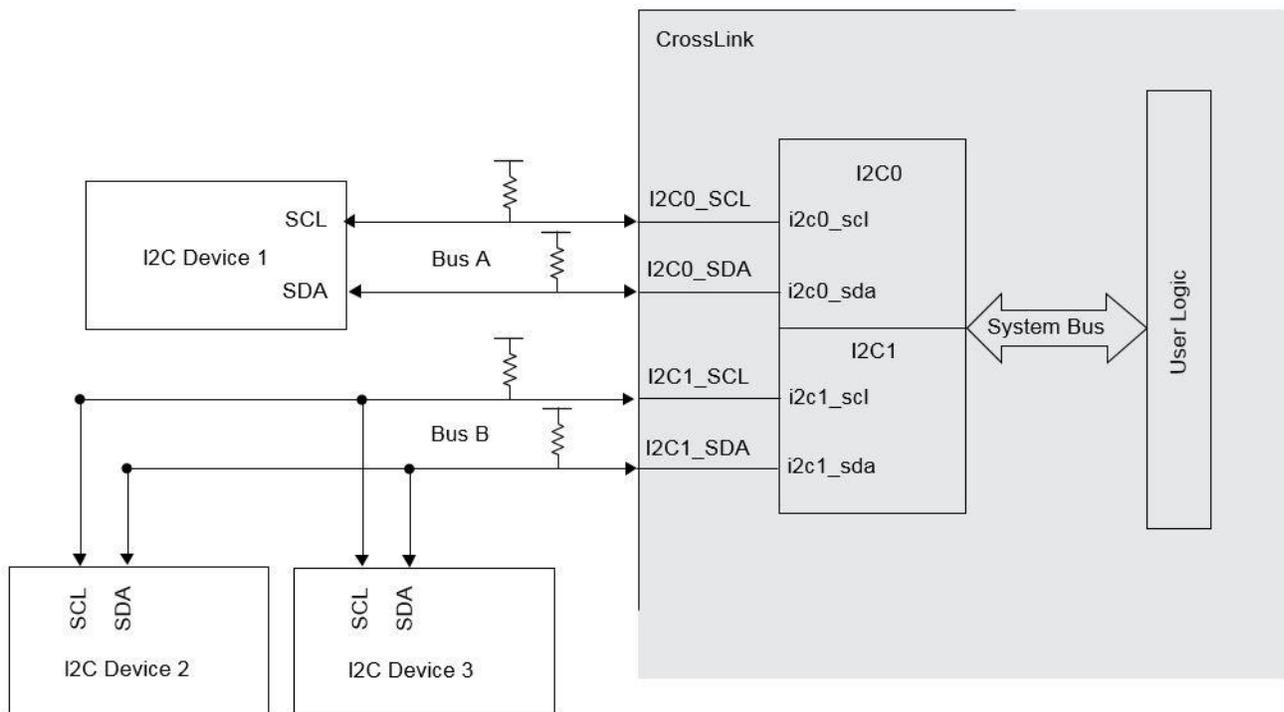


Figure 5.1. I²C Circuit

References

For more information, refer to the following documents:

- [CrossLinkPlus Family Data Sheet \(FPGA-DS-02054\)](#)
- [CrossLinkPlus High-Speed I/O Interface \(FPGA-TN-02102\)](#)
- [CrossLinkPlus Hardware Checklist \(FPGA-TN-02105\)](#)
- [CrossLinkPlus Programming and Configuration Usage Guide \(FPGA-TN-02103\)](#)
- [CrossLinkPlus sysCLOCK PLL/DLL Design and Usage Guide \(FPGA-TN-02109\)](#)
- [CrossLinkPlus sysI/O Usage Guide \(FPGA-TN-02108\)](#)
- [CrossLinkPlus Memory Usage Guide \(FPGA-TN-02110\)](#)
- [Power Management and Calculation for CrossLinkPlus Devices \(FPGA-TN-02111\)](#)
- [Advanced CrossLinkPlus I2C Hardened IP Reference Guide \(FPGA-TN-02135\)](#)

Technical Support Assistance

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

Revision History

Revision 1.0, August 2019

Section	Change Summary
All	Production release.



www.latticesemi.com