FTCSIRX100

MIPI CSI Receiver Controller

Block Data Sheet
Rev.: 1.12
Issue Date: July 2014

FARADAY

REVISION HISTORY

FTCSIRX100 Block Data Sheet

Date	Rev.	From	To
Apr. 2012	0.1	-	Original
Aug. 2012	1.0	-	• Updated Sections 4.1.4, 5.2.2, 5.4, 5.4.1, 5.4.6, 5.5.2, and 5.9 • Updated Table 2-7, Table 3-1, Table 3-4, Table 3-5, and Table 4-1 • Added Section 4.1.23
Oct. 2012	1.1	-	• Added Section 4.1.5, 5.6.5, 5.10, and 6.3 • Modified Table 2-4, Table 2-5, Table 2-7, Table 3-3, Table 3-4, Table 4-1, Table 4-4, Table 4-6, Table 4-27, Table 4-28, Table 4-33, Table 5-3, and Table 5-4. • Updated Sections 4.1, 5.2.2, and 5.9 • Updated Figure 5-9 and Figure 6-1
Mar. 2013	1.2	-	• Updated Section 5.2, Section 5.4, Section 5.6.2, and Section 5.10 • Updated Table 2-1., Table 2-4, Table 2-7, Table 3-1, Table 3-5, and Table 4-1 • Added Section 1.6 and Section 4.1.14 • Fixed typo
Apr. 2013	1.3	-	Updated the IP version to (1.1.0)
Jun. 2013	1.4	-	• Updated Table 2-4, Table 3-4, Table 4-4, Table 4-15, Table 4-19, and Table 4-28 • Updated Sections 3.1, 3.6, 4.1, 5.4.3, 5.4.6, 5.5.1, 5.6, 5.6.1, 5.6.2, 6.1, and 6.2
Jul. 2013	1.5	-	Updated the IP version to (1.2.0)
Nov. 2013	1.6	-	• Updated Table 2-3, Table 2-4, Table 2-6, Table 2-7, Table 2-9, Table 3-1, Table 3-4, Table 4-1, Table 4-25, and Table 4-29 • Updated Sections 1.3, 1.5, 1.6, 3.3, 3.4, 3.5, 4.1, 4.1.17, 4.1.19, 4.1.20, 4.1.21, 5.5.2, 5.6.1, 5.6.2, 5.9, and 5.10.1 • Added Table 2-12, Table 2-13, and Table 4-32 • Added Sections 4.1.15, 4.1.22, 4.1.25, 4.1.26, 4.1.27, 5.10.2, 5.11, and 5.12
Nov. 2013	1.7	-	Updated the IP version to (1.3.0)

Date	Rev.	From	To
Feb. 2014	1.8	-	• Updated the IP version to (1.4.0) • Added Section 4.1 • Updated Table 2-1, Table 2-3, Table 2-6, Table 4-5, Table 4-24, Table 4-26, Table 4-27, Table 4-28, Table 4-30, and Table 5-6 • Updated Chapter 2, Section 4.2.15, Section 4.2.19, Section 4.2.20, Section 5.5.3, Section 5.6.1.3, Section 5.9, and Section 5.11
May 2014	1.9	-	• Added Section 4.2.8 • Updated Figure 5-9 • Updated Sections 4.2.9 and 5.6.1 • Updated Table 2-4, Table 2-5, Table 2-7, Table 3-4, Table 4-1, Table 4-2, Table 4-27, and Table 5-3
Jun. 2014	1.10	-	• Updated Table 2-6, Table 3-1, Table 3-4, Table 3-5, Table 4-2, Table 4-24, and Table 4-36 • Updated Chapter 2 • Updated Sections 3.4, 3.6, 4.2.14, 4.2.22, 5.3, 5.5.2, 5.5.3, and 5.6.2 • Added Table 2-14, Table 2-15, and Table 2-17, Table 2-18, and Table 4-37, • Added Sections 5.5.1 • Added Figure 5-3, Figure 5-4, Figure 5-6, and Figure 5-8 • Fixed typo
Jun. 2014	1.11		• Updated the IP version to (1.6.0) • Supported the quad-pixel mode • Updated Table 3-4 and Table 3-5 • Updated Section 5.5.1.3
Jul. 2014	1.12		• Updated the IP version to (1.7.0) • Supported single RGB pixel transmitting by one pixel clock in the quad-pixel mode • Updated Table 2-6, Table 2-16, Table 2-17, Table 2-18, Table 3-4, Table 4-5, and Table 4-37 • Updated Chapter 2 • Updated Sections 4.2.4, 4.2.14, 4.2.22, 4.2.25, 5.3, 5.5.1.1, 5.5.1.2, 5.5.1.3, 5.5.2, 5.6.1.2, and 5.6.2 • Added Sections 5.5.2.4 and 5.5.2.5

Faraday Technology Corporation
No. 5, Li-Hsin Road III, Hsinchu Science Park, Hsinchu City, Taiwan 300, R.O.C.

Faraday's home page can be found at:
http://www.faraday-tech.com

TABLE OF CONTENTS

List of Tables

List of Figures

Chapter 1

Introduction

This chapter contains the following sections:

1.1 Version of the IP

IP release version: 1.7.0

1.2 Terminology

Terminology	Description
AP	Application Layer
CSI	Camera Serial Interface
DLW	Data Lane Width It is the value of the hardmacro "FTCSIRX100_LANE_NUM" which is described in Section 3.4.
DWORD	32-bit Data
DPI	Display Pixel Interface
ECC	Error Correcting Code
EoT	End of Transmission
HS-RX	High-Speed receiver (Low-swing differential)
LP-RX	Low-Power receiver (Large-swing single-end)
LSB	Least Significant Bit
ML	Media Access Layer
MSB	Most Significant Bit
PL	Protocol Layer
PPI	PHY Protocol Interface This is defined in MIPI Alliance Specification for D-PHY [MIPI01].
ULPS	Ultra-Low Power State
VC	Virtual Channel

1.3 Features

1.3.1 Specifications

- Compliant with MIPI Alliance Specification for Camera Serial Interface (CSI-2), Version 1.1
- Compliant with MIPI Alliance Standard for Display Pixel Interface (DPI-2), Version 2.00
- Compliant with MIPI Alliance Specification for D-PHY, Version 1.1
- Implements Logical PHY-Protocol Interface (PPI) according D-PHY, Version 1.1 Annex A
- Compliant with AMBA APB Protocol Specification, Version 2.0
- Follows I^{2}C Bus Specification, Version 2.1

1.3.2 D-PHY Interface

- Supports up to four data lanes
- Supports High-Speed RX (HS-RX) and Low-Power RX (LP-RX)
- Supports ULPS and Receive Trigger event in Escape mode
- Supports PPI HS-RX byte clock (RxByteClkHS) stopping after next clock of last valid data byte
- Supports data lane sequence swap

1.3.3 CSI Features

- Supports four virtual channels
- Supports horizontal line interleaving stream among virtual channels
- Supports maximum pixel of the horizontal line to be 4096 pixels for RAW format and 2048 pixels for RGB/YUV formats
- Supports data formats: YUV422 8-bit, YUV422 10-bit, RGB888, RGB565, RAW8, RAW10, RAW12, RAW14, Generic 8-bit Long Packet Data Types, Generic Short Packet Data Types, and User Defined Data Types
- Supports error logging
- Supports ECC and CRC checking
- Supports CRC checking turn-on/off
- Supports HS RX timeout detection
- Supports DPCM decoding

1.3.4 I^{2}C Features

- Supports 7-bit address mode, slave transmit mode, and slave receive mode
- Supports standard (100 Kbit/s) and fast (400 Kbit/s) modes

1.3.5 DPI Features

- Programmable DPI Sync Pulse Width
- Provides pixel FIFO threshold control

1.4 Block Diagram

Figure 1-1 is the functional block diagram of FTCSIRX100. This controller consists of three layers: ML, PL, and AP. ML is responsible for lane management at receive (Rx) side. The yellow bars show the clock domain partitions.

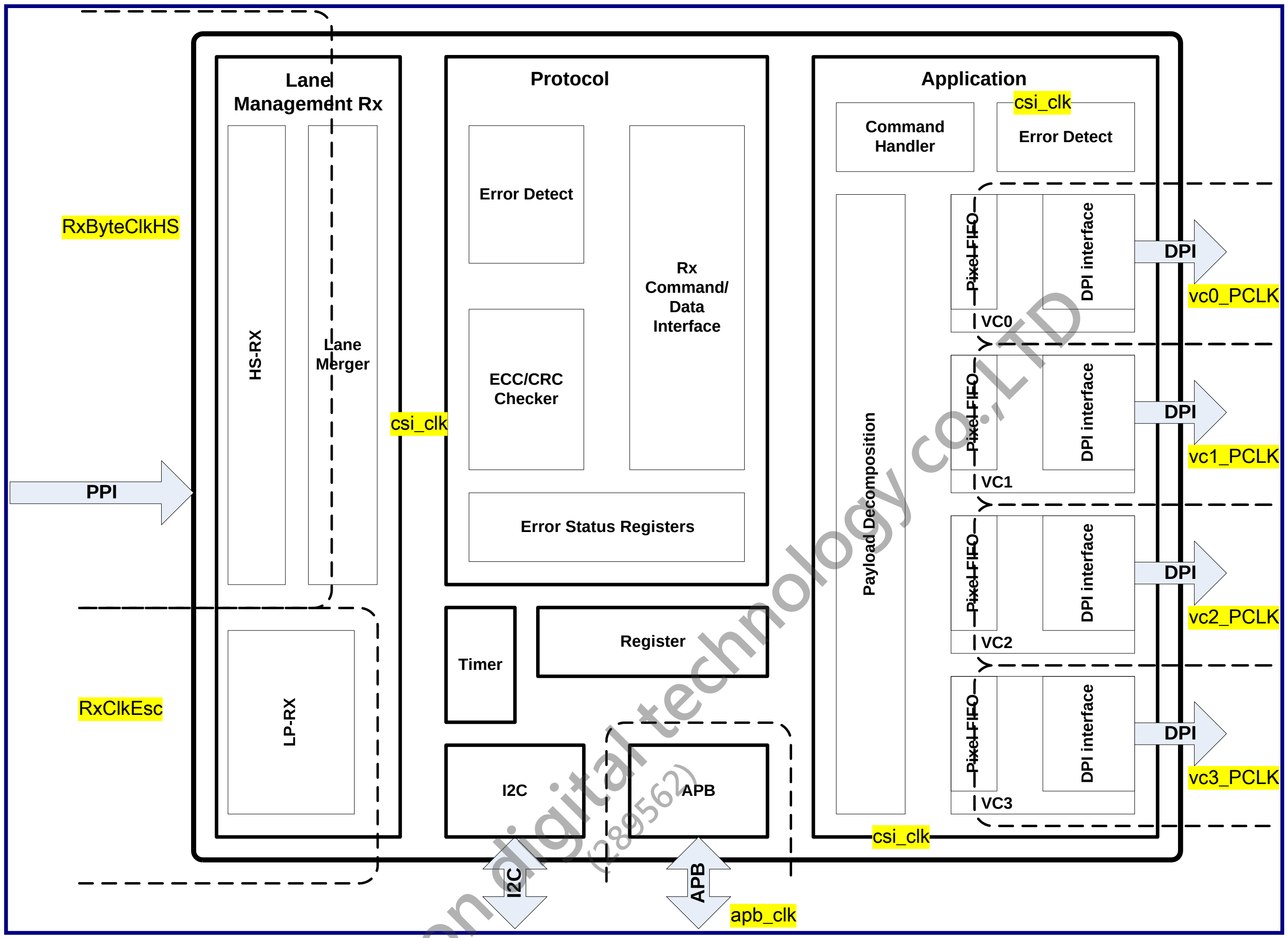

Figure 1-1. Functional Block Diagram

1.5 Overview

FTCSIRX100 is a high-speed and high-resolution interconnection for the CSI receiver. It is compliant with the Camera Serial Interface (CSI-2) [MIPI02] and supports the logical PHY Protocol Interface (PPI) of D-PHY [MIPI01]. It provides the Display Pixel Interface (DPI) [MIPI03] for the image processor. The supported data rate is up to 1.5 Gbps per lane and is scalable from one to four data lanes. The maximum throughput will be 6 Gbps when four data lanes are active in the High-Speed RX (HS-RX) mode. Various resolutions and pixel formats are supported. The DPI interface can connect up to four CSI virtual channels. FTCSIRX100 uses the layer architecture, where implements the lane management layer, protocol layer, and application layer. Depending on the hardware configuration, several clocks domains exist in FTCSIRX100: “RxByteClkHS” and “RxClkEsc” are defined in PPI of the D-PHY specification [MIPI01].

"RxByteClkHS" is the High-Speed receive byte clock. "RxClkEsc" is the Escape mode receive clock. "vc***n***_PCLK" is the pixel clock (PCLK) for each virtual channel ***n*** in DPI interface. "csi_clk" is the clock for the internal modules usage. "apb_clk" is the APB clock. The clock partition provides the flexible clock usage for power saving and can accept that the "RxByteClkHS" frequency is different from the pixel clock.

1.6 References

- [MIPI01] MIPI Alliance Specification for D-PHY, Version 1.1
- [MIPI02] MIPI Alliance Specification for Camera Serial Interface (CSI-2), Version 1.1
- [MIPI03] MIPI Alliance Standard for Display Pixel Interface (DPI-2), Version 2.00
- [APB] Compliant with AMBA APB Protocol Specification, Version 2.0
- [I2C] I^{2}C Bus Specification, Version 2.1

FARADAY

Chapter 2

External Signal Description

In this chapter, DLW (Data Lane Width) is the value of the hardmacro, "FTCSIRX100_LANE_NUM", which is described in Section 3.4. The I/O type of each signal is listed as "Input" or "Output". Signals with the I/O type "Output" are driven by FTCSIRX100. Signals with the I/O type "Input" are FTCSIRX100 inputs. In this chapter, the data lane#***n*** means the physical data lane#***n*** if it is not specified.

Table 2-1 lists and describes the clock and reset signals of FTCSIRX100.

Table 2-1. Clock and Reset Signals

Signal Name	Width (Bit)	I/O Type	Description
pwr_rst_n	1	Input	Power-On Reset for FTCSIRX100 This reset is low active and can reset FTCSIRX100 to the default state.
sys_rst_n	1	Input	System Reset for FTCSIRX100 This reset is low active and can reset FTCSIRX100 to the default state, except the internal registers, I^2C, APB and APB synchronization modules in FTCSIRX100.
apb_rst_n	1	Input	APB Reset PRESETn is defined in AMBA APB Protocol specification [APB]. This signal only exists when the micro, FTCSIRX100_APB, is defined.

Signal Name	Width (Bit)	I/O Type	Description
RxClkEsc	1	Input	Escape Mode Receive Clock This is the PPI RxClkEsc signal and connects to the PPI signal RxClkEsc of data lane#0. This clock is used to synchronize the received signals in the Escape mode. This signal exists when the hardmacro FTCSIRX100_LANE_SWAP is **not** defined.
RxClkEsc_p	DLW	Input	Escape Mode Receive Clock The signal RxClkEsc_p[***n***] is the PPI RxClkEsc signal which connects with the PPI RxClkEsc of the physical data lane#***n***. According to DLMR setting, FTCSIRX100 selects one signal of RxClkEsc_p as the clock (PPI RxClkEsc for the logical data lane#0). This clock is used to synchronize the received signals in the Escape mode. This bus exists only when the hardmacro FTCSIRX100_LANE_SWAP is defined.
RxByteClkHS	1	Input	High-Speed Receive Byte Clock This clock is used to synchronize the signals at PPI in the High-Speed receive clock domain. FTCSIRX100 treats the high speed receive signals of all data lanes that are synchronous with this byte clock.
apb_clk	1	Input	APB Clock PCLK is defined in AMBA APB Protocol specification [APB]. This clock only exists when the micro, FTCSIRX100_APB, is defined.
csi_clk	1	Input	CSI Clock This is a local clock in FTCSIRX100.
vc0_PCLK	1	Input	DPI VC0 Pixel Clock Pixel clock for the VC0 display interface (DPI).
vc1_PCLK	1	Input	DPI VC1 Pixel Clock Pixel clock for the VC1 display interface (DPI). This clock exists only when VC1 exists.
vc2_PCLK	1	Input	DPI VC2 Pixel Clock Pixel clock for the VC2 display interface (DPI). This clock exists only when VC2 exists.
vc3_PCLK	1	Input	DPI VC3 Pixel Clock Pixel clock for the VC3 display interface (DPI). This clock exists only when VC3 exists.

Table 2-2 lists and describes the PPI High-Speed receive signals connected to FTCSIRX100. These are the RxByteClkHS clock domain signals. For the high-speed data transmission of the multiple data lanes, the first byte of each data lane shall be valid at the same RxByteClkHS clock cycle. In this table, data lane means the physical data lane.

Table 2-2. PPI High-Speed Receive Signals

Signal Name	Width (Bit)	I/O Type	Description
RxDataHS	DLW*8	Input	High-Speed Receive Data The signal connected to RxDataHS[0] is received first in the D-PHY serial interface. Data are latched at the rising edges of RxByteClkHS. RxDataHS[7:0] is used for data lane#0, RxDataHS[15:8] is used for data lane#1, RxDataHS[23:16] is used for data lane#2, and RxDataHS[31:24] is used for data lane#3. If the corresponding lane is disabled, the input signals should be tied to '0'. The lane number can be scalable by using the hard macro definition and RxDataHS bus width can be 32-bit, 24-bit, 16-bit, or 8-bit.
RxValidHS	DLW	Input	High-Speed Receive Data Valid This active-high signal indicates that D-PHY is driving data to FTCSIRX100 on the RxDataHS output. There is no "RxReadyHS" signal, and the protocol is expected to capture RxDataHS at every rising edge of RxByteClkHS when RxValidHS is asserted. There is no provision for the protocol to slow down ("throttle") the receive data. If the corresponding lane is disabled, the input signals should be tied to '0'. The lane number can be scalable by using the hard macro definition and RxValidHS bus width can be 4-bit, 3-bit, 2-bit, or 1-bit. RxValidHS[***n***] represents RxValidHS of data lane#***n***.
RxActiveHS	DLW	Input	High-Speed Reception Active This active-high signal indicates that D-PHY is actively receiving a High-Speed transmission from the lane interconnect. If the corresponding lane is disabled, the input signals should be tied to '0'. The lane number can be scalable by using the hard macro definition and RxActiveHS bus width can be 4-bit, 3-bit, 2-bit, or 1-bit. RxActiveHS[***n***] represents RxActiveHS of data lane#***n***.
RxSyncHS	DLW	Input	Receiver Synchronization Observed This active-high signal indicates that D-PHY has seen an appropriate synchronization event. In a typical High-Speed transmission, RxSyncHS will be high for one cycle of RxByteClkHS at the beginning of a High-Speed transmission when RxActiveHS is asserted. If the corresponding lane is disabled, the input signals should be tied to '0'. The lane number can be scalable by using the hard macro definition and RxSyncHS bus width can be 4-bit, 3-bit, 2-bit, or 1-bit. RxSyncHS[***n***] represents RxSyncHS of data lane#***n***.

Table 2-3 lists and describes the receive signals in the PPI escape mode, which are connected to FTCSIRX100. FTCSIRX100 uses the PPI signal, UlpsActiveNot, to record the ULP state of the D-PHY data lane.

Table 2-3. PPI Escape Mode Receive Signals

Signal Name	Width (Bit)	I/O Type	Description
RxTriggerEsc	4 or 4*DLW	Input	Escape Mode Receive Trigger 0-3 This is the PPI RxTriggerEsc signal for the logical data lane#0. If the hardmacro, FTCSIRX100_LANE_SWAP, is not defined, the width will be 4 bits and it connects to RxTriggerEsc of the data lane#0. Otherwise, the width is 4 * DLW and RxTriggerEsc[3+***n****4:***n****4] connects to the physical data lane#***n***.
RxUlpsEsc	1 or DLW	Input	Escape Ultra-Low Power (Receive) Mode on Data Lane This is the PPI RxUlpsEsc signal for the logical data lane#0. If the hardmacro, FTCSIRX100_LANE_SWAP, is not defined, the width will be 1 bit and it connects to RxUlpsEsc of the data lane#0. Otherwise, the width is DLW and RxUlpsEsc[***n***] connects to the physical data lane#***n***.

Table 2-4 lists and describes the PPI control signals connected to FTCSIRX100. In this table, data lane means the physical data lane.

Table 2-4. PPI Control Signals

Signal Name	Width (Bit)	I/O Type	Description
ForceRxmode	1	Output	Force Lane Module into Receive Mode/Wait for Stop State This is the PPI ForceRxmode signal. FTCSIRX100 only outputs the single ForceRxmode signal for all data lanes. This signal is synchronous with csi_clk and is glitch-free. The period of the assertion is controlled by the ITR register (Address = 0x12 ~ 0x13) . When this signal asserts, FTCSIRX100 does not drop the data in the high speed receive interface of PPI.
Stopstate	DLW	Input	Data Lane is at Stop State This is the PPI Stopstate signal of the data lane. This active-high signal indicates that the data lane module is currently at the Stop state. The lane number can be scalable by using the hard macro definition and Stopstate bus width can be 4-bit, 3-bit, 2-bit, or 1-bit. Stopstate[***n***] represents the Stopstate of data lane ***n***.

Signal Name	Width (Bit)	I/O Type	Description
ppi_Enable	DLW	Input	Enable Lane Module This is the data lane enable signal. The data lane number can be scalable by using the hard macro definition and ppi_Enable bus width can be 4-bit, 3-bit, 2-bit, or 1-bit. This hardmacro definition is described in Section 3.4. "ppi_Enable[***n***]" represents Enable of data lane ***n***. ppi_Enable value cannot be changed after pwr_rst_n is released. For data lane enable, please refer to Sections 5.10 and 5.11 for details.
UlpsActiveNot	DLW	Input	ULP State (Not) Active for Data Lane This is the PPI UlpsActiveNot signal of the data lane. The lane number can be scalable by using the hard macro definition and UlpsActiveNot bus width can be 4-bit, 3-bit, 2-bit, or 1-bit. UlpsActiveNot[***n***] represents the UlpsActiveNot of data lane ***n***.
ClkStopstate	1	Input	Clock lane is in Stop State This is the PPI Stopstate signal of the clock lane.
ClkUlpsActiveNot	1	Input	ULP State (Not) Active for Clock Lane This is the PPI UlpsActiveNot signal of the clock lane.
RxUlpsClkNot	1	Input	Receive Ultra-Low Power State on Clock Lane This is the PPI RxUlpsClkNot signal of the clock lane.
ftcsirx100_ppi_Enable	4	Output	Enable PHY Lane Module This signal drives the lane enable for the physical data lane of PHY. Please refer to Section 5.10 for details.

Table 2-5 lists and describes the PPI error signals of FTCSIRX100. ErrSotHS, ErrSotSyncHS, and ErrEotSyncHS are synchronous with the RxByteClkHS clock. In this table, data lane means the physical data lane.

Table 2-5. PPI Error Signals

Signal Name	Width (Bit)	I/O Type	Description
ErrSotHS	DLW	Input	Start-of-Transmission (SoT) Error If the High-Speed SoT leader sequence is corrupted in a way that proper synchronization can still be achieved, this active-high signal will be asserted for one cycle of RxByteClkHS. This is considered to be a "soft error" in the leader sequence and the payload data are reduced. The lane number can be scalable by using the hard macro definition and ErrSotHS bus width can be 4-bit, 3-bit, 2-bit, or 1-bit. ErrSotHS[***n***] represents the ErrSotHS of data lane ***n***.

Signal Name	Width (Bit)	I/O Type	Description
ErrSotSyncHS	DLW	Input	Start-of-Transmission Synchronization Error If the High-Speed SoT leader sequence is corrupted in a way that proper synchronization cannot be expected, this active-high signal will be asserted for one cycle of RxByteClkHS. The lane number can be scalable by using the hard macro definition and ErrSotSyncHS bus width can be 4-bit, 3-bit, 2-bit, or 1-bit. ErrSotSyncHS[***n***] represents ErrSotSyncHS of data lane ***n***.
ErrEotSyncHS	DLW	Input	End-of-Transmission Synchronization Error This active-high signal indicates when the last bit of a transmission doesn't match the byte boundary. This signal can only be effective when EoT detecting LP-11 and is latched at the rising edge of RxByteClkHS. The lane number can be scalable by using the hard macro definition and ErrEotSyncHS bus width can be 4-bit, 3-bit, 2-bit, or 1-bit. ErrEotSyncHS[***n***] represents the ErrEotSyncHS of data lane ***n***.
ErrEsc	DLW	Input	Escape Entry Error If an unrecognized escape entry command is received, this active-high signal will be asserted and remains asserted until the next change at the line state. The lane number can be scalable by using the hard macro definition and ErrEsc bus width can be 4-bit, 3-bit, 2-bit, or 1-bit. ErrEsc[***n***] represents the ErrEsc of data lane ***n***.
ErrControl	DLW	Input	Receiver Data Lane Control Error This active-high signal will be asserted when an incorrect line state sequence is detected at the receiver data lane. For example, if a turn-around request or escape-mode request is immediately followed by a Stop state, instead of the required Bridge state, this signal will be asserted and remains asserted until the next change at the line state. The lane width is scalable by using the hard macro definition and can be 4-bit, 3-bit, 2-bit, or 1-bit. ErrControl[***n***] represents the ErrControl of data lane ***n***.
ErrContentionLP0	DLW	Input	LP0 Contention Error This is PPI ErrContentionLP0 signal. The lane width is scalable by using the hard macro definition and can be 4-bit, 3-bit, 2-bit, or 1-bit. ErrContentionLP0[***n***] represents the ErrContentionLP0 of data lane ***n***.
ErrContentionLP1	DLW	Input	LP1 Contention Error This is PPI ErrContentionLP1 signal. The lane width is scalable by using the hard macro definition and can be 4-bit, 3-bit, 2-bit, or 1-bit. ErrContentionLP1[***n***] represents the ErrContentionLP1 of data lane ***n***.

Table 2-6 lists and describes the DPI signals and the extend signals for each virtual channel. All signals are synchronous with the vc***n***_PCLK clock. The pixel data bus mapping is shown in Table 2-10 through Table 2-16.

In Table 2-6, '***n***' means the virtual channel number and the supported image packets are YUV422 8-bit, YUV422 10-bit, RGB888, RGB565, RAW8, RAW10, RAW12, and RAW14.

Table 2-6. DPI and Extend Signals

Signal Name	**Width (Bit)**	**I/O Type**	**Description**
dpi_vc***n***_Vsync	1	Output	Vertical Sync(Vsync) for VC***n*** Vertical synchronization timing signal for the virtual channel ***n***. It asserts after receiving a VC***n*** Frame Start packet. The deassertion is controlled by VSCR.VSPC***n*** bit (Address = 0x05).This is a low-active signal. It doesn't exist when VC***n*** doesn't exist.
dpi_vc***n***_Hsync	1	Output	Horizontal Sync(Hsync) for VC***n*** Horizontal synchronization timing signal for the virtual channel ***n***. This is a low-active signal. This signal doesn't exist when VC***n*** doesn't exist. Depending on the setting of CR.HEG, FTCSIRX100 asserts this signal for receiving supported VC***n*** image data packet, user defined byte-based packet (Packet type = 0x30 ~ 0x37) and generic 8-bit long packet (packet type = 0x12 ~ 0x17). The low-active pulse width is controlled by HSTR***n*** (Address = 0x15 + ***n***).
dpi_vc***n***_DE	1	Output	Data Enable(DE) for VC***n*** The assertion of this signal is used to indicate the valid pixels at dpi_vc***n***_D bus for the virtual channel ***n***. This is a high-active signal. This signal doesn't exist when VC***n*** doesn't exist.
dpi_vc***n***_D	24,28, or 56	Output	Pixel Data(D) for VC***n*** Pixel data for VC***n.*** For the single-pixel mode, the bus width is 24-bit. For the dual-pixel mode, the bus width is 28-bit. For the quad-pixel mode, the bus width is 56-bit. This bus doesn't exist when VC***n*** doesn't exist.
csi_vc***n***_lnum	16	Output	Line Number for VC***n*** When csi_ vc***n***_lineval is asserted, this bus indicates the line number captured from the latest line start packet for VC***n***. The default value is zero before getting the first Line Start packet for VC***n***. The value of this bus can be cleared by software reset or hardware reset (pwr_rst_n or sys_rst_n is active). This bus doesn't exist when VC***n*** doesn't exist.

Signal Name	Width (Bit)	I/O Type	Description
csi_vc***n***_lineval	1	Output	Line Valid for VC***n*** This signal asserts as high after receiving a Line Start packet for VC***n*** and deasserts after receiving a Line End packet for VC***n***. This is a high-active signal. The value of this bus can be cleared by software reset or hardware reset (pwr_rst_n or sys_rst_n is active). This signal doesn't exist when VC***n*** doesn't exist.
csi_vc***n***_fnum	16	Output	Frame Number for VC***n*** This bus indicates the frame number captured from the latest Frame Start packet for VC***n***. The default value is zero before getting the first Frame Start packet for VC***n***. The value of this bus can be cleared by software reset or hardware reset (pwr_rst_n or sys_rst_n is active). This bus doesn't exist when VC***n*** doesn't exist.
csi_vc***n***_pt	6	Output	Image Data Packet Type for VC***n*** When dpi_vc***n***_DE is high, this bus indicates the received image data packet type for VC***n***, excluding generic and user defined packets. This bus doesn't exist when VC***n*** doesn't exist.
csi_vc***n***_gt	6	Output	Generic Packet Data Type for VC***n*** When csi_vc***n***_gt is high, this bus indicates the received generic short/long and user defined packets for VC***n***. This bus doesn't exist when VC***n*** doesn't exist.
csi_vc***n***_gDE	1	Output	Data Enable for VC***n*** Generic Packet Data The assertion of this signal is used to indicate the valid generic packet data at dpi_vc***n***_D bus and Packet Data type at csi_vc***n***_gt bus for the virtual channel ***n***. This is a high-active signal. This signal doesn't exist when VC***n*** doesn't exist.

Table 2-7 lists the miscellaneous signals, including interrupt, error reporting, and data checking signals. The signals listed in Table 2-7 are synchronous with csi_clk clock.

Table 2-7. Miscellaneous Signals

Signal Name	Width (Bit)	I/O Type	Description
csi_rx_int_r	1	Output	FTCSIRX100 Interrupt FTCSIRX100 will assert this interrupt when either of the interrupt status (INTSTS, Address = 0x33) is set and its corresponding interrupt enable (INTER, Address = 0x3C) is also active. This interrupt will be deasserted when none of the interrupt status is set with its corresponding interrupt enable = 1. This signal is the flip-flop output and high active.

Signal Name	Width (Bit)	I/O Type	Description
csi_err_r	16	Output	CSI Error Report Bits This bus reflects the value of Error Report Bits which are defined in the CSI Error Report Registers (Address = 0x30 and 0x31). “csi_err_r[7:0]” maps to CSIERR0 register (Address = 0x30) Bit#7~0 and “csi_err_r[15:8]” maps to CSIERR1 register (Address = 0x31) Bit#7~0.
csi_sw_flag_r	1	Output	Software Flag This signal reflects the value of ECR.SF (Address = 0x06). This signal is synchronous with the csi_clk clock and glitch-free.
csi_chk_req	1	Input	PPI Interface Data Checking Test Request This signal only exists when the macro, FTCSIRX100_CHK_DATA, is defined. This input signal must be level signal and glitch-free. When this signal is high, this PPI Interface Data Checking test mode is triggered. Please refer to Section 5.9 for details.
csi_chk_en_sych	1	Output	Enable PPI Interface Data Checking Test Mode This signal only exists when the macro, FTCSIRX100_CHK_DATA, is defined. When this signal is high, this PPI Interface Data Checking test mode is active. Otherwise, it is disabled. This signal is asserted when either input signal csi_chk_req or the value of register ECR.PCE (Address = 0x06) is high. Please refer to Section 5.9 for details.
csi_chk_pass_r	1	Output	PPI Interface Data Checking Pass This signal only exists when the macro, FTCSIRX100_CHK_DATA, is defined. The assertion of this flag indicates that this data checking has passed. Please refer to Section 5.9 for details.

Table 2-8 lists and describes the I^{2}C interface signals. FTCSIRX100 contains the I^{2}C slave. The characteristics of the SDA and SCL I/O stages shall meet I^{2}C bus specification [I2C].

Table 2-8. I^{2}C Interface Signals

Signal Name	Width (Bit)	I/O Type	Description
csi_scl_in	1	Input	I^{2}C Clock This is the serial clock (SCL) line in of I^{2}C and shall be the Schmitt-trigger input.
csi_sda_in	1	Input	I^{2}C Data Input This is the serial data (SDA) line in of I^{2}C and shall be the Schmitt-trigger input.

Signal Name	Width (Bit)	I/O Type	Description
csi_devaddr	7	Input	I^{2}C Device Address It represents the I^{2}C slave address of the device. At I^{2}C bus, FTCSIRX100 can compare the first seven bits after the START condition from the I^{2}C master with this slave address. If the address matches, the device treats itself addressed by the master as a slave-receiver or slave-transmitter.
csi_sda_out	1	Output	I^{2}C Data Output This is the serial data (SDA) line out of I^{2}C.
csi_sda_oe	1	Output	I^{2}C Data Output Enable The output enable for the serial data line
csi_gsvalue	5	Input	Glitch Suppression Value This input shall be hardwired. The bus determines the pulse width of the signals csi_scl_in and csi_sda_in spike which are suppressed by the FTCSIRX100 digital core logic. The pulse width = the value of this bus * the period of csi_clk clock. The pulse width, function and value setting have the limitation. Please refer to section 5.8 for details. When this bus is hardwired as zero, FTCSIRX100 doesn't perform the spike suppression.

Table 2-9 indicates the APB signals. All signals are synchronous with APB PCLK (apb_clk) and only exist when the macro, FTCSIRX100_APB, is defined.

Table 2-9. APB Interface Signals

Signal Name	Width (Bit)	I/O Type	Description
apb_clk	1	Input	APB Clock This is the PCLK which is defined in AMBA APB Protocol specification [APB].
apb_rst_n	1	Input	APB Reset This is PRESETn which is defined in AMBA APB Protocol specification [APB].
apb_addr	32	Input	APB Address Bus This is PADDR which is defined in AMBA APB Protocol specification [APB].
apb_sel	1	Input	APB Slave Select This is PSELx which is defined in AMBA APB Protocol specification [APB]. It indicates that the slave device is selected and a data transfer is required.

Signal Name	Width (Bit)	I/O Type	Description
apb_enable	1	Input	APB Enable This is the PENABLE which is defined in AMBA APB Protocol specification [APB]. It indicates the second and subsequent cycles of an APB transfer.
apb_write	1	Input	APB Write This is PWRITE which is defined in AMBA APB Protocol specification [APB]. It indicates an APB write access when HIGH and an APB read access when LOW.
apb_wdata	32	Input	APB Write Data This is PWDATA which is defined in AMBA APB Protocol specification [APB].
apb_strb	4	Input	APB Write Strobe This is PSTRB which is defined in AMBA APB Protocol specification [APB]. There is one write strobe for each eight bits of the write data bus. apb_strb[n] corresponds to apb_wdata [(8n + 7): (8n)]. Write strobes must not be active during a read transfer.
apb_ready	1	Output	APB Ready This is PREADY which is defined in AMBA APB Protocol specification [APB]. This signal is used to extend an APB transfer.
apb_rdata	32	Output	APB Read Data This is PRDATA which is defined in AMBA APB Protocol specification [APB]. FTCSIRX100 drives this bus during read cycles when apb_write is LOW.
apb_slverr	1	Output	APB Slave Error This is PSLVERR which is defined in AMBA APB Protocol specification [APB]. This signal indicates a transfer failure.

Table 2-10 through Table 2-18 indicate the pixel bits mapping to the DPI D bus. Each virtual channel D bus pin, "D***m***", means the output signal of FTCSIRX100 IP pin, "dpi_vcn_D[***m***]", and n is the virtual channel number. By setting the MCR Register (Address = 0x1C), the RAW pixel output can align to LSB or MSB.

Table 2-10. Single-pixel Mode DPI Data Pin Mapping for RAW (Align to LSB)

Signal	RAW8	RAW10	RAW12	RAW14
D23	0	0	0	0
D22	0	0	0	0
D21	0	0	0	0
D20	0	0	0	0
D19	0	0	0	0
D18	0	0	0	0
D17	0	0	0	0
D16	0	0	0	0
D15	0	0	0	0
D14	0	0	0	0
D13	0	0	0	R13
D12	0	0	0	R12
D11	0	0	R11	R11
D10	0	0	R10	R10
D9	0	R9	R9	R9
D8	0	R8	R8	R8
D7	R7	R7	R7	R7
D6	R6	R6	R6	R6
D5	R5	R5	R5	R5
D4	R4	R4	R4	R4
D3	R3	R3	R3	R3
D2	R2	R2	R2	R2
D1	R1	R1	R1	R1
D0	R0	R0	R0	R0

Table 2-11. Single-pixel Mode DPI Data Pin Mapping for RAW (Align to MSB)

Signal	RAW8	RAW10	RAW12	RAW14
D23	0	0	0	0
D22	0	0	0	0
D21	0	0	0	0
D20	0	0	0	0
D19	0	0	0	0
D18	0	0	0	0
D17	0	0	0	0
D16	0	0	0	0
D15	0	0	0	0
D14	0	0	0	0
D13	R7	R9	R11	R13
D12	R6	R8	R10	R12
D11	R5	R7	R9	R11
D10	R4	R6	R8	R10
D9	R3	R5	R7	R9
D8	R2	R4	R6	R8
D7	R1	R3	R5	R7
D6	R0	R2	R4	R6
D5	0	R1	R3	R5
D4	0	R0	R2	R4
D3	0	0	R1	R3
D2	0	0	R0	R2
D1	0	0	0	R1
D0	0	0	0	R0

Table 2-12. Dual-pixel Mode DPI Data Pin Mapping for RAW (Align to LSB)

Signal	RAW8	RAW10	RAW12	RAW14
D27	0	0	0	Re_13
D26	0	0	0	Re_12
D25	0	0	Re_11	Re_11
D24	0	0	Re_10	Re_10
D23	0	Re_9	Re_9	Re_9
D22	0	Re_8	Re_8	Re_8
D21	Re_7	Re_7	Re_7	Re_7
D20	Re_6	Re_6	Re_6	Re_6

Signal	RAW8	RAW10	RAW12	RAW14
D19	Re_5	Re_5	Re_5	Re_5
D18	Re_4	Re_4	Re_4	Re_4
D17	Re_3	Re_3	Re_3	Re_3
D16	Re_2	Re_2	Re_2	Re_2
D15	Re_1	Re_1	Re_1	Re_1
D14	Re_0	Re_0	Re_0	Re_0
D13	0	0	0	Ro_13
D12	0	0	0	Ro_12
D11	0	0	Ro_11	Ro_11
D10	0	0	Ro_10	Ro_10
D9	0	Ro_9	Ro_9	Ro_9
D8	0	Ro_8	Ro_8	Ro_8
D7	Ro_7	Ro_7	Ro_7	Ro_7
D6	Ro_6	Ro_6	Ro_6	Ro_6
D5	Ro_5	Ro_5	Ro_5	Ro_5
D4	Ro_4	Ro_4	Ro_4	Ro_4
D3	Ro_3	Ro_3	Ro_3	Ro_3
D2	Ro_2	Ro_2	Ro_2	Ro_2
D1	Ro_1	Ro_1	Ro_1	Ro_1
D0	Ro_0	Ro_0	Ro_0	Ro_0

Note: "Ro" denotes the first pixel; "Re" denotes the second pixel. FTCSIRX100 receives the pixel "Ro" prior to the pixel "Re" from PPI.

Table 2-13. Dual-pixel Mode DPI Data Pin Mapping for RAW (Align to MSB)

Signal	RAW8	RAW10	RAW12	RAW14
D27	Re_7	Re_9	Re_11	Re_13
D26	Re_6	Re_8	Re_10	Re_12
D25	Re_5	Re_7	Re_9	Re_11
D24	Re_4	Re_6	Re_8	Re_10
D23	Re_3	Re_5	Re_7	Re_9
D22	Re_2	Re_4	Re_6	Re_8
D21	Re_1	Re_3	Re_5	Re_7
D20	Re_0	Re_2	Re_4	Re_6
D19	0	Re_1	Re_3	Re_5
D18	0	Re_0	Re_2	Re_4
D17	0	0	Re_1	Re_3

Signal	RAW8	RAW10	RAW12	RAW14
D16	0	0	Re_0	Re_2
D15	0	0	0	Re_1
D14	0	0	0	Re_0
D13	Ro_7	Ro_9	Ro_11	Ro_13
D12	Ro_6	Ro_8	Ro_10	Ro_12
D11	Ro_5	Ro_7	Ro_9	Ro_11
D10	Ro_4	Ro_6	Ro_8	Ro_10
D9	Ro_3	Ro_5	Ro_7	Ro_9
D8	Ro_2	Ro_4	Ro_6	Ro_8
D7	Ro_1	Ro_3	Ro_5	Ro_7
D6	Ro_0	Ro_2	Ro_4	Ro_6
D5	0	Ro_1	Ro_3	Ro_5
D4	0	Ro_0	Ro_2	Ro_4
D3	0	0	Ro_1	Ro_3
D2	0	0	Ro_0	Ro_2
D1	0	0	0	Ro_1
D0	0	0	0	Ro_0

Note: "Ro" denotes the first pixel, "Re" denotes the second pixel. FTCSIRX100 will send the pixel "Ro" prior to the pixel "Re" to PPI.

When FTCSIRX100_QUAD_PIXEL is defined, the bus width of dpi_vc***n***_D is 56-bit. Assume that FTCSIRX100 outputs "R0", "R1", "R2", and "R3" pixels at the same vc***n***_PCLK clock, "R0" is the first pixel, "R1" is the second one, "R2" is the third one, and "R3" is the fourth pixel.

Table 2-14. Quad-pixel Mode DPI Data Pin Mapping for RAW (Align to MSB)

Signal	RAW8	RAW10	RAW12	RAW14
dpi_vc***n***_D[55:42]	{R3[7:0], 0b000000}	{R3[9:0], 0b0000}	{R3[11:0], 0b00}	R3[13:0]
dpi_vc***n***_D[41:28]	{R2[7:0], 0b000000}	{R2[9:0], 0b0000}	{R2[11:0], 0b00}	R2[13:0]
dpi_vc***n***_D[27:14]	{R1[7:0], 0b000000}	{R1[9:0], 0b0000}	{R1[11:0], 0b00}	R1[13:0]
dpi_vc***n***_D[13:0]	{R0[7:0], 0b000000}	{R0[9:0], 0b0000}	{R0[11:0], 0b00}	R0[13:0]

Table 2-15. Quad-pixel Mode DPI Data Pin Mapping for RAW (Align to LSB)

Signal	RAW8	RAW10	RAW12	RAW14
dpi_vc***n***_D[55:42]	{0b000000, R3[7:0]}	{0b0000, R3[9:0]}	{0b00, R3[11:0]}	R3[13:0]
dpi_vc***n***_D[41:28]	{0b000000, R2[7:0]}	{0b0000, R2[9:0]}	{0b00, R2[11:0]}	R2[13:0]

Signal	RAW8	RAW10	RAW12	RAW14
dpi_vc***n***_D[27:14]	{0b000000, R1[7:0]}	{0b0000, R1[9:0]}	{0b00, R1[11:0]}	R1[13:0]
dpi_vc***n***_D[13:0]	{0b000000, R0[7:0]}	{0b0000, R0[9:0]}	{0b00, R0[11:0]}	R0[13:0]

Table 2-16 shows the DPI data pin mapping for transmitting a single RGB pixel by one pixel clock (Please refer to Section 5.5.2.4) or a couple YUV pixels by two pixel clocks (Please refer to Section 5.5.2.2). D24 ~ D27 are always zero for YUV and RGB.

Table 2-16. DPI Data Pin Mapping for YUV (2 pixels by 2 clocks) and RGB (Single pixel by single clock)

Signal	YUV422 8-bit	YUV422 10-bit	RGB565	RGB888
D23	0	0	0	R7
D22	0	0	0	R6
D21	0	Y9	0	R5
D20	0	Y8	R4	R4
D19	Y7	Y7	R3	R3
D18	Y6	Y6	R2	R2
D17	Y5	Y5	R1	R1
D16	Y4	Y4	R0	R0
D15	Y3	Y3	0	G7
D14	Y2	Y2	0	G6
D13	Y1	Y1	G5	G5
D12	Y0	Y0	G4	G4
D11	0	0	G3	G3
D10	0	0	G2	G2
D9	0	U9/V9	G1	G1
D8	0	U8/V8	G0	G0
D7	U7/V7	U7/V7	0	B7
D6	U6/V6	U6/V6	0	B6
D5	U5/V5	U5/V5	0	B5
D4	U4/V4	U4/V4	B4	B4
D3	U3/V3	U3/V3	B3	B3
D2	U2/V2	U2/V2	B2	B2
D1	U1/V1	U1/V1	B1	B1
D0	U0/V0	U0/V0	B0	B0

When FTCSIRX100_QUAD_PIXEL is defined (Quad-pixel mode is selected), the bus width of dpi_vc***n***_D is 56-bit. When FTCSIRX100 outputs a couple of the pixels (UYVY) in the same vc***n***_PCLK clock (Please refer to Section 5.5.2.3), "U1" denotes the component U, "V1" denotes the component V, "Y1" and "Y2" denote the component Y, and FTCSIRX100 receives "Y1" prior to "Y2". Therefore, FTCSIRX100 receives the components in the sequence: U1 -> Y1 -> V1 -> Y2.

Table 2-17. DPI Data Pin Mapping for Transmitting a Couple YUV Pixel By One Pixel Clock

Signal	YUV422 8-bit	YUV 422 10-bit
dpi_vc***n***_D[55:42]	{0b000000,Y2[7:0] }	{0b0000, Y2[9:0]}
dpi_vc***n***_D[41:28]	{0b000000, V1[7:0]}	{0b0000, V1[9:0]}
dpi_vc***n***_D[27:14]	{0b000000, Y1[7:0]}	{0b0000, Y1[9:0]}
dpi_vc***n***_D[13:0]	{0b000000, U1[7:0]}	{0b0000, U1[9:0]}

When FTCSIRX100 outputs two RGB pixels in the same vc***n***_PCLK clock (Please refer Section 5.5.2.5), "R1" denotes the component R of the first pixel, "G1" denotes the component G of the first pixel, and "B1" denotes the component R of the first pixel. "R2" denotes the component R of the second pixel, "G2" denotes the component G of the second pixel, and "B2" denotes the component R of the second pixel. FTCSIRX100 receives the first pixel prior to the second pixel.

Table 2-18. DPI Data Pin Mapping for Transmitting a Couple RGB Pixels By One Pixel Clock

Signal	RGB565	RGB888
dpi_vc***n***_D[55:52]	0b0000	0b0000
dpi_vc***n***_D[51:44]	{0b000, R2[4:0]}	R2[7:0]
dpi_vc***n***_D[43:36]	{0b00, G2[5:0]}	G2[7:0]
dpi_vc***n***_D[35:28]	{0b000, B2[4:0]}	B2[7:0]
dpi_vc***n***_D[27:24]	0b0000	0b0000
dpi_vc***n***_D[23:16]	{0b000, R1[4:0]}	R1[7:0]
dpi_vc***n***_D[15:8]	{0b00, G1[5:0]}	G1[7:0]
dpi_vc***n***_D[7:0]	{0b000, B1[4:0]}	B1[7:0]

Chapter 3

Configuration Parameters

This chapter contains the following sections:

3.1 Hardware Configurable Macros

The hardware macros are defined in the "*FTCSIRX100_DEF.vh*" file of the FTCSIRX100 database. The hardware configuration macros are described in the tables of this chapter. In the macro definitions of this table, the width of the assigned integer value and the consistent value with the required width is used to avoid the DWORD (32-bit) alignment issue in the RTL code of FTCSIRX100.

For example, to define Vender ID as 0x8B, users can refer to Table 3-1 for the required constant that is an 8-bit integer, and the macro should be defined as below:

```
`define FTCSIRX100_VENDID          8'h8B
```

3.2 Register Default Value Configurations

This section describes the hard macros for the default values of the registers.

Table 3-1. Register Default Value Configurations

Defined Macro Name	Description
`define FTCSIRX100_VENDID	Vendor ID This macro defines the value of the Vender ID Register (VIDR, Address = 0x00). The default setting is as follows: `define FTCSIRX100_VENDID **8**'h8B
`define FTCSIRX100_DEVID	Device ID This macro defines the value of the Device ID Register (DIDR, Address = 0x01). The default setting is as follows: `define FTCSIRX100_DEVID **8**'h01
`define FTCSIRX100_TINIT	Initialization Timer This macro defines the default value of the Initialization Timer Register (ITR, Address = 0x12 ~ 0x13). The default setting is as follows: `define FTCSIRX100_TINIT **10**'h20

Defined Macro Name	Description
`define FTCSIRX100_VC***n***_HPN	VC***n*** Horizontal Pixel Number The macro defines the default value of the register HPNR***n*** (Address = 0x20 ~ 0x27). The value of the lower 8 bits is the default of the register HPNLR***n*** and the value of the higher 5 bits is the default of the register HPNHR***n***. Here ***n*** is the virtual channel number which can be 0, 1, 2, or 3 when the macro, FTCSIRX100_VC***n***, is defined. The default setting is as follows: `define FTCSIRX100_VC0_HPN ***13***'h0 `define FTCSIRX100_VC1_HPN ***13***'h0 `define FTCSIRX100_VC2_HPN ***13***'h0 `define FTCSIRX100_VC3_HPN ***13***'h0
`define FTCSIRX100_PECR	PPI Enable Control Register The macro defines the default value of the register PECR.PEC. (PECR, Address = 0x28). The default setting is as follows: `define FTCSIRX100_PECR ***4***'h0
`define FTCSIRX100_VC***n***_DPT	DPCM Packet Type This macro defines the default value of the register DPCMR (Address = 0x48 ~ 0x4B) for the virtual channel ***n***. The default setting is as follows: `define FTCSIRX100_VC0_DPT ***6***'h39 `define FTCSIRX100_VC1_DPT ***6***'h39 `define FTCSIRX100_VC2_DPT ***6***'h39 `define FTCSIRX100_VC3_DPT ***6***'h39
`define FTCSIRX100_VC***n***_VLN	Vertical Line Number for DPI Pattern Generator This macro defines the default value of the register BPGR (address = 0x90 ~0x97). The default setting is as follows: `define FTCSIRX100_VC0_VLN ***12***'h400 `define FTCSIRX100_VC1_VLN ***12***'h400 `define FTCSIRX100_VC2_VLN ***12***'h400 `define FTCSIRX100_VC3_VLN ***12***'h400

3.3 Virtual Channel Configurations

FTCSIRX100 supports four virtual channel configurations. The virtual channel#0 (VC0) is always supported by FTCSIRX100 and the hard macro, FTCSIRX100_VC0, is defined by default. Others configurations are listed in Table 3-2.

Table 3-2. Virtual Channel Configurations

Defined Macro Name	Description
`define FTCSIRX100_VC3	Virtual Channel#0~3 exist When this macro is defined, FTCSIRX100 supports four virtual channels (VC3 ~ VC0) and the macros, FTCSIRX100_VC2, FTCSIRX100_VC1 and FTCSIRX100_VC0, are automatically defined in the FTCSIRX100_DEF.vh file. This macro is turn-off by default.
`define FTCSIRX100_VC2	Virtual Channel#0~2 exist When this macro is defined, FTCSIRX100 supports three virtual channels (VC2 ~ VC0) and the macros, FTCSIRX100_VC1 and FTCSIRX100_VC0, are automatically defined in the FTCSIRX100_DEF.vh file. This macro is turn-off by default.
`define FTCSIRX100_VC1	Virtual Channel#0~1 exist When this macro is defined, FTCSIRX100 supports two virtual channels (VC1 ~ VC0) and the macro, FTCSIRX100_VC0, is automatically defined in the FTCSIRX100_DEF.vh file. This macro is turn-off by default.

3.4 Lane Number Configurations

By defining the hard macros, FTCSIRX100 supports four kinds of the lane number configurations and the hardmacro, "FTCSIRX100_LANE_NUM", is also automatically defined. These hardmacros determine the number of data lanes existed in the hardware of FTCSIRX100. For data lane enable, please refer to Section 5.10 for details. Here, the data lane number is the physical data lane number. The hardmacro, "FTCSIRX100_FOUR_LANE", is defined by default.

Table 3-3. Lane Number Configurations

Hard Definition	Description
`define FTCSIRX100_FOUR_LANE	Four data lanes (data lane#0 ~ #3) exist at the IP interface. The value of FTCSIRX100_LANE_NUM is 4 (DLW = 4).
// `define FTCSIRX100_FOUR_LANE `define FTCSIRX100_THREE_LANE	Three data lanes (data lane#0 ~ #2) exist at the IP interface. The value of FTCSIRX100_LANE_NUM is 3 (DLW = 3).

FARADAY

Hard Definition	Description
// `define FTCSIRX100_FOUR_LANE //`define FTCSIRX100_THREE_LANE `define FTCSIRX100_TWO_LANE	Two data lanes (data lane#0 ~ #1) exist at the IP interface. The value of FTCSIRX100_LANE_NUM is 2 (DLW = 2).
// `define FTCSIRX100_FOUR_LANE //`define FTCSIRX100_THREE_LANE //`define FTCSIRX100_TWO_LANE `define FTCSIRX100_ONE_LANE	Single data lane (data lane#0) exists at the IP interface. The value of FTCSIRX100_LANE_NUM is 1 (DLW = 1).

3.5 Feature Configurations

This section describes the hardware feature configurations.

Table 3-4. Feature Configurations

Defined Macro Name	Description
`define FTCSIRX100_CRC_CHK	CRC Checking If this macro is defined, FTCSIRX100 will check CRC (Checksum) of the received packets. Otherwise, FTCSIRX100 will treat CRC (Checksum) of each received packet to be correct. This macro is turn-on by default.
`define FTCSIRX100_I2C	Supporting I^{2}C Interface When this macro is defined, the I^{2}C slave interface exists. This macro is turn-on by default.
`define FTCSIRX100_APB	Supporting APB Interface When this macro is defined, the APB slave interface exists. This macro is turn-on by default.
`define FTCSIRX100_ASC	APB Clock is Asynchronous to csi_clk. When this macro is defined, a. The APB synchronization module in FTCSIRX100 exists. This module is used to synchronize the signals between the apb_clk and csi_clk domains. b. The clock apb_clk can be asynchronous to the clock csi_clk. c. The AMBA 2 APB Specification (APB2) is not supported. The AMBA 3 APB Protocol Specification, Version 1.0 (APB3), and AMBA APB Protocol Specification, Version 2.0 (APB4), are supported. d. The macro, FTCSIRX100_APB, is also automatically defined in the "*FTCSIRX100_DEF.vh*" file. When this macro is **not** defined, a. The APB synchronization module in FTCSIRX100 **doesn't** exist. b. The clock apb_clk **must** have the same frequency as the clock csi_clk and be synchronous with csi_clk. c. APB2, APB3, and APB4 are supported. This macro is turn-on by default.

Defined Macro Name	Description
`define FTCSIRX100_CHK_DATA	PPI Interface Data Checking Test Mode Exist When this macro is defined, the logic and signals will exist for the PPI interface data checking test mode. This macro is turn-off by default. Please refer to Sections 5.6.2 and 5.9 for details.
`define FTCSIRX100_DUAL_PIXEL	Dual-Pixel Mode When this macro is defined, the dual-pixel mode is enabled. In this mode, FTCSIRX100 transmits two pixels per pixel clock for RAW8, RAW10, RAW12, and RAW14 at DPI and two bytes per pixel clock at the generic interface. This macro is turn-off by default.
`define FTCSIRX100_QUAD_PIXEL	Quad-Pixel Mode When this macro is defined, the quad-pixel mode is enabled. In this mode, FTCSIRX100 transmits four pixels per pixel clock for RAW8, RAW10, RAW12, and RAW14 at DPI and four bytes per pixel clock at the generic interface. Please refer to Section 5.5.1 for further information. This macro is turn-off by default.
`define FTCSIRX100_MONO_RGB_EN	Single RGB Pixel Mode This macro is used to configure the RGB pixel output at DPI in the quad-pixel mode. When this macro is not defined and FTCSIRX100_QUAD_PIXEL is defined, FTCSIRX100 will output dual pixels per pixel clock at DPI. Please refer to Section 5.5.2.5 for more detailed information. Otherwise, FTCSIRX100 will output single pixel per pixel clock at DPI. Please refer to Section 5.5.2.4 for more detailed information. This macro is turn-off by default.
`define FTCSIRX100_DPCM_EN	DPCM Mode When this macro is defined, DPCM decoding is supported. This macro is turn-off by default.
`define FTCSIRX100_FRC	Frame Rate Control When this macro is defined, FTCSIRX100 supports the frame rate control. This macro is turn-off by default.
`define FTCSIRX100_DPI_PG	DPI Pattern Generator When this macro is defined, FTCSIRX100 can generate the built-in test pattern at DPI. This macro is turn-off by default.
`define FTCSIRX100_LANE_SWAP	Data Lane Swap When this hardmacro is defined, FTCSIRX100 can support the data lane swapping. This macro is turn-off by default.

3.6 Memory Configurations

This section describes the memory configurations. The hardware stationary macros cannot be modified by users. If other configurations are required, please contact the Faraday service team for required information.

Table 3-5. Memory Configurations

Name	Size (Bit)	Type	Macro Description and Definition (Fixed)
DPI FIFO	32x30x2	Synchronous two-port SRAM	For both the single-pixel and dual-pixel modes, FTCSIRX100 uses a couple of the memory block for each virtual channel. The memory block size is 32x30 bits. Therefore, the size totals 32x30x2 bits per virtual channel.
	32x60		For the quad-pixel mode, FTCSIRX100 uses a single memory block for each virtual channel. The memory block size is 32x60 bits per virtual channel.
Lane alignment buffer	16x34	Synchronous two-port SRAM	Lane alignment buffer has 16 entries and each entry contains 34 bits.

3.6.1 Memory Delay Options

These macros provide the input values for the delay options of the Faraday memory blocks.

The default settings are

`define FTCSIRX100_DVSE 1'b0

`define FTCSIRX100_DVS0 1'b0

`define FTCSIRX100_DVS1 1'b0

`define FTCSIRX100_DVS2 1'b0

`define FTCSIRX100_DVS3 1'b0

`FTCSIRX100_DVSE is the DVSE input when the memory block contains this input.

`FTCSIRX100_DVS0 is the DVS0 input when the memory block contains this input.

`FTCSIRX100_DVS1 is the DVS1 input when the memory block contains this input.

`FTCSIRX100_DVS2 is the DVS2 input when the memory block contains this input.

`FTCSIRX100_DVS3 is the DVS3 input when the memory block contains this input.

Chapter 4

Register Definition

This chapter contains the following sections:

4.1 Register Abbreviation

Table 4-1. Definition of Register Abbreviation

Register Type	Description
RO	Read-only: Register bits are read-only and can not be altered by software. Register bits are initialized by hardware mechanisms, such as the pin strapping, hardware configuration, or serial EEPROM.
RW	Read-Write: Register bits are read-write and can be set or cleared by software to the desired state.
RW1C	Write-1-to-clear status: Register bits indicate status when read, a set bit indicating a status event may be cleared by writing a '1'. Writing a '0' to RW1C bits has no effect.
RW1S	Write-1-to-set status: Register bits indicate status when read. Writing a '0' to RW1S bits has no effect.
Rsvd	Reserved: Reserved for future RO implementations. Registers or memory that shall be treated as read-only by the system software. Reserved registers shall return '0' when read. Software shall ignore the value read from these bits. Default value: 0
RsvdO	Reserved and Opaque: Reserved for exclusive IP use. Registers values may be modified by FTCSIRX100 at any time. Software manipulation of this register can cause undetermined results. Software shall not write this register unless explicitly allowed by the Faraday-specific instruction.
Default Value	**Description**
HwInit	Hardware Initialized: The default value of register bits are initialized by firmware or hardware mechanisms, such as the pin strapping or serial EEPROM.
HwCfg	Hardware Configurable: The default value of register bits are hardware configurable and determined by FTCSIRX100_DEF.vh.

4.2 FTCSIRX100 Registers

Table 4-2 summarizes the registers in FTCSIRX100. The software can access these registers through the I^{2}C (If FTCSIRX100_I2C is defined) or APB interface (If FTCSIRX100_APB is defined). The reserved registers or fields are read only as zero.

Besides, the status register bits with RW1C attribute, Interrupt Enable Register (INTER, Address = 0x3C), and Frame Rate Control Register (FRCR, Address = 0x80 ~ 0x87), after programming all registers, CR.SR (Address = 0x04) shall be set to re-initialize FTCSIRX100.

Table 4-2. Summary of FTCSIRX100 Registers

Offset (Hex)	Name	Size (Byte)	Type	Description	Default Value (Hex)
0x00	VIDR	1	RO	Vendor ID Register	HwCfg
0x01	DIDR	1	RO	Device ID Register	HwCfg
0x04	CR	1	RO, RW, RW1S	Control Register	HwCfg
0x05	VSCR	1	RO, RW	DPI V Sync Control Register	0
0x06	ECR	1	RO, RW	Extended Control Register	0
0x08	TCNR	2	RO, RW	Timer Count Number Register	0x64
0x0A	HRTVR	1	RW	HS RX Timeout Value Register	0x64
0x0B	FIUR	3	RsvdO	FTC Internal Use Register	0
0x12	ITR	2	RO, RW	Initialization Timer Register	HwCfg
0x14	VSTR0	1	RW	DPI VC0 V Sync Timing Register	0x02
0x15	HSTR0	1	RW	DPI VC0 H Sync Timing Register	0x08
0x16	VSTR1	1	RO or RW	DPI VC1 V Sync Timing Register	HwCfg
0x17	HSTR1	1	RO or RW	DPI VC1 H Sync Timing Register	HwCfg
0x18	VSTR2	1	RO or RW	DPI VC2 V Sync Timing Register	HwCfg
0x19	HSTR2	1	RO or RW	DPI VC2 H Sync Timing Register	HwCfg
0x1A	VSTR3	1	RO or RW	DPI VC3 V Sync Timing Register	HwCfg
0x1B	HSTR3	1	RO or RW	DPI VC3 H Sync Timing Register	HwCfg
0x1C	MCR	1	RO, RW	DPI Mapping Control Register	0

Offset (Hex)	Name	Size (Byte)	Type	Description	Default Value (Hex)
0x1E	VSTER	2	RO, RW	DPI V Sync Timing Extended Register	0
0x20	HPNR	8	RO, RW	DPI Horizontal Pixel Number Register	HwCfg
0x28	PECR	1	RO, RW	PPI Enable Control Register	HwCfg
0x2A	DLMR	2	RO, RW	Data Lane Mapping Register	HwCfg
0x30	CSIERR	2	RO	CSI Error Report Register	0
0x33	INTSTS	1	RO, RW1C	Interrupt Status register	0
0x34	ESR	2	RO	Escape Mode and Stop State Register	HwInit
0x38	DPISR	4	RO, RW1C	DPI Status Register	0
0x3C	INTER	1	RW	Interrupt Enable Register	0x41
0x40	FFR	8	RO	FTCSIRX100 Feature Register	HwCfg, HwInit
0x48	DPCMR	4	RO, RW	DPCM Register	HwCfg
0x4C	FRR	4	RO	FTCSIRX100 Revision Register	HwCfg
0x50	PFTR	4	RO, RW	Pixel FIFO Threshold Register	0
0x80	FRCR	8	RO, RW	Frame Rate Control Register	HwCfg
0x88	FNR	8	RO	Frame Number Register	HwInit
0x90	BPGR	8	RO, RW	DPI Built-in Pattern Generator Register	HwCfg

4.2.1 Vender ID Register (VIDR, Address = 0x00)

This register shows the vendor ID.

Table 4-3. Vendor ID Register (VIDR, Address = 0x00)

Bit	Name	Type	Default Value	Description
[7:0]	VID	RO	HwCfg	Vendor ID This register identifies the manufacture of a device. The value of this register is defined by the hard macro, FTCSIRX100_VENDID.

4.2.2 Device ID Register (DIDR, Address = 0x01)

This register shows the device ID.

Table 4-4. Device ID Register (DIDR, Address = 0x01)

Bit	Name	Type	Default Value	Description
[7:0]	DID	RO	HwCfg	Device ID This register identifies a particular device. The value of this register is defined by the hard macro, FTCSIRX100_DEVID.

4.2.3 Control Register (CR, Address = 0x04)

This register contains the control bits of FTCSIRX100 and these bits affect all valid virtual channels.

Table 4-5. Control Register (CR, Address = 0x04)

Bit	Name	Type	Default Value	Description
[7:6]	-	RO	0x0	Reserved
5	HEG	RW	0	Hsync Enable for Generic Long and User Defined Packets When this bit is 1: dpi_vc***n***_Hsync can assert for receiving virtual channel ***n*** generic long (Packet type = 0x12 ~ 0x17) or user-defined packets (Packet type = 0x30 ~ 0x37). 0: dpi_vc***n***_Hsync doesn't assert for receiving the virtual channel ***n*** generic long or user defined packets.
4	EAPBL	RW	0	Enable Auto-Padding for Broken Line When this bit is 1: When the line with the image stream is broken (i.e. dpi_vc***n***_DE deasserting), FTCSIRX100 will treat this event as the exception and automatically pad the pixels up to the number defined in the corresponding DPI VC***n*** Horizontal Pixel Number Register (HPNR***n***, Address = 0x20 ~ 0x27). 0: Auto-padding is disabled.
3	ECCCE	RW	1	ECC Checking Enable When this bit is set, FTCSIRX100 can report the ECC single bit error (CSIERR1.ECCES) and multiple-bit error (CSIERR1.ECCEM). Otherwise, the reporting is disabled.
2	CRCCE	RW or RO	1 for Type = RW 0 for Type = RO	CRC Checking Enable This bit controls CRC checking. When this bit is set, FTCSIRX100 calculates CRC (checksum) and checks the result with the checksum of the received packet. If the CRC error is found, it will reflect at CSIERR1.CE. When this bit is cleared, it doesn't check the result. If the hard macro FTCSIRX100_CRC_CHK is not defined, this field is read-only as zero.

Bit	Name	Type	Default Value	Description
1	TME	RW	0	Test Mode Enable This test mode is only for Faraday internal use.
0	SR	RW1S	0	Software Reset Writing one to this bit will trigger FTCSIRX100 to perform the software reset. This bit is still read as zero after writing one to it. Setting this bit is not effective when the clock of the corresponding clock domain is not valid. For example, to clear the pixel FIFO of the virtual channel ***n***, the csi_clk and vc***n***_PCLK clocks shall be valid. Please refer to Section 5.6.1.2 for details.

4.2.4 DPI V Sync Control Register (VSCR, Address = 0x05)

This register determines the deassertion condition and the time unit of pulse width of the dpi_vc***n***_Vsync signal. Here ***n*** is the virtual channel number and ***n*** can be 0, 1, 2, or 3.
FTCSIRX100 asserts dpi_vc***n***_Vsync (This signal is low active) for receiving VC***n*** Frame Start short packet.

For the deassertion condition:
When the VSPC***n*** bit of this register is set, FTCSIRX100 will deassert this signal after receiving VC***n*** Frame End.

When this bit is cleared, the low-active pulse width of dpi_vc***n***_Vsync is controlled by DPI VC***n*** V Sync Timing Register (Section 4.2.10), DPI V Sync Timing Extended Register (Section 4.2.13) and VSTU***n*** bit of this register. That is,

VC***n*** (***n*** = 0 or 1) V Sync width = {VSTER0.VC***n***[3:0], VSTR***n***.W[7:0]} or
4096 when both VSTER0.VC***n*** and VSTR***n*** are zero.
VC***n*** (***n*** = 2 or 3) V Sync width = {VSTER1.VC***n***[3:0], VSTR***n***.W[7:0]} or
4096 when both VSTER1.VC***n*** and VSTR***n*** are zero.

In the above expression, the time unit is "pixel clock (vc***n***_PCLK)" for VSCR.VSTU***n*** = '1' and the time unit is "horizontal line" for VSCR.VSTU***n*** bit = '0'. When the time unit is "horizontal line", the pulse width counter increases by each dpi_vc***n***_Hsync deassertion.
The bits, VSPC***n*** and VSTU***n***, are read-only as zero when VC***n*** doesn't exist.

Table 4-6. DPI V Sync Control Register (VSCR, Address = 0x05)

Bit	Name	Type	Default Value	Description
7	VSPC3	RW or RO	0	V Sync Pulse Control for VC3 This bit determines dpi_vc3_Vsync signal behavior. Please refer to the above description.
6	VSPC2	RW or RO	0	V Sync Pulse Control for VC2 This bit determines dpi_vc2_Vsync signal behavior. Please refer to the above description.
5	VSPC1	RW or RO	0	V Sync Pulse Control for VC1 This bit determines dpi_vc1_Vsync signal behavior. Please refer to the above description.
4	VSPC0	RW	0	V Sync Pulse Control for VC0 This bit determines dpi_vc0_Vsync signal behavior. Please refer to the above description.
3	VSTU3	RW or RO	0	V Sync Time Unit for VC3 This bit determines the time unit of dpi_vc3_Vsync signal width. When this bit is set, the time unit is the pixel clock (vc3_PCLK). Otherwise, the time unit is the horizontal line.
2	VSTU2	RW or RO	0	V Sync Time Unit for VC2 This bit determines the time unit of dpi_vc2_Vsync signal width. When this bit is set, the time unit is the pixel clock (vc2_PCLK). Otherwise, the time unit is the horizontal line.
1	VSTU1	RW or RO	0	V Sync Time Unit for VC1 This bit determines the time unit of dpi_vc1_Vsync signal width. When this bit is set, the time unit is the pixel clock (vc1_PCLK). Otherwise, the time unit is the horizontal line.
0	VSTU0	RW	0	V Sync Time Unit for VC0 This bit determines the time unit of dpi_vc0_Vsync signal width. When this bit is set, the time unit is the pixel clock (vc0_PCLK). Otherwise, the time unit is the horizontal line.

4.2.5 Extended Control Register (ECR, Address = 0x06)

Table 4-7. Extended Control Register (ECR, Address = 0x06)

Bit	Name	Type	Default Value	Description
[7:2]	-	RO	0	Reserved
1	SF	RW	0	Software Flag The value of this bit is reflected to the output signal, csi_sw_flag_r.
0	PCE	RW or RO	0	PPI Data Checking Enable This bit is the software enable bit for PPI Data Checking. When this bit is 0: PPI Interface Data Checking is not enabled by software. 1: PPI Interface Data Checking is enabled by software. The normal operation is disabled. This bit is read-only as zero when the macro, FTCSIRX100_CHK_DATA, is not defined.

4.2.6 Timer Count Number Register (TCNR, Address = 0x08 ~ 0x09)

This register determines the count number of the csi_clk clock for 1 μs. The default value of the timer count number is 0x64 (= 100 in decimal), which means that FTCSIRX100 counts 100 csi_clk clocks as 1 μs. When the value is set as zero, this timer will be disabled and the mechanisms which count μs or ms will not work at this situation.

Table 4-8. Timer Count Number Lower Register (TCNLR, Address = 0x08)

Bit	Name	Type	Default Value	Description
[7:0]	TCN	RW	0x64	Timer Count Number This field is the lower bits (Bit 7 to bit 0) of the timer count number for 1 μs.

Table 4-9. Timer Count Number Higher Register (TCNHR, Address = 0x09)

Bit	Name	Type	Default Value	Description
[7:2]	-	RO	0	Reserved
[1:0]	TCN	RW	0	Timer Count Number This field is the higher bits (Bit 9 to bit 8) of the timer count number for 1 μs.

4.2.7 HS RX Timeout Value Register (HRTVR, Address = 0x0A)

The timeout value of HRX_TO timer is (Value of this register) * 128 µs. When the value of this field is zero, this timer is disabled and never expires. The default value is 0x64 (The decimal is 100) so the default of the timeout value is 128 * 100 µs = 12.8 ms.

Table 4-10. HS RX Timeout Value Register (HRTVR, Address = 0x0A)

Bit	Name	Type	Default Value	Description
[7:0]	TV	RW	0x64	Timeout Value This value of this register indicates the timeout value of HRX_TO timer.

4.2.8 FTC Internal Use Registers (FIUR0, Address = 0x0B ~ 0x0D)

These registers are for Faraday internal use only. Software shall not write them.

Table 4-11. FTC Internal Use Register 0 (FIUR0, Address = 0x0B)

Bit	Name	Type	Default Value	Description
[7:0]	V	RsvdO	0	For Faraday internal use only

Table 4-12. FTC Internal Use Register 1 (FIUR1, Address = 0x0C)

Bit	Name	Type	Default Value	Description
[7:0]	V	RsvdO	0	For Faraday internal use only

Table 4-13. FTC Internal Use Register 2 (FIUR2, Address = 0x0D)

Bit	Name	Type	Default Value	Description
[7:0]	V	RsvdO	0	For Faraday internal use only

4.2.9 Initialization Timer Register (ITR, Address = 0x12 ~ 0x13)

Both registers, ITHR and ITLR, determine the period of asserting PPI signal, ForceRxmode, after the power-on reset, system reset, or software reset (Please refer to notes in the following page). Please refer to Section 5.6 for more details. The period is {ITHR.IT[1:0], ITLR.IT[7:0]} * 1 μs. Here, "1 μs" length is defined in TCNR (Address = 0x08 ~ 0x09). The initialization timer counts for the period of all PPI Stopstate signals continuously keeping as high. Here "all PPI Stopstate signals" means all enabled data lanes' Stopstate signals. This timer is cleared when either of below clear conditions is TRUE:

- Power-on reset is active.
- System reset is active.
- Software reset (CR.SR) is triggered.
- Either of enabled data lane's Stopstate deasserted.

For example, if {ITHR.IT[1:0], ITLR.IT[7:0]} = 0x3FF (1023 in decimal), after the power-on reset, system reset, or software reset (Please refer to notes in the following page), ForceRxmode can be asserted until FTCSIRX100 detecting all PPI Stopstate signals continuously keeping at high for 1023 μs.

The default value is defined in the hard macro, FTCSIRX100_TINIT.

Notes:

For the case {ITHR.IT[1:0], ITLR.IT[7:0]} = 0, the PPI signal, ForceRxmode, behaves as:

1. FTCSIRX100 still asserts ForceRxmode 1~3 csi_clk clocks after both pwr_rst_n and sys_rst_n are releasing. Here, ForceRxmode pulse variation depends on the timing relationship among csi_clk, pwr_rst_n, and sys_rst_n.
2. ForceRxmode isn't asserted by software reset.

Table 4-14. Initialization Timer Lower Register (ITLR, Address = 0x12)

Bit	Name	Type	Default Value	Description
[7:0]	IT	RW	HwCfg	Initialization Timer Value The field is the lower bits (bit 7 to bit 0) of the initialization timer value.

Table 4-15. Initialization Timer Higher Register (ITHR, Address = 0x13)

Bit	Name	Type	Default Value	Description
[7:2]	-	RO	0	Reserved
[1:0]	IT	RW	HwCfg	Initialization Timer Value The field is the higher bits (bit 9 to bit 8) of the initialization timer value.

4.2.10 DPI V Sync Timing Register (VSTR, Address = 0x14, 0x16, 0x18, 0x1A)

When VSCR.VSPC***n*** register bit (Address = 0x05) is clear, the DPI VC***n*** V Sync Timing Register can determine the V Sync pulse width of VC***n*** at DPI. The register VSTR***n*** is read-only as zero when the hardmacro, FTCSIRX100_VC***n***, isn't defined. ***n*** is the virtual channel number which can be 0, 1, 2, or 3.

Table 4-16. DPI VC*n* V Sync Timing Register for *n* = 0 or 1 (VSTR0, Address = 0x14; VSTR1, Address = 0x16)

Bit	Name	Type	Default Value	Description
[7:0]	W	RW or RO	0x02	V Sync Width VC***n*** V Sync width = {VSTER0.VC***n***[3:0], VSTR***n***.W[7:0]} The unit is "Line", which is defined in the DPI specification [MIPI03]. VSTER0 register locates at the address 0x1E. The default value is 0x002, which means that DPI Vsync signal width is 2 lines. If the value of {VSTER0.VC***n***[3:0], VSTR***n***.W[7:0]} is set as zero, the width is 4096 lines.

Table 4-17. DPI VC*n* V Sync Timing Register for *n* = 2 or 3 (VSTR2, Address = 0x18; VSTR3, Address = 0x1A)

Bit	Name	Type	Default Value	Description
[7:0]	W	RW or RO	0x02	V Sync Width VC***n*** V Sync width = {VSTER1.VC***n***[3:0], VSTR***n***.W[7:0]} The unit is "Line", which is defined in the DPI specification [MIPI03]. VSTER1 register locates at the address 0x1F. The default value is 0x002, which means that DPI Vsync signal width is 2 lines. If the value of {VSTER1.VC***n***[3:0], VSTR***n***.W[7:0]} is set as zero, the width is 4096 lines.

4.2.11 DPI H Sync Timing Register (HSTR, Address = 0x15, 0x17, 0x19, 0x1B)

The DPI VC***n*** H Sync Timing Register determines the H Sync pulse width of VC***n*** at DPI. This register is read-only as zero when VC***n*** doesn't exist. ***n*** is the virtual channel number which can be 0, 1, 2 or 3.

Table 4-18. DPI VC*n* H Sync Timing Register (HSTR*n*, Address = 0x15 + 2n*)**

Bit	Name	Type	Default Value	Description
[7:0]	W	RW or RO	0x08	H Sync Width VC***n*** H Sync width = HSTR***n***.W[7:0] The unit is vc***n***_PCLK (DPI pixel clock). The default value is 0x08, which means that DPI Hsync signal width is 8 vc***n***_PCLK cycles. If this register is set as zero, the width is 256 vc***n***_PCLK cycles.

4.2.12 DPI Mapping Control Register (MCR, Address = 0x1C)

Table 4-19. DPI Mapping Control Register (MCR, Address = 0x1C)

Bit	Name	Type	Default Value	Description
7	-	RO	0	Reserved
6	M3	RW or RO	0	Mapping Control for VC3 When this bit is set, the VC3 pixel outputs for all RAW types are aligned to dpi_vc3_D[13] (MSB), so the pixel output for RAW8 is dpi_vc3_D[13:6]; for RAW10 is dpi_vc3_D[13:4]. When this bit is zero, the VC3 pixel outputs for all RAW types are aligned to dpi_vc3_D[0] (LSB), so the pixel output for RAW8 is dpi_vc3_D[7:0]; for RAW10 is dpi_vc3_D[9:0]. This bit is read-only as zero when VC3 doesn't exist.
5	-	RO	0	Reserved
4	M2	RW or RO	0	Mapping Control for VC2 When this bit is set, the VC2 pixel outputs for all RAW types are aligned to dpi_vc2_D[13] (MSB), so the pixel output for RAW8 is dpi_vc2_D[13:6]; for RAW10 is dpi_vc2_D[13:4]. When this bit is zero, the VC2 pixel outputs for all RAW types are aligned to dpi_vc2_D[0] (LSB), so the pixel output for RAW8 is dpi_vc2_D[7:0]; for RAW10 is dpi_vc2_D[9:0]. This bit is read-only as zero when VC2 doesn't exist.
3	-	RO	0	Reserved

Bit	Name	Type	Default Value	Description
2	M1	RW or RO	0	Mapping Control for VC1 When this bit is set, the VC1 pixel outputs for all RAW types are aligned to dpi_vc1_D[13] (MSB), so the pixel output for RAW8 is dpi_vc1_D[13:6]; for RAW10 is dpi_vc1_D[13:4]. When this bit is zero, the VC1 pixel outputs for all RAW types are aligned to dpi_vc1_D[0] (LSB), so the pixel output for RAW8 is dpi_vc1_D[7:0]; for RAW10 is dpi_vc1_D[9:0]. This bit is read-only as zero when VC1 doesn't exist.
1	-	RO	0	Reserved
0	M0	RW	0	Mapping Control for VC0 When this bit is set, the VC0 pixel outputs for all RAW types are aligned to dpi_vc0_D[13] (MSB), so the pixel output for RAW8 is dpi_vc0_D[13:6]; for RAW10 is dpi_vc0_D[13:4]. When this bit is zero, the VC0 pixel outputs for all RAW types are aligned to dpi_vc0_D[0] (LSB), so the pixel output for RAW8 is dpi_vc0_D[7:0]; for RAW10 is dpi_vc0_D[9:0].

4.2.13 DPI V Sync Timing Extended Register (VSTER, Address = 0x1E ~ 0x1F)

These registers are used to extend the active pulse width of DPI V Sync signals.

Table 4-20. DPI V Sync Timing Extended Register 0 (VSTER0, Address = 0x1E)

Bit	Name	Type	Default Value	Description
[7:4]	VC1	RW or RO	0	VC1 Extended Timing It is used to extend the width of dpi_vc1_Vsync and effective only when VSCR.VSPC1 register bit (Address = 0x05) is clear. This field is read-only when the macro, FTCSIRX100_VC1, is not defined. Please refer to Section 4.2.4 for details.
[3:0]	VC0	RW	0	VC0 Extended Timing It is used to extend the width of dpi_vc0_Vsync and effective only when VSCR.VSPC0 register bit (Address = 0x05) is clear. Please refer to Section 4.2.4 for details.

Table 4-21. DPI V Sync Timing Extended Register 1 (VSTER1, Address = 0x1F)

Bit	Name	Type	Default Value	Description
[7:4]	VC3	RW or RO	0	VC3 Extended Timing It is used to extend the width of dpi_vc3_Vsync and effective only when VSCR.VSPC3 register bit (Address = 0x05) is clear. This field is read-only when the macro, FTCSIRX100_VC3, is not defined. Please refer to Section 4.2.4 for details.
[3:0]	VC2	RW or RO	0	VC2 Extended Timing It is used to extend the width of dpi_vc2_Vsync and effective only when VSCR.VSPC2 register bit (Address = 0x05) is clear. This field is read-only when the macro, FTCSIRX100_VC2, is not defined. Please refer to Section 4.2.4 for details.

4.2.14 DPI Horizontal Pixel Number Register (HPNR, Address = 0x20 ~ 0x27)

The DPI VC***n*** Horizontal Pixel Number Register (HPNR***n***) only exists when the hard macro, FTCSIRX100_VC***n***, is defined, here ***n*** is the virtual channel number which can be 0, 1, 2, or 3.

HPNR0 contains HPNLR0 (Address = 0x20) and HPNHR0 registers (Address = 0x21).
HPNR1 contains HPNLR1 (Address = 0x22) and HPNHR1 registers (Address = 0x23).
HPNR2 contains HPNLR2 (Address = 0x24) and HPNHR2 registers (Address = 0x25).
HPNR3 contains HPNLR3 (Address = 0x26) and HPNHR3 registers (Address = 0x27).

These registers determine the VC***n*** DPI horizontal pixel number. The default value is defined by the hardmacro, FTCSIRX100_VC***n***_HPN. When the value is zero, it means that the DPI horizontal pixel number is 8192.

When CR.EAPBL (Address = 0x04) is set and the line with the image stream is broken (i.e. dpi_vc***n***_DE deassert), FTCSIRX100 will pad the pixels up to the number defined in the corresponding pixel number. Please refer to the description of CR.EAPBL for the detailed information.

The pixel number of the received line shall consist with the corresponding pixel number setting; otherwise, the pixel number exceeding (DPISR***n***.PNE) and discontinuous data enable (DPISR***n***.DDE) flags are not effective.

FARADAY

If the number of the received pixel is smaller than the setting of the DPI horizontal pixel number(HPNR***n***), FTCSIRX100 can still output the image packet without any error flags. For example, when HPNR0 = 2048 (in decimal) and the number of the received horizontal line pixels is 1024 (in decimal), FTCSIRX100 can output 1024 pixels to DPI for this horizontal line without any error flags.

If the number of the received pixel is larger than the setting of DPI horizontal pixel number(HPNR***n***) and the value of HPNR***n*** is 4****m*** (***m*** is an integer), FTCSIRX100 will cut the image packet according to the setting of HPNR***n***, drop the residue pixels, and set the error flag, "DPISR***n***.PNE". For example, when HPNR0 = 1024 (in decimal) and the number of the received horizontal line pixels is 1048 (in decimal), FTCSIRX100 only outputs 1024 pixels to DPI and drop the residue pixels.
If the number of the received pixel is larger than the setting of the DPI horizontal pixel number(HPNR***n***) and the value of HPNR***n*** is 4****m*** + 2 (***m*** is an integer), it will cause the behavior of FTCSIRX100, and the registers, DPISR***n***.PNE and DPISR***n***.DDE, to be ambiguous.

Table 4-22. VC*n* Horizontal Pixel Number Lower Register (HPNLR*n*, Address = 0x20 + 2n*)**

Bit	Name	Type	Default Value	Description
[7:0]	HPN	RW or RO	HwCfg	Horizontal Pixel Number This field is the lower bits (Bit 7 to bit 0) of the DPI horizontal pixel number. This field is read-only when the macro, FTCSIRX100_VC***n***, is not defined.

Table 4-23. VC*n* Horizontal Pixel Number Higher Register (HPNHR*n*, Address = 0x21 + 2n*)**

Bit	Name	Type	Default Value	Description
7	-	RO	0	Reserved
6	VP	RW or RO	0	Vsync Polarity 0: dpi_vc***n***_Vsync is low active. 1: dpi_vc***n***_Vsync is high active. This field is read-only when the macro, FTCSIRX100_VC***n***, is not defined.
5	HP	RW or RO	0	Hsync Polarity 0: dpi_vc***n***_Hsync is low active. 1: dpi_vc***n***_Hsync is high active. This field is read-only when the macro, FTCSIRX100_VC***n***, is not defined.

Bit	Name	Type	Default Value	Description
[4:0]	HPN	RW or RO	HwCfg	Horizontal Pixel Number This field is the higher bits (Bit 12 to bit 8) of the DPI horizontal pixel number. This field is read-only when the macro, FTCSIRX100_VC***n***, is not defined.

4.2.15 PPI Enable Control Register (PECR, Address = 0x28)

Table 4-24. PPI Enable Control Register (PECR, Address = 0x28)

Bit	Name	Type	Default Value	Description
[7:4]	-	RO	0	Reserved
[3:0]	PEC	RW	HwCfg	PPI Enable Control This field controls the output signal ftcsirx100_ppi_Enable. The default value is configured by the hardmacro, FTCSIRX100_PECR. Please refer to Section 5.10 for details.

4.2.16 Data Lane Mapping Register (DLMR, Address = 0x2A ~ 0x2B)

These registers (DLMR0 and DLMR1) determine the logic data lanes mapping to the physical data lanes. DLMR0 register contains the data lane#0 and #1 mapping and DLMR1 register contains the data lane#2 and #3.

By default, the logical data lane#***n*** maps to the physical data lane#***n***. If DLMR0.L0 = 3 and the hardmacro, FTCSIRX100_LANE_NUM = 4 (Please refer to Section 3.4), it means that the logical data lane#0 maps to the physical data lane#3, that is, FTCSIRX100 treats the physical data lane#3 as the data lane#0 in the internal logic design. Please refer to Section 5.11 for details. This register can not be dynamically changed during normal operating.

Table 4-25. Data Lane Mapping Register 0 (DLMR0, Address = 0x2A)

Bit	Name	Type	Default Value	Description
[7:6]	-	RO	0	Reserved
[5:4]	L1	RW or RO	1 or 0	Logical Data Lane#1 Mapping This field indicates the physical data lane number mapping to the logical data lane#1. If the hardmacro, FTCSIRX100_LANE_SWAP, isn't defined, this field is read-only as zero.

Bit	Name	Type	Default Value	Description
[3:2]	-	RO	0	Reserved
[1:0]	L0	RW or RO	0	Logical Data Lane#0 Mapping This field indicates the physical data lane number mapping to the logical data lane#0. If the hardmacro, FTCSIRX100_LANE_SWAP, isn't defined, this field is read-only as zero.

Table 4-26. Data Lane Mapping Register 1 (DLMR1, Address = 0x2B)

Bit	Name	Type	Default Value	Description
[7:6]	-	RO	0	Reserved
[5:4]	L3	RW or RO	3 or 0	Logical Data Lane#3 Mapping This field indicates the physical data lane number mapping to the logical data lane#3. If the hardmacro, FTCSIRX100_LANE_SWAP, isn't defined, this field is read-only as zero.
[3:2]	-	RO	0	Reserved
[1:0]	L2	RW or RO	2 or 0	Logical Data Lane#2 Mapping This field indicates the physical data lane number mapping to the logical data lane#2. If the hardmacro, FTCSIRX100_LANE_SWAP, isn't defined, this field is read-only as zero.

4.2.17 CSI Error Report Register (CSIERR, Address = 0x30 ~ 0x31)

Besides CSIERR0.PCR bit (Address = 0x30, bit#4), the bits of this register are the error report bits. FTCSIRX100 can log the error events in the registers, CSIERR0 and CSIERR1. These bits (Except CSIERR0.PCR bit) are cleared after write-one-clear to INTSTS.CSI bit (Address = 0x33) or write-one to CR.SR bit (Address = 0x04). Please refer to Section 5.4 for details. For CSIERR0.PCR bit, please refer to Section 5.10 for details.

Table 4-27. CSI Error Report Register 0 (CSIERR0, Address = 0x30)

Bit	Name	Type	Default Value	Description
7	CD	RO	0	Contention Detected When this bit is set, it indicates that either of the data lane signals "ErrContentionLP0" or "ErrContentionLP1" has asserted. To detect this event, the high pulse of ErrContentionLP0 or ErrContentionLP1 shall be longer than two csi_clk clock cycles and csi_clk clock shall exist when the event occurs.

Bit	Name	Type	Default Value	Description
6	FCE	RO	0	False Control Error When this bit is set, it indicates either of ErrControl[***n***] has asserted. To detect this event, the high pulse of ErrControl[***n***] shall be longer than two csi_clk clock cycles and csi_clk clock shall exist when the event occurs. ErrControl[***n***] is the PPI signal ErrControl for the physical data lane#***n***. If the corresponding data lane ***n*** doesn't exist, ErrControl[***n***] is treated as low.
5	HSRT	RO	0	High-Speed Receiver Timeout When this bit is set, it indicates that "HS RX Timeout" (HRX-TO) has occurred. To detect this event, csi_clk clock shall exist for the timeout counter.
4	PCR	RO	0	PPI Checking Result This bit reflects the value of the output signal, csi_chk_pass_r. This bit is read-only as zero when the macro, FTCSIRX100_CHK_DATA, is not defined.
3	EMECE	RO	0	Escape Mode Entry Command Error When this bit is set, it indicates either of ErrEsc[***n***] has ever asserted. To detect this event, the high pulse of ErrControl[***n***] shall be longer than two csi_clk clock cycles and csi_clk clock shall exist when the event occurs. ErrEsc[***n***] is the PPI signal ErrEsc for the physical data lane#***n***. If the corresponding lane ***n*** doesn't exist, ErrEsc[***n***] is treated as low.
2	ESE	RO	0	EoT Sync Error When this bit is set, it indicates either of ErrEotSyncHS[***n***] has ever asserted. To detect this event, both csi_clk and RxByteClkHS clocks shall exist until logging this event in this bit. ErrEotSyncHS[***n***] is the input signal ErrEotSyncHS for the physical data lane#***n***. If the corresponding lane ***n*** doesn't exist, ErrEotSyncHS[***n***] is treated as low.
1	SSE	RO	0	SoT Sync Error When this bit is set, it indicates either of ErrSotSyncHS[***n***] has ever asserted. To detect this event, both csi_clk and RxByteClkHS clocks shall exist until logging this event in this bit. ErrSotSyncHS[***n***] is the PPI signal ErrSotSyncHS for the physical data lane#***n***. If the corresponding lane ***n*** doesn't exist, ErrSotSyncHS[***n***] is treated as low.
0	SE	RO	0	SoT Error When this bit is set, it indicates either of ErrSotHS[***n***] has ever asserted. To detect this event, both csi_clk and RxByteClkHS clocks shall exist until logging this event in this bit. ErrSotHS[***n***] is the PPI signal ErrSotHS for the physical data lane#***n***. If the corresponding lane ***n*** doesn't exist, ErrSotHS[***n***] is treated as low.

FARADAY

Table 4-28. CSI Error Report Register 1 (CSIERR1, Address = 0x31)

Bit	Name	Type	Default Value	Description
[7:6]	-	RO	0	Reserved
5	ITL	RO	0	Invalid Transmission Length
4	CSIVIDI	RO	0	CSI VC ID Invalid
3	UPT	RO	0	Unsupported Packet Type
2	CE	RO	0	Checksum Error (Long packet only)
1	ECCEM	RO	0	ECC Error, multi-bit (Detected but not corrected)
0	ECCES	RO	0	ECC Error, single-bit (Detected and corrected)

4.2.18 Interrupt Status Register (INTSTS, Address = 0x33)

This register records the interrupt status. It affects the output signal csi_rx_int_r. Please refer to Table 2-7 for details.

Table 4-29. Interrupt Status Register (INTSTS, Address = 0x33)

Bit	Name	Type	Default Value	Description
7	-	RO	0	Reserved
6	RT	RW1C	0	Receive Trigger This bit is set when FTCSIRX100 receives the trigger events from the PPI signal, RxTriggerEsc. The bus value of RxTriggerEsc can be found in the register ESR1 (Address = 0x35).
5	ULPS_E	RW1C	0	ULPS Entry This bit is set when either the enabled data lanes or clock lane enters ULPS. This event is determined by detecting the assertion of UlpsActiveNot and ClkUlpsActiveNot. The enabled data lanes mean the data lanes with the corresponding ppi_Enable bit being active.
4	ULPS_X	RW1C	0	ULPS Exit This bit is set when all data lanes and clock lane exit from ULPS. This event is determined by detecting the deassertion of UlpsActiveNot and ClkUlpsActiveNot. Here the enabled data lanes mean the data lanes with the corresponding ppi_Enable bit being active.
3	FS	RW1C	0	Receive Frame Start Packet This bit is set when FTCSIRX100 receives a frame start packet.

Bit	Name	Type	Default Value	Description
2	DPI	RW1C	0	DPI Events This bit reflects the occurrence of DPI events. The DPI events are logged in the register DPISR***n*** (***n*** = 0, 1, 2, or 3, Address = 0x38 + ***n***). Combining the setting of the INTER.DEM bit (Address = 0x3C), this bit behaves as: When INTER.DEM is set, INTSTS.DPI asserts only when DPISR***n***.DPI_OF (Address = 0x38 + ***n***) is set. Otherwise, INTSTS.DPI can assert when either bit of the register DPISR***n*** is set. Please note that Write-one-clear this bit can also clear the registers DPISR***n***.
1	-	RO	0	Reserved
0	CSI	RW1C	0	CSI Error Events This bit is set when the CSI error event occurs. The CSI error events are reported in the registers, CSIERR0 (Address = 0x30) and CSIERR1 (Address = 0x31). Please note that write-one-clear this bit can also clear the CSI Error Report Registers 0 and 1 (CSIERR0 and CSIERR1).

4.2.19 Escape Mode and Stop State Register (ESR, Address = 0x34 ~ 0x35)

These registers are used to record the PPI signals about the escape mode and stop state so the software can reach these signals state at PPI through APB or I^2C interface.

Table 4-30. Escape Mode and Stop State Register 0 (ESR0, Address = 0x34)

Bit	Name	Type	Default Value	Description
[7:4]	STOP	RO	HwInit	Data Lane Stopstate This field reflects the input PPI signals Stopstate for the logical data lanes. Bit 4 indicates the PPI signal Stopstate for the logical data lane#0; bit 5 indicates the PPI signal Stopstate for the logical data lane#1; bit 6 indicates the PPI signal Stopstate for the logical data lane#2; bit 7 indicates the PPI signal Stopstate for the logical data lane#3.
[3:0]	ULPS	RO	HwInit	Data Lane UlpsActiveNot This field reflects the input PPI signals UlpsActiveNot for the logical data lanes. Bit 0 indicates the PPI signal UlpsActiveNot for the logical data lane#0; bit 1 indicates the PPI signal UlpsActiveNot for the logical data lane#1; bit 2 indicates the PPI signal UlpsActiveNot for the logical data lane#2; and bit 3 indicates the PPI signal UlpsActiveNot for the logical data lane#3.

Table 4-31. Escape Mode and Stop State Register 1 (ESR1, Address = 0x35)

Bit	Name	Type	Default Value	Description
7	CSTOP	RO	HwInit	Clock Lane Stopstate This bit reflects the PPI signal ClkStopState.
6	CUAN	RO	HwInit	Clock Lane ULP State (Not) Active The bit reflects the PPI signal ClkUlpsActiveNot.
5	RUE	RO	HwInit	Data Lane Escape Ultra-Low Power (Receive) Mode This bit reflects the PPI signal RxUlpsEsc for the logical data lane#0.
4	RUCN	RO	HwInit	Clock Lane Receive Ultra-Low Power State This bit reflects the PPI signal RxUlpsClkNot.
[3:0]	RTE	RO	HwInit	Receive Trigger Event The value of this field is the bus content of the PPI signal RxTriggerEsc for the logical data lane#0.

4.2.20 DPI Status Register (DPISR, Address = 0x38 ~ 0x3B)

This register records the status occurred at Virtual Channel ***n*** of DPI. Here ***n*** is the virtual channel number which can be 0, 1, 2, or 3. The DPI VC***n*** Status Register only exists when the hard macro FTCSIRX100_VC***n*** is defined. These DPI events are not affected by the FRCR register (Address = 0x80 ~ 0x87) setting. That is, the DPI events are still reflected in this register even if the corresponding frame is not enabled by FRCR register.

Table 4-32. DPI VC*n* Status Register (DPISR*n*, Address = 0x38 + *n*)

Bit	Name	Type	Default Value	Description
[7:6]	-	RO	0	Reserved
5	DPI_OF	RW1C	0	DPI FIFO Overflow When this bit is set, it indicates that DPI VC***n*** FIFO is overflow.
4	DEOV	RW1C	0	Data Enable Overlaps Vsync Signal When this bit is set, it indicates that DPI Data Enable (dpi_vc***n***_DE) overlaps the Vertical Synchronization Timing (dpi_vc***n***_Vsync) at DPI. Otherwise, this event doesn’t occur.
3	DEOH	RW1C	0	Data Enable Overlaps Hsync Signal When this bit is set, it indicates that DPI Data Enable (dpi_vc***n***_DE) overlaps the Horizontal Synchronization Timing signal (dpi_vc***n***_Hsync) at DPI. Otherwise, this event doesn’t occur.

Bit	Name	Type	Default Value	Description
2	PNE	RW1C	0	Pixel Number Exceeding When this bit is set, it indicates that the pixel number of the VC***n*** received horizontal line exceeds the setting of the Horizontal Pixel Number Register (HPNR***n***). Otherwise, this bit isn't set. This bit is only effective when the value of HPNR***n*** registers is correct.
1	-	RO	0	Reserved
0	DDE	RW1C	0	Discontinuous Data Enable When this bit is set, it indicates that the discontinuous DPI Data Enable (dpi_vc***n***_DE) occurs during the pixel data (Excluding generic 8-bit long packets, user defined 8-bit data packets and short packets) transmission of a horizontal line at DPI. Otherwise, this event doesn't occur. This bit is only effective when the value of the Horizontal Pixel Number Register (HPNR***n***) registers is correct.

4.2.21 Interrupt Enable Register (INTER, Address = 0x3C)

Except bit#7 and bit#1, this register controls the interrupt enable. It affects the output signal csi_rx_int_r. Please refer to Table 2-7 for details. In Table 4-30, ***n*** is the virtual channel number which can be 0, 1, 2, or 3.

Table 4-33. Interrupt Enable Register (INTER, Address = 0x3C)

Bit	Name	Type	Default Value	Description
7	ACDF	RW	0	Automatically Clear DPI FIFO This it is not the interrupt enable bit. When this bit is set and VC***n*** frame start packet is received, FTCSIRX100 will clear the VC***n*** DPI FIFO. Otherwise, receiving VC***n*** frame start packet won't clear VC***n*** DPI FIFO.
6	RT	RW	1	Receive Trigger This is the interrupt enable bit for receiving the trigger event at the logical data lane#0 (INTSTS.RT). When this bit is set, the interrupt of this event will be enabled. Otherwise, it will be disabled.
5	ULPS_E	RW	0	ULPS Entry This bit is the interrupt enable bit for the event of ULPS entry (INTSTS.ULPS_E). When this bit is set, the interrupt of this event will be enabled. Otherwise, it will be disabled.

FARADAY

Bit	Name	Type	Default Value	Description
4	ULPS_X	RW	0	ULPS Exit This bit is the interrupt enable bit for the event of ULPS exit (INTSTS.ULPS_X). When this bit is set, the interrupt of this event will be enabled. Otherwise, it will be disabled.
3	FS	RW	0	Receive Frame Start Packet This bit is the interrupt enable bit for the event of receiving the frame start packet (INTSTS.FS). When this bit is set, the interrupt of this event will be enabled. Otherwise, it will be disabled.
2	DPI	RW	0	DPI Events This bit is the interrupt enable bit of DPI events (INTSTS.DPI). When this bit is set, the interrupt of this event will be enabled. Otherwise, it will be disabled.
1	DEM	RW	0	DPI Event Mask This it is not the interrupt enable bit. This bit effects the register bit INTSTS.DPI (Address = 0x33). When this bit is set, INTSTS.DPI asserts only when DPISR***n***.DPI_OF (Address = 0x38 + ***n***) is set. Otherwise, INTSTS.DPI can assert when either bit of the register DPISR***n*** is set.
0	CSI	RW	1	CSI Error Events This bit is the interrupt enable bit of CSI error events (INTSTS.CSI). When this bit is set, the interrupt of this event will be enabled. Otherwise, it will be disabled.

4.2.22 FTCSIRX100 Feature Register (FFR, Address = 0x40 ~ 0x47)

These registers are read-only. The FFR0, FFR6 and FFR7 registers reflect the hard macro definitions . The other registers (FFR1 ~ FFR5) are only for Faraday internal use.

Table 4-34. FTCSIRX100 Feature Register 0 (FFR0, Address = 0x40)

Bit	Name	Type	Default Value	Description
[7:4]	DFAD	RO	HwCfg	DPI FIFO Address Depth The value of this field is *p* which is the hard macro value of FTCSIRX100_AP_VC0_ADDR_DEPTH. This hardmacro is only for internal use. The default value of this hardmacro is 0x5.

Bit	Name	Type	Default Value	Description
[3:2]	LN	RO	HwCfg	Lane Number The value of this field reflects the hardmacro (FTCSIRX100_LANE_CFG) configuration for the number of the data lane. The default value of this hardmacro is 0b11. The encoding of this field is 0b11: 4 data lanes 0b10: 3 data lanes 0b01: 2 data lanes 0b00: 1 data lane
[1:0]	VCN	RO	HwCfg	Virtual Channel Number The value of this field reflects the hardmacro (FTCSIRX100_VC_CFG) configuration for the number of the virtual channel. This hardmacro is the internal use only. The default value of this hardmacro is 0b11. The encoding of this field is 0b11: 4 virtual channels 0b10: 3 virtual channels 0b01: 2 virtual channels 0b00: 1 virtual channel

Table 4-35. FTCSIRX100 Feature Register 1~5 (FFR1, Address = 0x41; FFR2, Address = 0x42; FFR3, Address = 0x43; FFR4, Address = 0x44; FFR5, Address = 0x45)

Bit	Name	Type	Default Value	Description
[7:0]	CS	RO	HwInit	The value of this register is ignored. These registers are for internal-use only.

Table 4-36. FTCSIRX100 Feature Register 6 (FFR6, Address = 0x46)

Bit	Name	Type	Default Value	Description
7	ASC	RO	HwCfg	Hardmacro FTCSIRX100_ASC Indicator 1: This hardmacro is defined. 0: This hardmacro is not defined.
6	APB	RO	HwCfg	Hardmacro FTCSIRX100_APB Indicator 1: This hardmacro is defined. 0: This hardmacro is not defined.
5	I2C	RO	HwCfg	Hardmacro FTCSIRX100_I2C Indicator 1: This hardmacro is defined. 0: This hardmacro is not defined.
4	DPCM	RO	HwCfg	Hardmacro FTCSIRX100_DPCM_EN Indicator 1: This hardmacro is defined. 0: This hardmacro is not defined.

Bit	Name	Type	Default Value	Description
3	CD	RO	HwCfg	Hardmacro FTCSIRX100_CHK_DATA Indicator 1: This hardmacro is defined. 0: This hardmacro is not defined.
2	DP	RO	HwCfg	Hardmacro FTCSIRX100_DUAL_PIXEL Indicator 1: This hardmacro is defined. 0: This hardmacro is not defined.
1	FRC	RO	HwCfg	Hardmacro FTCSIRX100_FRC Indicator 1: This hardmacro is defined. 0: This hardmacro is not defined.
0	PG	RO	HwCfg	Hardmacro FTCSIRX100_DPI_PG Indicator 1: This hardmacro is defined. 0: This hardmacro is not defined.

Table 4-37. FTCSIRX100 Feature Register 7 (FFR7, Address = 0x47)

Bit	Name	Type	Default Value	Description
[7:4]	-	RO	0	Reserved
3	MR	RO	HwCfg	Hardmacro FTCSIRX100_MONO_RGB_EN Indicator 1: This hardmacro is defined. 0: This hardmacro is not defined.
2	CRC	RO	HwCfg	Hardmacro FTCSIRX100_CRC_CHK Indicator 1: This hardmacro is defined. 0: This hardmacro is not defined.
1	SWAP	RO	HwCfg	Hardmacro FTCSIRX100_LANE_SWAP Indicator 1: This hardmacro is defined. 0: This hardmacro is not defined.
0	QP	RO	HwCfg	Hardmacro FTCSIRX100_ QUAD_PIXEL Indicator 1: This hardmacro is defined. 0: This hardmacro is not defined.

4.2.23 DPCM Register (DPCMR, Address = 0x48 ~ 0x4B)

This register only exists when the hardmacro, FTSCIRX100_DPCM_EN, is defined and the corresponding virtual channel exists. Only DPCM 10-8-10 and 12-8-12 decode schemes with the predictor1 are supported.

FTCSIRX100 doesn't support padding the pixels (Please refer to CR.EAPBL, Address = 0x04) and detecting the pixel number exceeding (Please refer to DPISR***n***.PNE, Address = 0x38 + ***n***, ***n*** is the virtual channel number) for receiving DPCM packets.

Table 4-38. VC*n* DPCM Register (DPCMR*n*, Address = 0x48+*n*)

Bit	Name	Type	Default Value	Description
7	DE	RW or RO	0	DPCM Enable When this bit is: 1: Enable the DPCM decoder. FTCSIRX100 will decode DPCM data and transmit the pixel at DPI. If DPCMR.DDSS = '0', FTCSIRX100 will decode DPCM as RAW10 pixel and transmit the pixel at DPI. If DPCMR.DDSS = '1', FTCSIRX100 will decode DPCM as RAW12 pixel and transmit the pixel at DPI. 0: Disable the DPCM decoder This register is read-only when VC***n*** doesn't exist.
6	DDSS	RW or RO	0	DPCM Decode Scheme Selection When this bit is: 0: 10-8-10 scheme is selected. 1:12-8-12 scheme is selected. This register is read-only when VC***n*** doesn't exist.
[5:0]	DPT	RW or RO	HwCfg	DPCM Packet Type This field indicates the data type is treated as the DPCM packet. The default value is defined by the hardmacro, FTCSIRX100_VC***n***_DPT This register is read-only when VC***n*** doesn't exist.

4.2.24 FTCSIRX100 Revision Register (FRR, Address = 0x4C ~ 0x4F)

This register is read-only and only for Faraday internal use.

Table 4-39. FTCSIRX100 Revision Register 0 (FRR0, Address = 0x4C)

Bit	Name	Type	Default Value	Description
[7:0]	B0	RO	HwCfg	Byte 0 of the hardmacro, FTCSIRX100_REVISION

Table 4-40. FTCSIRX100 Revision Register 1 (FRR1, Address = 0x4D)

Bit	Name	Type	Default Value	Description
[7:0]	B1	RO	HwCfg	Byte 1 of the hardmacro, FTCSIRX100_REVISION

Table 4-41. FTCSIRX100 Revision Register 2 (FRR2, Address = 0x4E)

Bit	Name	Type	Default Value	Description
[7:0]	B2	RO	HwCfg	Byte 2 of the hardmacro, FTCSIRX100_REVISION

Table 4-42. FTCSIRX100 Revision Register 3 (FRR3, Address = 0x4F)

Bit	Name	Type	Default Value	Description
[7:0]	B3	RO	HwCfg	Byte 3 of the hardmacro, FTCSIRX100_REVISION

4.2.25 Pixel FIFO Threshold Register (PFTR, Address = 0x50 ~ 0x53)

The pixel FIFO threshold is used to control the assertion of the DPI Data Enable (dpi_vc***n***_DE) signal for the first pixel of each horizontal line. Each virtual channel owns the individual pixel FIFO threshold.

Pixel FIFO threshold = (Value of the corresponding Pixel FIFO Threshold Register * 4) pixels

For example, when the Pixel FIFO Threshold Register 0 (PFTR0) = 0x5, the VC0 pixel FIFO threshold is 5 * 4 = 20 pixels. In this case, the corresponding VC (VC0) DPI Data Enable (dpi_vc***n***_DE) signal will assert to transmit the first pixel once the pixel FIFO contains at least 20 pixels and Hsync (output signal dpi_vc***n***_Hsync) signal is deasserting..

By default, there is no threshold for the DPI Data Enable (dpi_vc***n***_DE) signal. Consequently, DPI Data

Enable will assert to transmit the first pixel once the FIFO contains at least one pixel and Hsync (output signal dpi_vc***n***_Hsync) signal is deasserting. Please note that the threshold cannot exceed the entry number of its pixel FIFO.

Here, ***n*** is the virtual channel number which can be 0, 1, 2, or 3. The Pixel FIFO Threshold Register for VC***n*** will exist only when VC***n*** exist.

Table 4-43. Pixel FIFO Threshold Register for VC*n* (PFTR*n*, Address = 0x50 + *n*)

Bit	Name	Type	Default Value	Description
[7:0]	V	RW or RO	0	Pixel FIFO Threshold Setting The VC***n*** pixel FIFO threshold is (The value of this register * 4) pixels. This register is read-only when VC***n*** doesn't exist.

4.2.26 Frame Rate Control Register (FRCR, Address = 0x80 ~ 0x87)

Frame rate control registers include both "Frame Enable Register" and "Frame Control Wrap Register". The setting of these registers can control the corresponding frame transmitting at DPI. Both registers are read-only as zero when FTCSIRX100_FRC is not defined or VC***n*** doesn't exist.
When FCWR***n***.W = ***m***, it means FTCSIRX100 performs (***m***+1) frame enable bits wrapping for VC***n*** by using the register bits FER***n***.E[***m***:0]. The wrapping starts at the bit FER***n***.E[0] so the VC***n*** 1st frame is controlled by FER***n***.E[0].

For example, when FCWR0.W = 0b101 and FER0.E = 0b10111001, FTCSIRX100 will perform 6 frame enable bits wrapping by using FER0.E[5:0] for VC0. So the sequence of the frame enable will be FER0.E[0] (Enable), FER0.E[1] (Disable), FER0.E[2] (Disable), FER0.E[3] (Enable), FER0.E[4] (Enable), FER0.E[5] (Enable), FER0.E[0] (Enable), FER0.E[1] (Disable), …….

By default, FCWR***n***.W = 0 and FER***n***.E = 0xFF, FTCSIRX100 will depend on FER***n***.E[0] to transmit the frame so the frame transmitting is enabled by default.

Table 4-44. Frame Control Wrap Register for VC*n* (FCWR*n*, Address = 0x80 + *n2)**

Bit	Name	Type	Default Value	Description
[7:3]	-	RO	0	Reserved

FARADAY

Bit	Name	Type	Default Value	Description
[2:0]	W	RW or RO	0	Wrap Size This field indicates the wrap size. Please refer to the above description of this section. This field is read-only as zero when FTCSIRX100_FRC is not defined or VC***n*** doesn’t exist.

Table 4-45. **Frame Enable Register for VC*n* (FER*n*, Address = 0x81 + *n**2)**

Bit	Name	Type	Default Value	Description
[7:0]	E	RW or RO	0xFF	Frame Enable This field controls the corresponding frame enable. Please refer to the above description of this section. 1: Enable this frame transmitting. 0: Disable this frame transmitting This field is read-only as zero when FTCSIRX100_FRC is not defined or VC***n*** doesn't exist.

4.2.27 Frame Number Register (FNR, Address = 0x88 ~ 0x8F)

These registers reflect the value of the IP interface signal csi_vc***n***_fnum.

Table 4-46. **Frame Number Lower Register for VC*n* (FNLR*n*, Address = 0x88 + *n**2)**

Bit	Name	Type	Default Value	Description
[7:0]	N	RO	HwInit	Frame Number This field reflects the value of csi_vc***n***_fnum[7:0]. This field is read-only as zero when VC***n*** doesn't exist.

Table 4-47. **Frame Number Higher Register for VC*n* (FNHR*n*, Address = 0x89 + *n**2)**

Bit	Name	Type	Default Value	Description
[7:0]	N	RO	HwInit	Frame Number This field reflects the value of csi_vc***n***_fnum[15:8]. This field is read-only as zero when VC***n*** doesn't exist.

4.2.28 DPI Built-in Pattern Generator Register (BPGR, Address = 0x90 ~ 0x97)

DPI built-in pattern generator register defines the vertical line number, generator enable, and pattern type selection. About the horizontal size of the built-in pattern, it is defined by HPNR registers (Address = 0x20 ~ 0x27). MCR register is still effective for DPI built-in pattern.

The vertical line size of the VC***n*** DPI built-in pattern (Unit is line) is the value of register {BPGHR***n***.VLN, BPGLR***n***.VLN} and the default value is configured by the hardmacro, FTCSIRX100_VC***n***_VLN. When this value is zero, the vertical line size of the VC***n*** DPI built-in pattern is 4096 lines.

These registers only exist when the hardmacro, FTCSIRX100_DPI_PG, is defined and the corresponding virtual channel ***n*** exists. Otherwise, the registers are read-only as zero.

Table 4-48. DPI Built-in Pattern Generator Lower Register for VC*n* (BPGLR*n*, Address = 0x90 + 2*n)

Bit	Name	Type	Default Value	Description
[7:0]	VLN	RW or RO	HwCfg	Vertical Line Number This field defines the bit#7~0 of the vertical line number. This field is read-only as zero when VC***n*** doesn't exist.

Table 4-49. DPI Built-in Pattern Generator Higher Register for VC*n* (BPGHR*n*, Address = 0x91 + 2*n)

Bit	Name	Type	Default Value	Description
[7:6]	PT	RW or RO	0	Pixel Type This field selects the pixel type of the built-in pattern. When this field is: 0b00: Pixel type is RAW8. 0b01: Pixel type is RAW10. 0b10: Pixel type is RAW12. 0b11: Pixel type is RAW14. This field is read-only as zero when VC***n*** doesn't exist.
5	PS	RW or RO	0	Pattern Selection This bit selects the built-in pattern type. When this bit is: 0: No swap pattern. 1: Swap pattern per 15 frames. This field is read-only as zero when VC***n*** doesn't exist.

Bit	Name	Type	Default Value	Description
4	GE	RW or RO	0	Generator Enable This bit controls the built-in pattern generator enable. When this bit is: 0: Disable the built-in pattern generator. 1: Enable the built-in pattern generator. This field is read-only as zero when VC***n*** doesn't exist.
[3:0]	VLN	RW or RO	HwCfg	Vertical Line Number This field defines the bit#11 ~ 8 of the vertical line number. This field is read-only as zero when VC***n*** doesn't exist.

Chapter 5

Function Description

This chapter contains the following sections:

5.1 Layer Architecture

FTCSIRX100 contains the layer architecture. The layers are media access layer, protocol layers and application layers.

5.1.1 Media Access Layer (ML)

CSI is a lane-scalable interface. When the applications require more bandwidth than the provided one data lane, it can expand the data path to two-lane, three-lane or four-lane wide and obtain approximately linear increase in the peak bus bandwidth. Media Access layer (ML) collects bytes from the lanes and merges the bytes into a recombined data stream that restores the original stream sequence.

5.1.2 Protocol Layer (PL)

This layer receives the transmission from ML and transmits the headers and data of the packets to the application layer (AP). The protocol layer only contains the single clock domain (csi_clk).

5.1.3 Application Layer (AP)

The application layer comprehends as below:

- Decode CSI Frame Start, Frame End, Line Start, Line End short packet commands to the DPI Interface
- Manage the received pixel format data to DPI
- Support different DPI pin mapping
- Multiple virtual channel handling
- Record unexpected DPI behavior

5.2 PHY Interface

FTCSIRX100 follows both MIPI specifications of CSI-2 [MIPI02] and PPI D-PHY [MIPI01]. It uses the PPI to connect with D-PHY.

5.2.1 Data Lane Configuration

The data lane hardware configurations can be found in Table 3-3. Please refer to Section 5.10 for details.

5.2.2 Requirements for PPI Signals

In additional to the PPI standard in D-PHY specification [MIPI01], the following items are requirements for the PPI signals:

a. RxSyncHS must be asserted prior to the assertion of RxValidHS and de-asserted when RxValidHS is active high.
b. ErrSotSyncHS must not be asserted after the RxSyncHS signal is asserted. If FTCSIRX100 gets the error flag, the later data will be dropped until the next normal reception of the PPI High-Speed Receive at the Slave side.
c. For the high-speed data transmission of the multiple data lanes, the first byte of each data lane shall be valid at the same RxByteClkHS clock cycle.
d. The high-speed receive signals (RxDataHS, RxValidHS, RxActiveHS, and RxSyncHS) of the multiple data lanes are all synchronous with the same RxByteClkHS clock.

5.3 Supported Packet Types

The supported packet types are listed in Table 5-1. The length of each packet must follow the data size constraints, which are defined in the CSI-2 specification [MIPI02].

Table 5-1. Supported Packet Types

Data Type	Description
0x00	Frame Start
0x01	Frame End
0x02	Line Start
0x03	Line End
0x08	Generic Short Packet Code 1
0x09	Generic Short Packet Code 2
0x0A	Generic Short Packet Code 3
0x0B	Generic Short Packet Code 4

Data Type	Description
0x0C	Generic Short Packet Code 5
0x0D	Generic Short Packet Code 6
0x0E	Generic Short Packet Code 7
0x0F	Generic Short Packet Code 8
0x10	Generic 8-bit Long Packet - Null
0x11	Generic 8-bit Long Packet – Blanking Data
0x12	Generic 8-bit Long Packet – Embedded 8-bit Non Image Data
0x13	Generic 8-bit Long Packet – Reserved
0x14	Generic 8-bit Long Packet – Reserved
0x15	Generic 8-bit Long Packet – Reserved
0x16	Generic 8-bit Long Packet – Reserved
0x17	Generic 8-bit Long Packet – Reserved
0x1E	YUV422 8-bit Image Data
0x1F	YUV422 10-bit Image Data
0x22	RGB565 Image Data
0x24	RGB888 Image Data
0x2A	RAW8 Image Data
0x2B	RAW10 Image Data
0x2C	RAW12 Image Data
0x2D	RAW14 Image Data
0x30	User Defined 8-bit Data Type 1
0x31	User Defined 8-bit Data Type 2
0x32	User Defined 8-bit Data Type 3
0x33	User Defined 8-bit Data Type 4
0x34	User Defined 8-bit Data – Reserved
0x35	User Defined 8-bit Data – Reserved
0x36	User Defined 8-bit Data – Reserved
0x37	User Defined 8-bit Data – Reserved

5.4 CSI Error Reporting Events

These events in this section can be detected and handled only when the csi_clk clock exists. These events are described in the following subsections and indicated in the registers, CSIERR0 (Address = 0x30) and CSIERR1 (Address = 0x31).

5.4.1 High-Speed Receiver (HS-RX) Timeout Error

FTCSIRX100 has a HRX_TO timer to monitor the state of the data lanes. The timer will be initially loaded with the timeout value HRTVR (Address = 0x0A) and will be reloaded after all the receive data lanes leave the High-Speed mode (RxActiveHS is low) and will be decremented when RxActiveHS of either data lane keeps at high active.

The CSIERR0.HSRT bit (Address = 0x30) will be set when the HRX_TO timer expires, which means that the active duration of RxActiveHS will be longer than the setting of the register, HRTVR (Address = 0x0A). This error bit is useful for recovering from some transient errors that may result in common-mode fault. Consequently, the HRTVR register should be programmed to be longer than the maximum duration of a High-Speed transmission expected by FTCSIRX100.

This timer is clocked by csi_clk clock. If csi_clk clock doesn't exist, the timer will not count and this flag will not be effective.

5.4.2 ECC Checker and Correction

Each CSI packet contains the Error Correction Code (ECC). FTCSIRX100 can detect the ECC single-bit error or multiple-bit error. The single-bit error of ECC can be corrected and a 2-bit error is detectable in the packet header. When the ECC multiple-bit error is detected, it means that the uncorrectable error is found in the header of the packet. If the word count of the packet is not right, it can cause that the protocol layer lose the boundary of the packet boundary within this transmission.

If ECC multiple-bit error is detected, it will

- Drop the remainder of this packet and this transmission
- Set CSIERR1.ECCEM bit (Address = 0x31).

The ECC error reporting can be disabled by setting the register CR.ECCCE (Address = 0x04) as zero.

5.4.3 CRC Checker

If the checksum (CRC) error is detected, FTCSIRX100 sets the register bit, CSIERR1.CE (Address = 0x31). This function can be disabled by setting the register, CR.CRCCE (Address = 0x04), as zero.

5.4.4 Unsupported Packet Type

The "Unsupported Packet Type" error is caused by detecting a received Packet Type that is not supported by FTCSIRX100. The supported packet types are listed in Table 5-1. This error flag is indicated by the register bit, CSIERR1.UPT (Address = 0x31). If the word count of the long packet is zero, this packet is also treated as "Unsupported Packet Type".

5.4.5 CSI VC ID Invalid

The "CSI VC ID Invalid" error is caused by receiving a packet with an invalid VC ID.

5.4.6 Invalid Transmission Length

This error is indicated by the register, CSIERR1.ITL (Address = 0x31). The flag will be asserted when either of events occurs:

- Multi-bit ECC error
- CRC error

5.5 DPI and the Extended Signals

FTCSIRX100 supports multiple virtual channels, in this section, '***n***' means the virtual channel number (***n*** = 0, 1, 2 or 3) and the corresponding term only exists when the hard macro, FTCSIRX100_VC***n***, is defined. Each VC owns individual vc***n***_PCLK (DPI pixel clock, PCLK) clock domain, DPI and the extended signals. They are listed and described in Table 2-6.

5.5.1 Application Interface Modes

FTCSIRX100 provides the application interface modes, which can be configured to be the single-pixel mode, the dual-pixel mode, the quad-pixel mode, and DPCM mode. The pin mapping can be found in Table 2-10 through Table 2-18.

5.5.1.1 Single-pixel Mode

This mode means that FTCSIRX100 can transmit only single RAW pixel or byte at dpi_vc***n***_D bus per pixel clock. When both FTCSIRX100_DUAL_PIXEL and FTCSIRX100_QUAD_PIXEL are not defined, FTCSIRX100 will operate in this mode.

For the RGB image types, please refer to Section 5.5.2.4 for more detailed information. For the YUV image types, please refer to Section 5.5.2.2 for more detailed information.

5.5.1.2 Dual-pixel Mode

This mode means that FTCSIRX100 can transmit two RAW pixels or bytes at dpi_vc***n***_D bus per pixel clock. When the hardmacro, FTCSIRX100_DUAL_PIXEL, is defined, the dual-pixel mode is enabled. In this mode, the received image and non-image packets have some differences:

- For the image packets, RAW8, RAW10, RAW12, and RAW14, FTCSIRX100 transmits two pixels per pixel clock with dpi_vc***n***_DE = '1'. For the RGB image types, please refer to Section 5.5.2.4 for more detailed information. For the YUV image types, please refer to Section 5.5.2.2 for more detailed information.
- For the generic packets and user-defined 8-bit data packets, FTCSIRX100 transmits two bytes per pixel clock with dpi_vc***n***_gDE = '1'.

5.5.1.3 Quad-pixel Mode

This mode means that FTCSIRX100 can transmit four RAW pixels or bytes at dpi_vc***n***_D bus per pixel clock. When the hardmacro, FTCSIRX100_QUAD_PIXEL, is defined, the quad-pixel mode is enabled. In this mode, the received image and non-image packets have some differences:

- For the image packets, RAW8, RAW10, RAW12, and RAW14, FTCSIRX100 transmits four pixels per pixel clock with dpi_vc***n***_DE = '1'. For the YUV image types, FTCSIRX100 can transmit two pixels per pixel clock, please refer to Section 5.5.2.3 for more detailed information. For the RGB image types, FTCSIRX100 can transmit two pixels per pixel clock when the macro, FTCSIRX100_MONO_RGB_EN, is not defined. (Please refer to Section 5.5.2.5 for more detailed information.) FTCSIRX100 can transmit single pixel per pixel clock when the macro, FTCSIRX100_MONO_RGB_EN, is defined (Please refer to Section 5.5.2.4 for more detailed information.)
  For the generic packets and user-defined 8-bit data packets, FTCSIRX100 transmits four bytes per pixel clock with dpi_vc***n***_gDE = '1'.

5.5.1.4 DPCM Mode

In this mode, if FTCSITX100 transmits the image line at DPI and DPCMR.DE (Address = 0x48 ~ 0x4B) is active, the payload of this packet will be decoded according to the DPCM schemes and the setting of DPMCR register. Please refer to Section 4.2.23 for more detailed information. When the DPCM mode is enabled, to avoid the ambiguous conditions, please don't transmit DPCM packet from the MIPI CSI transmitter during the image blanking region.

5.5.2 Image Data Signal Mapping

The signal names of the FTCSIRX100 IP interface mapping to the DPI signals are shown in Table 5-2. FTCSIRX100 stores the pixels in the pixel FIFO and drives these signals to the DPI interface.

Table 5-2. DPI Signals Mapping to FTCSIRX100

Signal Name in DPI v2.0 Spec.	Signal Name in FTCSIRX100
Vsync	dpi_vc***n***_Vsync
Hsync	dpi_vc***n***_Hsync
DE	dpi_vc***n***_DE
PCLK	vc***n***_PCLK

Signal Name in DPI v2.0 Spec.	Signal Name in FTCSIRX100
D	dpi_vc***n***_D

After Hsync deasserting, when the SRAM buffer is not empty and meets the FIFO threshold, FTCSIRX100 will read the pixel FIFO and will drive the pixel to the DPI interface. The FIFO threshold is defined in the PFTR registers (Address = 0x50 ~ 0x53).

Depending on the pixel format and application interface modes, the image pixel bits mapping to dpi_vc***n***_D can be found in the tables of Chapter 2.

5.5.2.1 RAW Formats Mapping

For the RAW format, the pixel number of each horizontal line must be an even number and meet the CSI-2 specification [MIPI02].

FTCSIRX100 provides two kinds of the dpi_vc***n***_D bus alignment for the RAW formats, one is aligned to LSB LSB and the other is aligned to MSB, please refer to the

DPI Mapping Control Register (MCR, Address = 0x1C) for more detailed information.

Depending on the hardware configurations, the application modes can be single-pixel, dual-pixel, or quad-pixel mode. Each mode has the different mapping for the RAW formats. Please refer to the tables of Chapter 2.

5.5.2.2 Transmitting a Couple of YUV Pixels By Two Pixel Clocks

For both the single-pixel and dual-pixel modes, please refer to Figure 5-1 and Figure 5-2, FTCSIRX100 drives a couple of YUV pixels by two vc***n***_PCLK clocks with dpi_vc***n***_DE = '1'.

At the odd number pixel clock (vc***n***_PCLK) with dpi_vc***n***_DE = '1', Y-component (Y1, Y3, Y5 …) is driven at dpi_vc***n***_D[19:12] for YUV422 8-bit format and dpi_vc***n***_D[21:12] for YUV422 10-bit format. U-component (U1, U3, U5 …) is driven at dpi_vc***n***_D[7:0] for YUV422 8-bit format and dpi_vc***n***_D[9:0] for YUV422 10-bit format.

At the even number pixel clock (vc***n***_PCLK) with dpi_vc***n***_DE = 1, Y-component (Y2, Y4, Y6 …) is driven at dpi_vc***n***_D[19:12] for YUV422 8-bit format and dpi_vc***n***_D[21:12] for YUV422 10-bit format.

V-component (V2, V4, V6 …) is driven at dpi_vc***n***_D[7:0] for YUV422 8-bit format and dpi_vc***n***_D[9:0] for YUV422 10-bit format. Other bits of the dpi_vc***n***_D bus are not defined and are driven as zero.

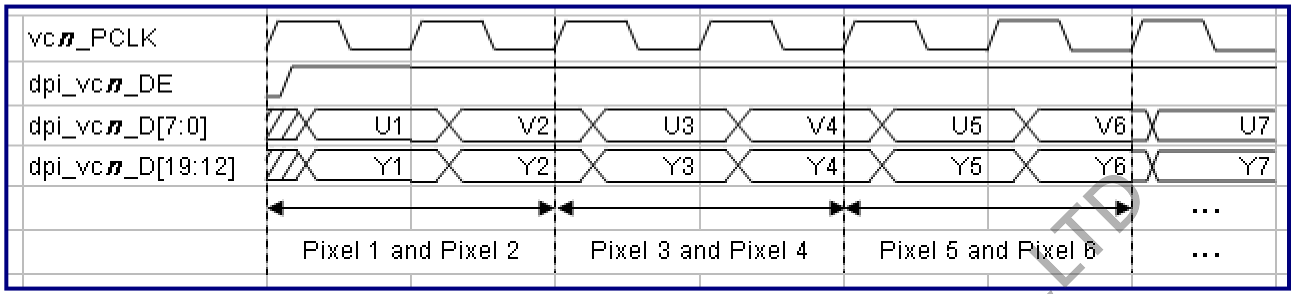

Figure 5-1. **Transmitting a Couple of YUV422 8-bit Pixels by Two Pixel Clock**

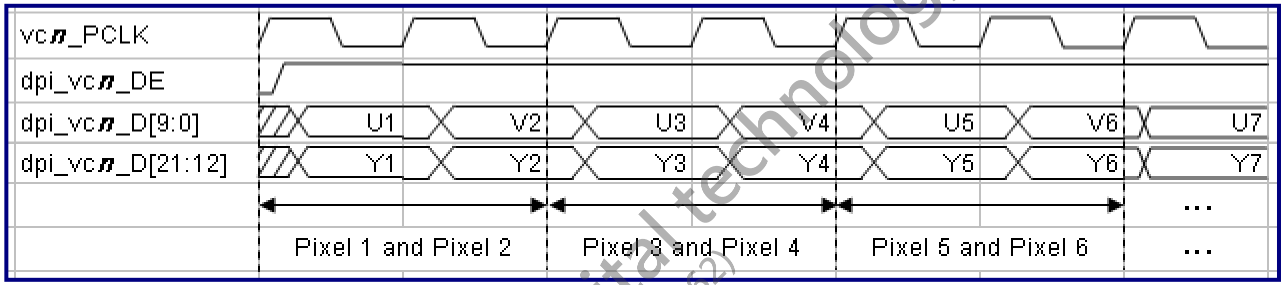

Figure 5-2. **Transmitting a Couple of YUV422 10-bit Pixels by Two Pixel Clock**

5.5.2.3 Transmitting a Couple of YUV Pixels by One Pixel Clocks

For the quad-pixel mode, please refer to Figure 5-3 and Figure 5-4. FTCSIRX100 transmits a couple YUV pixels by one vc***n***_PCLK clock with dpi_vc***n***_DE = 1.

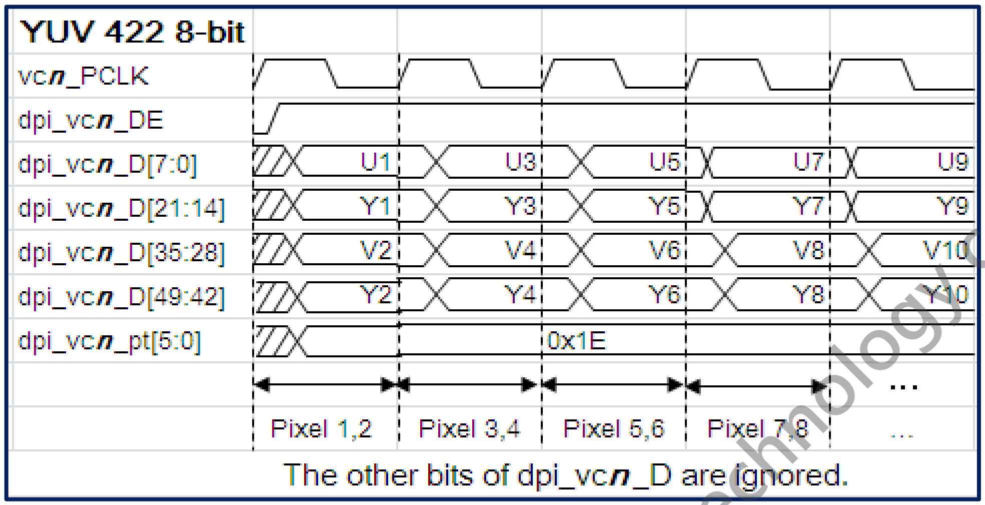

Figure 5-3. Transmitting a Couple of YUV422 8-bit Pixels by One Pixel Clock

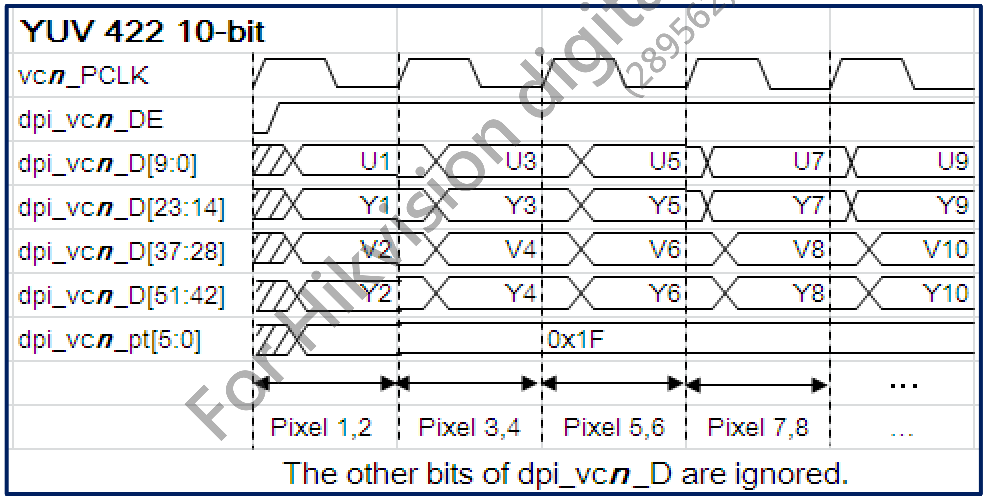

Figure 5-4. Transmitting a Couple of YUV422 10-bit Pixels by One Pixel Clock

5.5.2.4 Transmitting a Single RGB Pixel by One Pixel Clock

FTCSIRX100 can drive a single RGB pixel by one vc***n***_PCLK clock with dpi_vc***n***_DE = '1' under either of the below conditions:

- Single-pixel mode is configured.
- Dual-pixel mode is configured.
- Quad-pixel mode is configured and the macro, FTCSIRX100_MONO_RGB_EN, is defined.

5.5.2.5 Transmitting a Couple of RGB Pixels by One Pixel Clock

FTCSIRX100 can drive a couple of RGB pixels by one vc***n***_PCLK clock with dpi_vc***n***_DE = '1' only when the macro, FTCSIRX100_MONO_RGB_EN, is not defined and FTCSIRX100_QUAD_PIXEL is defined.

5.5.3 Generic Short Packet

For receiving the generic short packet with data type = 0x08 ~ 0x0F, FTCSIRX100 can indicate the packet. When csi_vc***n***_gDE is asserted, the content of the short packet data field can be found in dpi_vc***n***_D and the packet data type is indicated in csi_vc***n***_gt.

For the single-pixel mode as shown in Figure 5-5, Byte#1 is the least significant byte (First byte) of the short packet 16-bit data field and Byte#2 is the second byte of the short packet 16-bit data field.

For both the dual-pixel and quad-pixel modes as shown in Figure 5-6 , two bytes are outputted to the dpi_vc***n***_D bus.

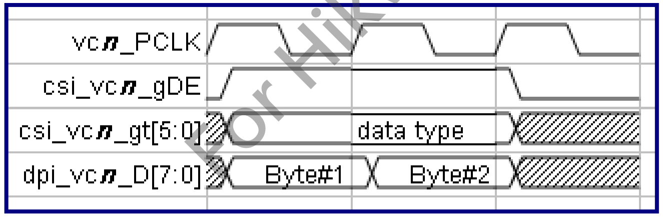

Figure 5-5. Single-pixel Mode Generic Short Packet Data Output

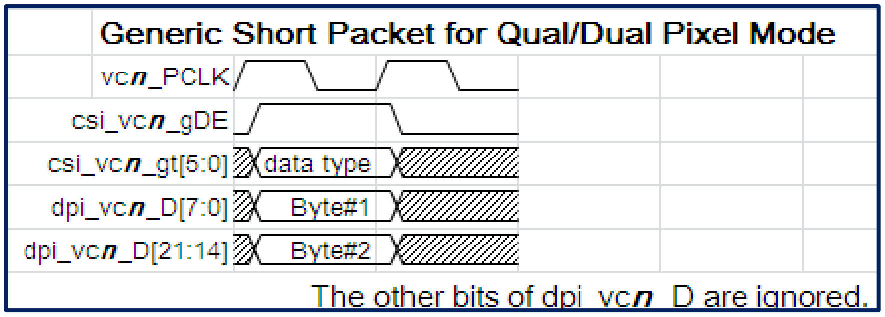

Figure 5-6. Dual-pixel and Quad-pixel Mode Generic Short Packet Data Output

5.5.4 Generic 8-bit Long Packet and User-defined Data Packet

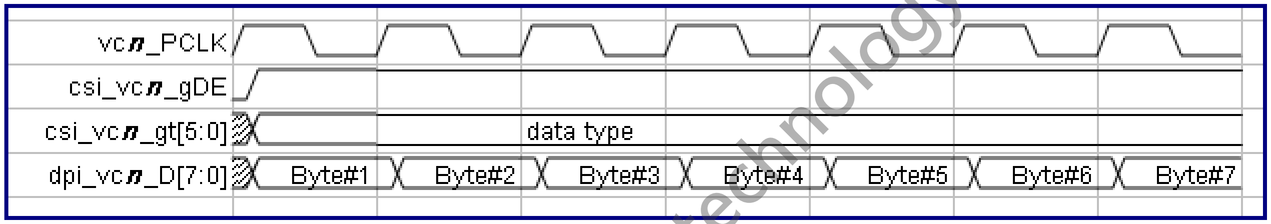

Figure 5-7. Single-pixel Mode Generic 8-bit Long Packet and User-defined Data Packet Output

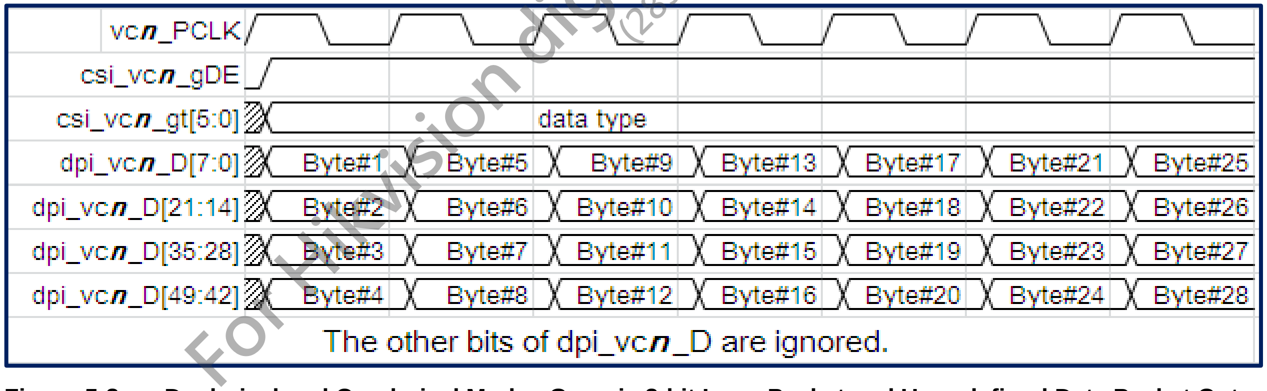
vc*n*_PCLK
csi_vc*n*_gDE
csi_vc*n*_gt[5:0]
data type
dpi_vc*n*_D[7:0]
Byte#1 Byte#5 Byte#9 Byte#13 Byte#17 Byte#21 Byte#25
dpi_vc*n*_D[21:14]
Byte#2 Byte#6 Byte#10 Byte#14 Byte#18 Byte#22 Byte#26
dpi_vc*n*_D[35:28]
Byte#3 Byte#7 Byte#11 Byte#15 Byte#19 Byte#23 Byte#27
dpi_vc*n*_D[49:42]
Byte#4 Byte#8 Byte#12 Byte#16 Byte#20 Byte#24 Byte#28
The other bits of dpi_vc*n*_D are ignored.

Figure 5-8. Dual-pixel and Quad-pixel Modes Generic 8-bit Long Packet and User-defined Data Packet Output

For receiving,

- the generic 8-bit long packet with data type = 0x12 (Embedded 8-bit non image data type) or 0x13~0x17 (Reserved) or
- the user-defined data packet with data type = 0x30 ~ 0x37,

FTCSIRX100 can indicate the packet as shown in Figure 5-7 and Figure 5-8. When csi_vc***n***_gDE is asserted, the data payload can be found in dpi_vc***n***_D and the packet data type is indicated in csi_vc***n***_gt. Byte#1 is the least significant byte (First byte) and Byte#2 is the second byte of the data payload.

For receiving generic 8-bit long packet with data type = 0x10 (Null data type) or 0x11 (Blanking data type), FTCSIRX100 ignores both data type packets and doesn't indicate at this interface.
When the DPCM mode is enabled, to avoid the ambiguous conditions, please don't transmit DPCM packet from MIPI CSI transmitter during the image blanking region.

5.6 Reset and Clock

FTCSIRX100 has several reset and clock signals. They are described in this section. For the multiple virtual channels in this section, '***n***' means the virtual channel number (***n*** = 0, 1, 2, or 3) and the corresponding term only exists when VC***n*** exists.

5.6.1 Reset

5.6.1.1 Hardware Reset Inputs

FTCSIRX100 has the reset inputs, "pwr_rst_n", "sys_rst_n", and "apb_rst_n". The "apb_rst_n" signal exists only when FTCSIRX100_APB is defined.
The "pwr_rst_n" reset signal is the power-on-reset, which resets all modules of FTCSIRX100.
The "sys_rst_n" reset signal can reset all modules of FTCSIRX100, except these parts - the internal registers, I^{2}C, APB and APB synchronization modules in FTCSIRX100.

The "apb_rst_n" reset signal is the Preset signal, which is defined in APB specification [APB]. It can reset the module of APB but doesn't reset the internal registers.

The reset synchronization is applied in the FTCSIRX100 internal modules. So the reset release edges of the flip-flops are synchronous with the corresponding clock.

5.6.1.2 Software Reset

Software can issue the software reset by the following two steps:
Step 1: Set the CR.SR bit.
Step 2: Polling the CR.SR bit until it is read as zero.
FTCSIRX100 performs the software reset as below when the related clocks (which are defined in Section 5.6.2) are valid. The software reset must be finished after the transactions of the above steps are finished.

- Reset the status registers: CSIERR0 (Address = 0x30), CSIERR1 (Address = 0x31), INTSTS (Address = 0x33), DPISR0 (Address = 0x38), DPISR1 (Address = 0x39), DPISR2 (Address = 0x3A), DPISR3 (Address = 0x3B) and FNR (Address = 0x88 ~ 0x8F).
- PPI interface returns to the initial state.
- Clear the pixel (DPI) FIFO(s) and the counter for the pixel number
- Clear timer counters mentioned in the register CSIERR0.HSRT bit (Address = 0x30)
- DPI interface signals return to the initial state.
- Clear csi_vc***n***_lnum and csi_vc***n***_fnum as zero
- Deassert csi_vc***n***_lineval
- Clear the interrupt csi_rx_int_r
- Assert the ForceRxmode signal when the values of the Initialization Timer Register (ITR, Address = 0x12 and 0x13) are not zero. Please refer to ITR register description for details.

5.6.2 Clocks

Depending on the number of the valid virtual channels, FTCSIRX100 contains several clock domains. The maximum supported frequencies are estimated by using the UMC 55-nm SP process library.
Each clock is described as below:

High-Speed Receive Byte Clock (RxByteClkHS)
High-speed receive byte clock is named as RxByteClkHS clock which is defined in D-PHY PPI [MIPI01]. This clock is used to sample the PPI signals and data for the PPI High-Speed Receive (HS-RX). FTCSIRX100 supports both the continuous and non-continuous clock schemes in the High-Speed (HS) mode. For the non-continuous clock behavior, the RxByteClkHS clock shall be valid before either of RxActiveHS signals goes high. If the hardmacro, FTCSIRX100_CHK_DATA, is not defined, RxByteClkHS clock can stop after the next clock of all RxActiveHS and RxValidHS signals are deasserted. If the hardmacro,

FTCSIRX100_CHK_DATA, is defined, RxByteClkHS clock can only be stopped after the next two clocks of all RxActiveHS and RxValidHS signals deasserting.

When RxByteClkHS clock is not valid, FTCSIRX100 cannot receive the data and setting CR.SR is not effective. The maximum frequency of this clock is 187.5 MHz.

Local Clock (csi_clk)

This input clock is mainly used in the ML, PL, AP, I^{2}C and APB modules. The maximum frequency of this clock is 187.5 MHz. For the throughput balance, the minimum frequency of this clock shall be faster than [the frequency of RxByteClkHS]*[(the number of active data lanes)/4]. If the I^{2}C function is used, this clock frequency must also meet the requirement in Section 5.8.2.

Escape Mode Receive Clock (RxClkEsc)

This is the PPI LP-RX byte clock which is defined in the MIPI D-PHY specification [MIPI01]. This clock is used to sample the PPI signals for the Escape Mode Receive Signals in PPI. FTCSIRX100 can support PPI LP-RX byte clock (RxClkEsc) stopping after the last valid data byte. The operation frequency of this clock shall be less than 1/4 of the csi_clk clock. According to D-PHY specification [MIPI01], the maximum data rate in the low power mode is 10 Mbps, so the maximum frequency of this clock is 1.25 MHz.

APB Clock (apb_clk)

This is the APB interface clock. When the hard macro, FTCSIRX100_ASC, is defined, apb_clk can be asynchronous to csi_clk and the related logic also exists for the signal synchronization. Otherwise, apb_clk must be synchronous to csi_clk. The maximum frequency of this clock is 200 MHz.

Pixel Clock (vc*n*_PCLK)

This clock is used to synchronize the signals at DPI and the generic interface. Each VC owns a dedicated PCLK input. The maximum frequency of this clock is 375 MHz for the single-pixel and dual-pixel modes; 187.5 MHz for the quad-pixel mode.

For the throughput balance, the minimum operation frequency of this clock ($F_{p\text{-}min)}$ shall be:

For “the single-pixel mode, RAW pixel formats” or “transmitting a single RGB pixel by one pixel clock” or “transmitting a couple YUV pixels by two pixel clocks”,

$F_{p\text{-}min}$ = ($F_{RxByteClkHS}$ * 8 *Active_Lane_Num)/bbp;

Or

For "the dual-pixel mode, RAW pixel formats" or "transmitting a couple of RGB by one pixel clock" or "transmitting a couple YUV pixels by one pixel clock",

$F_{p\text{-}min}$ = ($F_{RxByteClkHS}$ * 8 *Active_Lane_Num)/ (2*bbp);

Or

For "the quad-pixel mode, RAW pixel formats",

$F_{p\text{-}min}$ = ($F_{RxByteClkHS}$ * 8 *Active_Lane_Num)/ (4*bbp);

Terms description:

$F_{RxByteClkHS}$: The operation frequency of RxByteClkHS clock

Active_Lane_Num: The number of the active data lanes

bbp: bits per pixel transmitted at DPI or generic interface of FTCSIRX100.

For example:

- "Generic 8-bit Long Packet Data Types" or "User Defined Byte-based Data Types", bbp_{min} = 8.
- RGB888, bpp = 24; YUV422 8-bit, bpp = 16; RAW14, bpp = 14.

5.7 APB Supporting

The APB supporting is described at the hard macro descriptions of FTCSIRX100_ASC and FTCSIRX100_APB in Table 3-4.

5.8 I^{2}C Usage

FTCSIRX100 provides I^{2}C slave and implements this design in csi_clk clock domain. The section describes the spike suppression and the local clock requirement for I^{2}C function.
In this section, let the value of csi_gsvalue = G and the period of csi_clk clock = T.

5.8.1 Spike Suppression

This function is enabled when the input bus signal csi_gsvalue is hardwired as a non-zero value. Both csi_scl_in and csi_sda_in input signals' spikes with the pulse width ≤ G*T are suppressed. Since this function is implemented by the digital synchronous design, the pulse width is identified by csi_clk sampling the consecutive signal level. The narrow spike whose pulse width ≤ 1 * csi_clk period may not be detected. If the characteristics input I/O stages of I^{2}C SCL (The source of csi_scl_in) and SDA (The source of csi_sda_in) both meet the Table 4 of I^{2}C bus specification [I2C] or the spike never appears at SCL and SDA signals, both csi_scl_in and csi_sda_in signals shall not have the spike and this function can be disabled (Hardwire csi_gsvalue as zero).

Here are the examples of csi_gsvalue = 2 to show the spike suppression,
In Figure 5-9, the spikes of pulse width ≤ 2 * csi_clk periods are suppressed. The signals, csi_sda_ip and csi_scl_ip, in the figures of this section indicate that the I^{2}C inputs are referred by FTCSIRX100.

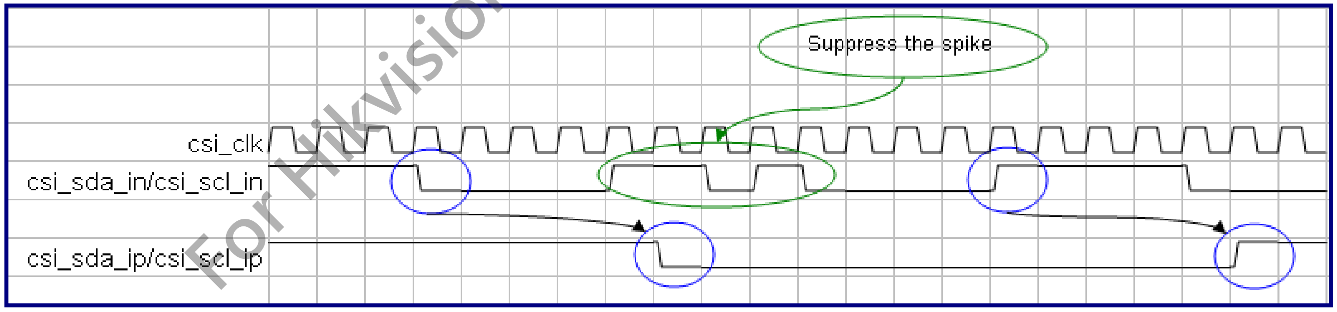

Figure 5-9. Suppress Spike at I^{2}C

In Figure 5-10, the narrow spikes cannot be detected.

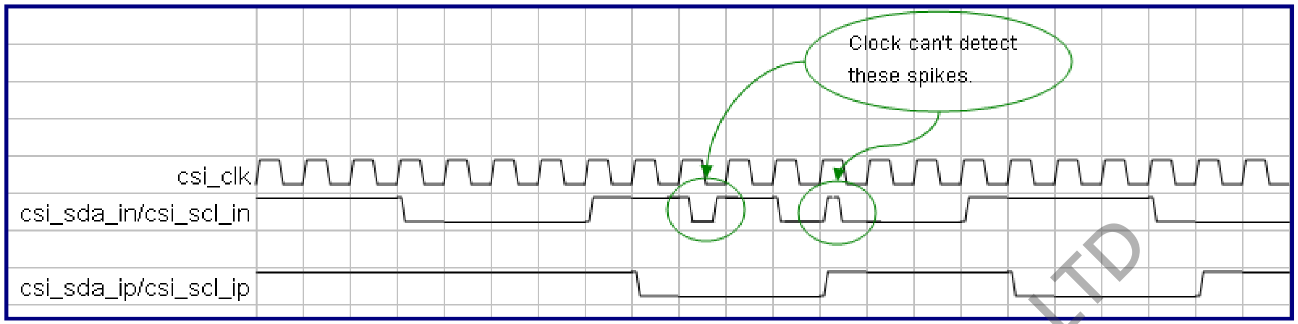

Figure 5-10. Narrow Spike at I²C

Figure 5-11 shows the signal latency when the csi_gsvalue = 2. The latency of SDA and SCL signals in the FTCSIRX100 internal module ranges from (G + 3) * T to (G + 5) * T.

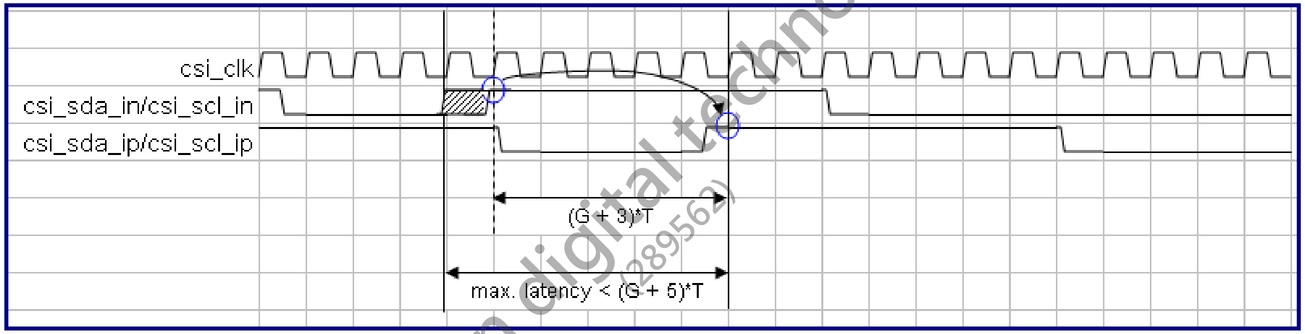

Figure 5-11. I²C Signal Latency for csi_gsvalue = 2

5.8.2 Local Clock Requirement for I²C Bus

When csi_gsvalue bus is hardwired as zero, the csi_clk clock frequency must be faster than 8.34 MHz for I²C fast mode and 1.25 MHz for I²C standard mode.

When csi_gsvalue is hardwired as the value of G,
The frequency for I²C standard mode shall be faster than both 1.25 MHz and [(8+G)/4.45] MHz.
The frequency for I²C fast mode shall be faster than both 8.34 MHz and [(8+G)/1.2] MHz.

For example, when G = 2,
The frequency for I²C standard mode shall be faster than 2.25 MHz.
The frequency for I²C fast mode shall be faster than 8.34 MHz.

5.9 PPI Interface Checking Test Mode

FTCSIRX100 provides a test mode for checking the PPI interface connection between FTCSIRX100 and D-PHY. The checking scheme is shown in Figure 5-12 and the hardware of this test mode only exists when the hard macro, FTCSIRX100_CHK_DATA, is defined. This test mode can be enabled by either of software or hardware. The software method is to set ECR.PCE bit (Address = 0x06) and the hardware method is to assert the input signal csi_chk_req. The input signal, csi_chk_req, must be a level signal and glitch-free. The signals of the IP connection mapping to Figure 5-12 are listed in Table 5-3. When the test mode is enabled, the input signal, ppi_Enable, will be ignored and all data lanes of FTCSIRX100 are checking.

In this section, the data lane means the physical data lane. Please keep the DLMR register as the default value when this test mode is active.

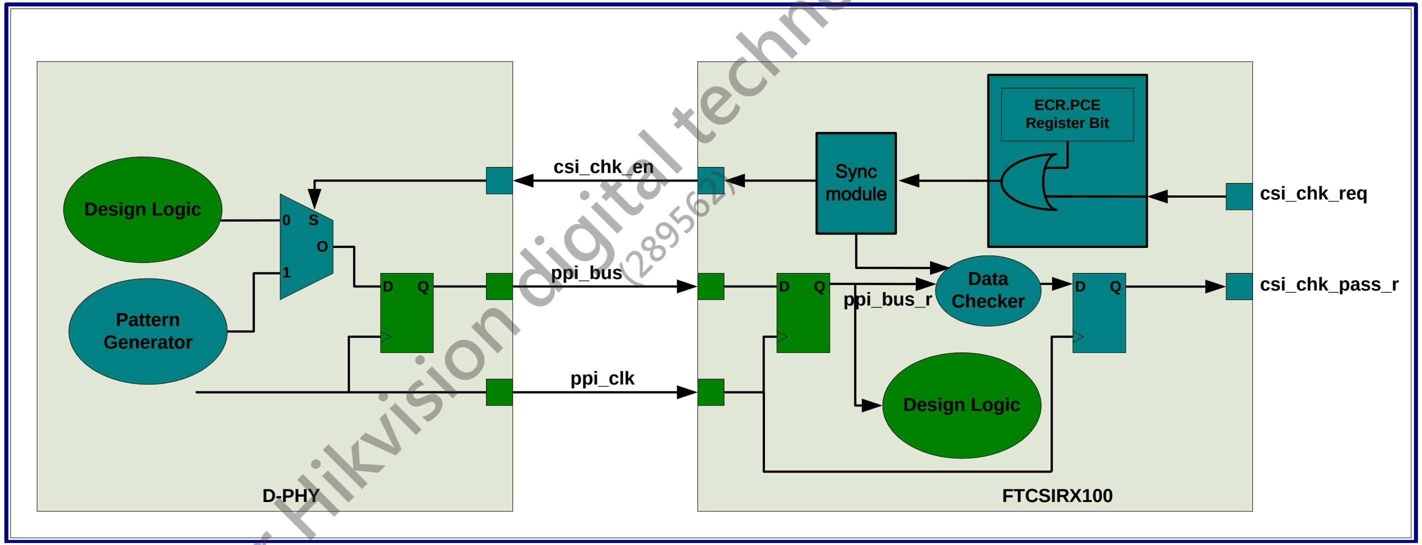

Figure 5-12. PPI Interface Checking Scheme

Table 5-3. Signal Mapping for PPI Interface Checking Scheme

Signal in Figure 5-12	FTCSIRX100 IP Signal
csi_chk_en	csi_chk_en_sych
csi_chk_req	csi_chk_req
csi_chk_pass_r	csi_chk_pass_r (Note)

Signal in Figure 5-12	FTCSIRX100 IP Signal
ppi_bus	For the data lane#***n***: {ErrSotHS[***n***], ErrSotSyncHS[***n***], RxSyncHS[***n***], RxActiveHS[***n***], RxValidHS[***n***], RxDataHS[***n****8+7:***n****8]}
ppi_clk	RxByteClkHS
P(Pattern for comparing)	0x5a5 for each data lane
~P(Pattern for comparing)	0x1a5a for each data lane

Note: csi_chk_pass_r will be asserted when ppi_bus of all data lanes existed in the FTCSIRX100 hardware match with patterns, P and ~P, as shown in Figure 5-13 or Figure 5-14.

The data checking schemes are shown in Figure 5-13 through Figure 5-15. If ppi_clk doesn't exist, these schemes can't work.

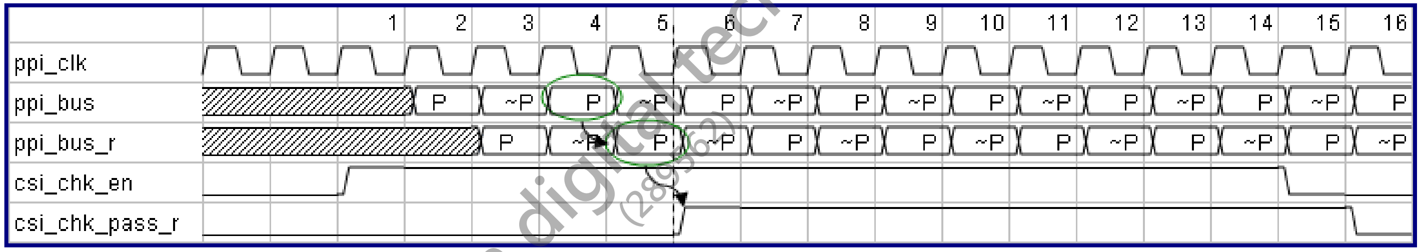

Figure 5-13. PPI Interface Data Checking Pass Case with Patterns = {P, ~P}

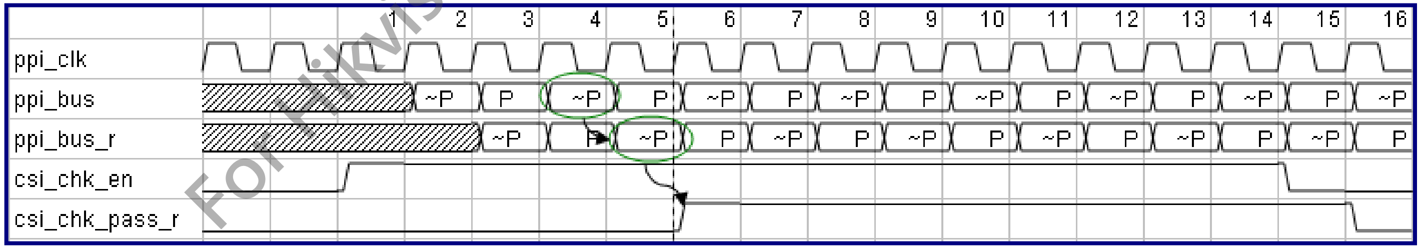

Figure 5-14. PPI Interface Data Checking Pass Case with Patterns = {~P, P}

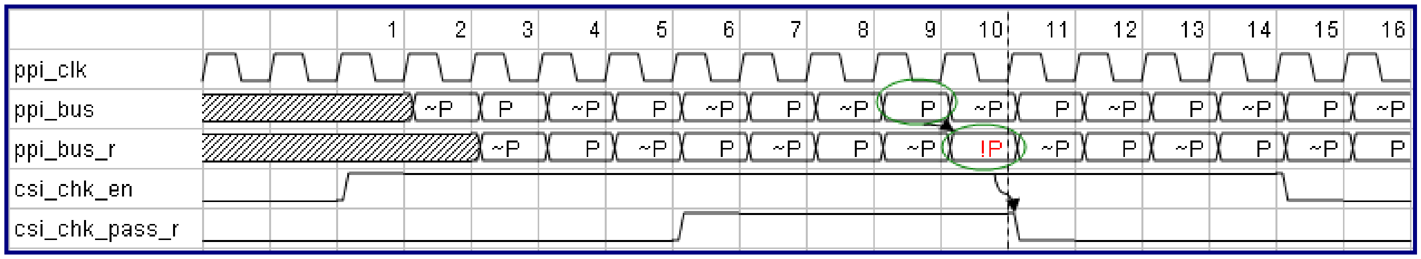

Figure 5-15. PPI Interface Data Checking Fail Case

Figure 5-13 and Figure 5-14 show the pass cases and Figure 5-15 shows the fail case.

Here are the notes:

- After the software or hardware enables the test mode, FTCSIRX100 will assert the output signal, csi_chk_en, to indicate that the test mode is enabled.
- Input ppi_bus data shall be ready at the 4th clock after csi_chk_en is asserted.
- For Figure 5-13, the input data pattern is P for the even number clock (4th, 6th, 8th …) and ~P for the odd number clock (3rd, 5th, 7th …) after csi_chk_en is asserted.
- For Figure 5-14, the input data pattern is ~P for the even number clock (4th, 6th, 8th …) and P for the odd number clock (3rd, 5th, 7th …) after csi_chk_en is asserted.
- FTCSIRX100 latches ppi_bus to ppi_bus_r by the ppi_clk clock and compares the expected data with ppi_bus_r. FTCSIRX100 begins to compare the data at the 5th clock after asserting csi_chk_en so that csi_chk_pass_r will be asserted at the 6th clock, as shown in Figure 5-13 or Figure 5-14.
- The output signal, csi_chk_pass_r, keeps as high when FTCSIRX100 constantly receives the expected data at ppi_bus_r until csi_chk_en is deasserted as shown in Figure 5-13 or Figure 5-14.
- Once the data mismatch is found, such as byte at 10th clock, as shown in Figure 5-15, csi_chk_pass_r will always be deasserted.
- Deasserting csi_chk_en can clear the test result and the csi_chk_pass_r flag.
- If the ppi_clk clock doesn't exist, this scheme can't work.

5.10 Determine Data Lane Enable

The hardmacro, FTCSIRX100_LANE_NUM, configures the number of the data lanes which exist in the hardware. Please refer to Table 3-3 for this hardmacro.

Either the ppi_Enable input signals or PECR.PEC register (Address= 0x28) can determine data lane enable for existed data lanes. To avoid the ambiguous conditions, they can not be used at the same time. The detailed rules are listed in the following subsections.

5.10.1 Data Lane Enable without Lane Swap

When FTCSIRX100_LANE_SWAP is not defined, the physical data lane is equivalent to the logical data lane. The data lane enable rules are:

- When ppi_Enable is used to determine the data lane enable, the value of register PECR.PEC must be set as zero. The accepted ppi_Enable values are 0xF, 0x7, 0x3, 0x1, and 0x0. ppi_Enable[***n***] = '1' means that the data lane#***n*** is treated as enabled by FTCSIRX100.
- When the PECR.PEC register is used to determine data lane enable, all bits of ppi_Enable must be tied to LOW. PECR.PEC[***n***] = '1' means that the data lane#***n*** is treated as enabled by FTCSIRX100.
- Output signal ftcsirx100_ppi_Enable[***n***] can connect to PPI "Enable" signal for D-PHY data lane#***n***. The bus width is always 4-bit and the corresponding bit of the non-existing data lane is driven as LOW. When the data lane#***n*** is treated as enabled by FTCSIRX100, ftcsirx100_ppi_Enable[***n***] = '1'; otherwise, ftcsirx100_ppi_Enable[***n***] = '0'. ftcsirx100_ppi_Enable[3:0] = 0b0000 means that no lane is enabled by this setting.
- Only five values of ppi_Enable for four data lanes configuration (DLW = 4) are accepted: 0b1111, 0b0111, 0b0011, 0b0001, 0b0000
- Only four values of ppi_Enable for three data lanes configuration (DLW = 3) are accepted: 0b111, 0b011, 0b001, 0b000
- Only three values of ppi_Enable for two data lanes configuration (DLW = 2) are accepted: 0b11, 0b01, 0b00
- Only two values of ppi_Enable for single data lane configuration (DLW = 1) are accepted: 1, 0

Here are some combinations for the examples in Table 5-4. In this table, N is the number of the enabled data lanes. For N = 4, all data lanes (#0~3) are enabled; N = 3 means that only the data lanes#0~2 are enabled; N = 2 means that only the data lanes#0~1 are enabled; N = 1 means only the data lane#0 is enabled; N = 0 means that none data lane is enabled.

Table 5-4. Examples for Data Lane Enable without Lane Swap

FTCSIRX100_LANE_NUM	PECR.PEC	ppi_Enable	ftcsirx100_ppi_Enable	N
4	0b0000	0b1111	0b1111	4
4	0b0000	0b0111	0b0111	3
4	0b0000	0b0011	0b0011	2
4	0b0000	0b0001	0b0001	1
4	0b1111	0b0000	0b1111	4

FTCSIRX100_LANE_NUM	PECR.PEC	ppi_Enable	ftcsirx100_ppi_Enable	N
4	0b0000	0b0000	0b0000	0
3	0b0000	0b111	0b0111	3
3	0b0011	0b000	0b0011	2
2	0b1111	0b00	0b0011	2
2	0b0000	0b01	0b0001	1
1	0b1111	0	0b0001	1

5.10.2 Data Lane Enable with Lane Swap

When FTCSIRX100_LANE_SWAP is defined, the physical data lane is mapped to the logical data lane by DLMR register. To describe the rules clearly, the terms "logical data lane" and "physical data lane" are defined in Section 5.11. The term "logical_ppi_Enable[***m***]" is defined as the corresponding ppi_Enable[***n***] of the logical data lane ***m***. For example, when DLW = 4 and the register DLMR1.L3 = 0, the data lane enable of the logical data lane 3 is logical_ppi_Enable[3] = ppi_Enable[0] (***m*** = 3, ***n*** = 0).

The data lane enable rules are:

- When ppi_Enable/logical_ppi_Enable is used to determine data lane enable, the value of register PECR.PEC must be set to '0'. The accepted logical_ppi_Enable values are 0xF, 0x7, 0x3, 0x1 and 0x0; logical_ppi_Enable[***m***] = '1' means that the logical data lane#***m*** is treated as enabled by FTCSIRX100.
- When PECR.PEC is used to determine data lane enable, all bits of ppi_Enable must be tied as LOW. PECR.PEC[***m***] = '1' means that the logical data lane#***m*** is treated as enabled by FTCSIRX100.
- Output signal ftcsirx100_ppi_Enable[***n***] can connect to PPI "Enable" signal for D-PHY physical data lane#***n***. The bus width is always 4-bit and the corresponding bit of the non-existing data lane is driven as LOW. FTCSIRX100 can automatically map logical data lane enable to the output ftcsirx100_ppi_Enable[***n***] of the physical data lane#***n***.
- Only five values of logical_ppi_Enable for four data lanes configuration (DLW = 4) are accepted: 0b1111, 0b0111, 0b0011, 0b0001, 0b0000
- Only four values of logical_ppi_Enable for three data lanes configuration (DLW = 3) are accepted: 0b111, 0b011, 0b001, 0b000
- Only three values of logical_ppi_Enable for two data lanes configuration (DLW = 2) are accepted: 0b11, 0b01, 0b00
- Only two values of logical_ppi_Enable for single data lane configuration (DLW = 1) are accepted: 1, 0

Here are some combinations for the examples in Table 5-5 and Table 5-6 . In this table, N is the number of the enabled data lanes. For N = 4, all logical data lanes (#0 ~ 3) are enabled; N = 3 means that only the logical data lanes#0~2 are enabled; N = 2 means that only the logical data lanes (#0 ~ 1) are enabled; N = 1 means that only the data lane#0 is enabled and DLMR0.L0 must be 0 for this case; N = 0 means that none data lane is enabled.

In Table 5-5, assuming the DLMR register setting is DLMR0.L0 = 3, DLMR0.L1 = 2, DLMR1.L2 = 1, and DLMR1.L3 = 0. Users can get logical_ppi_Enable[3:0] = {ppi_Enable[0], ppi_Enable[1], ppi_Enable[2], ppi_Enable[3]}.

Table 5-5. Examples for Data Lane Enable with Lane Swap, FTCSIRX100_LANE_NUM = 4

PECR.PEC	logical_ppi_Enable	ppi_Enable	ftcsirx100_ppi_Enable	N
0b0000	0b1111	0b1111	0b1111	4
0b0000	0b0111	0b1110	0b1110	3
0b0000	0b0011	0b1100	0b1100	2
0b0000	0b0001	0b1000	0b1000	1
0b1111	0b0000	0b0000	0b1111	4
0b0000	0b0000	0b0000	0b0000	0

In Table 5-6, assuming the DLMR register setting is DLMR0.L0 = 2, DLMR0.L1 = 0, and DLMR1.L2 = 1. Users can get logical_ppi_Enable[2:0] = {ppi_Enable[1], ppi_Enable[0], ppi_Enable[2]}.

Table 5-6. Examples for Data Lane Enable with Lane Swap, FTCSIRX100_LANE_NUM = 3

PECR.PEC	logical_ppi_Enable	ppi_Enable	ftcsirx100_ppi_Enable	N
0b0000	0b111	0b111	0b0111	3
0b0000	0b011	0b101	0b0101	2
0b0000	0b001	0b100	0b0100	1
0b0111	0b000	0b000	0b0111	3
0b0011	0b011	0b000	0b0101	2

5.11 Data Lane Sequence Swap

When the hardmacro, FTCSIRX100_LANE_SWAP, is turned on, FTCSIRX100 can support the feature of data lane sequence swap. The swap is configured by DLMR (Address = 0x2A and 0x2B). The rules are listed as follows and the behavior of FTCSIRX100 is undefined if the rules are violated.

Physical data lane

- The original data lane sequence from D-PHY.

Logical data lane

- The data lane number is assigned by DLMR registers. FTCSIRX100 follows this lane number for data lane sequence.
- If hardmacro, FTCSIRX100_LANE_SWAP, is turned off, the physical data lane is equivalent to logical data lane.

DLMR register setting

- The value of each filed can only be lower than the value of DLW (Please refer to Section 3.4). For example, if DLW = 2, DLMR0.L0 and DLMR0.L1 can only be set as 0 or 1. Again, if DLW = 3, the registers DLMR0.L0, DLMR0.L1 and DLMR1.L2 are never set as 3.
- The value of each field cannot be the same as each other.
- This register can not be dynamically changed during normal operating.

Data lane enable

Please refer to Section 5.10. It clarifies the relation of the data lane enable among pins ppi_Enable, ftcsirx100_ppi_Enable, register PECR.PEC.

PPI interface checking test mode

- Keep DLMR register as the default value during this test mode being active.

5.12 DPI Built-in Pattern Generation

FTCSIRX100 can output the built-in pattern to DPI when the generator enable bit (BPGR.GE, Address = 0x91+2****n***, ***n*** is the virtual channel number) is set. As shown in Figure 5-16 and Figure 5-17, the parameters of the pattern can be configured by the following registers:

1. Vsync active width(T_{vsa}) is controlled by the register VSTR (Address = 0x14, 0x16, 0x18, 0x1A). The unit is the DPI pixel clock regardless of the setting of VSCR (Address = 0x05).
2. Hsync active width(T_{hsa}) is controlled by the register HSTR (Address = 0x15, 0x17, 0x19, 0x1B). The unit is the DPI pixel clock.

FARADAY

3. Active pixel in a horizontal line(T_{adr}) is controlled by the register HPNR (Address = 0x20 ~ 0x27).
4. Horizontal front porch(T_{hfp}) is fixed to "0"; i.e. DE deassert and Hsync assert at the same DPI pixel clock.
5. Active vertical lines per frame is controlled by the register BPGR (Address = 0x90 ~ 0x97).
6. Vertical Front Porch(T_{vfp}) is fixed to 512 DPI pixel clocks.
7. Frame blanking(T_{fb}) is fixed to 4096 Pixel clocks.
8. Data alignment is controlled by the register MCR (Address = 0x1C).
9. Data content

The frame content is a color bar shown in Figure 5-18 when BPGR.PS is clear. Colors are changed per 128 pixels and the visual colors (From left to right) are white, yellow, cyan, green, magenta, red, blue and black. When BPGR.PS is high, two frame contents are output. One is the same as shown in Figure 5-18 and the other is the reverse visual color sequence as Figure 5-18. The frame content is changed every 15 frames.

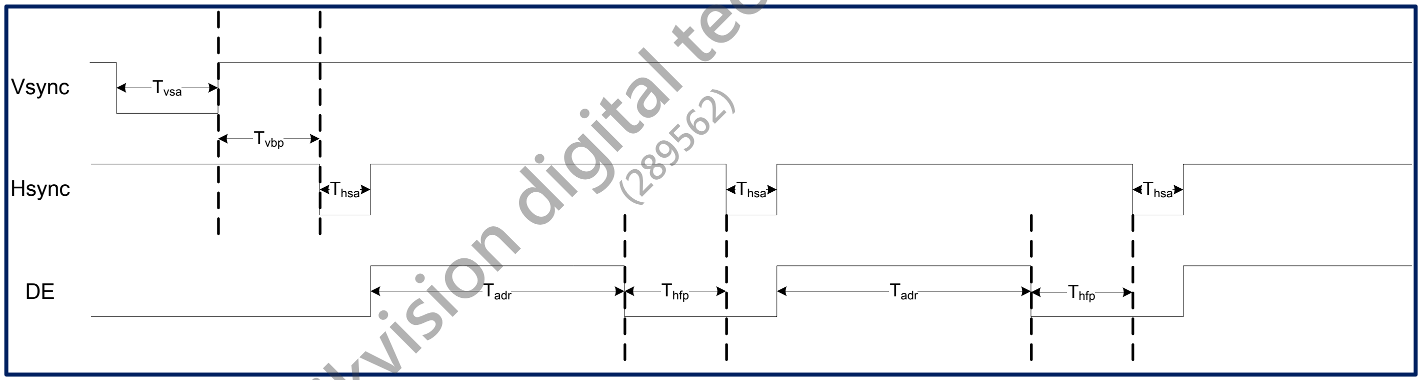

Figure 5-16. First and Subsequent Lines of Each Frame in DPI Built-in Pattern

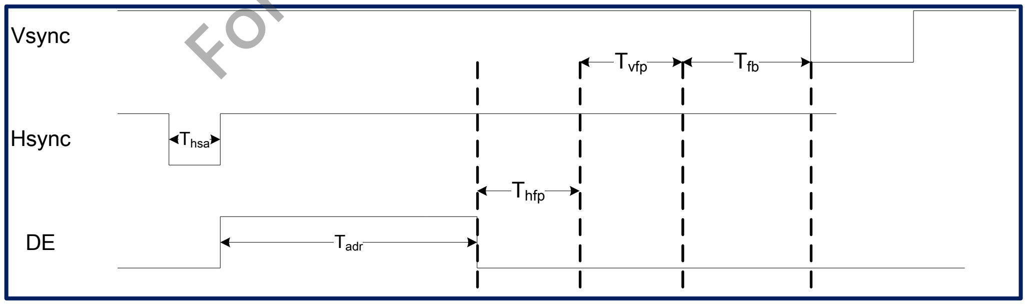

Figure 5-17. The Last Line of Each Frame in DPI Built-in Pattern

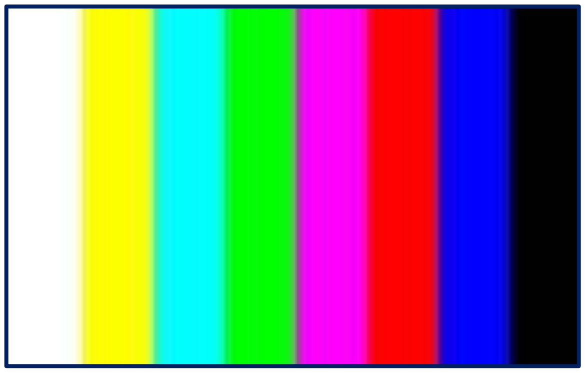

Figure 5-18. Vertical 8-color Stripe

Chapter 6

Initialization/Application Information

This chapter contains the following sections:

6.1 Configure by Programming Registers

FTCSIRX100 provides the registers for software configuration, which are described in Chapter 4. After finishing the register configuration, the software shall perform the software reset to reset the IP interface signals, state machine, FIFOs, counters and status registers. Please refer to Section 4.2.3 for the CR.SR bit and Section 5.6.1.2 for details.

6.2 ForceRxmode Signal for Initialization Period

"The Initialization period" is defined in the D-PHY specification [MIPI01]. The ITR registers (Address = 0x12 ~ 0x13) are used to adjust the assertion duration of PPI signal, ForceRxmode, to meet the $T_{INIT,\ SLAVE}$ timing parameter. The value of ITR registers may be changed for different Slave side PHY (CSI receiver) or Master side PHY (CSI transmitter).

6.3 Camera Control Interface (CCI)

CCI is a subset of the I^{2}C protocol. Combining Faraday I^{2}C bus interface controller (IP name is FTIIC010) can implement a CCI master. FTIIC010 provides the APB interface. The application block diagram is shown in Figure 6-1.

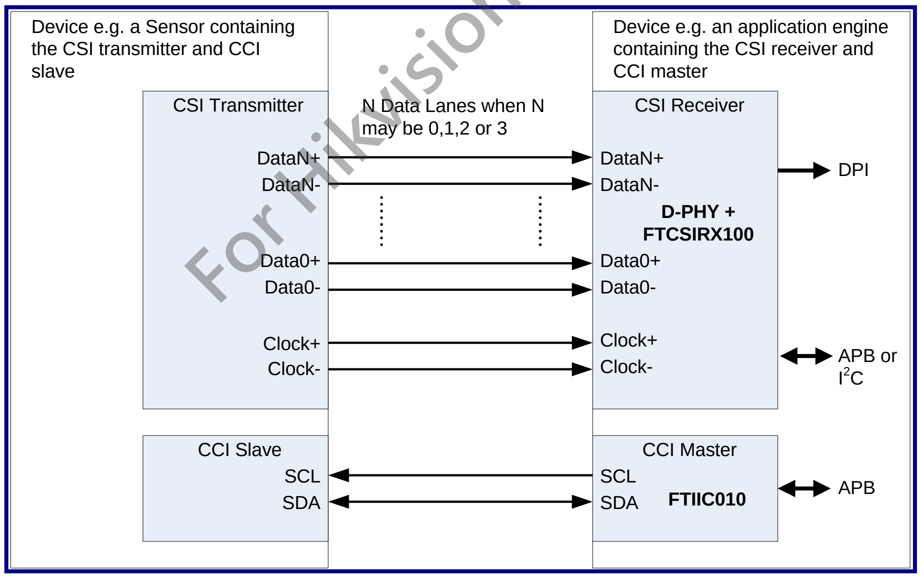
Device e.g. a Sensor containing the CSI transmitter and CCI slave
CSI Transmitter
DataN+
DataN-
Data0+
Data0-
Clock+
Clock-
N Data Lanes when N may be 0,1,2 or 3
Device e.g. an application engine containing the CSI receiver and CCI master
CSI Receiver
DataN+
DataN-
D-PHY + FTCSIRX100
Data0+
Data0-
Clock+
Clock-
DPI
APB or I2C
CCI Slave
SCL
SDA
CCI Master
SCL
SDA
FTIIC010
APB

Figure 6-1. Use FTIIC010 as CCI Master