

### 1 Features

- Conforms to *CSIX-L1: Common Switch Interface Specification-L1 1.0*
- Supports Utopia, POS, etc. interface at the Traffic Manager (TM) side and synchronous interface at the Fabric side
- User data channel and Flow Control channel
- Optimized for Altera APEX 20KC/APEX II architecture

### 2 General Description

The Common Switch Interface (CSIX) defines a physical interface for transferring information between a traffic manager (Network Processor) and a switching fabric. SOCmagic CSIX Interface Core implements the adaptation from traditional interfaces to the CSIX interface, as shown in figure 1.

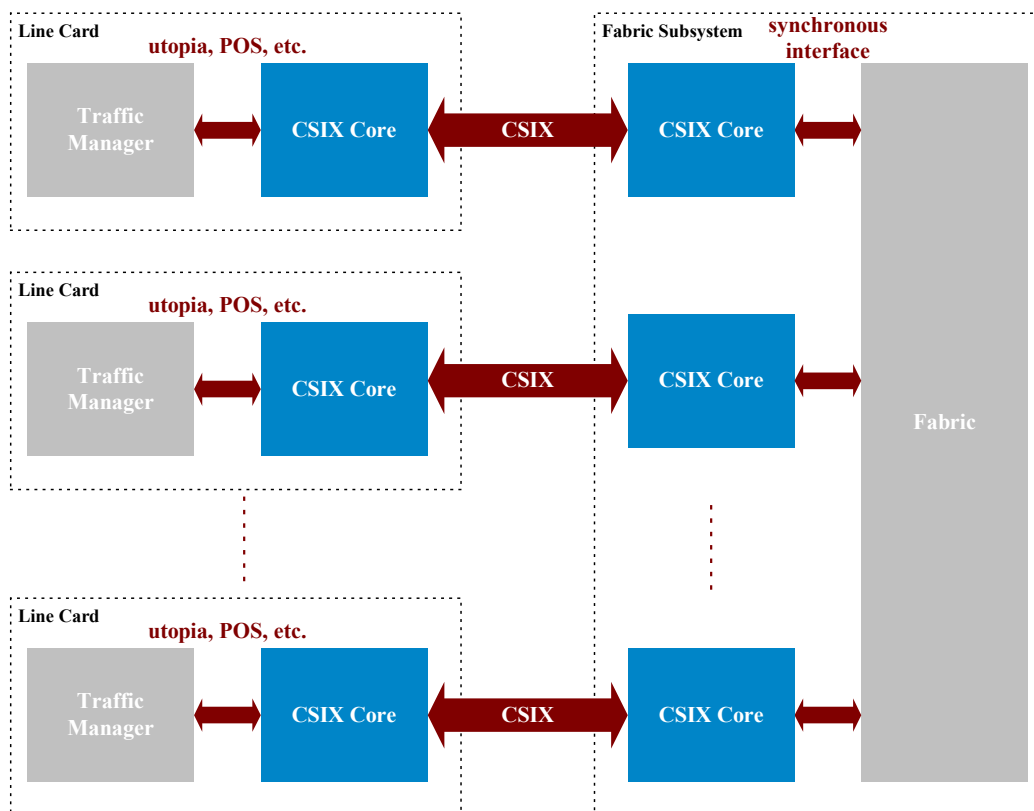


Figure 1 Function Block Diagram

The CSIX Interface Core consists of two parts: one is TM-to-CSIX which applied to the Line Card, the other is CSIX-to-Fabric which applied to the Fabric. The TM-to-CSIX adapts TM interface (It can be utopia, POS, etc) to CSIX and the CSIX-to-Fabric adapts CSIX to Fabric synchronous interface.

### 3 Signal Definitions

Figure 2 shows the Core's signal block diagram.

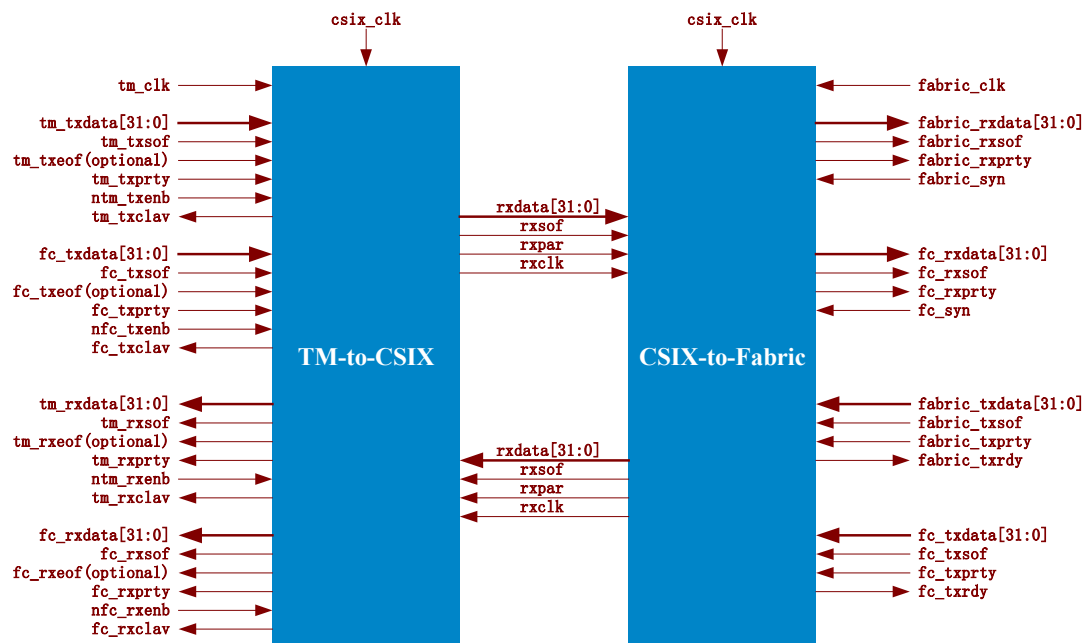


Figure 2 Signal Block Diagram

In the part of the TM-to-CSIX, the interface signal definitions of the TM user data interface is shown in table 1.

Name	Type	Description
tm_clk	Input	Clock of the Traffic Manager, all signals on the TM user data interface and Flow Control interface of the TM-to-CSIX part are synchronous to it.
tm_txdata[31:0]	Input	TM transmit data
tm_txsof	Input	TM transmit start of CFrame. Active-high signal asserted only when tm_txdata contains the first double word of CFrame.

tm_txeof	Input	TM transmit end of CFrame (Optional). Active-high signal asserted only when tm_txdata contains the last double word of CFrame.
tm_txprty	Input	TM transmit data odd parity.
ntm_txenb	Input	TM transmit enable. Active-low signal asserted when tm_txdata contains valid data.
tm_txclav	Output	TM transmit available, Active-high signal. It indicates that space for at least one CFrame is available in the TM tx buffer.
tm_rxdata[31:0]	Output	TM receive data
tm_rxsof	Output	TM receive start of CFrame. Active-high signal asserted only when the TM-to-CSIX drives the first double word of CFrame.
tm_rxeof	Output	TM receive end of CFrame (Optional). Active-high signal asserted only when the TM-to-CSIX drives the last double word of CFrame.
tm_rxprty	Output	TM receive data odd parity.
ntm_rxenb	Input	TM receive enable. Active-low signal asserted to enable the TM-to-CSIX drives data on tm_rxdata.
tm_rxclav	Output	TM receive available, Active-high signal. It indicates that at least one CFrame is available in TM rx buffer.

In the part of the TM-to-CSIX, the interface signal definitions of the Flow Control interface is shown in table 2.

<b>Table 2 The Flow Control interface of the TM-to-CSIX Part</b>		
<b>Name</b>	<b>Type</b>	<b>Description</b>
fc_txdata[31:0]	Input	Flow Control transmit data
fc_txsof	Input	Flow Control transmit start of CFrame. Active-high signal asserted only when fc_txdata contains the first double word of CFrame.
fc_txeof	Input	Flow Control transmit end of CFrame (Optional). Active-high signal asserted only when fc_txdata contains the last double word of CFrame.
fc_txprty	Input	Flow Control transmit data odd parity
nfc_txenb	Input	Flow Control transmit enable. Active-low signal

		asserted when fc_txdata contains valid data.
fc_txclav	Output	Flow Control transmit available, Active-high signal. It indicates that space for at least one CFrame is available in Flow Control tx buffer.
fc_rxdata[31:0]	Output	Flow Control receive data
fc_rxsof	Output	Flow Control receive start of CFrame. Active-high signal asserted only when the TM-to-CSIX drives the first double word of Flow Control Cframe.
fc_rxeof	Output	Flow Control receive end of CFrame (Optional). Active-high signal asserted only when the TM-to-CSIX drives the last double word of Flow Control Cframe.
tm_rxprty	Output	Flow Control receive data odd parity
ntm_rxenb	Input	Flow Control receive enable, Active-low signal asserted to enable the TM-to-CSIX drives data on fc_rxdata.
tm_rxclav	Output	Flow Control receive available, Active-high signal. It indicates that at least one CFrame is available in Flow Control rx buffer.

In the part of the TM-to-CSIX, the interface signal definitions of CSIX interface is shown in table 3. All these signals are defined by CSIX-L1 Specification.

**Table 3 The CSIX interface of TM-to-CSIX Part**

<b>Name</b>	<b>Type</b>	<b>Description</b>
rxclk	Output	Receive clock, generated from csix_clk.
rxdata	Output	Receive data
rxsof	Output	Receive start of frame
rxpar	Output	Receive data odd parity
txclk	Input	Transmit clock, all signals of the CSIX receiver in the TM-to-CSIX part are synchronous to it.
txdata	Input	Transmit data
txsof	Input	Transmit start of frame
txpar	Input	Transmit data odd parity

Table 4 shows the other signal definitions on the CSIX interface of TM-to-CSIX part.

**Table 4 The Misc. signal of TM-to-CSIX part**

Name	Type	Description
csix_clk	Input	CSIX clock, All signals on the CSIX state machine of TM-to-CSIX part are synchronous to it.
nreset	Input	Reset, low active

In the part of the CSIX-to-Fabric part, the interface signal definitions of the CSIX interface is shown in table 5. All these signals are defined by CSIX-L1 Specification.

**Table 5 The CSIX interface of CSIX-to-Fabric part**

Name	Type	Description
rxclk	Input	Receive clock, all signals of the CSIX receiver of CSIX-to-Fabric part are synchronous to it.
rxdata	Input	Receive data
rxsof	Input	Receive start of frame
rxpar	Input	Receive data odd parity
txclk	Output	Transmit clock, generated from csix_clk.
txdata	Output	Transmit data
txsof	Output	Transmit start of frame
txpar	Output	Transmit data odd parity

In the part of the CSIX-to-Fabric part, the interface signal definitions of the user data interface is shown in table 6.

**Table 6 The User data interface of CSIX-to-Fabric Part**

Name	Type	Description
fabric_clk	Input	Clock of the Fabric. All signals on the user data and Flow Control interface of CSIX-to-Fabric part are synchronous to it.
fabric_rxdata[31:0]	Output	Receive data
fabric_rxsof	Output	Receive start of frame. Active-high signal asserted only when the CSIX-to-Fabric drives the first double word of CFrame.
fabric_rxprty	Output	Receive data odd parity
fabric_syn	Input	Receive synchronous signal, Active-high signal. The CSIX-to-Fabric outputs the user data two cycles later

		if the fabric_syn is asserted.
fabric_txdata[31:0]	Input	Transmit data
fabric_txsof	Input	Transmit start of frame. Active-high signal asserted only when fabric_txdata contains the first double word of CFrame.
fabric_txprty	Input	Transmit data odd parity
fabric_txrdy	Output	Transmit ready for send a CFrame. Active-high signal. It indicates that space for at least one CFrame is available in Fabric tx buffer.

In the part of the CSIX-to-Fabric part, the interface signal definitions of the Flow Control interface is shown in table 7.

**Table 7 The Flow Control interface of CSIX-to-Fabric Part**

Name	Type	Description
fc_rxdata[31:0]	Output	Flow control receive data
fc_rxsof	Output	Flow control receive start of frame. Active-high signal asserted only when the CSIX-to-Fabric drives the first double word of Flow Control CFrame.
fc_rxprty	Output	Flow control receive data odd parity
fc_syn	Input	Flow control receive synchronous signal, Active-high signal. The CSIX-to-Fabric outputs the Flow Control Cframe two cycles later if the fc_syn is asserted.
fc_txdata[31:0]	Input	Flow control transmit data
fc_txsof	Input	Flow control transmit start of frame. Active-high signal asserted only when fc_txdata contains the first double word of CFrame.
fc_txprty	Input	Flow control transmit data odd parity
fc_txrdy	Output	Flow control transmit ready for send a CFrame. Active-high signal. It indicates that space for at least one CFrame is available in Flow Control tx buffer.

Table 8 shows the others signal definitions of the CSIX interface of CSIX Interface Core Fabric side.

**Table 8 The Misc. signal of the CSIX-to-Fabric part**

Name	Type	Description
csix_clk	Input	CSIX clock. All signals on the CSIX state machine of CSIX-to-Fabric part are synchronous to it.
nreset	Input	Reset, low active

Table 9 shows the parameter description of the CSIX interface of SOCMagic CSIX Core.

**Table 9 The Parameter description of SOCMagic CSIX Core**

Name	Type	Description
MAX_FRAME_PAYLOAD_SIZE	Integer	Defined by CSIX specification. The TM and the Fabric should be able to support a range of values for MAX_FRAME_PAYLOAD_SIZE form 1 byte up to the maximum supported size.

#### 4 CFrame Format

A CFrame is the base information unit transferred between Traffic Managers and a CSIX Fabric. A CFrame consists of a base header, an optional (determined by type) extension header, an optional payload and a 16-bit vertical field, as shown in table 10.

**Table 10 CFrame structure**

CFrame Component	Base Header	Extension Header	Payload	Vertical Parity
Length	2 bytes	0-4bytes	0-256bytes	2 bytes

##### 4.1 Base Header

Table 11 shows the layout of the base header. This layout is defined by CSIX specification.

<b>Table 11 Base Header</b>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Ready	Type			CR	P	Payload Length									

##### 4.1.1 Type field

The type field (4bits) encodes the type of CFrame being transferred. Table 12 shows the type field values. This table is defined by CSIX specification.

Type	Encoding
Idle	0x0
Unicast	0x1
Multicast Mask	0x2
Multicast ID	0x3
Multicast Binary Copy	0x4
Broadcast	0x5
Flow Control Frame	0x6
Command and Status	0x7
CSIX Received	0x8-0xb
Private	0xc-0xf

The SOCMagic CSIX Interface Core supports two kinds of CFrames, the user CFrame (type = 0x1-0x5) and the Flow Control (type = 0x6) CFrame. All others kinds of CFrames will be ignore. In the TM-to-CSIX part, only the user CFrame can be sent to the TM user data interface and only the Flow Control CFrame can be sent to the Flow Control interface. In the CSIX-to-Fabric part, only the user CFrame can be sent to the user data interface and only the Flow Control CFrame can be sent to the Flow Control interface.

#### **4.1.2 Ready Field**

The ready field (2 bits) is used to indicate when the transmitting entity is ready to receive data. A low (0) ready bit indicates that the entity is not ready to receive the specified traffic type; a high (1) ready bit indicates that the entity is ready to receive the specified traffic type.

There are 2 ready bits representing 2 link level queues. The two groups and the CFrame types assigned to these groups are given below.

Ready[0] (bit 6 of byte 0) is for Control traffic

- Command & Status CFrames (TBD in future CSIX work)
- Flow control CFrames

Ready[1] (bit 7 of byte 0) is for Data Traffic

- Unicast
- Multicast
- Broadcast

The ready field is filled by the CSIX state machine. The Traffic Manager does not need care it.

#### **4.1.3 Payload Length Field**

Payload length (8 bits) is the number of bytes in the payload of the message, excluding any padding (which is inserted between the payload and the vertical parity field.). For data CFrames, “0” indicates a payload of 256 bytes; for idle CFrames, the payload length field shall be set to “0” (A “0” payload length for idle frames does not indicate a payload of 256 bytes.). The payload length must be less than the MAX\_FRAME\_PAYLOAD\_SIZE parameter. If the length of the TM packet overruns the MAX\_FRAME\_PAYLOAD\_SIZE parameter, the Traffic Manager *must* cut it to smaller packet, or the SOCMagic CSIX core will ignore the redundant (The CSIX Interface Core will not cut the big packet to smaller).

#### **4.2 Extension Header**

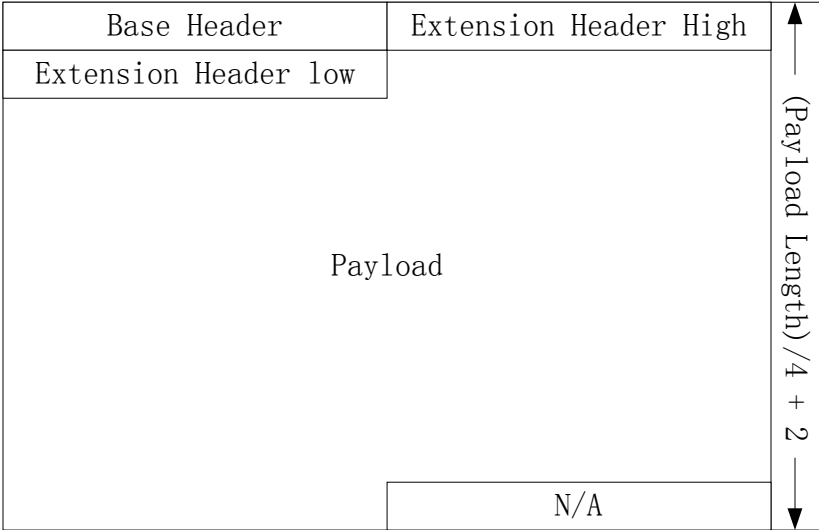
Table 13 shows the extension header of the user CFrame. These are defined by CSIX specification. For more information, please see the CSIX specification. The SOCMagic CSIX Interface core transfers all the fields of the extension header without any operation.

<i>Table 13 Extension Header</i>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Unicast															
Class								P	CSIX Reserved						
CSIX Reserved				Destination Address											
Multicast Mask															
Class								Bitmask header							
Bitmap															
Multicast ID															
Class								P	CR	Multicast ID					
Multicast ID															
Multicast Binary Copy															
Class								Left Destination Address							
LDA				Right Destination Address											
Broadcast															
Class								P	CSIX Reserved						
CSIX Reserved															

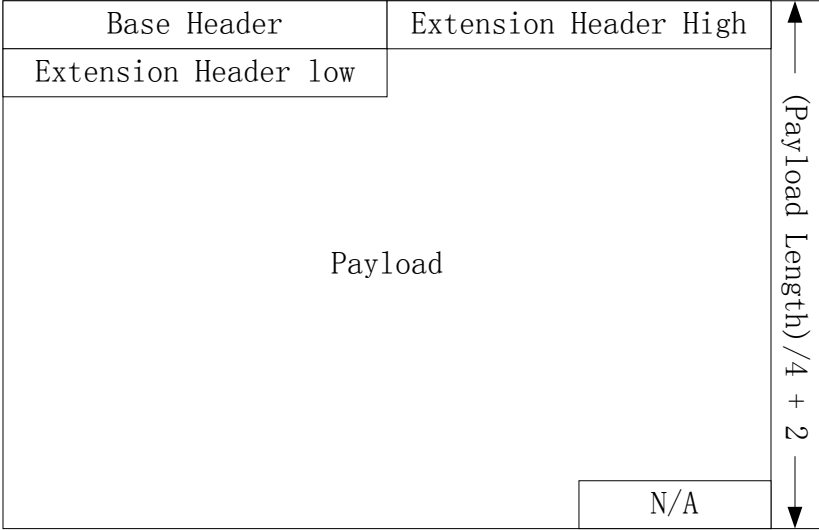
### 4.3 User CFrame

The user CFrame which is generated by Traffic Manager must consist of the base header, the extension header and the payload. In the base header, the type field and the payload length field are needed. The Traffic Manager neither need to calculate the vertical parity nor reserve the position for vertical parity. The SOCMagic CSIX core will calculate the vertical parity and insert it into the CFrame. Figure 3-6 show the format of the user CFrame generated by Traffic Manager.

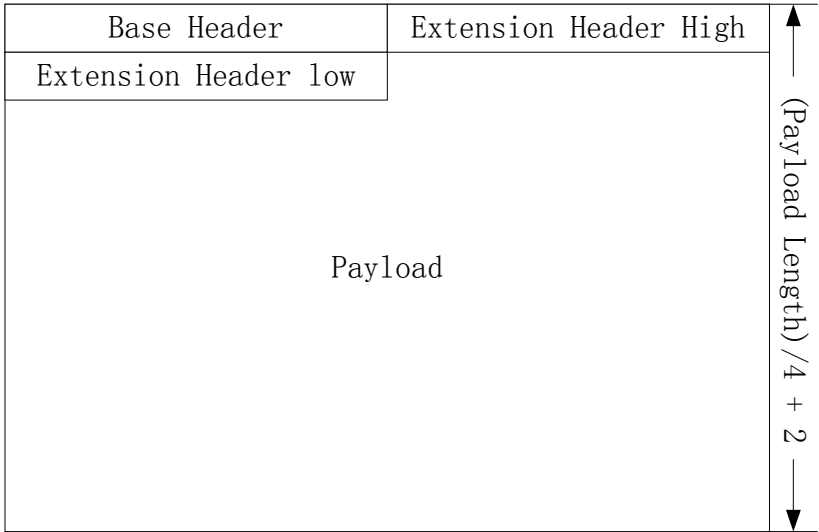
If the format of your TM packet is different from what we describe, please connect [SOCmagic](#), we can customize a translator for your packet format.



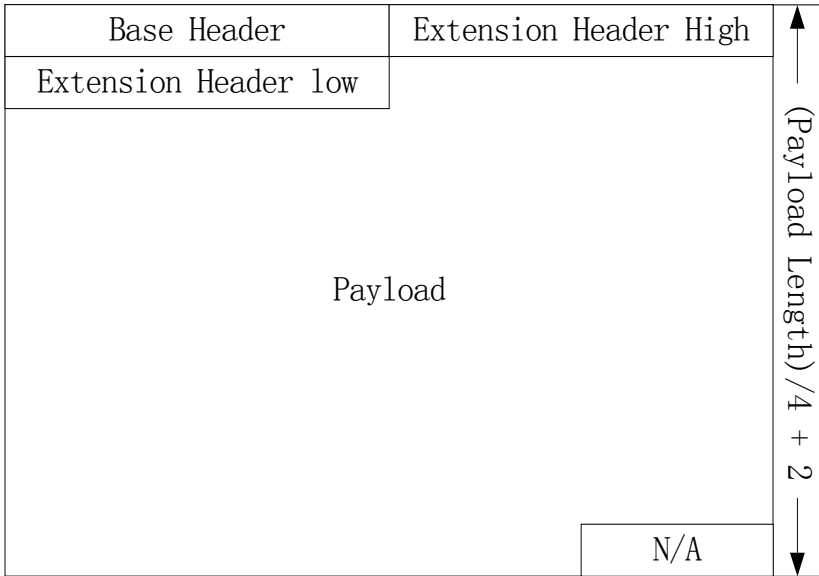
**Figure 3 User CFrame Format (Payload Length) mod 4 = 0**



**Figure 4 User CFrame Format (Payload Length) mod 4 = 1**

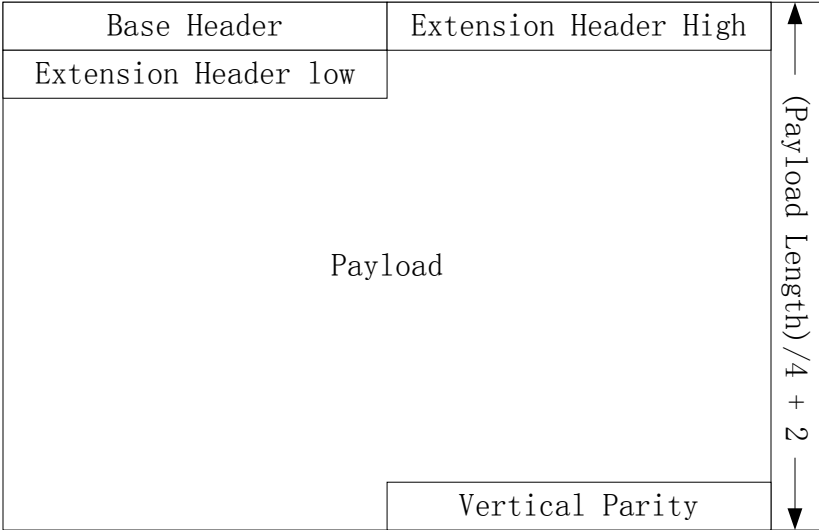


**Figure 5 User CFrame Format (Payload Length) mod 4 = 2**

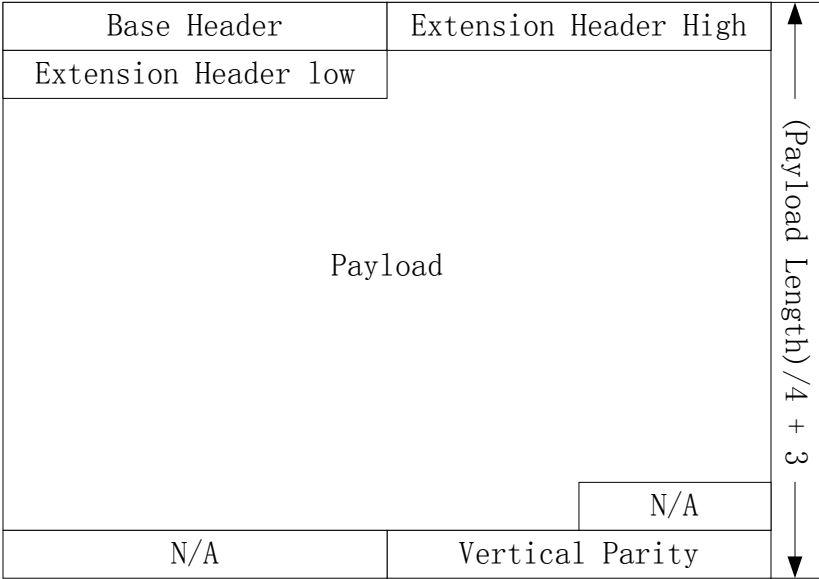


**Figure 6 User CFrame Format (Payload Length) mod 4 = 3**

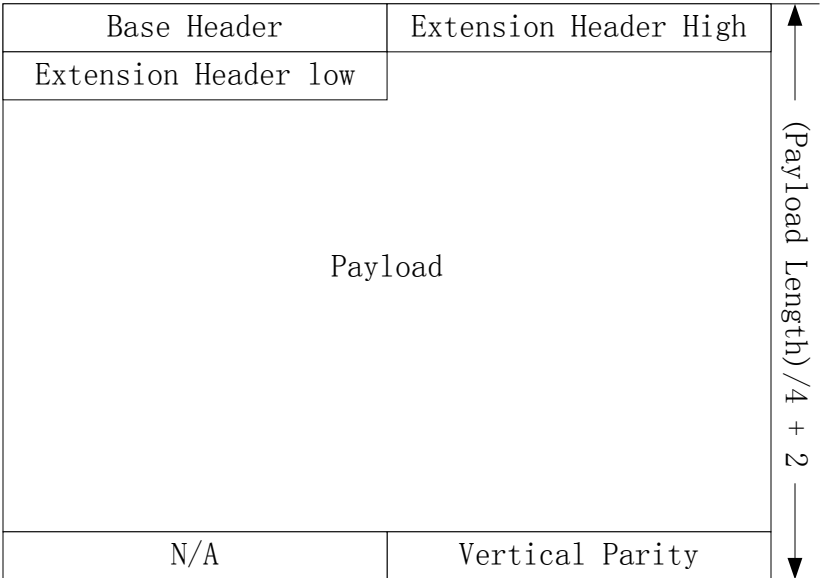
The format of user CFrame is shown in figure 7-10 after CSIX machine adds the vertical parity into the CFrame.



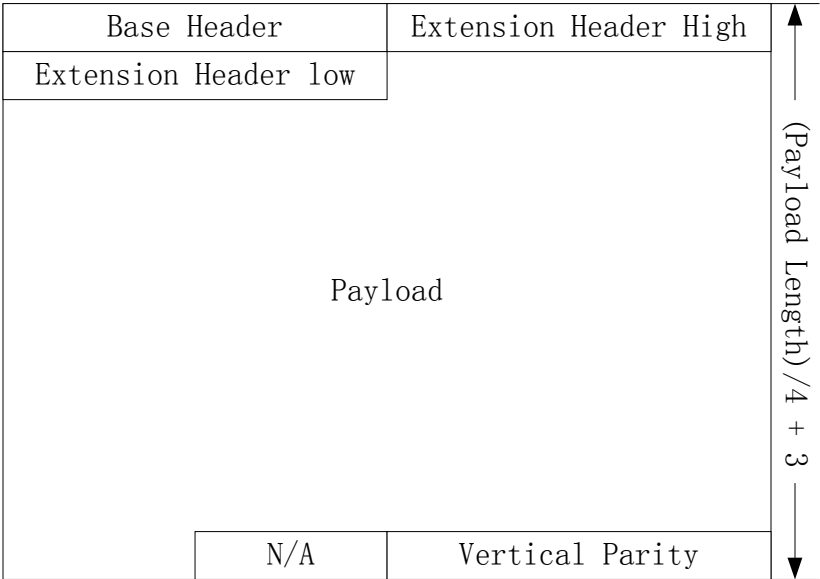
**Figure 7 User CFrame Format (Payload Length) mod 4 = 0**



**Figure 8 User CFrame Format (Payload Length) mod 4 = 1**



**Figure 9 User CFrame Format (Payload Length) mod 4 = 2**

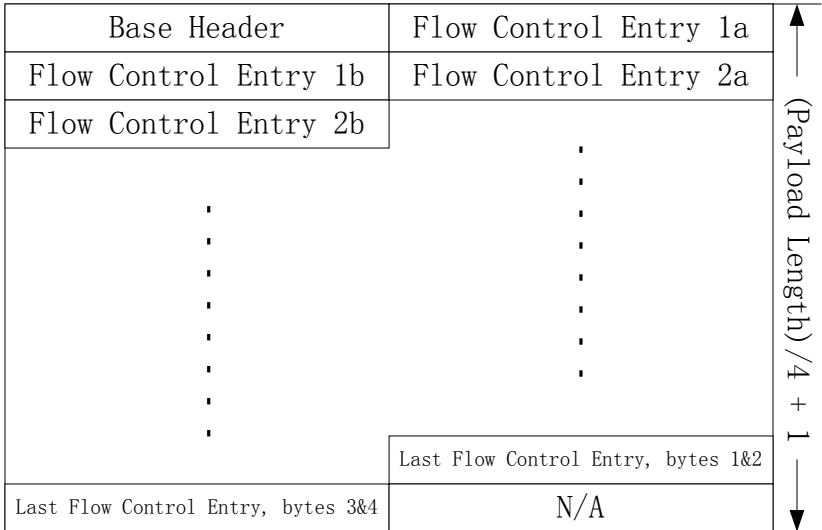


**Figure 10 User CFrame Format (Payload Length) mod 4 = 3**

**4.4 Flow Control CFrame**

The Flow Control CFrame generated by Traffic Manager must consist of the base header and the Flow Control Entries. In the base header, the type field and

the payload length field are needed. The Traffic Manager neither need to calculate the vertical parity nor reserve the position for vertical parity. The SOCMagic CSIX core will calculate the vertical parity and insert it into the flow control CFrame. Figure 11 shows the format of Flow Control CFrame generated by Traffic Manager. If the format of your Flow Control packet is different from what we describe, please connect [SOCmagic](#), we can customize a translator for your packet format.



**Figure 11 Flow Control CFrame Format**

After CSIX state machine adds the vertical parity into the CFrame, the Flow Control CFrame will be what shown in figure 12.



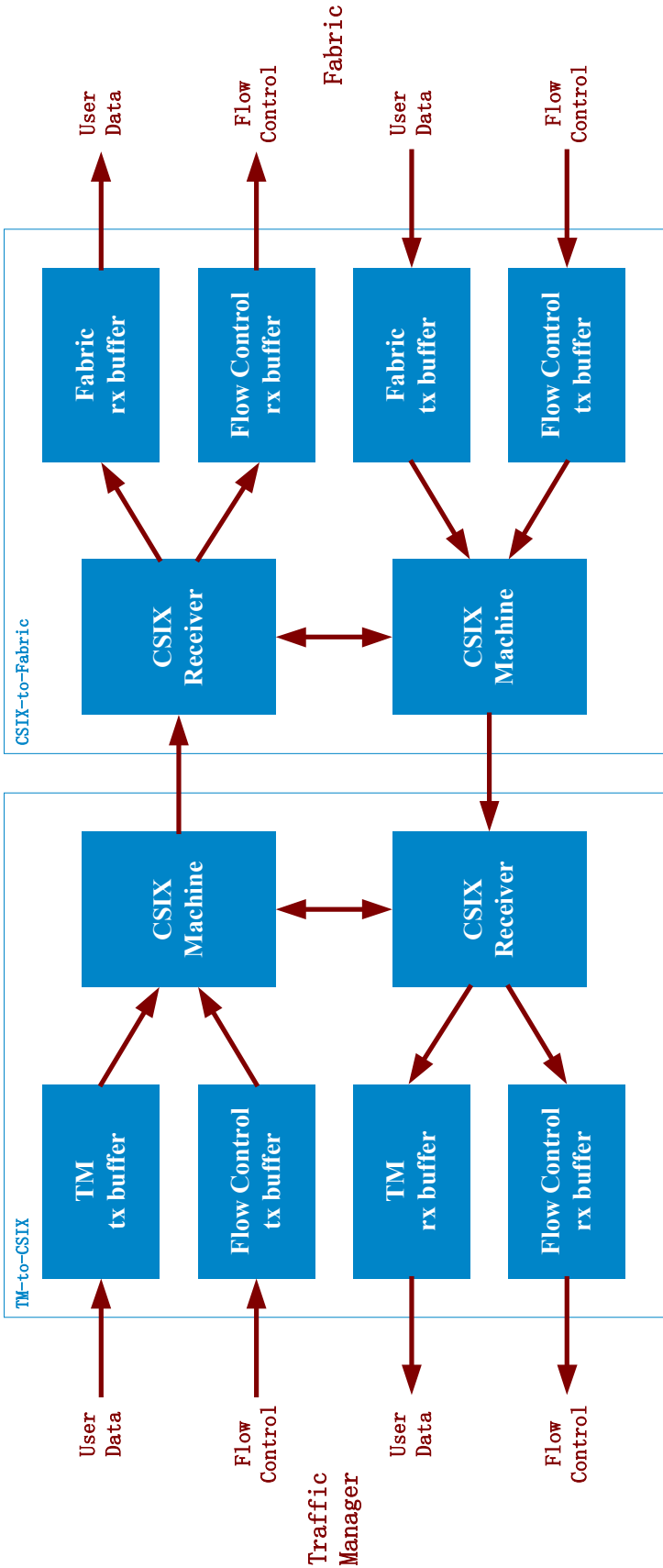
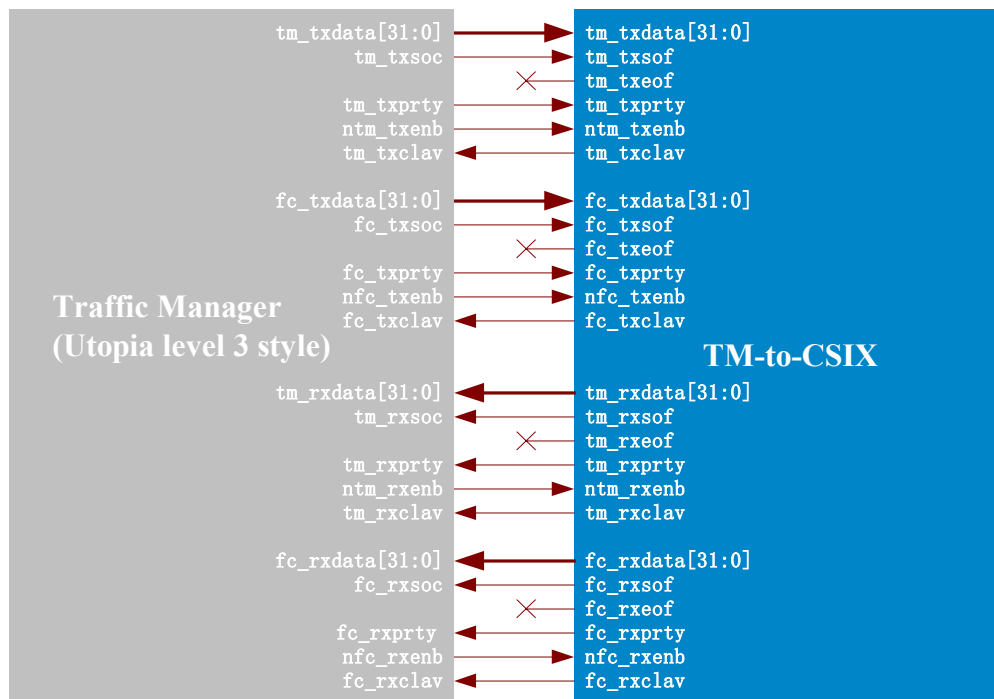


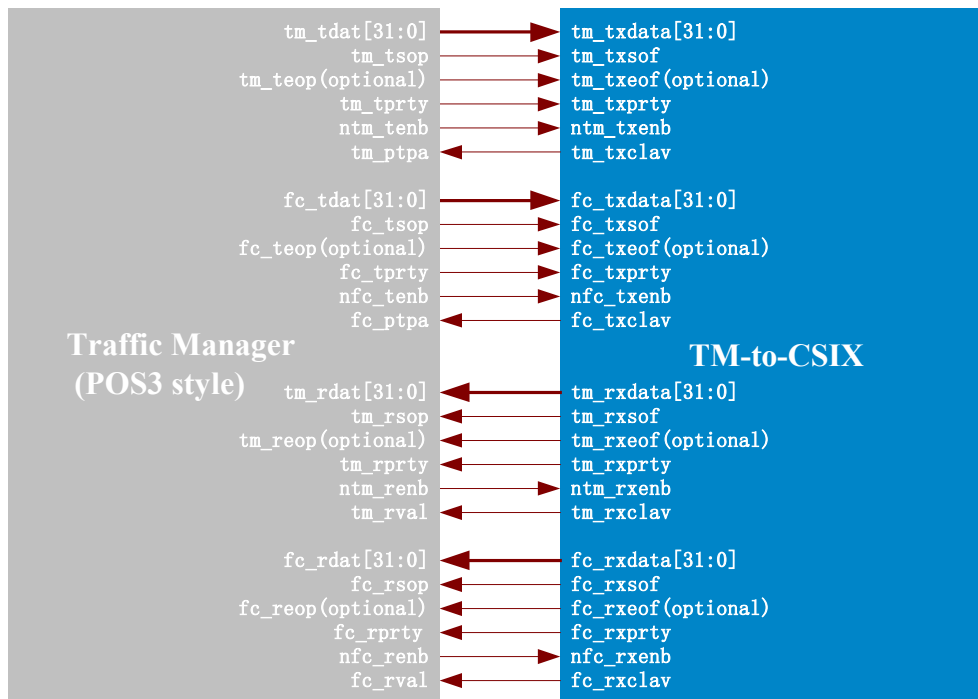
Figure 13 CSIX Core Block Diagram

## 5.1 TM-to-CSIX

The TM interface is a “clever” interface. It can be connected to many kinds of interfaces without any glue circuit, such as Utopia level 3, POS3, Intel IXPBus, etc. Figure 14 shows an example of a Utopia level 3 style TM’s connection to the SOCMagic TM-to-CSIX, and figure 15 shows an example of a POS3 style TM’s connection to the SOCMagic TM-to-CSIX. The txep/rxep signals of the TM are optional.



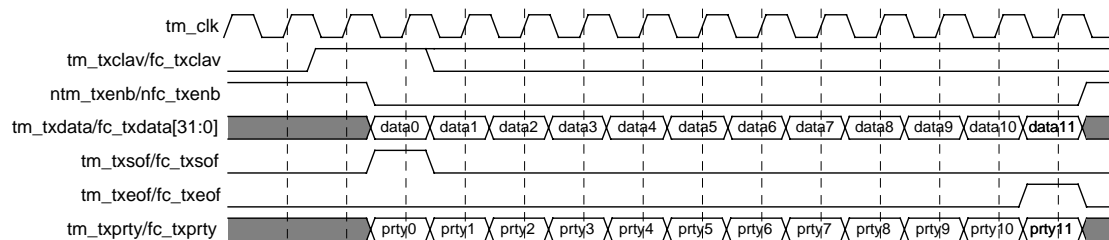
*Figure 14 The Example of a Utopia level 3 style TM connection*



**Figure 15 The Example of a POS3 style TM connection**

**5.1.1 TM tx buffer and Flow Control tx buffer**

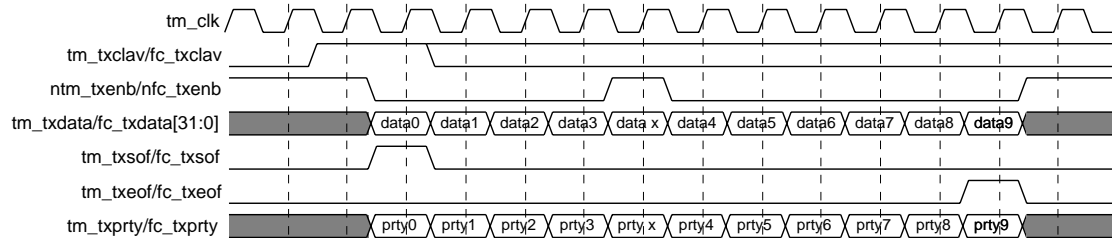
The circuits of the TM tx buffer and the Flow Control tx buffer are the same. Only user CFrame can be written into the TM tx buffer and only Flow Control CFrame can be written into Flow Control tx buffer. When the buffer has space for at least one CFrame, the buffer asserts its own txclav signal. When TM detects the txclav is high and TM has CFrame to send, TM asserts the ntxenb signal and output CFrame data, just like Utopia or POS. The timing diagram is shown in figure 16.



**Figure 16 TM-to-CSIX Transmit Timing Diagram: Normal**

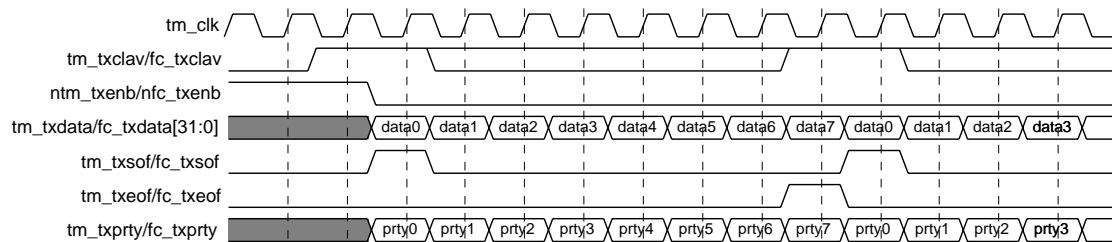
Once TM starts to send a CFrame, it must send a whole CFrame. If TM deasserts the ntxenb signal during the process of sending a CFrame, invalid data will be inserted into the CFrame and the tail of the CFrame will be lost. For example, as shown in

figure 17, if TM deasserts the ntxenb signal one cycle between data3 and data4, the data sequence that tx buffer gets then will be data0, data1, data2, data3, data x, data4, data5, data6, data7, data8 (Data x is inserted into the CFrame and data9 is lost).



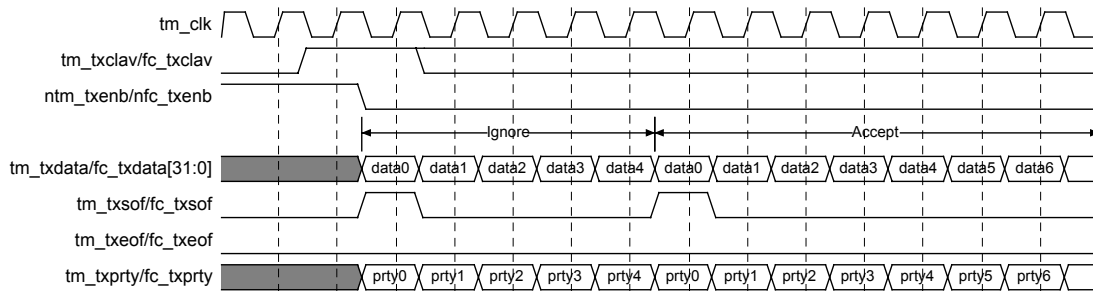
**Figure 17 TM-to-CSIX Transmit Timing Diagram: Error**

The SOCMagic CISX Core supports back-to-back transfer. If TM detects txclav signal asserted when it finishes the current transfer, TM can continue the next CFrame transfer without dead cycles, as shown in figure 18.



**Figure 18 TM-to-CSIX Transmit Timing Diagram: Back-to-back Mode**

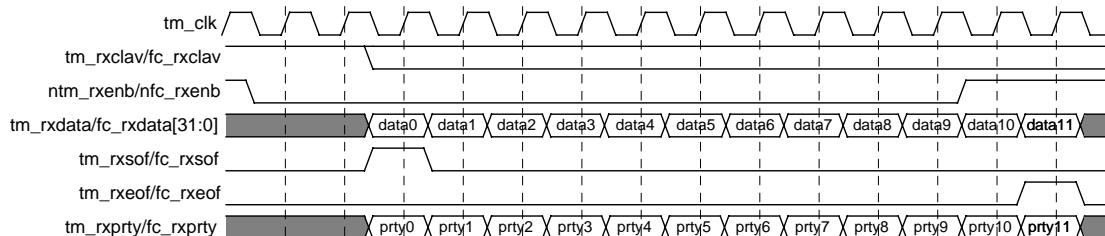
If the tx buffer encounters an unexpected SOF, it ignores the previous data and commence processing the CFrame signaled by the new SOF. For example, as shown in figure 19, TM sends an unexpected SOF without finishing the transfer of the first CFrame(assuming the payload length of the first CFrame is more than 4), the first unfinished CFrame will be ignored, and the second CFrame will be accepted.



**Figure 19 TM-to-CSIX Transmit Timing Diagram: Unexpected SOF**

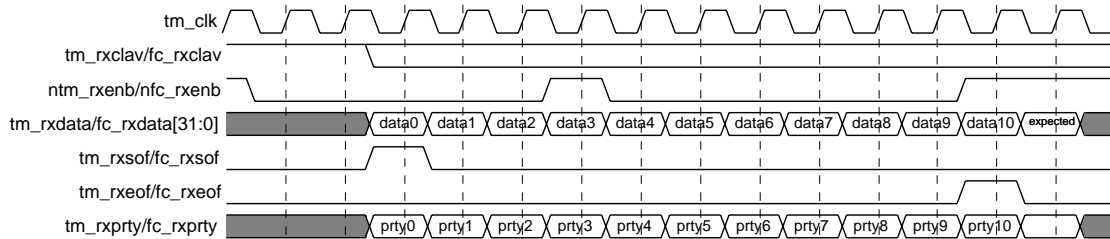
### 5.1.2 TM rx buffer and Flow Control rx buffer

The circuits of the TM rx buffer and the Flow Control rx buffer are the same. Only user CFrame can be output from the TM rx buffer and only Flow Control CFrame can be output from Flow Control rx buffer. When the buffer has at least one CFrame, the buffer asserts its own rxclav signal. When TM detects the rxclav asserted and TM has space for receiving CFrames, it asserts the nrxenb signal and rx buffer will output CFrame data two cycles later, just like Utopia or POS. The timing diagram is shown in figure 20.



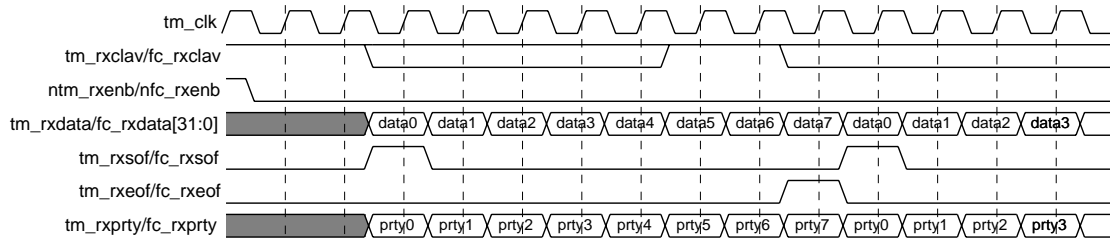
**Figure 20 TM-to-CSIX Receive Timing Diagram: Normal**

Once TM starts to receive a CFrame, it must receive a whole CFrame. If TM deasserts the nrxenb signal during the process of receiving a CFrame, valid data will be lost and another unexpected data will be appended to the tail of the CFrame. For example, as shown in figure 21, if TM deasserts the nrxenb signal one cycle between the fifth and the sixth cycle, the data sequence that TM gets then will be data0, data1, data2, data3, data4, data6, data7, data8, data9, data10, data “unexpected”. Data 5 is lost and data “unexpected” is appended to the tail of the CFrame.



**Figure 21 TM-to-CSIX Receive Timing Diagram: Error**

The SOCMagic CSIX Core supports back-to-back transfer. If TM detects rxclav signal asserted when it finishes the current transfer, it can continue the next CFrame transfer without dead cycles, as shown in figure 22.

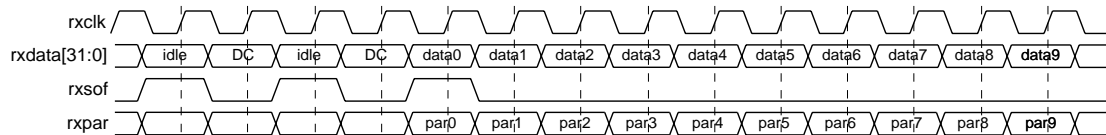


**Figure 22 TM-to-CSIX Receive Timing Diagram: Back-to-back Mode**

### 5.1.3 CSIX State Machine

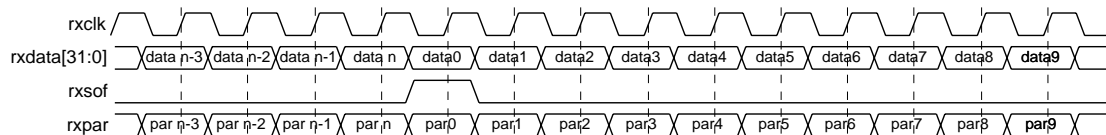
CSIX State Machine reads the CFrames from two rx buffers and then send them to CSIX interface. The Flow Control CFrames have higher priority.

CSIX state machine is designed Conforming to the CSIX Specification. At Power-Up or RESET, the CSIX state machine begins transmitting idle CFrames while holding both ready bits low. When the CSIX receiver detects receipt of idle CFrames, the CSIX state machine raises both ready bits to high and continues to send idle CFrames. When the CSIX receiver detects receipt of idle CFrames and high ready bits from its connected CSIX state machine of CSIX-to-Fabric, the CSIX state machine can send the user CFrame and the Flow Control CFrame. This is depicted in the timing diagram shown in figure 23.



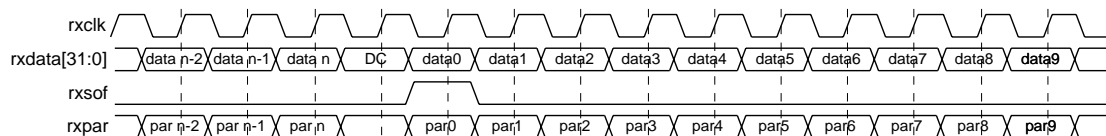
**Figure 23 CSIX State Machine Output Timing Diagram: Start-up**

Figure 24 shows the transmission of two consecutive CFrames with no break.

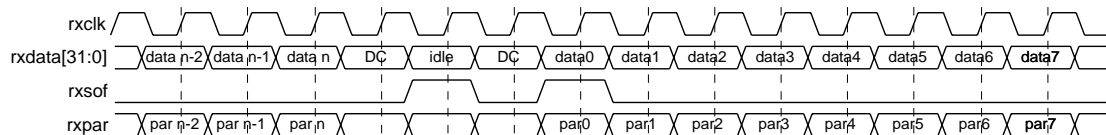


**Figure 24 CSIX State Machine Output Timing Diagram: Two Consecutive CFrames**

At any point in time the CSIX interface may contain a data, control or idle CFrame or a Dead Cycle. During times when no data is being transmitted, a pattern of alternating idle CFrames and Dead Cycles is transmitted to maintain synchronization and keep the ready bits alive. Figure 25 shows the transmission of two CFrames with a one-tick separation (a Dead Cycle is inserted). Figure 26 shows the transmission of two CFrames with a multi-tick separation (a idle CFrame is inserted).



**Figure 25 CSIX State Machine Output Timing Diagram: A One-tick Separation**



**Figure 26 CSIX State Machine Output Timing Diagram: A Multi-tick Separation**

At the same time of transmitting the CFrames, the CSIX state machine re-generates the parity bit and calculates the vertical parity. If the parity bit from rx buffer is error, the CSIX state machine will send an error parity bit to the CSIX

receiver of CSIX-to-Fabric in order that the CSIX receiver can operate this CFrame as an error CFrame. The vertical parity is appended to the end of the payload.

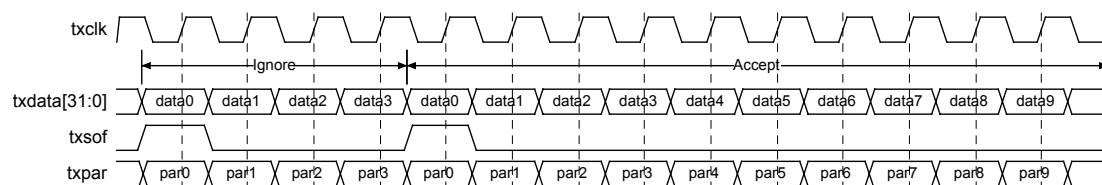
#### 5.1.4 CSIX Receiver

The CSIX Receiver receives the CFrames from its connected CSIX state machine of CSIX-to-Fabric and writes them into the TM tx buffers or the Flow Control tx buffer.

At Power-Up or RESET, the CSIX receiver waits for idle CFrames. When the CSIX receiver detects receipt of idle CFrames, it enables the CSIX state machine to send idle CFrames with high ready bits, and continues to wait for idle CFrame with high ready bits. When the CSIX receiver detects receipt of idle CFrames and high ready bits, it enables the CSIX state machine to send the user CFrames and the Flow Control CFrames.

When the CSIX receiver detects receipt of a non-idle CFrame, it determines to write the CFrame into the TM tx buffer or the Flow Control tx buffer based on the type field of the CFrame.

If the CSIX receiver encounters an unexpected SOF, it will ignore the previous data and commence processing the CFrame signaled by the new SOF. As shown in figure 27, CSIX receiver encounters an unexpected SOF without finishing the transfer of the first CFrame (assuming the payload length of the first CFrame is more than 4), the first unfinished CFrame will be ignored, and the second CFrame will be accepted.



**Figure 27 CSIX Receiver Timing Diagram: An Unexpected SOF**

When a parity error (horizontal or vertical) is detected on the CFrame, ready bits from the CFrame are ignored and the CSIX receiver interprets them as not ready. This means that the CSIX state machine stops sending further data or control CFrames until

it receives the next CFrame without any errors and corrects the state of the ready bits.

## 5.2 CSIX-to-Fabric

### 5.2.1 CSIX State Machine

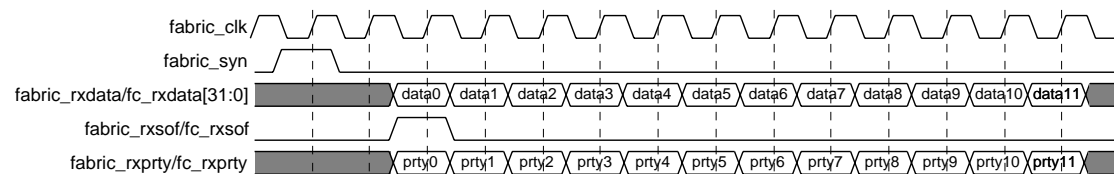
The CSIX state machine of CSIX-to-Fabric is the same as CSIX state machine of TM-to-CSIX. The function of this part is described in 5.1.3 with changing the prefix of all the signals from “rx” to “tx”.

### 5.2.2 CSIX Receiver

The CSIX receiver of CSIX-to-Fabric is the same as CSIX receiver of TM-to-CSIX. The function of this part is described in 5.1.4 with changing the prefix of all the signals from “tx” to “rx”.

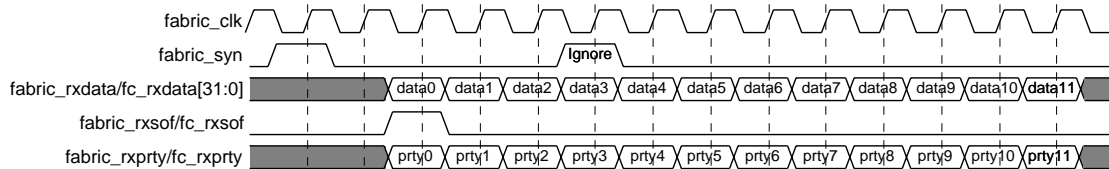
### 5.2.3 Fabric rx buffer and Flow Control rx buffer

The circuits of the Fabric rx buffer and the Flow Control rx buffer are the same. The Fabric rx interface is a synchronous interface. The Fabric generates a fabric\_syn signal every time-slot. When the rx buffer detects the fabric\_syn signal asserted, it will output the CFrame data two cycle later, as shown in figure 28. If there is no CFrame in the rx buffer, the rxsof signal will be deasserted.



**Figure 28 Fabric Receive Interface Timing Diagram: Normal**

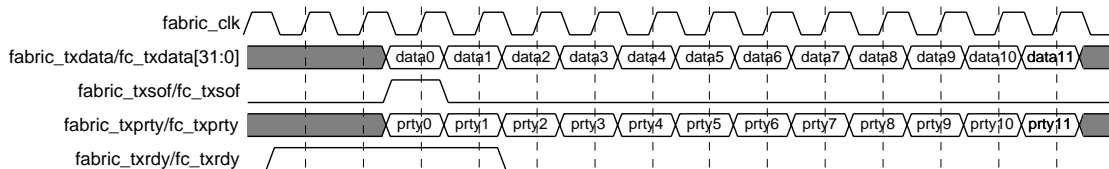
If the rx buffer detects an unexpected fabric\_syn when some problem occurs, it will ignore the unexpected fabric\_syn. As shown in figure 29, the second fabric\_syn high is ignored.



**Figure 29 Fabric Receive Interface Timing Diagram: An Unexpected Fabric\_syn**

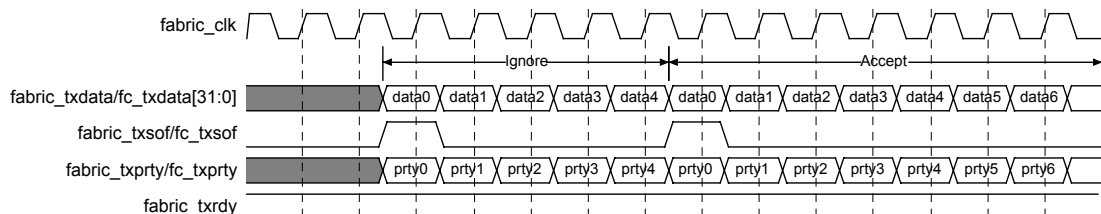
### 5.2.4 Fabric tx buffer and Flow control tx buffer

The circuits of the Fabric tx buffer and the Flow Control tx buffer are the same. When the txrdy signal is high, the Fabric can write CFrames into tx buffer. The Fabric only needs to supply txdata, txsof and txprty signals. The tx buffer accepts the CFrame if it detects txsof asserted, as shown in figure 30.



**Figure 30 Fabric Transmit Interface Timing Diagram: Normal**

If the tx buffer encounters an unexpected SOF, it ignores the previous data and commence processing the CFrame signaled by the new SOF. For example, as shown in figure 31, the fabric sends an unexpected SOF without finishing the transfer of the first CFrame (assuming the payload length of the first CFrame is more than 4), the first unfinished CFrame will be ignored, and the second CFrame will be accepted.



**Figure 31 Fabric Transmit Interface Timing Diagram: An Unexpected SOF**

## 6 Performance

The performance information in table 15 and 16 is generated with the Quartus II

2.0SP1 software. (MAX\_FRAME\_PAYLOAD\_SIZE = 56)

<b>Table 15 TM-to-CSIX Performance</b>						
<b>Target Device</b>	<b>Clocklocks</b>	<b>LEs</b>	<b>ESBs</b>	<b>Fmax(MHz)</b>		
				<b>TM_clk</b>	<b>rxclk</b>	<b>txclk</b>
EP2A15FC672-7	3	1321	26	124	134	169

<b>Table 16 CSIX-to-Fabric Performance</b>						
<b>Target Device</b>	<b>Clocklocks</b>	<b>LEs</b>	<b>ESBs</b>	<b>Fmax(MHz)</b>		
				<b>Fabric_clk</b>	<b>rxclk</b>	<b>txclk</b>
EP2A15FC672-7	3	1415	26	110	130	169