

Open-source  
SpaceWire CODEC IP Core  
User Manual  
Ver.0.40

Document No.230968001  
Date 13 Nov, 2013  
Shimafuji Electric Inc.  
Printed in Japan

## 改訂履歴

版 数	日 付	内 容	備 考
Ver.0.10	2013/11/13	第一版	
Ver.0.20	2013/11/15	ドキュメント名称変更 フォント変更 日本語「MS明朝」 英数時「Times New Roman」 表記変更 単語の間にスペースを追加 「SpaceWireCODECIP」→「SpaceWire CODEC IP」	
Ver.0.30	2013/12/12	10.2クロック 例2クロックの説明追加 図に図枠と補助線を追加	
Ver.0.40	2013/12/17	8.ファイル構成に、文字コードの注記を追加	

# 目次

1. 概要 .....	1
2. License と免責事項.....	1
3. ブロック図 .....	2
4. モジュールの機能.....	3
4.1. TRANSMITTER .....	3
4.2. RECEIVER .....	3
4.3. STATEMACHINE .....	4
4.4. TIMER .....	5
4.5. TIMECODE .....	5
4.6. STATISTICAL INFORMATION .....	5
4.7. TRANSMITFIFO .....	5
4.8. RECEIVEFIFO .....	5
5. インターフェース信号.....	6
6. タイミング図.....	9
6.1. TRANSMITFIFO .....	9
6.2. RECEIVEFIFO .....	9
6.3. 送信タイムコード .....	10
6.4. 受信タイムコード .....	10
7. パッケージ .....	11
8. ファイル構成.....	12
9. パフォーマンス.....	13
10. タイミング制約.....	14
10.1. UCF ファイル.....	14
10.2. クロック .....	15

## 1. 概要

SpaceWireは、宇宙機（衛星）内で、搭載コンポーネント間のデータ通信を行うための通信I/Fおよび通信プロトコルの仕様です。

SpaceWire CODEC IP は Xilinx 、Altera の FPGA をターゲットに SpaceWire の実装を目的とした VHDL コアです。

ECSS-E-ST-50-12C に適合するように設計されています、通信レートは最大 100Mbps まで対応。

RMAP Target IP や SpaceWire Router IP に組み込む事でテレメトリデータの収集や SpaceWire ルータ機能の実現が可能です。

## 2. Licenseと免責事項

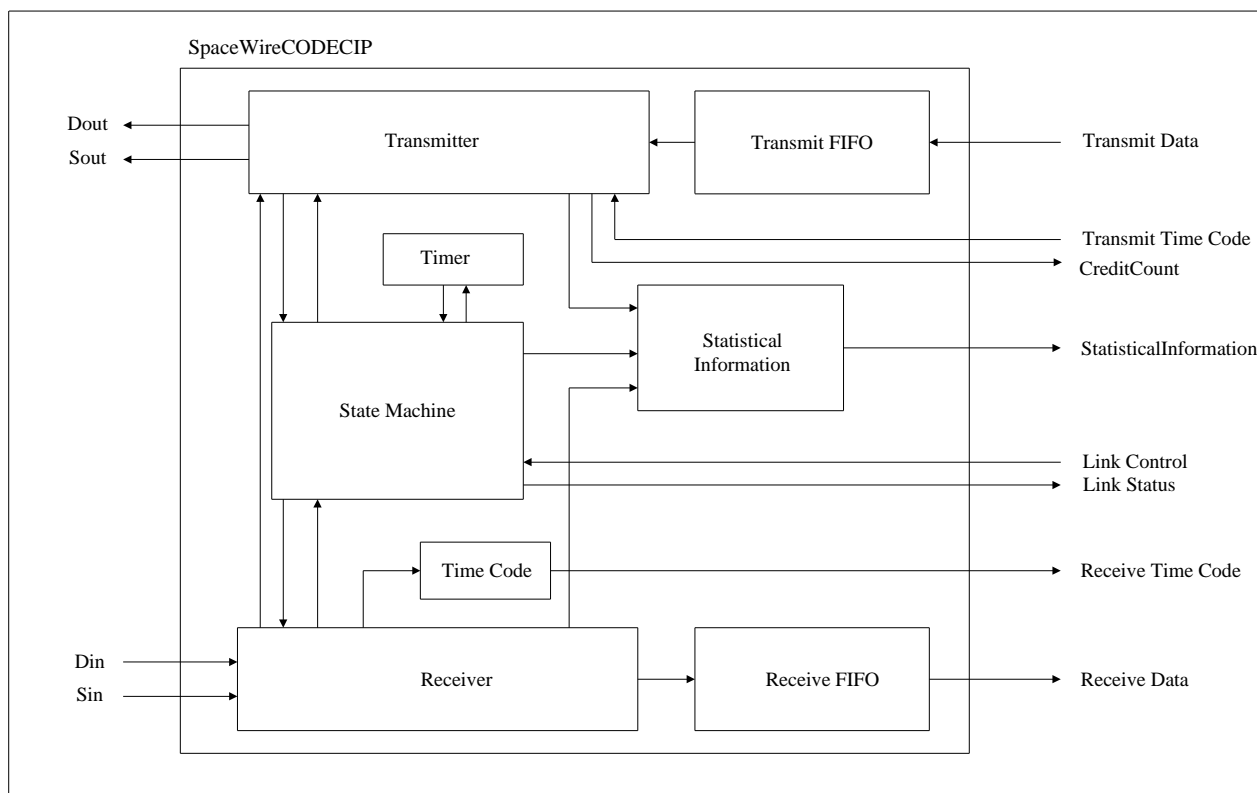
本ソフトウェアおよび関連文書のファイル（以下「ソフトウェア」）の複製を取得するすべての人に対し、ソフトウェアを無制限に扱うことを無償で許可します。これには、ソフトウェアの複製を使用、複写、変更、結合、掲載、頒布、サブライセンス、および/または販売する権利、およびソフトウェアを提供する相手に同じことを許可する権利も無制限に含まれます。

上記の著作権表示および本許諾表示を、ソフトウェアのすべての複製または重要な部分に記載するものとします。

ソフトウェアは「現状のまま」で、明示であるか暗黙であるかを問わず、何らの保証もなく提供されます。ここでいう保証とは、商品性、特定の目的への適合性、および権利非侵害についての保証も含みますが、それに限定されるものではありません。 作者または著作権者は、契約行為、不法行為、またはそれ以外であろうと、ソフトウェアに起因または関連し、あるいはソフトウェアの使用またはその他の扱いによって生じる一切の請求、損害、その他の義務について何らの責任も負わないものとします。

### 3. ブロック図

SpaceWire CODEC IP ブロック図を示します。



SpaceWire からの入力を Receiver がデコードし ReceiveFIFO に書き込みます。  
TransmitFIFO に書かれているデータを Transmitter がデータ及びストローブ信号に変換し  
SpaceWire に出力します。

## 4. モジュールの機能

### 4.1. Transmitter

StateMachine からの送信許可がある時だけ、TransmitFIFO に書き込まれているデータを受信した FCT 数分の N-Char だけデータ及びストローブ信号に変換し SpaceWire に出力します、送信するデータがない時又は送信可能数が 0 の時は NULL を出力しています。

FCT の送受信数は Transmitter が管理しています。

Receiver が FCT 受信するごとに Transmitter は送信可能数を 8 ずつ加算しています、Transmitter が N-Char を送信すると送信可能数は 1 つ減算します。

また、Transmitter が FCT を送信すると受信可能数を 8 ずつ加算しています、Receiver が N-Char を受信すると受信可能数は 1 つ減算します。

送信可能数が 56 以上になった時又は受信可能数が 0 の時に Receiver が N-Char を受信すると StateMachine にクレジットエラーを出力します。

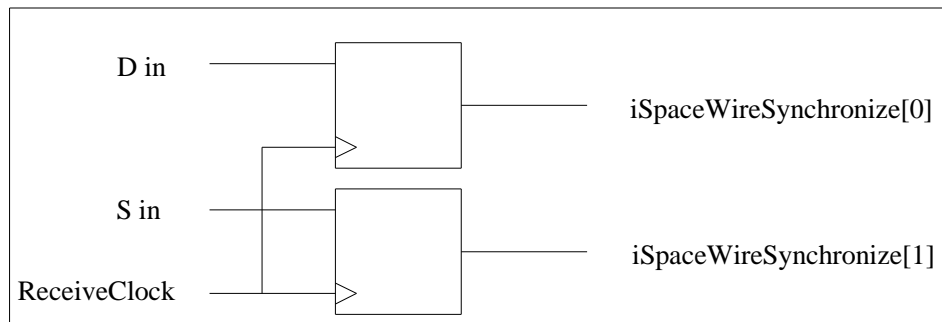
送信レートは TransmitClock と transmitClockDivideValue 値で決まります。

LinkUP 中でも transmitClockDivideValue の値を変える事で送信レートを変更する事ができます、Link 初期化時は 10Mbps で動作します。

### 4.2. Receiver

データ及びストローブ信号を内部クロックに同期させてからデコードしています。

データ及びストローブ信号の同期化を下図に示します。



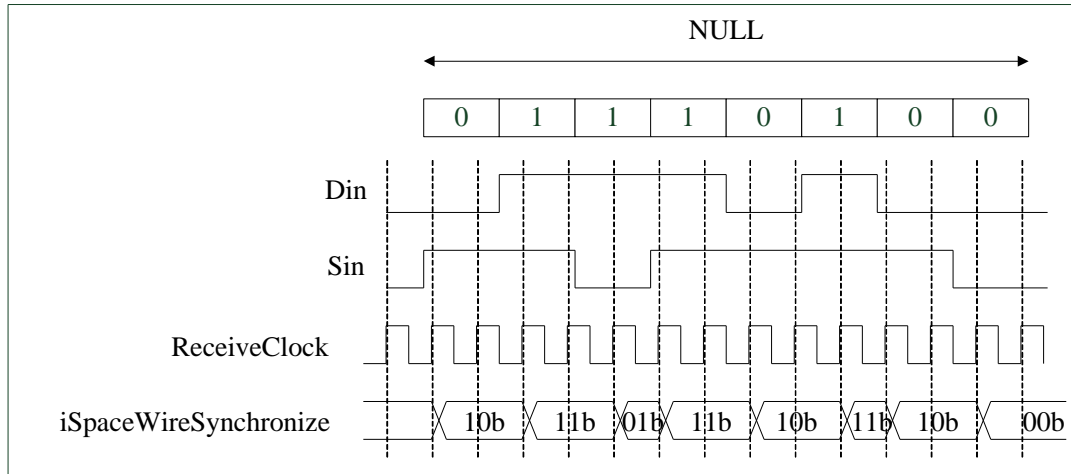
N-Char (データ、EOP、EEP) を受信した時はデータを ReceiveFIFO に書き込み StateMachine に通知されます。NULL、FCT、TimeCode を受信した時も StateMachine に通知されます、FCT のみ Transmitter にも通知されます。

N-Char (データ、EOP、EEP) 以外は ReceiveFIFO には書き込まれません。

エスケープエラー、パリティエラーを受信した時はデータを破棄し、StateMachine に通知されます。

また、データ及びストローブ信号が 850ns 変化しない時は断線エラーとして StateMachine に通知されます。

NULL受信時のReceiverタイミングチャートを下図に示します。



### 4.3. StateMachine

リンクインターフェースの初期化、通常動作、エラー復帰処理を管理しています。

- ErrorReset** . . . EnableTransmitter、EnableReceiverが“L”になり、Transmitter、Receiverの動作が停止します。  
6.4us経過後ErrorWaitに遷移します。
- ErrorWait** . . . EnableReceiveが“H”になり、Receiverが動作します。  
12.8us経過後Readyに遷移します。  
ReceiverがErrorを検知するとErrorResetに遷移します。
- Ready** . . . LinkEnableが“H”になるとStartedに遷移します。  
ReceiverがErrorを検知するとErrorResetに遷移します。
- Started** . . . EnableTransmit、SendNULLsが“H”になり、TransmitterはNULLを送信します。  
EnableTransmit、SendNULLsが“H”になり、TransmitterはNULLを送信します。  
ReceiverがNULLを受信するとConnectingに遷移します。  
ReceiverがErrorを検知又はStartedに遷移して12.8us経過後ErrorResetに遷移します。
- Connecting** . . . SendFCTsが“H”になり、TransmitterはFCTを送信します。  
ReceiverがFCTを受信するとRunに遷移します。  
ReceiverがErrorを検知又はConnectingに遷移して12.8us経過後ErrorResetに遷移します。
- Run** . . . SendTimeCode、SendNCharが“H”になり、TransmitterはTimeCode、N-Charの送信ができるようになります。  
ReceiverがErrorを検知するとErrorResetに遷移します。

#### 4. 4. **Timer**

Link 初期化に時使用する 6.4us、12.8us タイマーを生成しています。

#### 4. 5. **TimeCode**

Receiver が TimeCode を受信した場合、現在の値より 1 だけ大きい値かを確認し TickOut を出力します。  
無効タイムコードを受信した場合は現在の値を更新しますが、TickOut は出力しません。

#### 4. 6. **Statistical Information**

Transmitter が送信した EOP、EEP、N-Char。Receiver が受信した EOP、EEP、N-Char。  
リンクアップシーケンス時の LinkUP、LinkDown 回数をカウントしています。  
各情報は最大 4294967296 回カウントする事ができます。  
statisticalInformationClear を “H” にする事で積算回数をクリアする事ができます。

#### 4. 7. **TransmitFIFO**

ホスト側が送信したい N-Char を書き込む 9×64FIFO です。  
最大で 56 個の N-Char を書き込む事ができます、56 個のデータを書き込むと FULL が “H” になります。

#### 4. 8. **ReceiveFIFO**

Receiver がデコードした N-Char(データ、EOP、EEP)を書き込む 9×64FIFO です。  
最大で 56 個の N-Char が書き込まれています、FIFO にデータが書き込まれていない時は Empty が “H” になります。



## 5. インターフェース信号

SpaceWire CODEC IP は各 Clock の立ち上がりエッジで動作します。

入力信号は“H”にする事で、その機能が有効になります。

出力信号は“H”の時に、その機能が有効になります。

SpaceWire CODEC IP インターフェース信号を以下に示します。

信号名	I/O	説明
Clock	I	システムクロック
transmitClock	I	送信クロック
receiveClock	I	受信クロック
Reset	I	リセット
transmitFIFOWriteEnable	I	書き込み許可
transmitFIFODataIn[8:0]	I	書き込みデータ
transmitFIFOFull	O	FULL
transmitFIFODataCount[5:0]	O	データ保存数
receiveFIFOReadEnable	I	読み出し許可
receiveFIFODataOut[8:0]	O	読み出しデータ
receiveFIFOFull	O	FULL
receiveFIFOEmpty	O	Empty
receiveFIFODataCount[5:0]	O	データ保存数
tickIn	I	タイムコード送信
timeIn[5:0]	I	送信タイムコード
controlFlagsIn[1:0]	I	送信タイムコード制御フラグ
tickOut	O	受信タイムコード
timeOut[5:0]	O	受信タイムコード
controlFlagsOut[1:0]	O	受信タイムコード制御フラグ
linkStart	I	“H”を入力する事で LinkStart を有効にします。 リンクアップシーケンスは自動的に Ready から Started に遷移します。 “L”を入力する事で LinkStart を無効にします。
linkDisable	I	“H”を入力する事でリンクを Disable にします。 “L”を入力する事で Disable を解除します。 LinkStart 及び AutoStart は LinkDisable が “L” の時のみ有効になります。
autoStart	I	“H”を入力する事で AutoStart を有効にします。 NULL を受信する事でリンクが始動するようになります。 “L”を入力する事で AutoStart を無効にします。
transmitClockDivideValue[5:0]	I	リンクアップ後の転送レート設定値 $\text{転送レート} = \frac{\text{TransmitClock}[\text{MHz}]}{(\text{TransmitClockDivideValue} + 1)}$
creditCount[5:0]	O	リンク相手側クレジットカウンタ(送信可能数) Receiver が FCT を受信すると CreditCount を 8 ずつ加算します。 Transmitter が N-Char を送信すると CreditCount を 1 ずつ減算します。

outstandingCount[5:0]	O	自ノードのクレジットカウンタ(受信可能数) Transmitter が FCT を送信すると OutstandingCount を 8 ずつ加算します。 Receiver が N-Char を受信すると OutstandingCount を 1 ずつ減算します。
transmitActivity	O	データ読み出し中
receiveActivity	O	データ書き込み中
spaceWireDataIn	I	SpaceWireData 入力
spaceWireStrobeIn	I	SpaceWireStrobe 入力
spaceWireDataOut	O	SpaceWireData 出力
spaceWireStrobeOut	O	SpaceWireStrobe 出力
linkStatus[15:0]	O	リンクステータス情報
linkStatus[0]		EnableTransmit リンクアップシーケンスが Started に遷移すると “H” になります。 リンクダウン又はリンクアップシーケンス中のエラーで “L” になります。
linkStatus[1]		EnableReceive リンクアップシーケンスが ErrorWait に遷移すると “H” になります。 リンクダウン又はリンクアップシーケンス中のエラーで “L” になります。
linkStatus[2]		SendNULLs リンクアップシーケンスが Started に遷移すると “H” になります。 リンクダウン又はリンクアップシーケンス中のエラーで “L” になります。
linkStatus[3]		SendFCTs リンクアップシーケンスが Connecting に遷移すると “H” になります。 リンクダウン又はリンクアップシーケンス中のエラーで “L” になります。
linkStatus[4]		SendNChar リンクアップシーケンスが Run に遷移すると “H” になります。 リンクダウン又はリンクアップシーケンス中のエラーで “L” になります。
linkStatus[5]		SendTimeCode リンクアップシーケンスが Run に遷移すると “H” になります。 リンクダウン又はリンクアップシーケンス中のエラーで “L” になります。
linkStatus[6]		Reseve “0” 固定
linkStatus[7]		SpaceWireReset リンクアップシーケンスが ErrorReset に遷移すると “H” になります。 リンクアップシーケンスが ErrorWait に遷移すると “L” になります。
linkStatus[8]		リンク確立信号 リンクアップシーケンスが Run に遷移すると “H” になります。 リンクダウンすると “L” になります。
linkStatus[9]		Transmitter が N-Char を送信すると 1 クロックだけ “H” になります。
linkStatus[10]		Receiver が N-Char を受信すると 1 クロックだけ “H” になります。
linkStatus[11]		Receiver が NULL を受信すると 1 クロックだけ “H” になります。
linkStatus[12]		Receiver が FCT を受信すると 1 クロックだけ “H” になります。
linkStatus[13]		Receiver が EOP を受信すると 1 クロックだけ “H” になります。
linkStatus[14]		Receiver が EEP を受信すると 1 クロックだけ “H” になります。
linkStatus[15]		Reseve “0” 固定

errorStatus[7:0]	O	リンクエラー情報
errorStatus[0]		CharacterSequenceError リンクアップシーケンス中に予期しない FCT や N-Char を受信すると“H”になります。 リンクアップシーケンスが ErrorReset に遷移すると“L”になります。
errorStatus[1]		CreditError CreditCount が 56 以上になった時又は OutstandingCount が 0 の時に N-Char を受信すると“H”になります。 リンクアップシーケンスが ErrorReset に遷移すると“L”になります。
errorStatus[2]		RxErr EnableReceive が“H”の時に ParityError 又は DisconnectError 又は ESCError を検知すると“H”になります。 リンクアップシーケンスが ErrorReset に遷移すると“L”になります。
errorStatus[3]		Reserve “0” 固定
errorStatus[4]		Parity Error EnableReceive が“H”の時にパリティエラーを検知すると“H”になります。 リンクアップシーケンスが ErrorReset に遷移すると“L”になります。
errorStatus[5]		DisconnectError EnableReceive が“H”の時にデータ及びストロブ信号が 850ns 間変化がないと“H”になります。 リンクアップシーケンスが ErrorReset に遷移すると“L”になります。
errorStatus[6]		ESCError EnableReceive が“H”の時に ESC キャラクターの後に受信した制御キャラクターが FCT 以外の時に“H”になります。 リンクアップシーケンスが ErrorReset に遷移すると“L”になります。
errorStatus[7]		Reserve “0” 固定
statisticalInformationClear	I	“H”を入力する事で統計情報がクリアされます。 “L”を入力する事でカウントを開始します。
statisticalInformation 8×[31:0]	O	統計情報
statisticalInformation 0 [31:0]		Transmitter が EOP を送信した積算回数
statisticalInformation 1 [31:0]		Receiver が EOP を受信した積算回数
statisticalInformation 2 [31:0]		Transmitter が EEP を送信した積算回数
statisticalInformation 3 [31:0]		Receiver が EEP を受信した積算回数
statisticalInformation 4 [31:0]		Transmitter が N-Char を送信した積算回数
statisticalInformation 5 [31:0]		Receiver が N-Char を受信した積算回数
statisticalInformation 6 [31:0]		リンクアップシーケンスがリンクアップした積算回数
statisticalInformation 7 [31:0]		リンクアップシーケンスがリンクダウンした積算回数

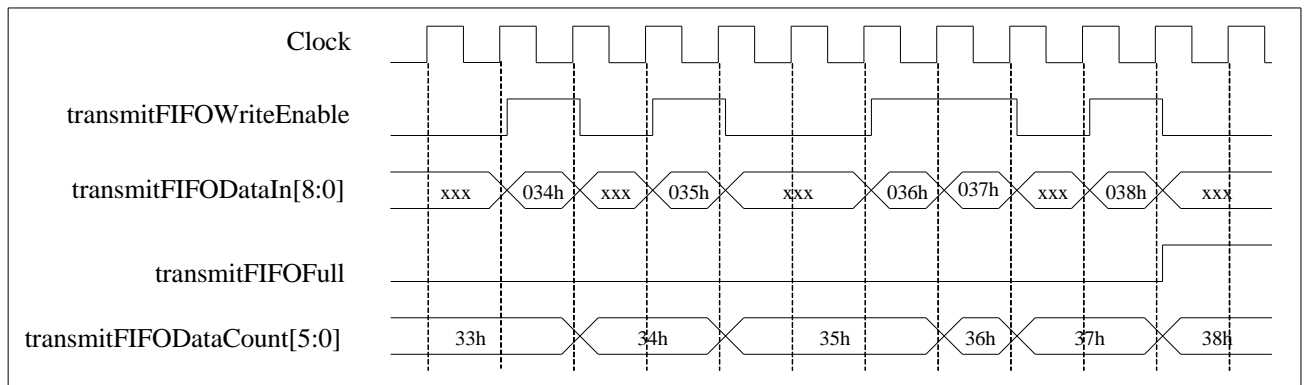
## 6. タイミング図

SpaceWire CODEC IP インターフェース信号のタイミングチャートを示します。

### 6.1. TransmitFIFO

ホスト側が送信したい N-Char を書き込む FIFO です。

TransmitFIFO 書き込み時のタイミングチャートを下図に示します。



Clock の立ち上がり時に WriteEnable が “H” であれば FIFO にデータを書き込む事ができます。  
56 データ書き込むと FIFOFull が “H” になります。

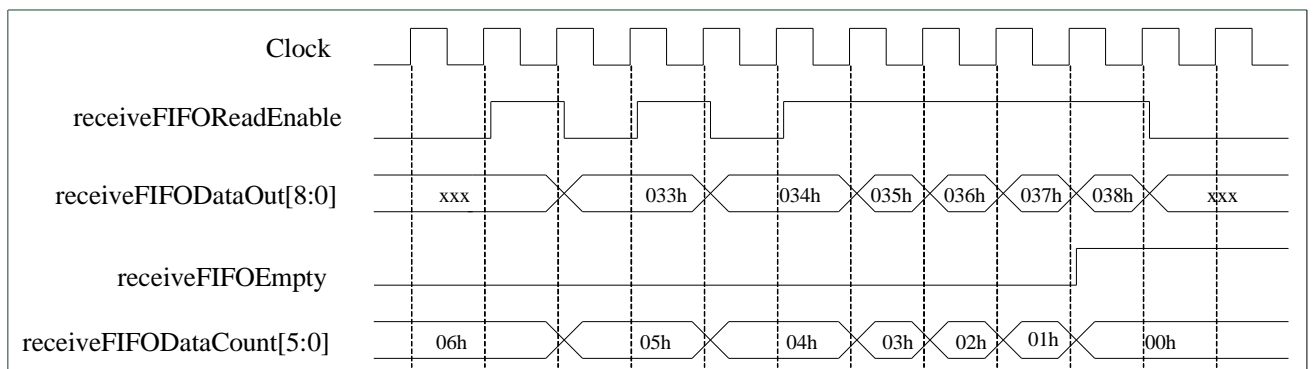
※FIFOFull が “H” の場合でもその後、7 データまでは書き込む事ができます、ただし 8 データ以上書き込むと DataCount がズレてしまい正しく書き込む事ができなくなりますので注意してください。

### 6.2. ReceiveFIFO

SpaceWire で受信した N-Char が書き込まれている FIFO です。

EOP、EEP 以外の制御キャラクターは ReceiveFIFO には書き込まれません。

ReceiveFIFO 読み出し時のタイミングチャートを下図に示します。

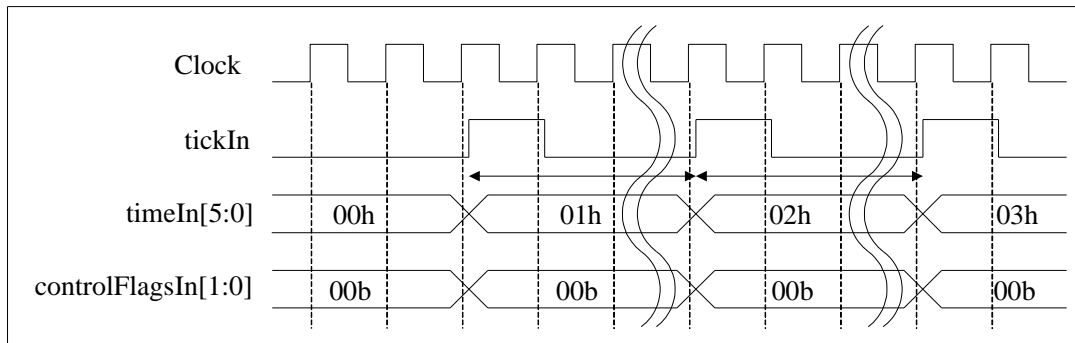


Clock の立ち上がり時に ReadEnable が “H” であれば FIFO からデータを読み出す事ができます。

※FIFOEmpty が “H” の時はデータを読み出す事はできません、連続読み出しをする時は注意してください。

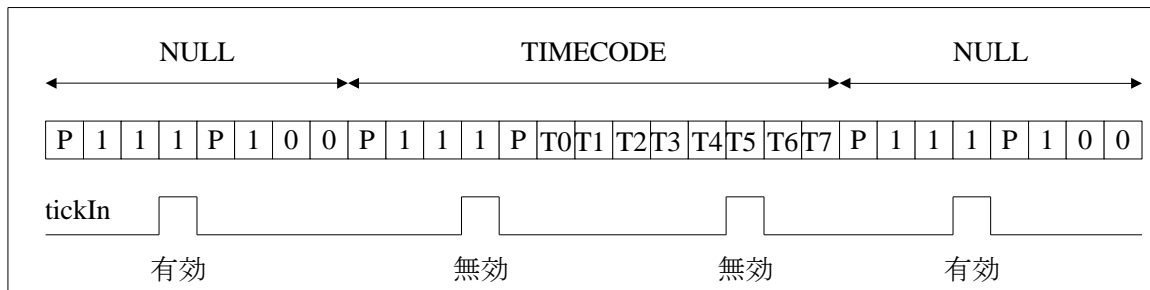
### 6.3. 送信タイムコード

タイムコード送信時のタイミングチャートを下図に示します。



Clock 同期の TickIn 信号を入力すると TimeIn、ControlFlagsin の値をタイムコードとして送信します。TimeIn、ControlFlagsIn の値は TickIn 毎に更新してください。

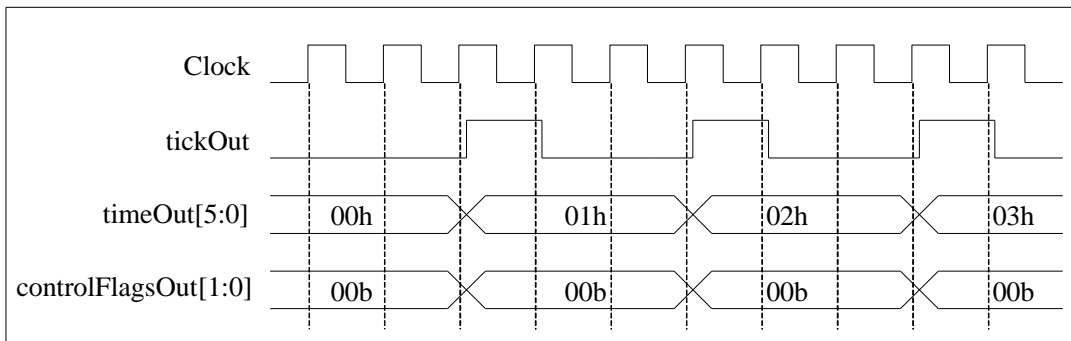
TickIn 信号のタイミングチャートを下図に示します。



Transmitter はタイムコード送信中の TickIn 信号を無視します。TickIn の周期は送信レートの周期 25 倍以上の時間を空けて下さい。

### 6.4. 受信タイムコード

タイムコード受信時のタイミングチャートを下図に示します。



タイムコードを受信した場合 Clock に同期した TickOut 信号が出力します。受信したタイムコードの値が TimeOut、ControlFlagsOut に出力します。

## 7. パッケージ

パッケージ内で宣言されている Constant 宣言は入力クロックによって設定値を変更してください。  
Constant 宣言を以下に示します。

Constant 名	type	Bit 幅	説明
gDisconnectCountValue	integer	0 to 255	<p>断線エラー時間設定値。 DS 信号が約 850ns 間変化しないと断線エラーを発生させます。</p> $\text{設定値} = \frac{850}{T} \quad T = \text{ReceiveClock の周期}$ <p>デフォルト値は 141(166MHz)に設定されています。</p>
gTimer6p4usValue	integer	0 to 1023	<p>6.4After 信号発生時間設定値。 ステートマシンの状態遷移に必要な after6p4us 信号を 6.4us で発生させます。</p> $\text{設定値} = \frac{6400}{T} \quad T = \text{Clock の周期}$ <p>デフォルト値は 320(50MHz)に設定されています。</p>
gTimer12p8usValue	integer	0 to 2047	<p>12.8After 信号発生時間設定値。 ステートマシンの状態遷移に必要な after12p8us 信号を 12.8us で発生させます。</p> $\text{設定値} = \frac{12800}{T} \quad T = \text{Clock の周期}$ <p>デフォルト値は 640(50MHz)に設定されています。</p>
gInitializeTransmitClock DivideValue	std_logic _vecto	6	<p>初期化時の送信レート設定値。 初期化時の送信レートを 10Mbps になるように設定します。</p> $\text{設定値} = \frac{f[\text{MHz}]}{10} - 1 \quad f = \text{TransmitClock}$ <p>デフォルト値は 9(100MHz)に設定されています。</p>

## 8. ファイル構成

SpaceWire CODEC IP ファイル構成を以下に示します。

ファイル名	説明
SpaceWireCODECIP.vhdl	SpaceWire CODEC IP トップモジュール
SpaceWireCODECIPReceiverSynchronize.vhdl	SpaceWire 受信モジュール
SpaceWireCODECIPTransmitter.vhdl	SpaceWire 送信モジュール
SpaceWireCODECIPStateMachine.vhdl	リンクインターフェースステートマシンモジュール
SpaceWireCODECIPTimer.vhdl	タイマーモジュール
SpaceWireCODECIPTimeCodeControl.vhdl	タイムコード受信モジュール
SpaceWireCODECIPStatisticalInformationCount.vhdl	統計情報カウンタモジュール
SpaceWireCODECIPFIFO9x64.vhdl	FIFO _RTL モジュール
SpaceWireCODECIPLinkInterface.vhdl	リンクインターフェーストップモジュール
SpaceWireCODECIPSynchronizeOnePulse.vhdl	クロック同期モジュール
SpaceWireCODECIPPackage.vhdl	Constant 宣言のパッケージ

※ 上記ソースの文字コードは UTF-8 です。

## 9. パフォーマンス

SpaceWire CODEC IP のリソースの使用率を以下に示します。

- デバイス Xilinx 製 Spartan-6 XC6SLX75
- 合成ツール ISE 14.4

リソースの使用率は合成オプションやデバイスによって異なります。

モジュール名	Flip-Flops	LUT	Slices	RAM (LUT 内)
SpaceWireCODECIP	691	662	306	24
SpaceWireCODECIPReceiverSynchronize	59	104	39	0
SpaceWireCODECIPTransmitter	114	116	63	0
SpaceWireCODECIPStateMachine	28	30	27	0
SpaceWireCODECIPTimer	24	25	24	0
SpaceWireCODECIPTimeCodeControl	25	9	9	0
SpaceWireCODECIPStatisticalInformationCount	274	266	86	0
SpaceWireCODECIPFIFO9x64	82	77	25	12



## 10. タイミング制約

### 10.1. ucf ファイル

タイミング制約は、合成回路の正確なタイミングを確保するために必要とされます。  
全てのクロック乗せ換えにはタイミング制約が必要です。

通信レート 100Mbps 時のタイミング制約を以下に示します。

```
clock          = 50MHz
transmitClock  = 100MHz
receiveClock   = 166.6MHz
```

```
NET "clock"      TNM_NET  = "SYS_CLK";
NET "transmitClock"TNM_NET  = "TX_CLK";
NET "receiveClock"TNM_NET  = "RX_CLK";
```

```
TIMESPEC TS_SYS_CLK_to_TX_CLK = FROM "SYS_CLK" TO "TX_CLK" TIG;
TIMESPEC TS_TX_CLK_to_SYS_CLK = FROM "TX_CLK" TO "SYS_CLK" TIG;
TIMESPEC TS_TX_CLK_to_RX_CLK  = FROM "TX_CLK" TO "RX_CLK" TIG;
TIMESPEC TS_RX_CLK_to_TX_CLK  = FROM "RX_CLK" TO "TX_CLK" TIG;
TIMESPEC TS_SYS_CLK_to_RX_CLK = FROM "SYS_CLK" TO "RX_CLK" TIG;
TIMESPEC TS_RX_CLK_to_SYS_CLK = FROM "RX_CLK" TO "SYS_CLK" TIG;
```

```
TIMESPEC "TS_clk"      = PERIOD  "SYS_CLK" 20 ns HIGH 50 %;
TIMESPEC "TS_txclk "   = PERIOD  "TX_CLK"  10 ns HIGH 50 %;
TIMESPEC "TS_rxclk "   = PERIOD  "RX_CLK"   6 ns  HIGH 50 %;
```

## 10.2. クロック

SpaceWire CODEC IP には Clock、TransmitClock、ReceiveClock を供給する必要があります。

通信レート 100Mbps 時の推奨クロックを以下に示します。

Clock	50-100MHz
TransmitClock	100.0MHz
ReceiveClock	166.6MHz
gInitializeTransmitClockDivideValue	001001
gDisconnectCountValue	141

入力クロックの条件

- ReceiveClock は受信レートの 1.5 倍程度の周波数を入力してください。
- TransmitClock を 100MHz 以下で使用する時は ReceiveClock も同じ比率で下げてください。  
TransmitClock を下げた時に ReceiveClock を同じ比率で下げないと使用する送受信レートによって制御コードの FCT を連続で受信した場合正しく処理出来ない事があります。

例 1: TransmitClock を 100MHz 以下で使用する時。

Clock	50.0MHz
TransmitClock	50.0MHz
ReceiveClock	83.3MHz
gInitializeTransmitClockDivideValue	000100
gDisconnectCountValue	71

例 2: FCT 連続受信を正しく処理出来ないクロックの組み合わせ。

Clock	50.0MHz
TransmitClock	10.0MHz
ReceiveClock	166.6MHz
gInitializeTransmitClockDivideValue	000000
gDisconnectCountValue	141

例2の入力クロックで送信レート 10Mbps、受信レート 100Mbps で使用すると TransmitClock の周期 100ns 以内に FCT を連続受信した場合、2 個以上の FCT 受信があるにも関わらず、1 個分の FCT として処理されます。

この状態が続くと最終的にクレジットカウントが 8 より増えない可能性があります。