サイプレスはインフィニオン テクノロジーズになりました

この表紙に続く文書には「サイプレス」と表記されていますが、これは同社が最初にこの製品を開発したからです。新規および既存のお客様いずれに対しても、引き続きインフィニオンがラインアップの一部として当該製品をご提供いたします。

文書の内容の継続性

下記製品がインフィニオンの製品ラインアップの一部として提供されたとしても、それを理由としてこの文書に変更が加わることはありません。今後も適宜改訂は行いますが、変更があった場合は文書の履歴ページでお知らせします。

注文時の部品番号の継続性

インフィニオンは既存の部品番号を引き続きサポートします。ご注文の際は、データシート記載の注文部品番号をこれまで通りご利用下さい。

EZ-USB® CX3 テクニカル リファレンス マニュアル

(EZ-USB FX3 テクニカル リファレンス マニュアル補完文書)

文書番号 001-92482 Rev. *B

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
www.cypress.com

サイプレス EZ-USB CX3

1.1 はじめに

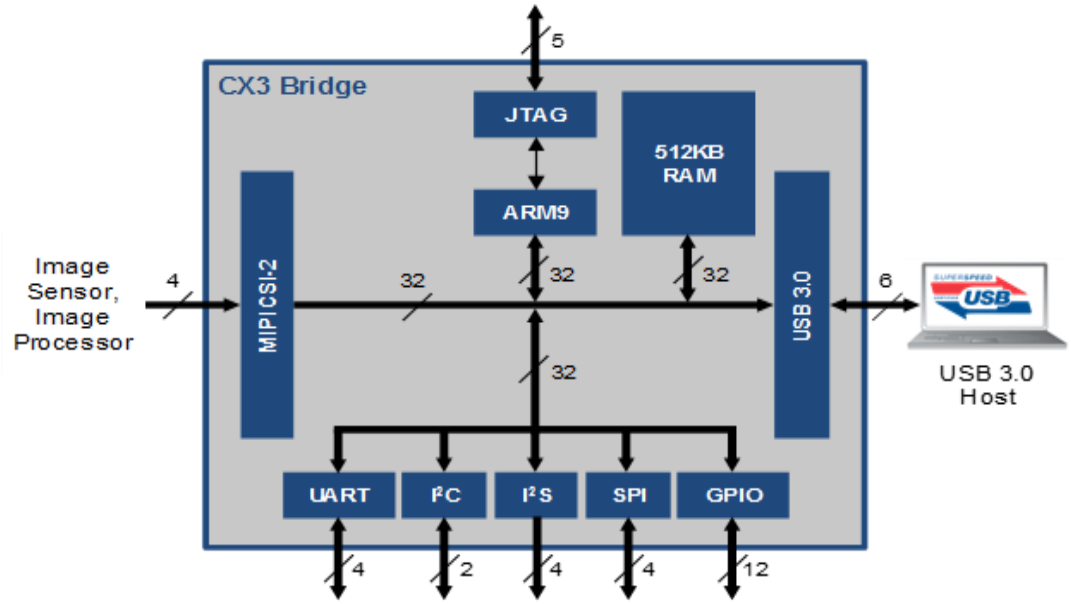
サイプレス EZ-USB® CX3 は、USB 3.0 カメラ コントローラであり、開発者が MIPI アライアンス (携帯機器産業プロセッサ インターフェース協会) のカメラ シリアル インターフェース 2 (CSI-2) 規格に準拠したイメージ センサーに USB 3.0 の接続機能を追加することができます。これは MIPI CSI-2-USB ブリッジとして機能します。

EZ-USB CX3 は、汎用プログラマブル インターフェース II (GPIF II) に接続された MIPI CSI-2 レシーバを内蔵した EZ-USB FX3 デバイスの応用品種です。CX3 により、MIPI CSI-2 インターフェースに対応したイメージ センサーにスーパースピード USB 接続機能を追加することができます。

CX3 は、MIPI CSI-2 仕様 (バージョン 1.01) に準拠しており、最大 4 本のデータ レーンでそれぞれの最大速度が 1 ギガビット毎秒 (Gbps)、合計 4Gbps の帯域幅に対応しています。

CX3 は、高解像度または高速のイメージ キャプチャ アプリケーションに最適であり、30fps で最大 1080p または 60fps で最大 720p の非圧縮のビデオをストリーミングすることができます。CX3 は、RAW8/10/12/14、YUV422、RGB888/666/565、ユーザー定義 8 ビットを含む幅広いイメージ フォーマットに対応しています。

図 1: EZ-USB® CX3 デバイス



実績のある EZ-USB FX3 プラットフォームに基づいて、CX3 は 200MIPS の計算能力を供給する ARM9™ CPU と 512KB SRAM を内蔵しています。CX3 は、I²C、SPI、UART など多くのペリフェラル インターフェースに対応しています。これらのインターフェースは、オートフォーカス、パン、チルト、ズーム (PTZ)、その他のカメラ制御機能に対応するようにプログラムすることができます。

CX3 は FX3 と同じアプリケーション開発ツールを使用します。FX3 ソフトウェア開発キットは、CX3 のサポートおよびサンプル アプリケーションを提供しており、市場投入までの時間を短縮することができます。CX3 は USB 3.0 v1.0 仕様に準拠しているため、USB 2.0 との後方互換性があります。

このテクニカル リファレンス マニュアル (TRM) は *EZ-USB FX3 TRM* の補完文書であり、追加された CX3 MIPI CSI-2 レシーバ機能ブロックの詳細を説明します。この TRM 補完文書の 1.5 節～1.9 節は、MIPI CSI-2 レシーバ機能ブロックおよび機能固定 GPIF II ステート マシンについて説明します。1.10 節は、MIPI CSI-2 レシーバ インターフェースのレジスタの詳細を説明します。CPU サブシステム、メモリ グローバル コントローラ、DMA、USB、低帯域幅 (シリアルと GPIO) ペリフェラルなどの既存の FX3 機能ブロックの詳細な説明は、*EZ-USB FX3 TRM* に記載されています。この TRM で使用する技術用語は「用語集」に定義されています。

図 2 は、FX3 のブロック図です。一部の機能ブロックが CX3 に含まれていません。これらは赤色で塗られています。青色で塗られた GPIF II は CX3 に含まれていますが、カメラ インターフェース固有の機能のみを持っています。

図 2: FX3 ブロック図

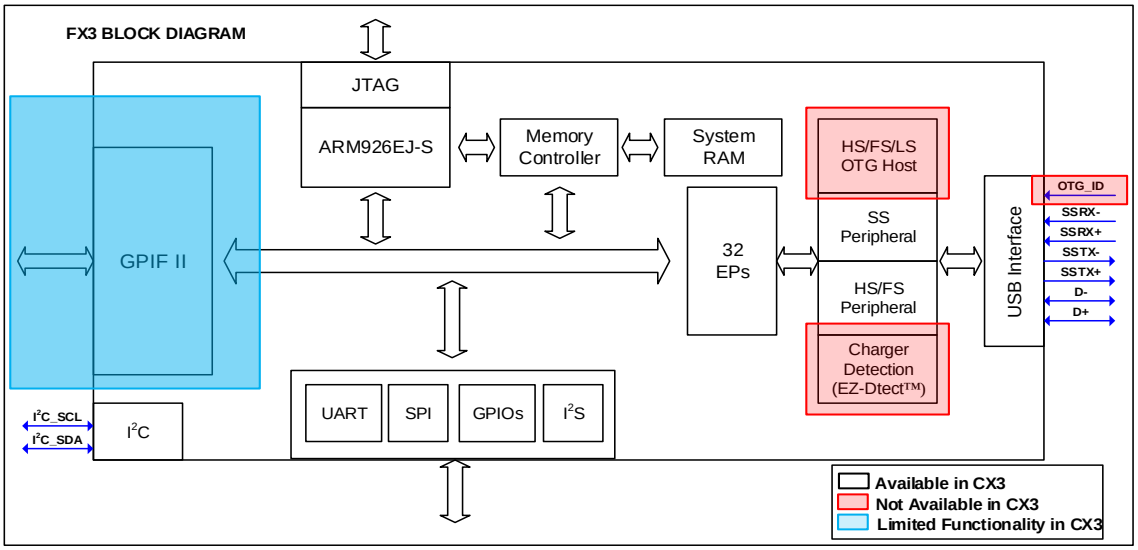
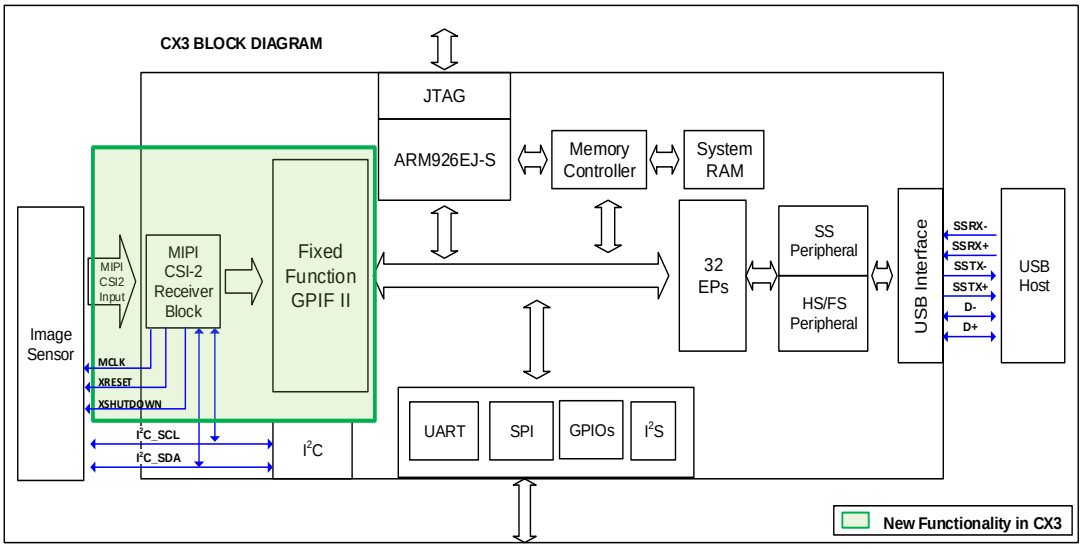


図 3 は、CX3 内の追加された MIPI CSI-2 機能ブロックと機能固定 GPIF II インターフェースを含む CX3 ブロック図を示します。これらは緑色で強調表示されています。

図 3: CX3 ブロック図



CX3 と FX3 デバイスの差異は以下の通りです。

表 1: FX3 と CX3 デバイスの差異

機能	FX3 でのサポート	CX3 でのサポート
P ポートのサポート		
GPIF II に実装された同期 ADMux	有	無
GPIF II に実装された非同期 SRAM	有	無
GPIF II に実装された非同期 ADMux	有	無
MMC スレーブ	有	無
GPIF II	有	CX3 MIPI CSI-2 レシーバ ブロックと通信するように実装された機能固定 GPIF II ステート マシン
低帯域幅ペリフェラル		
I^2S マスタ (トランスミッタのみ)	有	有
SPI マスタ	有	有
UART	有	有
I^2C マスタ コントローラ	有	有
U ポートのサポート		
USB 3.0 ペリフェラル	有	有
USB 2.0 ペリフェラル	有	有
32 個の物理エンドポイント	有	有
Charger Detection 1.1 (充電器検出) のサポート (EZ-Dtect™)	有	無
アクセサリ充電器アダプタ (ACA) のサポート	有	無
内蔵ハイスピード USB スイッチ	有	無
Carkit UART パススルー モード	有	無
USB OTG (ハイスピード、フルスピード、ロースピードのホストまたはペリフェラル)	有	無
クロック入力周波数:	19.2MHz、 26MHz、 38.4MHz、 52MHz	19.2MHz
水晶発振器入力: 19.2MHz	有	無
デバッグ専用 JTAG サポート (バウンダリ スキャンに非対応)	有	有
VBUS、VBAT 信号	5V と 3.3V 動作する個別の VBUS と VBAT 信号	VBUS と VBAT は VUSB に接続。3.3V と 5V 動作に対応
GPIO	42	12
イメージ センサー用のクロック出力	無	有
MIPI CSI-2 入力	無	有

1.2 CX3 の特長

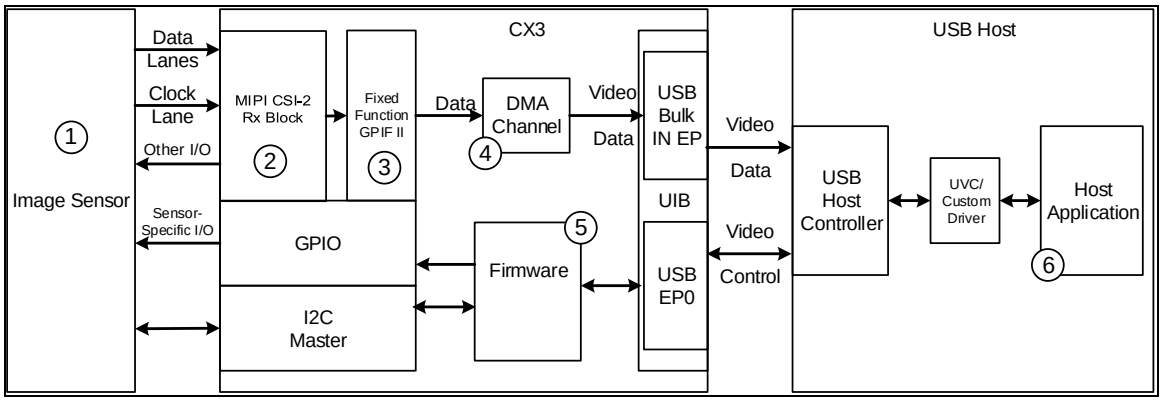
CX3 の特長は以下の通りです。

- USB 3.0 仕様 v1.0 に準拠した USB 3.0 と USB 2.0 ペリフェラル コントローラ
- MIPI CSI-2 RX インターフェース
  - MIPI CSI-2 準拠 (v1.01 Rev. 0.04;2009 年 4 月 2 日付け)
  - 最大 4 本のデータ レーンに対応、それぞれ最大 1Gbps に対応
  - イメージ センサー設定用のカメラ制御インターフェース (I^2C を介して) に対応
- 以下のビデオ データ フォーマットに対応
  - RAW 8/10/12/14
  - YUV 422 (8/10 ビット)
  - RGB 888 / 666 / 565
  - ユーザー定義 8 ビット
- カメラ関連の機能 (照明、同期入力、同期出力など) の制御用の 12 個の GPIO
- FX3 と同様に、I^2C、SPI、I^2S 出力、UART インターフェースに対応
- デバッグ用 JTAG インターフェース

1.3 ブロック図

図4は、CX3を使用してデータをイメージ センサーからUSBホストに転送する標準的なシステムの詳細なブロック図です。

図 4: システムのブロック図



主なブロックに番号を付け、それぞれが実行するタスクを以下に説明します。

1. MIPI CSI-2 ベースのイメージ センサーは、CX3 に接続しており、カメラ制御インターフェース (I^2C) バスを使用して設定されます。
2. CX3 MIPI CSI-2 レシーバ ブロックは、イメージ センサーからのデータを読み出してデシリアライズ (パラレル化) し、レーンを結合し、パケットを取り出し、それをパラレル入力として機能固定 GPIF II ブロックに送信します。
3. GPIF II ブロックとその機能固定ステート マシンは、DMA チャネルを使用してイメージ センサー データを USB インターフェースに転送します。
4. DMA チャネルは、イメージ データを GPIF II ブロックから USB インターフェース ブロック (UIB) に転送します。
5. CX3 ファームウェアは、CX3 ハードウェア ブロックを初期化し、イメージ センサーと MIPI CSI-2 コントローラを設定し、USB インターフェースを制御し、全ての USB プロトコル要求を処理します。CX3 ファームウェアは、ビデオ ストリーム データを USB インターフェースに転送する前にそれにクラス固有のヘッダを追加するようカスタマイズすることができます。
6. イメージ信号プロセッサまたはビデオ ストリーム プレーヤーなどのホスト アプリケーションは、ホスト PC 上でビデオ ストリームとセンサーを設定するよう制御要求を発行し、ビデオ ストリームを処理して表示します。

1.4 MIPI CSI-2 ブロックのコンフィギュレーション API

MIPI CSI-2 レシーバ ブロックのコンフィギュレーション API により、ユーザー アプリケーションは、カメラインターフェースを初期化、設定し、それに対して電源管理を行うことができます。これらの API は、I^2C バスを介してカメラと通信します。

モジュール化のため、MIPI CSI-2 のコンフィギュレーション API は個別の API ライブラリ (*cyu3mipicsi.a*) にコンパイルされています。このライブラリはアプリケーションの機能が必要となる場合にのみアプリケーションとリンクされます。

API ライブラリで提供される主な API は以下の通りです。

表 2: EZ-USB FX3 SDK における CX3 固有の API

API 名	API の説明
CyU3PMipicsiInit()	MIPI CSI-2 ブロックを初期化する API
CyU3PMipicsiDeInit()	MIPI CSI-2 ブロックを非初期化する API
CyU3PMipicsiSetIntfParams()	MIPI CSI-2 ブロックでクロックとインターフェースを設定する API
CyU3PMipicsiQueryIntfParams()	MIPI CSI-2 ブロックの設定を問い合わせる API
CyU3PMipicsiSleep()	MIPI CSI-2 ブロックを低消費電力スリープ モードに移行させる API
CyU3PMipicsiWakeup()	MIPI CSI-2 ブロックを低消費電力スリープ モードからアクティブ モードに復帰させる API
CyU3PCx3DeviceReset()	MIPI CSI-2 ブロックをリセットする API
CyU3PCx3DeviceReset()	CX3 デバイスに対しウォーム リセットまたはコールド リセットを実行する API
CyU3PMipicsiSetSensorControl()	MIPI CSI-2 XRESET と XSHUTDOWN 信号をイメージ センサーに駆動する API XRESET は、イメージ センサーをリセットするのに使用される CX3 出力信号 XSHUTDOWN は、イメージ センサーの電力モードを制御するのに使用される CX3 出力信号
CyU3PMipicsiGpifLoad()	機能固定 GPIF II 波形をロードし、機能固定 GPIF II バス幅と DMA バッファ サイズを設定する API

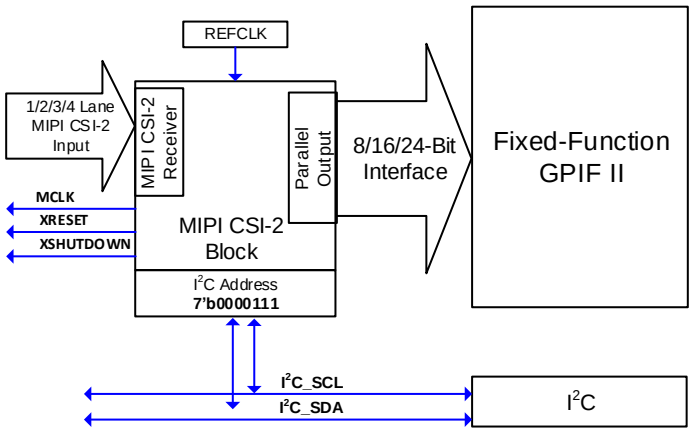
1.11 節で API を詳しく説明します。

API の詳細は、EZ USB FX3 SDK 同梱の「*EZ-USB FX3 SDK Firmware API Guide*」の「*MIPI CSI-2 and Fixed-Function GPIF Interface for CX3*」の節に記載されています。

1.5 MIPI CSI-2 ブロック

CX3 デバイスは MIPI CSI-2 ブロックを内蔵しています。このブロックは、片側が GPIF II インターフェースに固定接続され、MIPI CSI-2 に準拠したイメージ センサーと通信するために反対側が MIPI CSI-2 インターフェースを備えています。これは最大 4 本の MIPI CSI-2 データ レーンに対応しており、レーンごとの最大速度は 1Gbps です。MIPI CSI-2 レシーバは、最大 100MHz のクロックを供給される 8 ビット、16 ビットまたは 24 ビット データ バスを介して機能固定 GPIF II コントローラに接続します。MIPI CSI-2 ブロックは、I²C を介して設定され、7 ビットの I²C スレーブ アドレス 7'b0000111 で CX3 I²C バスで使用可能になります。

図 5. CX3 MIPI CSI-2 ブロック



1.6 CX3 MIPI CSI-2 ストリーム フォーマット

CX3 内の MIPI CSI-2 レシーバ ブロックは以下のストリーム フォーマットと出力モードに対応しています。

表 3: CX3 MIPI CSI-2 ストリーム フォーマット

フォーマットとモード	CX3 ファームウェアのストリーム フォーマット名	フォーマットの説明、ピクセル深度	CSI-2 データ タイプa	GPIF II バス幅	出力ストリーム
RAW8	CY_U3P_CSI_DF_RAW8	RAW フォーマット、8 ビット/ピクセル	0x2A	8 ビット	RAW [7:0]
RAW10	CY_U3P_CSI_DF_RAW10	RAW フォーマット、10 ビット/ピクセル	0x2B	16 ビット	6'b0、RAW [9:0]
RAW12	CY_U3P_CSI_DF_RAW12	RAW フォーマット、12 ビット/ピクセル	0x2C	16 ビット	4'b0、RAW [11:0]
RAW14	CY_U3P_CSI_DF_RAW14	RAW フォーマット、14 ビット/ピクセル	0x2D	16 ビット	2'b0、RAW[13:0]
RGB888	CY_U3P_CSI_DF_RGB888	RGB 888 フォーマット、24 ビット/ピクセル	0x24	24 ビット	R[7:0]、G[7:0]、B[7:0]
RGB666 モード 0	CY_U3P_CSI_DF_RGB666_0	RGB 666 フォーマット、24 ビット/ピクセル	0x23	24 ビット	2'b0、R[5:0]、2'b0、G[5:0]、2'b0、B[5:0]
RGB666 モード 1	CY_U3P_CSI_DF_RGB666_1	RGB 666 フォーマット、24 ビット/ピクセル	0x23	24 ビット	6'b0、R[5:0]、G[5:0]、B[5:0]
RGB565 モード 0	CY_U3P_CSI_DF_RGB565_0	RGB 565 フォーマット、24 ビット/ピクセル	0x22	24 ビット	2'b0、R[4:0]、3'b0、G[5:0]、2'b0、B[4:0]、1'b0
RGB565 モード 1	CY_U3P_CSI_DF_RGB565_1	RGB 565 フォーマット、24 ビット/ピクセル	0x22	24 ビット	3'b0、R[4:0]、2'b0、G[5:0]、3'b0、B[4:0]
RGB565 モード 2	CY_U3P_CSI_DF_RGB565_2	RGB 565 フォーマット、16 ビット/ピクセル	0x22	16 ビット	R[4:0]、G[5:0]、B[4:0]
YUV422 8 ビット モード 0	CY_U3P_CSI_DF_YUV422_8_0	YUV422 フォーマット、16 ビット/ピクセル	0x1E	8 ビット	P[7:0] データ順序: U1、Y1、V1、Y2、U3、Y3、・・・
YUV422 8 ビット モード 1	CY_U3P_CSI_DF_YUV422_8_1	YUV422 フォーマット、16 ビット/ピクセル	0x1E	16 ビット	P[15:0] データ順序: {U1、Y1}、{V1、Y2}、{U3、Y3}、{V3、Y4}・・・

[a] MIPI CSI-2 で定義されたデータ タイプ コード。詳細は、MIPI CSI-2 仕様を参照してください。

フォーマットとモード	CX3 ファームウェアのストリーム フォーマット名	フォーマットの説明、ピクセル深度	CSI-2 データ タイプa	GPIF II バス幅	出力ストリーム
YUV422 8 ビット モード 2	CY_U3P_CSI_DF_YUV422_8_2	YUV422 フォーマット、16 ビット/ピクセル	0x1E	16 ビット	P[15:0] データ順序: {Y1、U1}、 {Y2、V1}、 {Y3、U3}、 {Y4、V3}・・・
YUV422 10 ビット	CY_U3P_CSI_DF_YUV422_10	YUV422 フォーマット、20 ビット/ピクセル	0x1F	16 ビット	6'b0、P[9:0] データ順序: U1、Y1、V1、 Y2、U3、Y3、 V3、Y4

[a] MIPI CSI-2 で定義されたデータ タイプ コード。詳細は、MIPI CSI-2 仕様を参照してください。

選択された GPIF II バス幅が出力ストリームの幅より大きい場合（例えば、24 ビット GPIF II バス幅が CY_U3P_CSI_DF_YUV422_8_1 タイプに使用される場合)、GPIF II の上位ビットには 0 が付加されます。

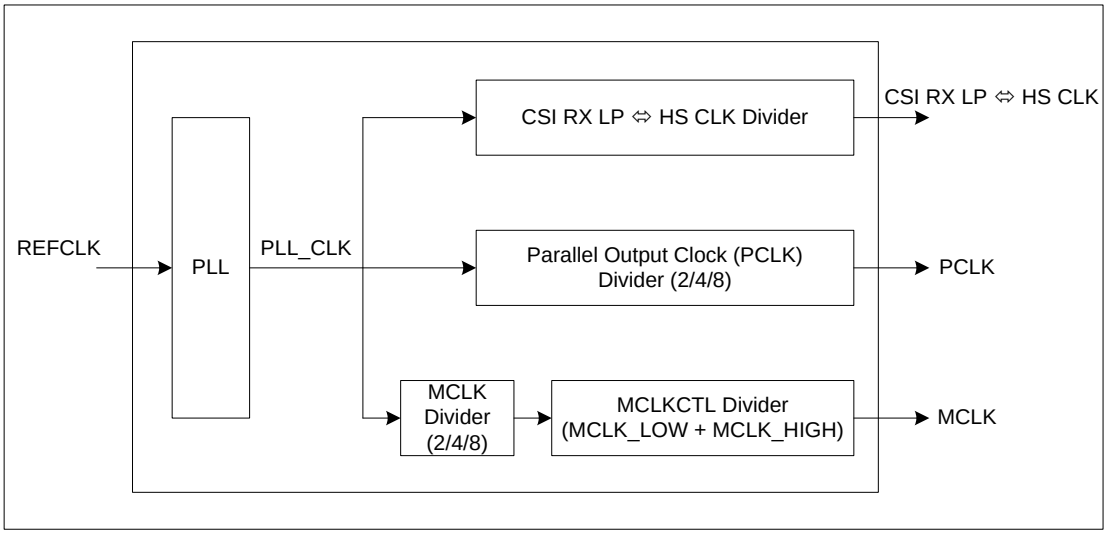
MJPEG やカスタム フォーマット ストリームなど表 3 に示されていないストリーム フォーマットは、8 ビット幅ストリームの場合、RAW8 ストリーム フォーマットで転送され、24 ビット幅ストリームの場合、RGB888 フォーマット ストリームとして取り扱って転送することができます。GPIF バス幅とバッファはストリーム出力に一致するように適切にセットアップする必要があります。1.11.12 節では、GPIF バス幅と DMA バッファのセットアップについて詳しく説明します。

PCLK ごとに 2 個以上の画素を圧縮することができます。例えば、「RAW8」に設定された「MIPI CSI 入力 – データ フォーマット」および 24 ビット フォーマット (RGB888) に設定された「MIPI インターフェース コンフィギュレーション – データ フォーマット」を選択すると、PCLK ごとに 3 個の画素データが出力されます。同様に、24 ビット出力フォーマットを使用して、PCLK ごとに 2 個の 10 ビットまたは 12 ビット画素を圧縮して出力することができます。

1.7 MIPI CSI-2 ブロックのクロック

図 6 は、MIPI CSI-2 ブロックの CX3 クロックを示します。このインターフェースは、リファレンス クロックを入力とし、複数のクロック分周器が続いた PLL を使用して必要なクロックを生成します。CX3 アプリケーション ファームウェアでは、クロック コンフィギュレーション パラメータは *CyU3PMipicsiCfg_t* 構造体の一部であり、CSI-2 ブロックを設定するために *CyU3PMipicsiSetIntfParams()* API に渡されます。この構造体とコンフィギュレーション API の詳細については、EZ-USB FX3 SDK ファームウェア API ガイドを参照してください。

図 6. CX3 MIPI CSI-2 ブロックのクロック



以下の節では、各クロックを簡単に説明します。

1.7.1 リファレンス クロック (REFCLK)

これは MIPI CSI-2 ブロックに供給されるリファレンス クロック入力です。この入力クロックは 6～40MHz です。

1.7.2 PLL クロック (PLL_CLK)

PLL_CLK は、MIPI CSI-2 ブロックの一次クロックです。PLL クロックの値は 62.5MHz～1GHz です。他の全ての内部と出力クロックはこのクロックから派生します。

PLL クロック周波数は、以下の式で入力リファレンス クロックから生成されます。

PLL_CLK = REFCLK * [(PLL_FBD + 1) / (PLL_PRD + 1)] / (2^ PLL_FRS)

ここに

PLL_FBD は、0～0x1FF のフィードバック分周器です。

PLL_PRD は、0～0x0F の入力分周器です。

PLL_FRS は、以下の値を持つ周波数範囲選択パラメータです。

「0」: PLL クロックが 500MHz～1GHz である場合。

「1」: PLL クロックが 250MHz～500MHz である場合。

「2」: PLL クロックが 125MHz～250MHz である場合。

「3」: PLL クロックが 62.5MHz～125MHz である場合。

以下の表は、19.2MHz の REFCLK の場合の PLL クロック周波数の計算例です。

表 4: PLL クロック周波数の計算例

REFCLK (MHz 単位)	PLL_PRD	PLL_FBD	PLL_FRS	PLL_CLK (MHz 単位)
19.2	1	69	0	672
19.2	1	69	2	168
19.2	3	201	1	484.8
19.2	2	97	3	78.4
19.2	2	125	1	403.2

1.7.3 CSI RX LP ←→ HS クロック

このクロックは、CSI リンクのロー パワー (LP) ←→ ハイ スピード (HS) の遷移を検出するのに使用されます。これは PLL_CLK を 2、4、または 8 で分周して生成します。CSI リンクの LP ←→ HS 遷移の詳細については、MIPI アライアンスの MIPI CSI-2 仕様を参照してください (http://www.mipi.org/specifications/camera-interface)。

このクロックの最大値は 125MHz です。

このクロック周波数は「MIPI Receiver configuration tool」により自動的に計算されます。

1.7.4 出力パラレル クロック (PCLK)

このクロックは PCLK 出力であり、CX3 上の機能固定 GPIF II インターフェースを駆動します。これは PLL_CLK を 2、4、または 8 で分周して生成します。

このクロックの最大値は 100MHz です。

このクロック周波数は「MIPI Receiver configuration tool」により自動的に計算されます。

1.7.5 イメージ センサー リファレンス クロック (MCLK)

MCLK は、オプションのクロック出力であり、イメージ センサーの入力リファレンス クロックとして使用できます。これは PLL_CLK を mClkRefDiv (2/4/8) で分周した後、さらに MCLKCTL 分周器で分周して生成します。MCLKCTL 分周器に、PLL_CLK を分周するための上位カウントと下位カウントを指定します。

上位 8 ビットは上位のカウント (1～255) を定義し、下位 8 ビットは下位のカウント (1～255) を定義します。

MCLK は以下の式で計算されます。

MCLK = (PLL_CLK/mClkRefDiv) / [(HighByte (mClkCtl) + 1) + (LowByte (mClkCtl) + 1)]

例えば、

PLL_CLK が 672MHz である場合、24MHz の MCLK を生成するために、mClkRefDiv を 4 に、mClkCtl を 0x0203 にセットします。

従って、

MCLK = (672 / 4) / ((2 + 1) + (3 + 1))

= 168 / 7 = 24MHz

MCLK は HighByte (MClkCtl) と LowByte (MclkCtl) が共に 0 でない時にのみ出力されます。

このクロック周波数は「MIPI Receiver configuration tool」により自動的に計算されます。

1.8 CX3 MIPI CSI-2 ブロックの電力モード

CX3 上の MIPI CSI-2 ブロックは、以下の電力モードに対応しています。

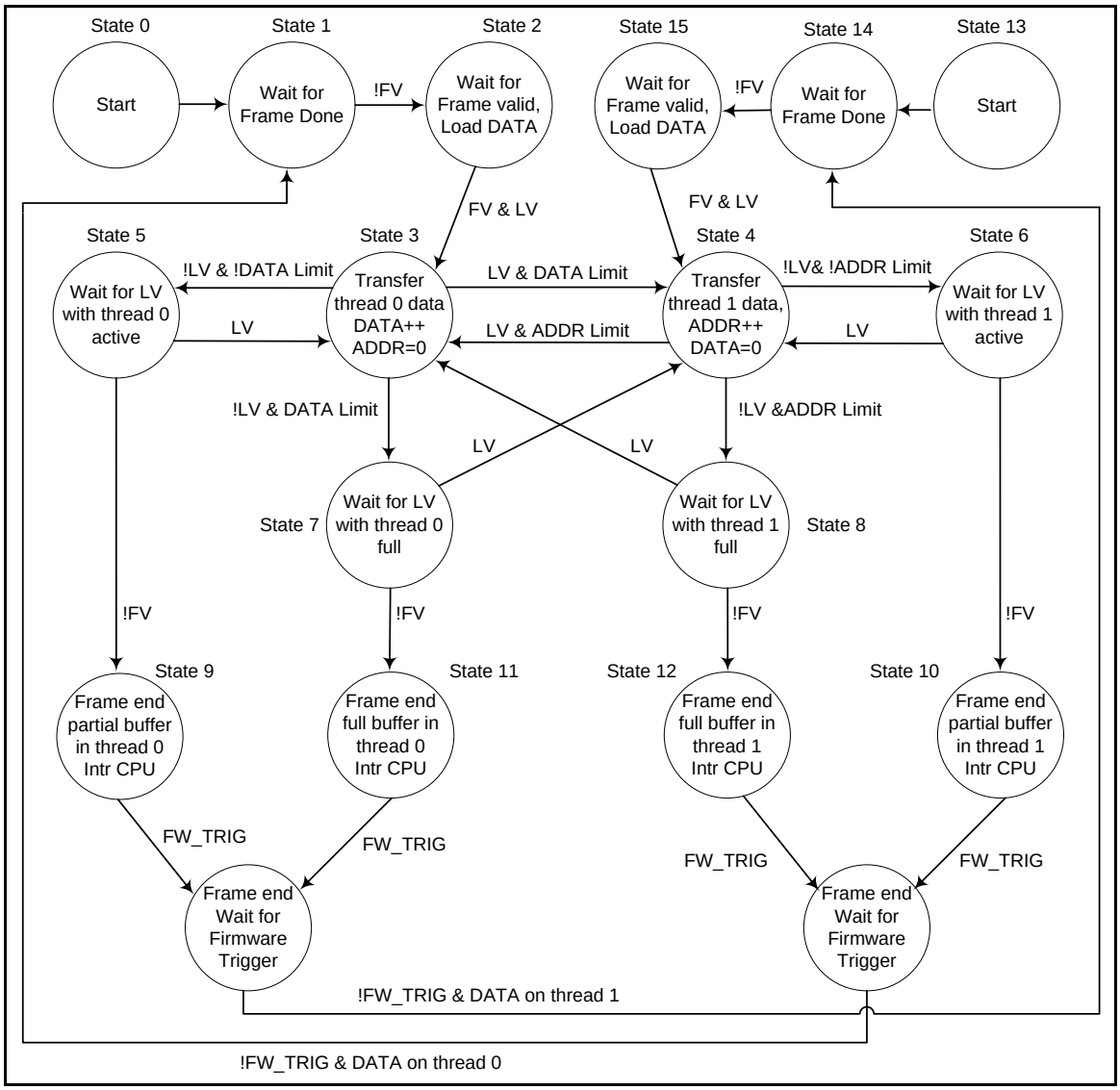
- **アクティブ**: このモードでは、MIPI CSI-2 レシーバ ブロックはアクティブとなり、データをイメージ センサーから GPIF II インターフェースに転送します。
- **低消費電力スリープ**: このモードでは、MIPI CSI-2 ブロック上の全てのデータ転送クロックが停止され、データ転送はありません。全てのレジスタ設定はこの状態で維持されます。
- **ソフト リセット**: この状態は、MIPI CSI-2 ブロックを、インターフェース上の全てのクロックが停止されるリセット モードに移行させます。MIPI CSI-2 ブロック コンフィギュレーション レジスタの状態は、ソフト リセット中に維持されます。
- **ハード リセット**: この状態では、MIPI CSI-2 コントローラおよび全てのクロックは停止されます。MIPI CSI-2 ブロック コンフィギュレーション レジスタは、ハード リセットによりそれらの初期状態にリセットされます。ハード リセットは、MIPI CSI-2 レシーバの初期化中に自動的に実行されます。

1.9 CX3 の機能固定 GPIF II インターフェース

CX3 の機能固定 GPIF II ステート マシンを図 7 に示します。このステート マシンは、MIPI CSI-2 レシーバが提供したパラレル データを 2 個のソケットを介して転送できるようにします。これらのソケットは手動の Many-to-One (多対一) DMA チャネルに接続されることがあります。

このステート マシンの機能は、アプリケーション ノート「*AN75779 - How to Implement an Image Sensor Interface with EZ-USB® FX3™ in a USB Video Class (UVC) Framework*」で説明される GPIF II ステート マシンに似ています。

図 7. CX3 GPIFII ステート マシン



FX3 SDK (ソフトウェア開発キット) は、この機能固定ステート マシンを CX3 GPIF II ブロックにロードするために *CyU3PMipicsiGpifLoad()* API を提供します。この API により、MIPI CSI-2 ブロック上のパラレル インターフェースの出力に一致するためにバス幅 (8、16 または 24 ビット) を選択することができ、また GPIF II インターフェースに使用されるために DMA バッファ サイズを設定することができます。 バッファ サイズの計算は、*1.11.12* 節 (*CyU3PMipicsiGpifLoad() API*)で説明されます。

1.10 MIPI CSI-2 ブロック レジスタ

MIPI CSI-2 ブロックは、I²C を介してブロックを設定するレジスタ セットを提供しています。このブロックは、7 ビット アドレス **7'b0000111** (読み出しアドレス **0x0F**; 書き込みアドレス **0x0E**) で I²C スレーブに接続されます。このブロックは、100kHz と 400kHz I²C 動作に対応しています。

注: MIPI CSI-2 ブロックと同じアドレスを持つ他の I²C デバイスを CX3 デバイスの I²C バスに接続しないでください。

これらのレジスタは 16 ビットに揃えられ、データ転送は最上位ビット(MSB) から行われます。

図 8 と図 9 は、MIPI CSI-2 ブロックへの標準的な I²C 読み出しと書き込み転送を示します。

図 8. 書き込み転送シーケンス

S	SLAVE ADDRESS 7 bit (0000_111)	0	A	REG_ADDR_VAL [15:8]	A	REG_ADDR_VAL [7:0]	A	DATA[15:8]	A	DATA[7:0]	A	P

図 9. 読み出し転送シーケンス

S	SLAVE ADDRESS 7 bit (0000_111)	0	A	REG_ADDR_VAL [15:8]	A	REG_ADDR_VAL [7:0]	A	S	SLAVE ADDRESS 7 bit (0000_111)	1	A	DATA[15:8]	A	DATA[7:0]	A	P

表 5 に CX3 MIPI CSI-2 ブロックのコンフィギュレーション レジスター覧を示します。

表 5: MIPI CSI-2 ブロック コンフィギュレーション レジスタ

レジスタ アドレス	レジスタ名	説明
0x0002	CX3_SYSTEM_CTRL	システム制御レジスタ
0x0004	CX3_CONFIG_CTRL	コンフィギュレーション制御レジスタ
0x0006	CX3_FIFO_CTRL	FIFO 制御レジスタ
0x0008	CX3_DATA_FMT	データ フォーマット制御レジスタ
0x000C	CX3_MCLK_CTRL	MCLK 制御レジスタ
0x0010	CX3_CSI_ SENSOR_SIG_EN	MIPI CSI-2 の信号 (XRESET と XSHUTDOWN) を有効にするレジスタ
0X0014	CX3_CSI_SENSOR_SIG_VAL	MIPI CSI-2 の信号 (XRESET と XSHUTDOWN) を制御するレジスタ
0x0016	CX3_PLL_CTRL0	PLL クロック制御レジスタ 0
0x0018	CX3_PLL_CTRL1	PLL クロック制御レジスタ 1
0x0020	CX3_CLK_CTRL	クロック制御レジスタ
0x0022	CX3_BYTE_COUNT	バイト カウント レジスタ
0x0060	CX3_PHY_TIME_DELAY	MIPI THS-Settle タイマの設定をセットするレジスタ

以降の節では、CX3 MIPI CSI-2 のコンフィギュレーション レジスタの詳細を説明します。

注: 予約済みレジスタ ビットを変更してはなりません。予約済みビットを含むレジスタへの書き込みは、書き込みにより予約済みビットの値を変更していないことを確認する必要があります。望ましくは書き込み前に読み出しを行い、書き込みで予約済みビットの値を維持します。

1.10.1 CX3_SYSTEM_CTRL (レジスタ アドレス: 0x0002)

このレジスタは、MIPI CSI-2 ブロックのためのスリープとソフトウェア リセット機能を制御します。

このレジスタは、*CyU3PMipicsiSleep()*、*CyU3PMipicsiWakeup()*、および *CyU3PMipicsiReset()* API により使用されます。

ビット	15	14	13	12	11	10	9	8
名称	予約済み							
ビット	7	6	5	4	3	2	1	0
名称	予約済み						スリープ	リセット

レジスタ フィールド	ビット	説明
予約済み	[15:2]	予約済み。ファームウェアは、レジスタを読み出し、予約済み以外のビットを変更し、そしてレジスタを再度書き込むことにより、このビットの設定を保持しなければならない
スリープ	[1]	スリープ制御: MIPI CSI-2 ブロックを低消費電力のスリープ モードに移行 0: 通常動作 1: スリープ モード
リセット	[0]	ソフトウェア リセット: MIPI CSI-2 ブロックを強制的にソフトウェア リセットする。コンフィギュレーション レジスタをクリアしないでください。 0: 通常動作 1: リセット動作

1.10.2 CX3_CONFIG_CTRL (レジスタ アドレス: 0x0004)

このレジスタは、データレーンの数および MIPI CSI-2 ブロックの出力データ モードを制御します。

このレジスタは、*CyU3PMipicsiSetIntfParams()* API を使って設定され、*CyU3PMipicsiQueryIntfParams()* API を使って問い合わせられます。

ビット	15	14	13	12	11	10	9	8
名称	予約済み						データ モード	
ビット	7	6	5	4	3	2	1	0
名称	予約済み	OUTEN	予約済み				データ レーン	

レジスタ フィールド	ビット	説明
予約済み	[15:10]	予約済み。ファームウェアは、レジスタを読み出し、予約済み以外のビットを変更し、そしてレジスタを再度書き込むことにより、このビットの設定を保持しなければならない
データ モード	[9:8]	データ モードの選択: CX3_DATA_FMT レジスタのデータ フォーマット ビットを使用して選択されたデータ フォーマットのデータ モードを設定する。表 3 に定義されたようにデータ モードとデータ フォーマットの組み合わせは出力ストリームを選択するために使用される。データ フォーマットの設定は出力データ フォーマットを決め、データ モードの設定は表 3 で定義しているようにデータ タイプのデータ パックとバイト順序付けを決める 2'b00: モード 0 2'b01: モード 1 2'b10: モード 2 2'b11: 予約済み
予約済み	[7]	予約済み。ファームウェアは、レジスタを読み出し、予約済み以外のビットを変更し、そしてレジスタを再度書き込むことにより、このビットの設定を保持しなければならない
OUTEN	[6]	パラレル出力を有効にする: MIPI CSI-2 レシーバ ブロックから機能固定 GPIF II へのパラレル出力を有効にする 0: パラレル出力を無効にする 1: パラレル出力を有効にする
予約済み	[5:2]	予約済み。ファームウェアは、レジスタを読み出し、予約済み以外のビットを変更し、そしてレジスタを再度書き込むことにより、このビットの設定を保持しなければならない
データ レーン	[1:0]	MIPI CSI-2 データ レーンの選択: 使用するデータレーンの数を選択 2'b00: 1 データ レーン 2'b01: 2 データ レーン 2'b10: 3 データ レーン 2'b11: 4 データ レーン

1.10.3 CX3_FIFO_CTRL (レジスタ アドレス：0x0006)

このレジスタは、MIPI CSI-2 ブロックのパラレル出力バッファからデータ出力を開始するための FIFO トリガー レベルを決定します。MIPI CSI-2 ブロックは、GPIF II インターフェースにデータを転送する前に、パラレル出力バッファがこのレジスタにより指定されたレベルに到達するまで待機します。

このレジスタは、*CyU3PMipicsiSetIntfParams()* API を使って設定され、*CyU3PMipicsiQueryIntfParams()* API を使って問い合わせられます。

ビット	15	14	13	12	11	10	9	8
名称	予約済み							FIFO レベル[8]
ビット	7	6	5	4	3	2	1	0
名称	FIFO レベル[7:0]							

レジスタ フィールド	ビット	説明
予約済み	[15:9]	予約済み。ファームウェアは、レジスタを読み出し、予約済み以外のビットを変更し、そしてレジスタを再度書き込むことにより、このビットの設定を保持しなければならない
FIFO レベル	[8:0]	FIFO レベル: FIFO 書き込みトリガー レベルを決める。MIPI CSI-2 ブロックは、出力バッファがこのレベルに達した時にのみ GPIF II へのパラレル出力を開始 範囲：0x000～0x1FF

1.10.4 CX3_DATA_FMT (レジスタ アドレス: 0x0008)

このレジスタは、MIPI CSI-2 ブロックの出力データ モードを制御します。

このレジスタは、*CyU3PMipicsiSetIntfParams()* API を使って設定され、*CyU3PMipicsiQueryIntfParams()* API を使って問い合わせられます。

ビット	15	14	13	12	11	10	9	8
名称	予約済み							
ビット	7	6	5	4	3	2	1	0
名称	データ フォーマット				予約済み			データ フォーマット イネーブル

レジスタ フィールド	ビット	説明
予約済み	[15:8]	予約済み。ファームウェアは、レジスタを読み出し、予約済み以外のビットを変更し、そしてレジスタを再度書き込むことにより、このビットの設定を保持しなければならない
データ フォーマット	[7:4]	データ形式の選択: 出力データ形式を選択 4'b0000: RAW8 4'b0001: RAW10 4'b0010: RAW12 4'b0011: RGB888 4'b0100: RGB666 4'b0101: RGB565 4'b0110: YUV422-8 ビット 4'b0111: RESERVED 4'b1000: RAW14 4'b1001: YUV422-10 ビット 4'b1010-1111: RESERVED このフィールドと CX3_CONFIG_CTRL レジスタ内のデータ モード フィールドは、表 3 に定義された出力ストリーム フォーマットを選択するために使用 MJPEG やカスタム フォーマット ストリームなど表 3 に示されていないストリーム フォーマットの場合、8 ビット幅ストリームは RAW8 ストリーム フォーマットで転送され、24 ビット幅ストリームは RGB888 フォーマット ストリームとして取り扱って転送することが可能
予約済み	[3:1]	予約済み。ファームウェアは、レジスタを読み出し、予約済み以外のビットを変更し、そしてレジスタを再度書き込むことにより、このビットの設定を保持しなければならない
データ フォーマット イネーブル	[0]	データ フォーマットを有効にする: 0: サポートされない 1: 通常動作

1.10.5 CX3_MCLK_CTRL (レジスタ アドレス: 0x000C)

このレジスタは、MCLK 分周器を設定し、MIPI CSI-2 ブロックからのイメージ センサー リファレンス クロック (MCLK) の出力を制御します。MCLK 値を算出の詳細については、1.7.5 節で説明しました。

このレジスタは、*CyU3PMipicsiSetIntfParams()* API を使って設定され、*CyU3PMipicsiQueryIntfParams()* API を使って問い合わせられます。

ビット	15	14	13	12	11	10	9	8
名称	MCLK HIGH							
ビット	7	6	5	4	3	2	1	0
名称	MCLK LOW							

レジスタ フィールド	ビット	説明
MCLK HIGH	[15:8]	MCLK 分周カウント上位: MCLK 分周器の分周カウントの上位バイトこのフィールドの詳細は 1.7.5 節を参照してください。 MCLK は MCLK HIGH と MCLK LOW が共にゼロでない場合にのみ有効になる
MCLK LOW	[7:0]	MCLK 分周カウント下位: MCLK 分周器の分周カウントの下位バイトこのフィールドの詳細は 1.7.5 節を参照してください。 MCLK 分周器 = (MCLK HIGH +1) + (MCLK LOW +1)

1.10.6 CX3_CSI_SENSOR_SIG_EN (レジスタ アドレス: 0x0010)

このレジスタは MIPI CSI-2 ブロックからの XSHUTDOWN と XRESET 信号の出力を有効または無効にします。

このレジスタは *CyU3PMipicsiInit()*と *CyU3PMipicsiReset()* API を使って設定されます。

ビット	15	14	13	12	11	10	9	8
名称	予約済み							
ビット	7	6	5	4	3	2	1	0
名称	予約済み					XSHUTDOWN イネーブル	XRESET イネーブル	予約済み

レジスタ フィールド	ビット	説明
予約済み	[15:3]	予約済み。ファームウェアは、レジスタを読み出し、予約済み以外のビットを変更し、そしてレジスタを再度書き込むことにより、このビットの設定を保持しなければならない
XSHUTDOWN イネーブル	[2]	MIPI CSI-2 XSHUTDOWN 信号を有効にする: 0: MIPI CSI-2 XSHUTDOWN 信号の出力を有効にする 1: MIPI CSI-2 XSHUTDOWN 信号の出力を無効にする 信号の論理は CX3_CSI_SENSOR_SIG_VAL (レジスタ アドレス: 0x0014) [2]
XRESET イネーブル	[1]	MIPI CSI-2 XRESET 信号を有効にする: 0: MIPI CSI-2 XRESET 信号の出力を有効にする 1: MIPI CSI-2 XRESET 信号の出力を無効にする 信号の論理は CX3_CSI_SENSOR_SIG_VAL (レジスタ アドレス: 0x0014) [1]
予約済み	[0]	予約済み。ファームウェアは、レジスタを読み出し、予約済み以外のビットを変更し、そしてレジスタを再度書き込むことにより、このビットの設定を保持しなければならない

1.10.7 CX3_CSI_SENSOR_SIG_VAL (レジスタ アドレス: 0x0014)

このレジスタは MIPI CSI-2 ブロックの XSHUTDOWN と XRESET 信号の論理を設定します。

このレジスタは *CyU3PMipicsiSetSensorControl()* API を使って設定されます。

ビット	15	14	13	12	11	10	9	8
名称	予約済み							
ビット	7	6	5	4	3	2	1	0
名称	予約済み					XSHUTDOWN 出力	XRESET 出力	予約済み

レジスタ フィールド	ビット	説明
予約済み	[15:3]	予約済み。ファームウェアは、レジスタを読み出し、予約済み以外のビットを変更し、そしてレジスタを再度書き込むことにより、このビットの設定を保持しなければならない
XSHUTDOWN 出力	[2]	MIPI CSI-2 XSHUTDOWN 信号の駆動: 0: XSHUTDOWN をローレベルに駆動する 1: XSHUTDOWN をハイレベルに駆動する この信号は CX3_CSI_SENSOR_SIG_EN (レジスタ アドレス: 0x0010) [2]が 0 である場合にのみ駆動される
XRESET 出力	[1]	MIPI CSI-2 XRESET 信号の駆動: 0: XRESET をローレベルに駆動する 1: XRESET をハイレベルに駆動する この信号は CX3_CSI_SENSOR_SIG_EN (レジスタ アドレス: 0x0010) [1]が 0 である場合にのみ駆動される
予約済み	[0]	予約済み。ファームウェアは、レジスタを読み出し、予約済み以外のビットを変更し、そしてレジスタを再度書き込むことにより、このビットの設定を保持しなければならない

1.10.8 CX3_PLL_CTRL0 (レジスタ アドレス: 0x0016)

このレジスタは MIPI CSI-2 ブロックの PLL クロックを設定します。CX3_PLL_CTRL0 (レジスタ アドレス: 0x0016) と CX3_PLL_CTRL1 (レジスタ アドレス: 0x0018)の値に基づいた PLL クロックの生成方法の詳細は 1.7.2 節で説明しました。

このレジスタは *CyU3PMipicsiSetIntfParams()* API を使って設定され、*CyU3PMipicsiQueryIntfParams()* API を使って問い合わせられます。

ビット	15	14	13	12	11	10	9	8
名称	PLL PRD				予約済み			PLL FBD [8]
ビット	7	6	5	4	3	2	1	0
名称	PLL FBD [7:0]							

レジスタ フィールド	ビット	説明
PLL PRD	[15:12]	入力分周器: PLL 生成用入力分周器このフィールドの詳細は 1.7.2 節を参照してください。 範囲: 0x0～0xF
予約済み	[11:9]	予約済み。ファームウェアは、レジスタを読み出し、予約済み以外のビットを変更し、そしてレジスタを再度書き込むことにより、このビットの設定を保持しなければいけません。
PLL FBD	[8:0]	フィードバック分周器: PLL 生成用フィードバック分周器。このフィールドの詳細は 1.7.2 節を参照してください。 範囲: 0x000～0x1FF

1.10.9 CX3_PLL_CTRL1 (レジスタ アドレス: 0x0018)

このレジスタは MIPI CSI-2 ブロックの PLL クロックを設定します。CX3_PLL_CTRL0 (レジスタ アドレス: 0x0016) と CX3_PLL_CTRL1 (レジスタ アドレス: 0x0018)の値に基づいた PLL クロックの生成方法の詳細は、1.7.2 節で説明しました。

このレジスタは *CyU3PMipicsiSetIntfParams()* API を使って設定され、*CyU3PMipicsiQueryIntfParams()* API を使って問い合わせられます。

ビット	15	14	13	12	11	10	9	8
名称	予約済み				PLL FRS		予約済み	
ビット	7	6	5	4	3	2	1	0
名称	予約済み			クロック イネーブル	予約済み		PLL イネーブル	

レジスタ フィールド	ビット	説明
予約済み	[15:12]	予約済み。ファームウェアは、レジスタを読み出し、予約済み以外のビットを変更し、そしてレジスタを再度書き込むことにより、このビットの設定を保持しなければいけません。
PLL FRS	[11:10]	周波数範囲の選択: PLL 周波数範囲を決定します。このフィールドの詳細は 1.7.2 節を参照してください。 2'b00: 500～1000MHz PLL 周波数 2'b01: 250～500MHz PLL 周波数 2'b10: 125～250MHz PLL 周波数 2'b11: 62.5～250MHz PLL 周波数
予約済み	[9:5]	予約済み。ファームウェアは、レジスタを読み出し、予約済み以外のビットを変更し、そしてレジスタを再度書き込むことにより、このビットの設定を保持しなければいけません。
クロック イネーブル	[4]	クロックを有効にします: 0: MIPI CSI-2 ブロックのクロックをオフにします 1: MIPI CSI-2 ブロックのクロックを有効にします (通常動作)
予約済み	[3:2]	予約済み。ファームウェアは、レジスタを読み出し、予約済み以外のビットを変更し、そしてレジスタを再度書き込むことにより、このビットの設定を保持しなければいけません。
PLL イネーブル	[1:0]	PLL クロックを有効にします: PLL クロックを有効にする設定は以下の通りです。 2'b00: PLL クロックが無効 2'b11: PLL クロックが有効 (通常動作) 2b'01～2b'10: 予約済み

1.10.10 CX3_CLK_CTRL (レジスタ アドレス: 0x0020)

このレジスタは MIPI CSI-2 ブロックのインターフェース クロック分周器を設定します。

このレジスタは *CyU3PMipicsiSetIntfParams()* API を使って設定され、 *CyU3PMipicsiQueryIntfParams()* API を使って問い合わせられます。

ビット	15	14	13	12	11	10	9	8
名称	予約済み							
ビット	7	6	5	4	3	2	1	0
名称	予約済み		CSI RX CLK DIV		MCLK REF DIV		PAR OUT CLK DIV	

レジスタ フィールド	ビット	説明
予約済み	[15:6]	予約済み。ファームウェアは、レジスタを読み出し、予約済み以外のビットを変更し、そしてレジスタを再度書き込むことにより、このビットの設定を保持しなければいけません。
CSI RX CLK DIV	[5:4]	CSI RX LP←→ HS 遷移クロック用のクロック分周器: PLL クロックから分周してこのクロックを生成する。詳細は 1.7.3 節を参照してください。 2'b00: PLL クロック/8 2'b01: PLL クロック/4 2'b10: PLL クロック/2 2'b00: 予約済み この周波数は 66～125MHz の範囲でなければいけません。
MCLK REF DIV	[3:2]	MCLK リファレンス クロック用のクロック分周器: PLL クロックから分周して MCLK リファレンス クロックを生成する。詳細は 1.7.5 節を参照してください。 2'b00: PLL クロック/8 2'b01: PLL クロック/4 2'b10: PLL クロック/2 2'b00: 予約済み この周波数は 125MHz 未満の周波数でなければいけません。
PAR OUT CLK DIV	[1:0]	パラレル出力クロック (PCLK) 用のクロック分周器: PLL クロックから分周して、GPIF II インターフェースを駆動するパラレル出力クロックを生成する。詳細は 1.7.4 節を参照してください。 2'b00: PLL クロック/8 2'b01: PLL クロック/4 2'b10: PLL クロック/2 2'b00: 予約済み この周波数は 100MHz を超えてはいけません。

1.10.11 CX3_BYTE_COUNT (レジスタ アドレス: 0x0022)

このレジスタは MIPI CSI-2 ブロックの 1 ライン当たりのバイト数を設定します。

このレジスタは *CyU3PMipicsiSetIntfParams()* API を使って設定され、*CyU3PMipicsiQueryIntfParams()* API を使って問い合わせられます。

ビット	15	14	13	12	11	10	9	8
名称	バイト数[15:8]							
タイプ	R/W							
ビット	7	6	5	4	3	2	1	0
名称	バイト数[7:0]							
タイプ	R/W							

レジスタ フィールド	ビット	説明
バイト数	[15:0]	1 ライン当たりの総バイト数: 入力 1 ライン当たりのバイト数は次のように計算します。 水平画素数 (アクティブ ピクセル) x ピクセル当たりのバイト数 例えば 1920x1080 RGB888 ストリームの場合、以下のように計算されます。 1 ライン当たりのアクティブ ピクセル数: 1920 ピクセル当たりのバイト数: 3 バイト数 = 1920x3 = 5760 = 0x1680 バイト数[15:8] = 0x16 バイト数[7:0] = 0x80

1.10.12 CX3_PHY_TIME_DELAY (レジスタ アドレス: 0x0060)

このレジスタは CX3 の MIPI CSI-2 ブロックの MIPI CSI-2 レシーバ PHY の遅延パラメータを設定します。この設定は 1.7.3 節で定義された CSI_RX_CLK によります。

このレジスタは *CyU3PMipicsiSetIntfParams()* API を使って設定され、 *CyU3PMipicsiQueryIntfParams()* API を使って問い合わせられます。

ビット	15	14	13	12	11	10	9	8
名称	TC TERM	予約済み						
ビット	7	6	5	4	3	2	1	0
名称	TD TERM	THS SETTLE						

レジスタ フィールド	ビット	説明
TC TERM	[15]	TC TERM の選択: 通常動作時は 1 に設定され、0 は未使用
予約済み	[14:8]	予約済み。ファームウェアは、レジスタを読み出し、予約済み以外のビットを変更し、そしてレジスタを再度書き込むことにより、このビットの設定を保持しなければいけません。
TD TERM	[7]	TD TERM の選択: 0: データ レーン HS 終端は、LP から HS への遷移時に 2xCSI_Rx_Clk または 3xCSI_Rx_Clk サイクルの後に発生 1: データ レーン HS 終端は、LP から HS への遷移時に即時にセット(推奨)
THS SETTLE	[6:0]	THS SETTLE タイマ: LP から HS への遷移の遅延を制御するタイマ 範囲: 0x00～0x7F 遅延 = (THS SETTLE + 1) x CSI_RX_CLK

1.11 CX3 MIPI CSI-2 API

本節では EZ USB FX3 SDK に含まれる、CX3 に搭載された MIPI CSI-2 ブロックを構成し使用するための API について詳しく説明します。各 API および API が使用するエニュメレーションと構造体の型の詳細は、EZ USB FX3 SDK の一部である FX3/FX3S/CX3 SDK ファームウェア API ガイドを参照してください。

MIPI CSI-2 ブロック コンフィギュレーション API は cyu3mipicsi.a ライブラリファイルを介して使用可能であり、関数宣言、エニュメレーションおよび構造体の型は、ヘッダファイル *cyu3mipicsi.h* から入手できます。

EZ USB FX3 SDK 1.3 は、CX3 デバイスをベータ レベルでサポートします。CX3 デバイスへの完全対応は EZ-USB FX3 SDK 1.3.1 リリース以降、EZ-USB FX3 SDK の一部として提供されます。

1.11.1 CyU3PMipicsiInit()

CyU3PReturnStatus_t CyU3PMipicsiInit (void)

この関数は CX3 デバイスの MIPI CSI-2 ブロックを初期化します。MIPI CSI-2 ブロックに対して他の関数を呼び出す前に実行します。CX3 GPIO ブロック、PIB ブロックおよび I^2C ブロックは、この関数を呼び出す前に初期化される必要があります。

この呼び出しにより MIPI CSI-2 ブロックが低消費電力モードに移行され、MIPI CSI-2 ブロックのクロックを有効にするために *CyU3PMipicsiWakeup()*を呼び出す必要があります。

この関数は内部で MIPI CSI-2 ブロックの電源を切断し再投入します。

この関数は CX3_CSI_SENSOR_SIG_EN レジスタの XSHUTDOWN イネーブルと XRESET イネーブルビットを 0 にクリアして XRESET の出力と XSHUTDOWN 信号を有効にします。

この関数は CX3_CSI_SENSOR_SIG_VAL レジスタを初期状態 (0x0000) に移行させるため、MIPI XRESET と XSHUTDOWN 信号がローレベル状態に設定されます。これら信号のいずれかをセンサー動作のためにハイレベル状態に移行する必要がある場合、CyU3PMipicsiSetSensorControl() 関数を呼び出した後 *CyU3PMipicsiSetSensorControl()* API を使用してその信号をハイレベル状態に確実に設定する必要があります。

正常に実行された場合、この関数は呼び出し元の関数に CY_U3P_SUCCESS 値を返します。

1.11.2 CyU3PMipicsiDeInit()

CyU3PReturnStatus_t CyU3PMipicsiDeInit (void)

この関数は CX3 デバイスの MIPI-CSI インターフェース ブロックの初期化を解除し、*CyU3PsysEnterStandbyMode()*を呼び出す前に実行します。I^2C ブロックや GPIO ブロックの初期化が解除される前にこの関数を呼び出す必要があります。

MIPI XRESET と XSHUTDOWN 信号は関数を呼び出した後に CX3 により駆動されてはなりません。

正常に実行された場合、この関数は呼び出し元の関数に CY_U3P_SUCCESS 値を返します。

1.11.3 CyU3PMipicsiReset()

CyU3PReturnStatus_t CyU3PMipicsiReset (CyU3PMipicsiReset_t resetType)

この関数は CX3 の MIPI-CSI ブロックをリセットするために使用されます。MIPI-CSI ブロックは 2 つのリセット モードを提供します:ブロック全体の電源を切断し再投入する**ハードリセット**とブロックが使用する I^2C 通信チャネルをリセットしないソフトリセットです。リセット モードはこの関数に渡された *CyU3PMipicsiReset_t* タイプのパラメータを介して選択されます。

ブロックが初期化されるまで、ソフトリセットはインターフェース上に呼び出すことはできません。しかしハードリセットはいつでもインターフェース上で呼び出すことができます。

*CyU3PMipicsiInit()*の呼び出しは、MIPI-CSI ブロックを初期化する前にインターフェース上で内部でハードリセットを実行します。

ハードリセットは CX3_CSI_SENSOR_SIG_EN レジスタの XSHUTDOWN ENABLE と XRESET ENABLE ビットを 0 にクリアすることにより XRESET および XSHUTDOWN 信号の出力を有効にします。この関数は CX3_CSI_SENSOR_SIG_VAL レジスタをそのデフォルト状態 (0x0000) に移行させるため、MIPI XRESET と XSHUTDOWN 信号が状態「0」に設定されます。これら信号のいずれかをセンサー動作のために状態「1」に移行する必要がある場合、*CyU3PMipicsiReset(CY_U3P_CSI_HARD_RST)* 関数を呼び出した後 *CyU3PMipicsiSetSensorControl()*を使ってその信号を確実に状態「1」に設定する必要があります。

ソフトリセットは XRESET と XSHUTDOWN 信号の状態を変更しません。ソフトリセットは CX3_SYSTEM_CTRL レジスタのリセット ビットを 1 にセットし、CX3 において 5 回のタイマカウントを待機して、通常動作を再開するために**リセット**ビットを 0 にクリアします。

正常に実行された場合、この関数は呼び出し元の関数に CY_U3P_SUCCESS 値を返します。

1.11.4 CyU3PCx3DeviceReset()

void CyU3PCx3DeviceReset (CyBool_t isWarmReset,

CyBool_t sensorResetHigh)

この関数は FX3/FX3S デバイスに使用される *CyU3PDeviceReset ()* API と類似しています。この関数は *isWarmReset* パラメータを使用してデバイスがウォームリセットとコールドリセットの内どちらを行うか決定します。

CX3 デバイスをリセットするのみならず、この関数は CyU3PMipicsiSetSensorControl()を内部で呼び出して *CY_U3P_CSI_IO_XRES* 信号を *sensorResetHigh* に提供された値に応じてハイレベルかローレベルに設定することにより、イメージ センサーを CX3 デバイスがリセットされる前にリセットするために、MIPI XRESET 信号も駆動します。

この関数は戻りません。

1.11.5 CyU3PMipicsiSleep()

CyU3PReturnStatus_t CyU3PMipicsiSleep (void)

この機能は MIPI CSI-2 ブロックの PLL クロックを無効にし、低消費電力のスリープ モードに移行させるために使用されます。ブロックが低消費電力スリープ モードにある間、イメージ センサーから CX3 へのデータ転送は行われません。

この API は MIPI CSI-2 ブロックの CX3_SYSTEM_CTRL レジスタの SLEEP ビットを 1 にセットします。

ブロックをスリープ状態に正常に移行させた場合、この関数は CY_U3P_SUCCESS を返します。

1.11.6 CyU3PMipicsiWakeup()

CyU3PReturnStatus_t CyU3PMipicsiWakeup (void)

この関数は MIPI CSI-2 ブロックのクロックを有効にして、それを低消費電力のスリープ モードからアクティブ モードに復帰させるために使用されます。

この API は MIPI CSI-2 ブロックの CX3_SYSTEM_CTRL レジスタの SLEEP ビットを 0 にクリアします。

正常に実行された場合、この関数は CY_U3P_SUCCESS 値を返します。

1.11.7 CyU3PMipicsiSetSensorControl()

CyU3PReturnStatus_t CyU3PMipicsiSetSensorControl (CyU3PMipicsiSensorIo_t io,

CyBool_t value)

この関数は XRES および XSHUTDOWN 信号を CX3 からイメージ センサーに対して駆動するために使用されます。この関数により信号の論理をハイアクティブかローアクティブに設定します。

この関数は、1 つの信号を駆動するか同時に両方の信号を(同じ値に)駆動することが可能です。両方の信号を同じ値に設定するために、*io* の値としてマスク *(CY_U3P_CSI_IO_XRES | CY_U3P_CSI_IO_XSHUTDOWN)*を使用してください。2 つの信号を別の値に設定するにはこの関数を複数呼び出し、それぞれの信号に 1 回ずつ実行します。

この API は、CX3_CSI_SENSOR_SIG_VAL の適切なビットを 1 または 0 に設定し、センサーに出力される対応する MIPI 信号をハイレベルまたはローレベルに駆動します。

XRESET 信号をハイレベルまたはローレベルに駆動するために、**XRESET OUTPUT** ビットを 1 にセットあるいは 0 にクリアします。同様に、XSHUTDOWN 信号をハイレベルまたはローレベルに駆動するために、**XSHUTDOWN OUTPUT** ビットを 1 にセットあるいは 0 にクリアします。

正常に実行された場合、この関数は CY_U3P_SUCCESS 値を返します。

1.11.8 CyU3PMipicsiCheckBlockActive()

CyBool_t CyU3PMipicsiCheckBlockActive (void)

この関数は、MIPI CSI-2 ブロックがアクティブ モードであるか、低消費電力スリープ モードであるかを確認するために使用されます。

ブロックがアクティブの場合は CyTrue を、アクティブではない場合は CyFalse を返します。

この関数は *CyU3PMipicsiWakeup()* が呼び出された時に設定されたフラグのみを確認し、実際のインターフェース レジスタを確認しません。コンフィギュレーション レジスタを直接変更する場合はこの API を使用せず、CX3_SYSTEM_CTRL レジスタの SLEEP ビットの実際の値を確認してください。

1.11.9 CyU3PMipicsiSetIntfParams()

CyU3PReturnStatus_t CyU3PMipicsiSetIntfParams (CyU3PMipicsiCfg_t * csiCfg,

CyBool_t wakeOnConfigure)

この関数は I^2C インターフェースを通して MIPI CSI-2 のブロック パラメータを設定するために使用されます。

この関数は *CyU3PMipicsiCfg_t* タイプのオブジェクトを取り込んで、MIPI CSI-2 ブロックを設定します。この関数はインターフェース コンフィギュレーション レジスタを変更する前に *CyU3PMipicsiSleep()*を呼び出すことによってインターフェース クロックをオフにします。

この関数は表 6 に示すように、以下の MIPI CSI-2 ブロック レジスタを渡された *CyU3PMipicsiCfg_t* 構造体の値で設定します。

表 6:MIPI CSI-2 ブロック レジスタを設定する際に使用するパラメータ

CX3 MIPI CSI-2 ブロック レジスタ	設定に使用される CyU3PMipicsiCfg_t パラメータ
CX3_PLL_CTRL0	*pllPrd* と *pllFbd* パラメータ
CX3_PLL_CTRL1	*pllFrs* パラメータ
CX3_CLK_CTRL	*csiRxClkDiv*、*parClkDiv* および *mClkRefDiv* パラメータ
CX3_MCLK_CTRL	*mClkCtl* パラメータ
CX3_BYTE_COUNT	*hResolution* と *dataFormat* パラメータから計算
CX3_PHY_TIME_DELAY	*thsSettle* パラメータ FX3 SDK 1.3.3 から利用可能
CX3_DATA_FMT	*dataFormat* パラメータ
CX3_CONFIG_CTRL	*dataFormat* と *numDataLanes* パラメータ

パラメータ *wakeOnConfigure* は、設定が完了した直後にクロックをオンにする、またはクロックをオフのままにするために使用されます。クロックがオフのままにされている場合、クロックを起動するために *CyU3PMipicsiWakeup()* を呼び出す必要があります。

正常に実行された場合、この関数は CY_U3P_SUCCESS 値を返します。

CX3 コンフィギュレーション生成ツールは、EZ-USB SDK で提供される EZ-USB Suite（Eclipse ベース IDE）の一部として提供されます。このツールは、イメージ センサー入力とストリーム パラメータに基づいて MIPI CSI-2 ブロックを設定するための *CyU3PMipicsiCfg_t* 構造体の要素を生成するために使用されます。

コンフィギュレーション ツールの詳しい使用方法は、EZ-USB Suite ヘルプ メニューの一部、および EZ-USB FX3 SDK インストール パスのドキュメント フォルダ内のサイプレス EZ-USB FX3 クイック スタート ガイド（Getting Started with FX3 SDK.pdf）の一部として記載されています。

1.11.10 CyU3PMipicsiQueryIntfParams()

CyU3PReturnStatus_t CyU3PMipicsiQueryIntfParams (CyU3PMipicsiCfg_t * csiCfg)

この関数は、ブロックから MIPI-CSI インターフェース パラメータを読み出すために使用されます。読み出したパラメータは、呼び出し関数から渡された *CyU3PMipicsiCfg_t* タイプのポインタを介して呼び出し関数に提供されます。この関数は *CyU3PMipicsiSetIntfParams()*によって書き込まれたレジスタを読み出します。 csiCfg によって指示された *CyU3PMipicsiCfg_t* 構造体オブジェクトはこの API に渡される前に初期化される必要があります。

1.11.11 CyU3PMipicsiGetErrors()

CyU3PReturnStatus_t CyU3PMipicsiGetErrors (CyBool_t clrErrCnts,
CyU3PMipicsiErrorCounts_t * errorCounts)

この関数は MIPI CSI-2 ブロックから CSI-2 プロトコルおよび物理層エラーを取得するために使用されます。

この関数はインターフェース上のエラー カウントがクリアされたか否かをパラメータとして取ります。

各エラー タイプのエラー カウントがこの関数に渡された *CyU3PMipicsiErrorCounts_t* タイプのポインタを介して検索されます。各タイプのエラー カウントは 0xFF の最大カウントに達することができます。カウントは、関数が clrErrCnts を使用してカウントを明示的にクリアしない限り、呼び出すたびに既存のエラー値を報告し続けます。

errorCounts オブジェクトはこの関数に渡される前に初期化される必要があります。

1.11.12 CyU3PMipicsiGpifLoad()

CyU3PReturnStatus_t CyU3PMipicsiGpifLoad (CyU3PMipicsiBusWidth_t busWidth,
uint32 t bufferSize)

1.9 節 REF _Ref377754096 \r \h * MERGEFORMAT に示すように CX3 には、MIPI CSI-2 ブロックからイメージ センサーのデータを取得するために設計された機能固定 GPIF インターフェースがあります。

この関数により、GPIF 転送のために提供される GPIF データのバス幅を選択し、DMA バッファのサイズを設定することができます。PIB ブロックはこの関数を呼び出す前に初期化される必要があります。

DMA バッファのサイズはバス幅の倍数とします。例えば、GPIF バス幅が 24 ビット幅に設定された場合、DMA バッファは 24 ビット（3 バイト）の倍数になります。

さらに DMA バッファ サイズは CyU3PDmaChannelCreate API の要件を満たすために 16 バイトの倍数である必要があります。

この API に渡されるバス幅と DMA バッファ サイズは、データ転送のために選択されたデータ フォーマットの幅に基づく必要があります。

例えば、RGB 888 データ型はバス幅を 24 ビットに設定する必要がありますが、YUV 422 はバス幅を 16 ビットに、RAW 8 フォーマットはバス幅を 8 ビットに設定する必要があります。

この API で設定したバス幅が、選択されたデータ フォーマットの幅より小さな場合は、MIPI CSI-2 ブロックから転送されているデータ ビットの上位が失われます。この API で設定したバス幅が、選択されたデータ フォーマットの幅より大きな場合は、受信したデータの未使用の上位ビットが 0 になります。

1.12 その他の参考資料

データシートおよびアプリケーション ノートを含む CX3 のその他の情報については、http://www.cypress.com/cx3/ を参照してください。

データシート、アプリケーション ノートおよび EZ-USB FX3 テクニカル リファレンス マニュアル（TRM）の追加情報については、http://www.cypress.com/fx3/をご覧ください。

SDK の一部として提供される EZ-USB FX3 SDK のソフトウェア ダウンロードおよびドキュメンテーション（プログラマのマニュアル、ファームウェア API ガイドおよびクイック スタート ガイド等）については、http://www.cypress.com/?rID=57990 をご覧ください。

MIPI CSI-2 仕様の詳細については MIPI アライアンス ウェブサイト http://www.mipi.org/specifications/camera-interface を参照してください。

1.13 用語集

API	アプリケーション プログラミング インターフェースアプリケーションおよび低レベルのサービスと機能間のインターフェース（例えば、FX3/CX3 デバイスの動作を設定・制御するインターフェース）が含まれる一連のソフトウェア ルーチン。 EZ-USB FX3 SDK には、FX3 デバイスの動作を設定・制御するための API を提供する一連のライブラリがあります。
ベイヤー フィルタ	フォトセンサーの正方形グリッド上に RGB のカラー フィルタを配置するためのカラーフィルタ アレイ。フィルタのパターンは緑が 50%、赤が 25%、および青が 25%です。
CPU サブシステム	EZ-USB FX3/CX3 デバイスは 200MIPS の処理能力を提供する 32 ビットの組み込み用 ARM926EJ-S コアを備えている。この ARM コアは、命令キャッシュとデータ キャッシュ、密接合メモリ（TCM）および PL192 ベクタ割り込みコントローラ（VIC）で結合されている。詳細については、*EZ-USB FX3 TRM*FX3 CPU サブシステムの章を参照してください。
データ レーン	データ転送に使用される単方向 MIPI CSI-2 差動シリアル インターフェース。各レーンはデータ+とデータ-の 2 つの信号から構成されています。CX3 は最大 4 個のデータ レーンをサポートしており、それぞれが最大 1 ギガビット/秒の速度に対応できます。
DMA	ダイレクト メモリ アクセス（DMA）のサブシステムは、CPU の介入なしでメモリとペリフェラル間で高帯域幅のデータ転送を実行します。FX3/CX3 アーキテクチャには、様々なペリフェラル インターフェースとシステム メモリ間のデータを操作するために使用される DMA ファブリックが含まれます。DMA サブシステムの詳細については、*EZ-USB FX3 TRM* の FX3 DMA サブシステムの章を参照してください。
グローバル コントローラ	FX3/CX3 のグローバル コントローラ（GCTL）ユニットは、クロック生成ブロック、多機能 I/O、出力制御ブロックおよびリセット ブロックを含みます。クロック生成ブロックはすべてのデバイス周辺ペリフェラルにクロックを提供します。詳細については、*EZ-USB FX3 TRM* の FX3 グローバル コントローラ（GCTL）の章を参照してください。
GPIF II	CX3 の GPIF II インターフェースは FX3 にある汎用プログラマブル インターフェースの一部の機能を固定して実装したものです。CX3 GPIF II インターフェースは MIPI CSI-2 レシーバ ブロックからデータを転送するために使用されます。CX3 の機能固定 GPIF II ステート マシンの詳細は本書の 1.9 節に記載されています。詳細については *EZ-USB FX3 TRM* の汎用プログラマブル インターフェース（GPIF II）の章を参照してください。
I^2C	Philips Semiconductor 社によって作成された I^2C-バス プロトコルは、「インター インテグレーテッド サーキット バス」の略であり、2 線を使用して I^2C 対応デバイス間のデータ通信を可能にします。データ（SDA）線およびクロック（SCL）信号を使用して情報をシリアルに送信します。FX3/CX3 は、100kHz、400 kHz および 1 MHz の処理をサポートする I^2C マスタコントローラとして機能します。詳細については、*EZ-USB FX3 TRM* の「低帯域幅ペリフェラル」章の「I^2C インターフェース」の節を参照してください。
JTAG	ジョイント テスト アクション グループ（JTAG）は IEEE 1149.1 標準テスト アクセス ポートおよびバウンダリ スキャン アーキテクチャの一般名。FX3/CX3 の JTAG インターフェースは、JTAG デバッガへ接続するために標準の 5 ピン インターフェースを備えており、オンチップ デバッグ回路を使用してファームウェアをデバッグすることができます。ARM926EJ-S コア用の業界標準デバッグ ツールは、FX3/CX3 デバイスをデバッグするために使用されます。
低帯域幅ペリフェラル	FX3/CX3 のシリアル ペリフェラル I^2C、SPI、I^2S、UART および GPIO は低帯域幅ペリフェラル（LBP）を形成します。詳細については、*EZ-USB FX3 TRM* の低帯域幅ペリフェラルの章を参照してください。
MCLK	MCLK は CX3 からのクロック出力であり、イメージ センサーの入力リファレンス クロックとして使用されます。詳細は本書の 1.7.5 節に記載されています。
CPU およびメモリ サブシステム	CX3 メモリ サブシステムは、プログラムおよびデータ メモリとして機能するシステム RAM、SRAM コントローラ、および ARM CPU とハードウェア ブロックがこれらのメモリにアクセスできるようにする高度な高性能バス（AHB）ベースの相互接続から構成されています。メモリ マップ I/O（MMIO）相互接続は様々なペリフェラル ブロックでレジスタへのアクセスを提供します。詳細については *EZ-USB FX3 TRM* のメモリおよびシステム相互接続の章を参照してください。

MIPI CSI-2	モバイル業界プロセッサ インターフェース（MIPI）アライアンスのカメラ シリアル インターフェース 2（CSI-2）規格は MIPI アライアンスによって定義されたシリアル カメラ インターフェース。標準的なデータ転送を定義し、カメラのトランスミッタおよびレシーバ間のインターフェースを制御します。データ転送インターフェースはデータ信号とクロック信号との単方向の差動シリアル インターフェース。MIPI CSI-2 仕様の詳細については、MIPI アライアンスのウェブサイト http://www.mipi.org/specifications/camera-interface を参照してください。
MJPEG フォーマット	Motion-JPEG または MJPEG は動画フォーマットであり、各動画フレームが JPEG 静止画像圧縮アルゴリズムを使用して別々に圧縮されます。そのようなフレームのシーケンスは元の動画を表示します。
画素／ピクセル	画素は画面に示される画像の最も小さい制御可能な要素です。各画素は元の画像の サンプル。一般的にサンプルが多ければ多いほど元の画像の表示がより正確になります。一般的に画素数は画像センサーの分解能を示します。
RAW フォーマット	RAW フォーマットのデータは画像センサーからの直接な処理されていない出力を提供します。CX3 MIPI CSI-2 レシーバ インターフェースは RAW8、RAW10、RAW12、RAW14 フォーマットをサポートしており、数値は画像センサーの色深度ビット数を示します。カラーフィルタ アレイ（例えばベイヤー フィルタ）は一般的に RAW フォーマットの出力から RGB カラーの情報を得るために使用されます。
RGB フォーマット	赤、緑と青（R、G、B）コンポーネントに関する色空間を定義するカラー フォーマット。CX3 MIPI CSI-2 レシーバ インターフェースは RGB888、RGB666 および RGB565 ストリームをサポートしており、数値はストリームの色深度ビット数を示します（例えば RGB 565 は赤データが 5 ビット、緑データが 6 ビット、青データが 5 ビットあります）。
予約済みビット	予約済みレジスタ ビットはユーザーが変更してはいけません。 予約済みビットを含むレジスタへの書き込みは、書き込みにより予約済みビットの値を変更していないことを確認する必要があります。書き込み前に読み出しを行い、書き込みで予約済みビットの値を維持すること推奨します。
SDK （ソフトウェア開発キット）	EZ-USB FX3 SDK（ソフトウェア開発キット）はフル機能のファームウェア ソリューション。このソリューションにより、顧客は FX3/FX3S/CX3 デバイスの機能を活用し、短時間でカスタム設計を構築することができます。SDK は、これらのデバイス上の様々なハードウェア ブロック用のドライバやコンビニエンス API のセットから構成されるファームウェアのフレームワークを提供します。ドライバは複雑な FX3 デバイス アーキテクチャを要約し、顧客が FX3 デバイスの動作を設定・制御できるように使いやすい API を提供します。
ソケット	DMA ソケットは、デバイスとのデータパスの一端をマッピングする FX3 のデバイス構造物。詳細については *EZ-USB FX3 TRM* の FX3 DMA サブシステムの章を参照してください。
USB	FX3/CX3 の USB サブシステムに関する詳細は EZ-USB FX3 TRM の USB の章に記載されています。CX3 デバイスは USB 周辺機器としてのみ動作し、FX3 が対応する USB ホスト、OTG、充電器機能に対応していません。
YUV フォーマット	輝度（Y）と色差（UV）コンポーネントに関する色空間を定義するカラー フォーマット。CX3 MIPI CSI-2 レシーバ インターフェースは YUV422 8 ビットと 10 ビットのカラー フォーマットをサポートしています。
XRESET	画像センサーをリセットするのに使用される CX3 出力信号。
XSHUTDOWN	画像センサーの電力モードを制御するのに使用される CX3 出力信号。

変更履歴

本書の変更履歴

文書名: EZ-USB® CX3 テクニカル リファレンス マニュアル 文書番号: 001-92482			
版	**ECN 番号**	**変更者**	**変更内容**
**	4377479	HZEN	これは英語版 001-91492 Rev. **を翻訳した日本語版 Rev. **です。
*A	5784056	SSAS	最新のテンプレートおよびロゴに変更しました。
*B	6416250	YSAT	これは英語版 001-91492 Rev. *B を翻訳した日本語版 Rev. *B です。

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2014-2018. 本書面は, Cypress Semiconductor Corporation 及び Spansion LLC を含むその子会社 (以下「Cypress」という。) に帰属する財産である。本書面 (本書面に含まれ又は言及されているあらゆるソフトウェア若しくはファームウェア (以下「本ソフトウェア」という。) を含む) は, アメリカ合衆国及び世界のその他の国における知的財産法令及び条約に基づき Cypress が所有する。Cypress はこれらの法令及び条約に基づく全ての権利を留保し, 本段落で特に記載されているものを除き, その特許権, 著作権, 商標権又はその他の知的財産権のライセンスを一切許諾しない。本ソフトウェアにライセンス契約書が伴っておらず, かつ Cypress との間で別途本ソフトウェアの使用方法を定める書面による合意がない場合, Cypress は, (1) 本ソフトウェアの著作権に基づき, (a) ソースコード形式で提供されている本ソフトウェアについて, Cypress ハードウェア製品と共に用いるためにのみ, かつ組織内部でのみ, 本ソフトウェアの修正及び複製を行うこと, 並びに (b) Cypress のハードウェア製品ユニットに用いるためにのみ, (直接又は再販売者及び販売代理店を介して間接のいずれかで) 本ソフトウェアをバイナリーコード形式で外部エンドユーザーに配布すること, 並びに (2) 本ソフトウェア (Cypress により提供され, 修正がなされていないもの) が抵触する Cypress の特許権のクレームに基づき, Cypress ハードウェア製品と共に用いるためにのみ, 本ソフトウェアの作成, 利用, 配布及び輸入を行うことについての非独占的で譲渡不能な一身専属的ライセンス (サブライセンスの権利を除く) を付与する。本ソフトウェアのその他の使用, 複製, 修正, 変換又はコンパイルを禁止する。

適用される法律により許される範囲内で, Cypress は, 本書面又はいかなる本ソフトウェア若しくはこれに伴うハードウェアに関しても, 明示又は黙示をとわず, いかなる保証 (商品性及び特定の目的への適合性の黙示の保証を含むがこれらに限られない) も行わない。いかなるコンピューティングデバイスも絶対に安全ということはない。従って, Cypress のハードウェアまたはソフトウェア製品に講じられたセキュリティ対策にもかかわらず, Cypress は, Cypress 製品への権限のないアクセスまたは使用といったセキュリティ違反から生じる一切の責任を負わない。加えて, 本書面に記載された製品には, エラッタと呼ばれる設計上の欠陥またはエラーが含まれている可能性があり, 公表された仕様とは異なる動作をする場合がある。適用される法律により許される範囲内で, Cypress は, 別途通知することなく, 本書面を変更する権利を留保する。Cypress は, 本書面に記載のある, いかなる製品若しくは回路の適用又は使用から生じる一切の責任を負わない。本書面で提供されたあらゆる情報 (あらゆるサンプルデザイン情報又はプログラムコードを含む) は, 参照目的のためのみに提供されたものである。この情報で構成するあらゆるアプリケーション及びその結果としてのあらゆる製品の機能性及び安全性を適切に設計, プログラム, かつテストすることは, 本書面のユーザーの責任において行われるものとする。Cypress 製品は, 兵器, 兵器システム, 原子力施設, 生命維持装置若しくは生命維持システム, 蘇生用の設備及び外科的移植を含むその他の医療機器若しくは医療システム, 汚染管理若しくは有害物質管理の運用のために設計され若しくは意図されたシステムの重要な構成部分としての使用, 又は装置若しくはシステムの不具合が人身傷害, 死亡若しくは物的損害を生じさせるようなその他の使用 (以下「本目的外使用」という。) のためには設計, 意図又は承認されていない。重要な構成部分とは, それの不具合が装置若しくはシステムの不具合を生じさせるか又はその安全性若しくは実効性に影響すると合理的に予想できるような装置若しくはシステムのあらゆる構成部分をいう。Cypress 製品のあらゆる本目的外使用から生じ, 若しくは本目的外使用に関連するいかなる請求, 損害又はその他の責任についても, Cypress はその全部又は一部をとわず一切の責任を負わず, かつ Cypress はそれら一切から本書により免除される。Cypress は Cypress 製品の本目的外使用から生じ又は本目的外使用に関連するあらゆる請求, 費用, 損害及びその他の責任 (人身傷害又は死亡に基づく請求を含む) から免責補償される。

Cypress, Cypress のロゴ, Spansion, Spansion のロゴ及びこれらの組み合わせ, WICED, PSoC, CapSense, EZ-USB, F-RAM, 及び Traveo は, 米国及びその他の国における Cypress の商標又は登録商標である。Cypress のより完全な商標のリストは, cypress.com を参照すること。その他の名称及びブランドは, それぞれの権利者の財産として権利主張がなされている可能性がある。