



EZ-USB[®] CX3 技术参考手册

(EZ-USB FX3 技术参考手册的补充文件)

文档编号: 001-92471 版本**

赛普拉斯 EZ-USB CX3



1.1 简介

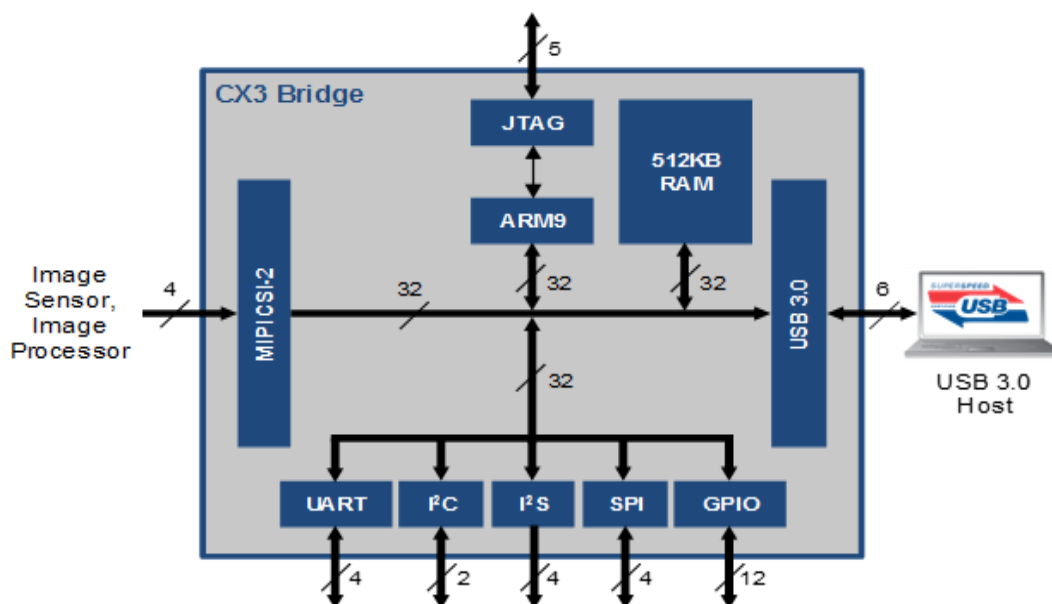
赛普拉斯 EZ-USB® CX3 是一个允许开发人员在任何图像传感器中添加 USB 3.0 连接的 USB 3.0 摄像机控制器。它符合移动工业处理器接口 (MIPI) 摄像机串行接口类型 2 (CSI-2) 标准, 并作为一个 MIPI CSI-2 至 USB 桥接器使用。

EZ-USB CX3 是 EZ-USB FX3 器件的一个变体。FX3 具有一个与通用编程接口 II (GPIF II) 相配的集成 MIPI CSI-2 接收器。可以使用 CX3 将超速 USB 连接添加到支持 MIPI CSI-2 接口的图像传感器内。

CX3 符合 MIPI CSI-2 标准 (版本 1.01), 并支持四个数据通道。总带宽为 4 Gbps, 且每通道的速度为每秒十亿比特。

CX3 非常适合高清或高速度图像捕获应用, 并具有流无压缩视频 (在 30 fps 为 1080p 或在 60 fps 为 720p) 的功能。CX3 支持各种图像格式, 如 RAW8/10/12/14、YUV422、RGB888/666/565 和用户定义的 8 位。

图 1: EZ-USB® CX3 器件



根据经过验证的 EZ-USB FX3 平台, CX3 包含一个 ARM9™ CPU 和 512 KB SRAM, 这些器件的运算能力为 200 MIPS。CX3 支持许多外设接口, 如 I²C、SPI 和 UART。可对这些接口进行编程, 使之支持自动对焦、平滑、倾斜、缩放 (PTZ) 等功能。

CX3 和 FX3 使用同一个应用开发工具。FX3 软件开发套件支持 CX3, 并提供各个应用示例, 以加快上市时间。由于在符合 USB 3.0 v1.0 规范的前提下编译 CX3, 所以该器件与 USB 2.0 相兼容。

本技术参考手册 (TRM) 是 [EZ-USB FX3 TRM](#) 的补充文件, 用于介绍有关新增的 CX3 MIPI CSI-2 接收器功能模块的详细信息。本技术参考手册补充文件中的第 1.5 节到第 1.9 节描述了 MIPI CSI-2 接收器功能模块 (包括固定功能 GPIF II 状态机在内)。第 1.10 节说明了有关 MIPI CSI-2 接收器接口的寄存器的详细信息。

EZ-USB FX3 TRM 中提供了现有 FX3 功能模块（如 CPU 子系统、存储器、全局控制器、DMA、USB 以及低带宽（串行和 GPIO）外设）的详细信息。本技术参考手册中的术语一节介绍了所使用的术语。

图 2 显示了 FX3 框图。红色加亮显示的一些功能模块不包含在 CX3 中。蓝色加亮显示的 GPIF II 则包含在 CX3 中，但该模块仅显示摄像机接口的特定功能。

图 2: FX3 框图

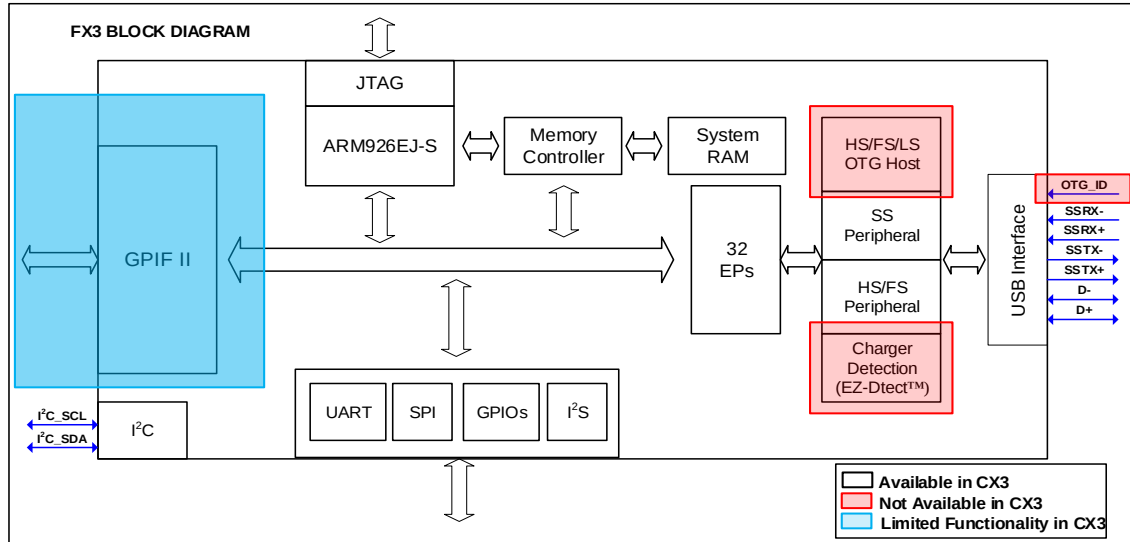
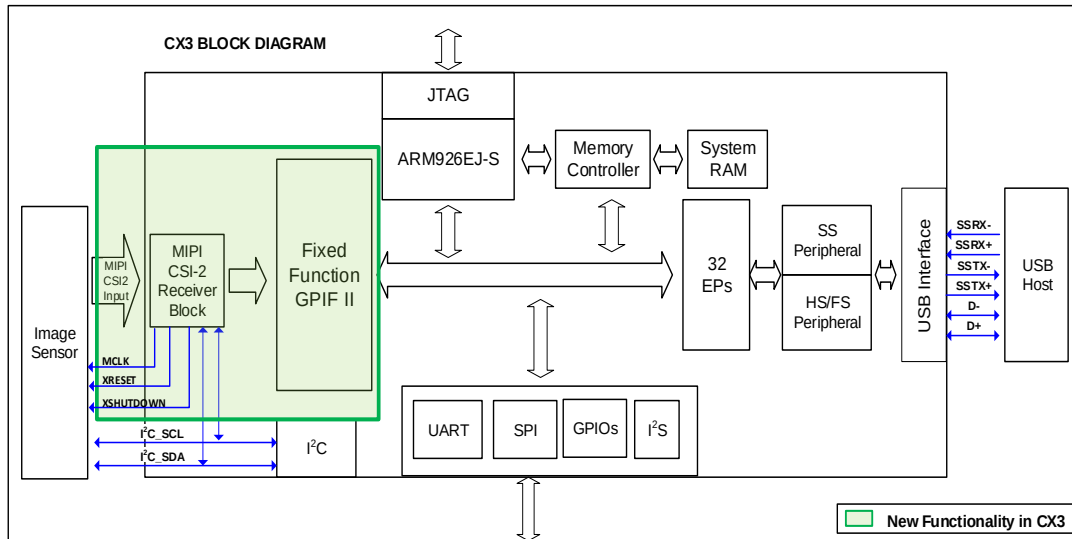


图 3 显示了 CX3 框图，其中包括新增的 MIPI CSI-2 功能模块和固定功能的 GPIF II 接口（绿色高亮显示）。

图 3: CX3 框图



下表列出了 CX3 和 FX3 器件的区别。

表 1: FX3 和 CX3 器件的区别

特性	在 FX3 中支持	在 CX3 中支持
P 端口支持		
通过 GPIF II 实现的同步 ADMux	有	无
通过 GPIF II 实现的异步 SRAM	有	无
通过 GPIF II 实现的异步 ADMux	有	无
MMC 从设备	有	无
GPIF II	有	实现固定功能的 GPIF II 状态机，以连接至 CX3 MIPI CSI-2 接收器模块。
低带宽外设		
I ² S 主设备（仅发送器）	有	有
SPI 主设备	有	有
UART	有	有
I ² C 主设备控制器	有	有
U 端口支持		
USB 3.0 外设	有	有
USB 2.0 外设	有	有
32 个物理端点	有	有
充电器检测 1.1 支持（EZ-Dtect™）	有	无
配件充电器适配器（ACA）支持	有	无
集成高速 USB 开关	有	无
车用套件 UART 通过模式	有	无
USB OTG（高速、全速、低速主机或外设）	有	无
时钟输入频率：	19.2 MHz、 26 MHz、 38.4 MHz、 52 MHz	19.2 MHz
晶体输入：19.2 MHz	有	无
支持的 JTAG 仅用于调试（不支持边界扫描）	有	有
VBUS、VBAT 信号	分别用于 5 V 和 3.3 V 操作的独立 VBUS、VBAT 信号	VBUS 与 VBAT 组合成一个单一信号 VUSB。支持 3.3 V 和 5 V 操作。
通用输入/输出	42	12
用于图像传感器的时钟输出	无	有
MIPI CSI-2 输入	无	有

1.2 CX3 特性

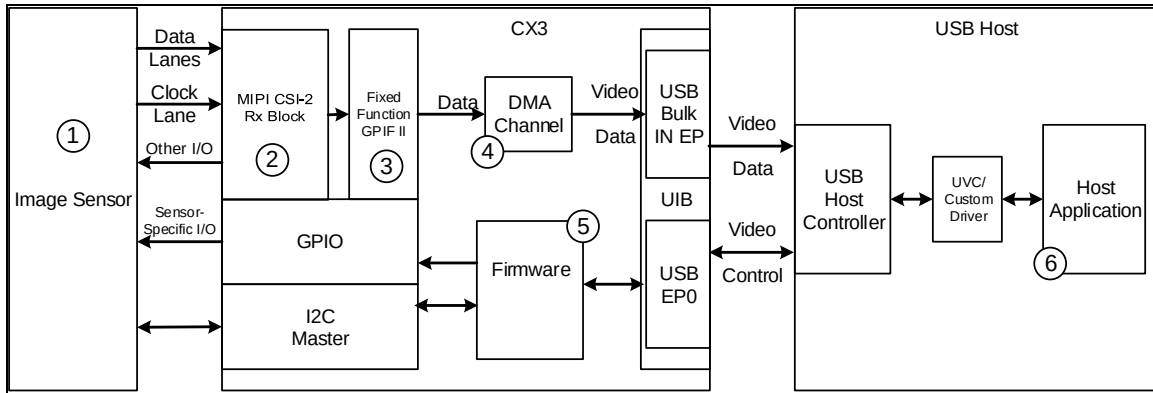
CX3 支持以下特性：

- USB 3.0 和 USB 2.0 外设控制器与 USB 3.0 规范 1.0 相兼容
- MIPI CSI-2 RX 接口
 - MIPI CSI-2 兼容（版本 1.01 修订版 0.04 — 2009 年 4 月 2 日）
 - 支持多达四个通道；每个通道的传输速度可达 1 Gbps
 - 支持用于配置图像传感器的摄像机控制接口（通过 I²C）
- 支持下面的视频数据格式
 - RAW 8/10/12/14
 - YUV 422（8/10 位）
 - RGB 888 / 666 / 565
 - 用户定义的 8 位
- 12 个 GPIO，用于控制摄像机相关功能（如照明、同步输入、同步输出，等等）
- 与 FX3 相同，该器件也支持 I²C、SPI、I²S 输出和 UART 接口
- 用于调试的 JTAG 接口

1.3 框图

图4显示了一个典型系统的详细框图，其中通过CX3将数据从图像传感器传输到USB主机。

图 4：系统框图



为框图的主要子模块进行编号，这些任务由下述的各个子模块执行：

1. 基于 MIPI CSI-2 的图像传感器与 CX3 相连接，并且通过使用摄像机控制接口（I²C）总线配置该传感器。
2. CX3 MIPI CSI-2 接收器模块从图像传感器读取数据，对其进行反序列化，然后合并各通道，拆开各数据包，最后将数据作为并行输入发送到固定功能的 GPIF II 模块。
3. GPIF II 模块及其固定功能状态机通过 DMA 通道将图像传感器的数据发送到 USB 接口。
4. DMA 通道将图像数据从 GPIF II 模块传输到 USB 接口模块（UIB）。
5. CX3 固件初始化 CX3 硬件模块、配置图像传感器和 MIPI CSI-2 控制器、控制 USB 接口以及处理所有的 USB 协议请求。在将视频流数据发送到 USB 接口前，通过自定义 CX3 固件，可以将类特性标题添加到该数据。
6. 主机应用（如图像信号处理器或视频流播放器）提供了控制请求，用以配置视频流和传感器并在主机 PC 上处理和显示视频流。

1.4 MIPI CSI-2 模块配置 API

使用 MIPI CSI-2 Rx 模块配置 API，用户应用可以初始化、配置以及执行摄像机接口的电源管理。这些 API 通过 I²C 总线与摄像机通信。

为实现模块化，MIPI CSI-2 配置 API 被编译到一个独立的 API 库（*cyu3mipicsi.a*）。仅在需要该库的功能时，才将其链接至应用。

API 库中的主要 API 如下：

表 2：EZ-USB FX3 SDK 中的 CX3 特定 API

API 名称	API 说明
<i>CyU3PMipicsiInit()</i>	用于初始化 MIPI CSI-2 模块的 API。
<i>CyU3PMipicsiDeInit()</i>	用于取消初始化 MIPI CSI-2 模块的 API。
<i>CyU3PMipicsiSetIntfParams()</i>	用于配置 MIPI CSI-2 模块的时钟和接口设置的 API。
<i>CyU3PMipicsiQueryIntfParams()</i>	从 MIPI CSI-2 模块查询设置的 API。
<i>CyU3PMipicsiSleep()</i>	用于将 MIPI CSI-2 模块置于低功耗睡眠模式的 API。
<i>CyU3PMipicsiWakeup()</i>	用于使 MIPI CSI-2 模块从低功耗睡眠模式进入活动模式的 API。
<i>CyU3PCx3DeviceReset()</i>	用于复位 MIPI CSI-2 模块的 API。
<i>CyU3PCx3DeviceReset()</i>	用于在 CX3 器件上执行热复位或冷复位的 API。
<i>CyU3PMipicsiSetSensorControl()</i>	用于将 MIPI CSI-2 XRESET 和 XSHUTDOWN 信号驱动到图像传感器的 API。 XRESET 是一个用于复位图像传感器的 CX3 输出信号。 XSHUTDOWN 是一个用于控制图像传感器功耗模式的 CX3 输出信号。
<i>CyU3PMipicsiGpifLoad()</i>	该 API 用于加载固定功能的 GPIF II 波形并配置固定功能 GPIF II 总线宽度和 DMA 缓冲器大小。

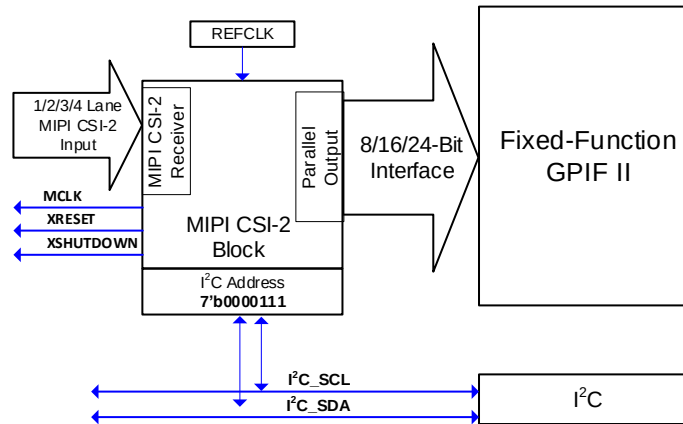
更多有关 API 的信息，请参阅第 1.11 节中的内容

有关 API 文档的详细信息，请查看《EZ-USB FX3 SDK 固件 API 指南》中的 CX3 的 MIPI CSI-2 和固定功能 GPIF 接口一节（位于 EZ USB FX3 SDK 中）。

1.5 MIPI CSI-2 模块

CX3 器件有一个集成的 MIPI CSI-2 模块，该模块的一侧硬连接到 GPIF II 接口，另一侧提供 MIPI CSI-2 接口，用以连接到支持 MIPI CSI-2 的图像传感器。该模块支持四个 MIPI CSI-2 数据通道，每通道的速度可达 1 Gbps。MIPI CSI-2 接收器通过 8 位、16 位或 24 位数据总线（时钟频率可达 100 MHz）连接至固定功能的 GPIF II 控制器。可通过 I²C 配置 MIPI CSI-2 模块，并且该模块在 CX3 I²C 总线上的位置为 7 位 I²C 从设备地址 7'b0000111。

图 5. CX3 MIPI CSI-2 模块



1.6 CX3 MIPI CSI-2 流格式

CX3 中的 MIPI CSI-2 Rx 模块内部支持以下流格式和输出模式：

表 3: CX3 MIPI CSI-2 流格式

格式和模式	CX3 固件流格式名称	格式说明、像素深度	CSI-2 数据类型	GPIO II 总线宽度	输出流
RAW8	CY_U3P_CSI_DF_RAW8	RAW 格式、每像素 8 位	0x2A	8 位	RAW [7:0]
RAW10	CY_U3P_CSI_DF_RAW10	RAW 格式、每像素 10 位	0x2B	16 位	6'b0、RAW [9:0]
RAW12	CY_U3P_CSI_DF_RAW12	RAW 格式、每像素 12 位	0x2C	16 位	4'b0、RAW [11:0]
RAW14	CY_U3P_CSI_DF_RAW14	RAW 格式、每像素 14 位	0x2D	16 位	2'b0、RAW[13:0]
RGB888	CY_U3P_CSI_DF_RGB888	RGB 888 格式、每像素 24 位	0x24	24 位	R[7:0]、G[7:0]、B[7:0]
RGB666 模式 0	CY_U3P_CSI_DF_RGB666_0	RGB 666 格式、每像素 24 位	0x23	24 位	2'b0、R[5:0]、2'b0、G[5:0]、2'b0、B[5:0]
RGB666 模式 1	CY_U3P_CSI_DF_RGB666_1	RGB 666 格式、每像素 24 位	0x23	24 位	6'b0、R[5:0]、G[5:0]、B[5:0]
RGB565 模式 0	CY_U3P_CSI_DF_RGB565_0	RGB 565 格式、每像素 24 位	0x22	24 位	2'b0、R[4:0]、3'b0、G[5:0]、2'b0、B[4:0]、1'b0
RGB565 模式 1	CY_U3P_CSI_DF_RGB565_1	RGB 565 格式、每像素 24 位	0x22	24 位	3'b0、R[4:0]、2'b0、G[5:0]、3'b0、B[4:0]
RGB565 模式 2	CY_U3P_CSI_DF_RGB565_2	RGB 565 格式、每像素 16 位	0x22	16 位	R[4:0]、G[5:0]、B[4:0]
YUV422 8 位模式 0	CY_U3P_CSI_DF_YUV422_8_0	YUV422 格式、每像素 16 位	0x1E	8 位	P[7:0] 数据顺序: U1、Y1、V1、Y2、U3、Y3、..
YUV422 8 位模式 1	CY_U3P_CSI_DF_YUV422_8_1	YUV422 格式、每像素 16 位	0x1E	16 位	P[15:0]数据顺序: {U1,Y1}、{V1,Y2}、{U3,Y3}、{V3,Y4}...
YUV422 8 位模式 2	CY_U3P_CSI_DF_YUV422_8_2	YUV422 格式、每像素 16 位	0x1E	16 位	P[15:0] 数据顺序: {Y1,U1}、{Y2,V1}、{Y3,U3}、{Y4,V3}....

格式和模式	CX3 固件流格式名称	格式说明、像素深度	CSI-2 数据类型	GPIF II 总线宽度	输出流
YUV422 10 位	CY_U3P_CSI_DF_YUV422_10	YUV422 格式、 每像素 20 位	0x1F	16 位	6'b0、P[9:0]数据 顺序: U1、Y1、 V1、Y2、U3、Y3 、V3、Y4。

^a MIPI CSI-2 定义的数据类型代码。更多信息，请参阅 MIPI CSI-2 规范。

如果所选的 GPIF II 总线宽度大于输出流的宽度（例如，如果将 24 位 GPIF II 总线宽度使用于 CY_U3P_CSI_DF_YUV422_8_1 类型），将使用各个位 0 填充 GPIF II 的高位。

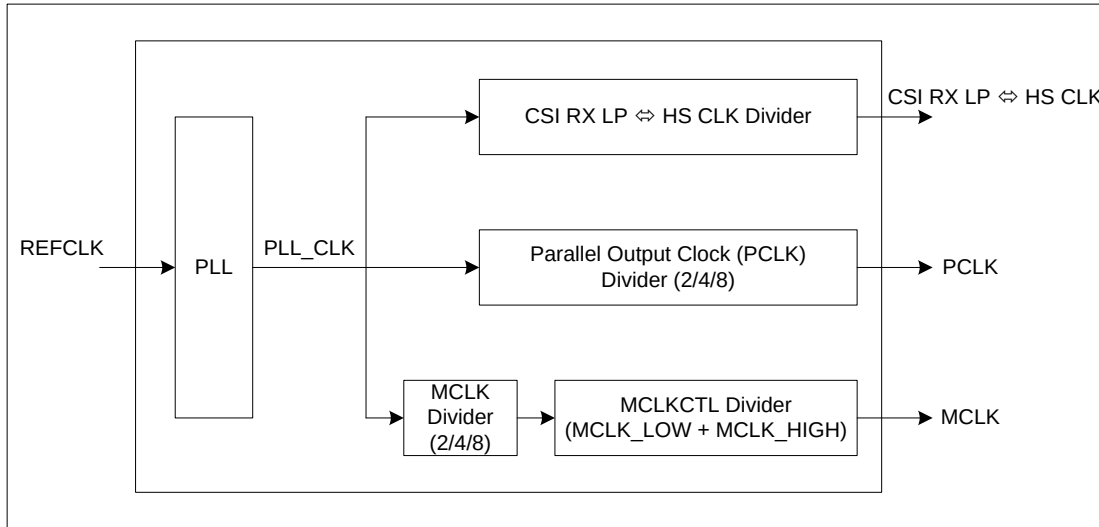
对于未显示在表 3 的流格式（如 MJPEG 或自定义格式流），以 RAW8 流格式传输 8 位的流数据，并以 RGB888 流格式传输 24 位的数据流。需要准确设置 GPIF 总线宽度和缓冲器，使之与流输出相匹配。更多有关设置 GPIF 总线宽度和 DMA 缓冲器的信息，请参考 1.11.12 一节。

可在每个 PCLK 中输出多个像素。例如，当将“MIPI CSI input -Data format”配置为“RAW8”，并将“MIPI interface configuration – data format”设置为 24 位格式（RGB888）时，可以在每个 PCLK 内输出三个像素。类似的，通过使用 24 位输出格式，可在每个 PCLK 中打包并输出两个 10 位或 12 位像素。

1.7 MIPI CSI-2 模块时钟

图 6 显示了 MIPI CSI-2 模块中的 CX3 时钟。该接口将一个参考时钟作为其输入，并使用后面跟着多个时钟分频器的 PLL 生成所需要的时钟。在 CX3 应用固件中，时钟配置参数是 `CyU3PMipicsiCfg_t` 结构的一部分，并被发送到 `CyU3PMipicsiSetIntfParams()` API 以配置 CSI-2 模块。有关结构和配置 API 的信息，请参考 [EZ-USB FX3 SDK 固件 API 指南](#)。

图 6. CX3 MIPI CSI-2 模块时钟



下面内容介绍了每个时钟的简要说明。

1.7.1 参考时钟（REFCLK）

这是为 MIPI CSI-2 模块提供的参考时钟输入。该时钟输入的频率范围应为 6 至 40 MHz。

1.7.2 PLL 时钟（PLL_CLK）

PLL_CLK 是 MIPI CSI-2 模块的主要时钟。PLL 时钟的频率范围应为 62.5 MHz 至 1 GHz。其他所有的内部和输出时钟都来源于该时钟。

通过使用以下公式，可以从输入参考时钟生成 PLL 时钟频率：

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

其中：

PLL_FBD 是反馈分频器，其范围值为 0 至 0x1FF。

PLL_PRD 是输入分频器，其范围值为 0 至 0x0F。

PLL_FRS 是频率范围选择的参数。其值可以为：

- ‘0’，如果 PLL 时钟频率的范围介于 500 MHz 和 1 GHz 之间。
- ‘1’，如果 if the PLL clock is between 250 MHz and 500 MHz.
- ‘2’，如果 PLL 时钟频率的范围介于 125 MHz 和 250 MHz 之间。
- ‘3’，如果 PLL 时钟频率的范围介于 62.5 MHz 和 125 MHz 之间。

下表显示的是 PLL 时钟频率的计算示例，其中 REFCLK 的值为 19.2 MHz：

表 4: PLL 时钟频率的计算示例

REFCLK (MHz)	PLL_PRD	PLL_FBD	PLL_FRS	PLL_CLK (MHz)
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 时钟

该时钟用于检测 CSI 链路低功耗 (LP) ↔ 高速 (HS) 的切换。通过对 PLL_CLK 进行 2、4 或 8 分频可以得到该时钟。更多有关 CSI 链路 LP ↔ HS 切换的信息，请查看 MIPI 联盟 (<http://www.mipi.org/specifications/camera-interface>) 中提供的 MIPI CSI-2 规范文档。

该时钟的最大频率为 125 MHz。

此时钟频率通过“MIPI 接收器配置工具”被自动计算。

1.7.4 输出并行时钟 (PCLK)

该时钟是 PCLK 输出，用于驱动 CX3 中功能固定的 GPIF II 接口。通过对 PLL_CLK 进行 2、4 或 8 分频可生成此时钟。

该时钟的最大值为 100 MHz。

此时钟频率通过“MIPI 接收器配置工具”被自动计算。

1.7.5 图像传感器的参考时钟 (MCLK)

MCLK 是一个可选的时钟输出，可将它作为图像传感器的输入参考时钟使用。PLL_CLK 先进行 mClkRefDiv (2/4/8) 分频，然后再进行 MCLKCTL 分频，从而生成该时钟。MCLKCTL 分频器指定了分频 PLL_CLK 计算的高电平时间和低电平时间。

高八位定义了高电平时间计数 (1-255)，低八位定义了低电平时间计数 (1-255)。

MCLK 可通过下面的公式计算得出：

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

例如：

如果 PLL_CLK 的频率为 672 MHz，那么要想生成频率为 24 MHz 的 MCLK，则需要将 mClkRefDiv 设置为 4，并将 mClkCtl 设置为 0x0203。

因此：

$$\begin{aligned} MCLK &= (672 / 4) / ((2 + 1) + (3 + 1)) \\ &= 168 / 7 = 24 \text{ MHz} \end{aligned}$$

当高字节 (MClkCtl) 和低字节 (MClkCtl) 均为非 0 值时，MCLK 是唯一的输出。

此时钟频率通过“MIPI 接收器配置工具”被自动计算。

1.8 CX3 MIPI CSI-2 模块的功耗模式

CX3 的 MIPI CSI-2 模块支持下列各功耗模式：

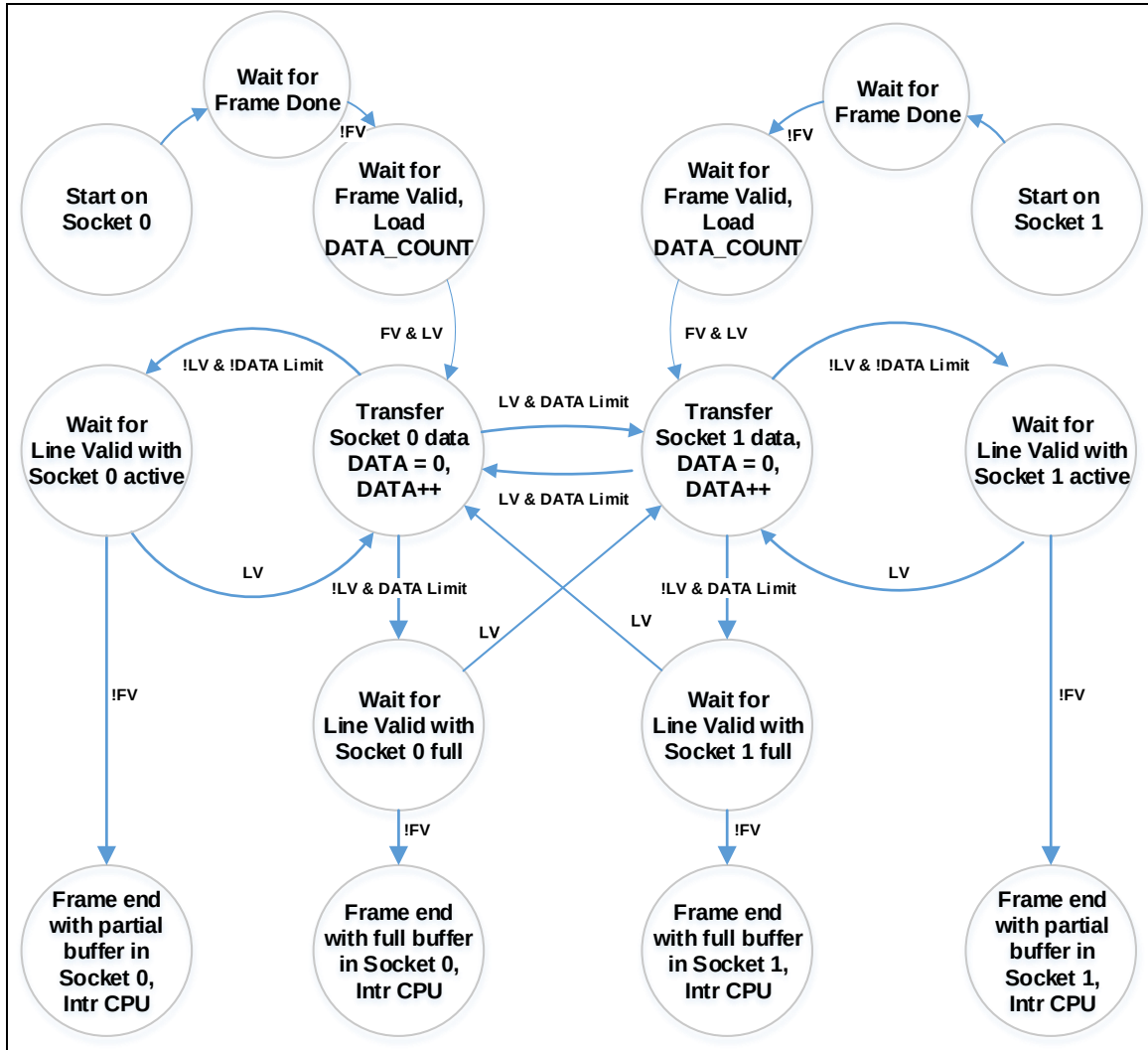
- **活动模式：**在该模式下，MIPI CSI-2 接收器模块有效，并且将数据从图像传感器传输到 GPIF II 接口。
- **低功耗睡眠模式：**在该模式下，MIPI CSI-2 模块上的所有数据传输时钟均被停止，因此没有数据传输。所有寄存器设置都会保留该状态。
- **软复位：**此状态会使 MIPI CSI-2 模块进入复位模式，其中接口上的所有时钟均被停止。进行软复位后，MIPI CSI-2 模块配置寄存器的状态被保留。
- **硬复位：**在该状态下，MIPI CSI-2 控制器和所有时钟均被停止。通过硬复位可将 MIPI CSI-2 模块配置寄存器复位到它们的默认状态。在 MIPI CSI-2 接收器初始化期间，会自动执行硬复位。

1.9 CX3 的功能固定 GPIF II 接口

CX3 的功能固定 GPIF II 状态机如图 7 所示。此状态机允许 MIPI CSI-2 接收器提供的并行数据传输两个以上的套接字，这些套接字能与手动多对一 DMA 通道相连。

此状态机和 GPIF II 状态机的功能是相同的。应用笔记 AN75779 — 如何将图像传感器连接到 USB 视频类别 (UVC) 框架中的 EZ-USB® FX3™ 介绍了该功能。

图 7. CX3 GPIFII 状态机



FX3 SDK (软件开发套件) 提供了 [CyU3PMipicsiGpifLoad\(\)](#) API，用于将该固定功能状态机加载到 CX3 GPIF II 模块。此 API 允许选择总线宽度 (8、16 或 24 位) 与 MIPI CSI-2 模块上的并行接口的输出相匹配，并且允许 GPIF II 接口使用 DMA 缓冲区大小的设置。缓冲区大小的计算在 [1.11.12](#) 章节中的 [CyU3PMipicsiGpifLoad\(\)](#) API 文档进行了介绍。

1.10 MIPI CSI-2 模块寄存器

MIPI CSI-2 模块提供了一组寄存器，以通过 I²C 进行配置。该模块在 7 位 I²C 从设备地址 **7'b0000111** 上有效（读地址 **0x0F**；写地址 **0x0E**）。该模块支持 100 kHz 和 400 kHz 的 I²C 操作。

注意：请勿将其他 I²C 器件（这些器件的地址与 MIPI CSI-2 模块的地址是相同的）连接到 CX3 器件的 I²C 总线上。

这些是 16 位对齐的寄存器，并且进行数据传输时都会优先传输最高有效位（MSB）。

图 8 和图 9 分别显示了 MIPI CSI-2 模块的典型 I²C 读和写传输。

图 8. 写传输序列

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
---	-----------------------------------	---	---	------------------------	---	-----------------------	---	------------	---	-----------	---	---

图 9. 读传输序列

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 显示的是 CX3 MIPI CSI-2 模块的配置寄存器列表。

表 5: MIPI CSI-2 模块配置寄存器

寄存器地址	寄存器名称	说明
0x0002	CX3_SYSTEM_CTRL	系统控制寄存器
0x0004	CX3_CONFIG_CTRL	配置控制寄存器
0x0006	CX3_FIFO_CTRL	FIFO 控制寄存器
0x0008	CX3_DATA_FMT	数据格式控制寄存器
0x000C	CX3_MCLK_CTRL	MCLK 控制寄存器
0x0010	CX3_CSI_SENSOR_SIG_EN	用于使能 MIPI CSI-2 信号的寄存器（XRESET 和 XSHUTDOWN）
0x0014	CX3_CSI_SENSOR_SIG_VAL	用于控制 MIPI CSI-2 信号的寄存器（XRESET 和 XSHUTDOWN）
0x0016	CX3_PLL_CTRL0	PLL 时钟控制寄存器 0
0x0018	CX3_PLL_CTRL1	PLL 时钟控制寄存器 1
0x0020	CX3_CLK_CTRL	时钟控制寄存器
0x0022	CX3_BYTE_COUNT	字节计数寄存器
0x0060	CX3_PHY_TIME_DELAY	用于设置 MIPI THS-Settle 定时器设置的寄存器。

以下内容介绍了每个 CX3 MIPI CSI-2 配置寄存器的详细情况。

注意：不能修改 RESERVED 寄存器位。将 RESERVED 位写入到某个寄存器时，需要确保它们的值在写操作期间不会发生改变（首选方法是在进行写操作前先进行读操作，并在写操作期间保持 RESERVED 位的值）。

1.10.1 CX3_SYSTEM_CTRL（寄存器地址：0x0002）

该寄存器控制 MIPI CSI-2 模块的睡眠和软件复位功能。

该寄存器被 *CyU3PMipicsiSleep()*、*CyU3PMipicsiWakeup()*和 *CyU3PMipicsiReset()* API 使用。

位	15	14	13	12	11	10	9	8
名称	RESERVED							
位	7	6	5	4	3	2	1	0
名称	RESERVED						SLEEP	RESET

寄存器字段	位	说明
RESERVED	[15:2]	已保留。固件必须通过读取这些位，修改未保留位和重新写入这些位来保留它们的设置。
SLEEP	[1]	睡眠控制： 将 MIPI CSI-2 模块置于低功耗睡眠模式。 0: 正常操作 1: 睡眠模式
RESET	[0]	软件复位： 强制对 MIPI CSI-2 模块进行软件复位，并且不会清除配置寄存器。 0: 正常操作 1: 复位操作

1.10.2 CX3_CONFIG_CTRL（寄存器地址：0x0004）

该寄存器控制数据通道的数量以及 MIPI CSI-2 模块的输出数据模式。

该寄存器由 `CyU3PMipicsiSetIntfParams()` API 设置。通过使用 `CyU3PMipicsiQueryIntfParams()` API 可以查询它。

位	15	14	13	12	11	10	9	8
名称	RESERVED						DATA MODE	
位	7	6	5	4	3	2	1	0
名称	RESERVED	OUTEN	RESERVED				DATA LANES	

寄存器字段	位	说明
RESERVED	[15:10]	已保留。固件必须通过读取这些位，修改未保留位和重新写入这些位保留它们的设置。
数据模式	[9:8]	数据模式选择： 使用 <code>CX3_DATA_FMT</code> 寄存器的 DATA FORMAT 位为选中的数据格式选择数据模式。通过使用 DATA MODE 和 DATA FORMAT 的结合可选择表 3 中所定义的输出流。DATA FORMAT 设置确定输出数据的格式，而 DATA MODE 设置可确定 DATA TYPE 的数据封装和字节顺序，如表 3 中的定义。 2'b00: 模式 0 2'b01: 模式 1 2'b10: 模式 2 2'b11: 保留
RESERVED	[7]	已保留。固件必须通过读取这些位，修改未保留位和重新写入这些位保留它们的设置。
OUTEN	[6]	使能并行输出： 使能从 MIPI CSI-2 接收器模块到固定功能 GPIF II 的并行输出。 0: 禁用并行输出。 1: 使能并行输出
RESERVED	[5:2]	已保留。固件必须通过读取这些位，修改未保留位和重新写入这些位保留它们的设置。
DATA LANES	[1:0]	MIPI CSI-2 数据通道选择： 选择要使用的数据通道数量。 2'b00: 一个数据通道 2'b01: 两个数据通道 2'b10: 三个数据通道 2'b11: 四个数据通道

1.10.3 CX3_FIFO_CTRL (寄存器地址: 0x0006)

此寄存器确定 FIFO 触发电平，用于初始化从 MIPI CSI-2 模块的并行输出缓冲区并行地输出数据。将数据传输到 GPIF II 接口前，MIPI CSI-2 模块会等待并行输出缓冲区达到该寄存器所指定的电平。

该寄存器由 `CyU3PMipicsiSetIntfParams()` API 设置。通过使用 `CyU3PMipicsiQueryIntfParams()` API 可以查询它。

位	15	14	13	12	11	10	9	8
名称	RESERVED							FIFO LEVEL[8]
位	7	6	5	4	3	2	1	0
名称	FIFO LEVEL [7:0]							

寄存器字段	位	说明
RESERVED	[15:9]	已保留。固件必须通过读取这些位，修改未保留位和重新写入这些位保留它们的设置。
FIFO LEVEL	[8:0]	FIFO 级别： 确定 FIFO 写触发电平。仅仅当输出缓冲达到该电平时，MIPI CSI-2 模块才会启动 GPIF II 的并行输出。 范围：0x000-0x1FF

1.10.4 CX3_DATA_FMT（寄存器地址：0x0008）

该寄存器控制 MIPI CSI-2 模块的输出数据格式。

该寄存器由 `CyU3PMipicsiSetIntfParams()` API 设置。通过使用 `CyU3PMipicsiQueryIntfParams()` API 可以查询它。

位	15	14	13	12	11	10	9	8
名称	RESERVED							
位	7	6	5	4	3	2	1	0
名称	DATA FORMAT				RESERVED			DATA FMT ENABLE

寄存器字段	位	说明
RESERVED	[15:8]	已保留。固件必须通过读取这些位，修改未保留位和重新写入这些位保留它们的设置。
DATA FORMAT	[7:4]	数据格式选择： 选择输出数据的格式。 4'b0000: RAW8 4'b0001: RAW10 4'b0010: RAW12 4'b0011: RGB888 4'b0100: RGB666 4'b0101: RGB565 4'b0110: YUV422-8 位 4'b0111: RESERVED 4'b1000: RAW14 4'b1001: YUV422-10 位 4'b1010-1111: RESERVED 该字段以及 CX3_CONFIG_CTRL 寄存器中的 DATA MODE 字段均用于选择表 3 中所定义的输出流格式。 对于未显示在表 3 的流格式（如 MJPEG 或自定义格式流），以 RAW8 流格式传输 8 位的流数据，并以 RGB888 流格式传输 24 位的数据流。
RESERVED	[3:1]	已保留。固件必须通过读取这些位，修改未保留位和重新写入这些位保留它们的设置。
DATA FMT ENABLE	[0]	数据格式使能： 0: 不受支持。 1: 正常操作。

1.10.5 CX3_MCLK_CTRL (寄存器地址: 0x000C)

该寄存器配置 MCLK 分频器，并控制 MIPI CSI-2 模块中的图像传感器参考时钟 (MCLK) 的输出。一节详细介绍了计算 MCLK 值的信息。

该寄存器由 *CyU3PMipicsiSetIntfParams()* API 设置。通过使用 *CyU3PMipicsiQueryIntfParams()* API 可以查询它。

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

寄存器字段	位	说明
MCLK HIGH	[15:8]	MCLK 高电平的时间计数: MCLK 分频器的高电平时间计数。有关该字段的详细内容请参考章节 1.7.5 。 只有 MCLK HIGH 和 MCLK LOW 为非零值时，MCLK 才有效。
MCLK LOW	[7:0]	MCLK 低电平的时间计数: MCLK 分频器的低电平时间计数。有关该字段的详细内容请参考章节 1.7.5 。 $MCLK \text{ 分频器} = (MCLK \text{ HIGH} + 1) + (MCLK \text{ LOW} + 1)$

1.10.6 CX3_CSI_SENSOR_SIG_EN（寄存器地址：0x0010）

通过该寄存器可以使能或禁用 MIPI CSI-2 模块中 XSHUTDOWN 和 XRESET 信号的输出。

该寄存器由 [CyU3PMipicsiInit\(\)](#) 和 [CyU3PMipicsiReset\(\)](#) API 设置。

位	15	14	13	12	11	10	9	8
名称	RESERVED							
位	7	6	5	4	3	2	1	0
名称	RESERVED					XSHUTDOWN ENABLE	XRESET ENABLE	RESERVED

寄存器字段	位	说明
RESERVED	[15:3]	已保留。固件必须通过读取这些位，修改未保留位和重新写入这些位保留它们的设置。
XSHUTDOWN ENABLE	[2]	使能 MIPI CSI-2 XSHUTDOWN 信号： 0：使能 MIPI CSI-2 XSHUTDOWN 信号的输出。 1：禁用 MIPI CSI-2 XSHUTDOWN 信号的输出。 信号驱动值由 CX3_CSI_SENSOR_SIG_VAL （寄存器地址：0x0014）[2]确定。
XRESET ENABLE	[1]	使能 MIPI CSI-2 XRESET 信号： 0：使能 MIPI CSI-2 XRESET 信号的输出。 1：禁用 MIPI CSI-2 XRESET 信号的输出。 信号驱动值由 CX3_CSI_SENSOR_SIG_VAL （寄存器地址：0x0014）[1]确定。
RESERVED	[0]	已保留。固件必须通过读取这些位，修改未保留位和重新写入这些位保留它们的设置。

1.10.7 CX3_CSI_SENSOR_SIG_VAL（寄存器地址：0x0014）

该寄存器将配置 MIPI CSI-2 模块中 XSHUTDOWN 和 XRESET 信号的驱动值。

通过使用 [CyU3PMipicsiSetSensorControl\(\)](#) API 可以设置该寄存器。

位	15	14	13	12	11	10	9	8
名称	RESERVED							
位	7	6	5	4	3	2	1	0
名称	RESERVED					XSHUTDOWN OUTPUT	XRESET OUTPUT	RESERVED

寄存器字段	位	说明
RESERVED	[15:3]	已保留。固件必须通过读取这些位，修改未保留位和重新写入这些位保留它们的设置。
XSHUTDOWN OUTPUT	[2]	驱动 MIPI CSI-2 XSHUTDOWN 信号： 0：将 XSHUTDOWN 驱动为低电平。 1：将 XSHUTDOWN 驱动为高电平。 只在 CX3_CSI_SENSOR_SIG_EN （寄存器地址：0x0010）[2]为零值时，才会驱动该信号。
XRESET OUTPUT	[1]	驱动 MIPI CSI-2 XRESET 信号： 0：将 XRESET 驱动为低电平。 1：将 XRESET 驱动为高电平。 只在 CX3_CSI_SENSOR_SIG_EN （寄存器地址：0x0010）[1]为 0 时，才驱动该信号。
RESERVED	[0]	已保留。固件必须通过读取这些位，修改未保留位和重新写入这些位保留它们的设置。

1.10.8 CX3_PLL_CTRL0（寄存器地址：0x0016）

该寄存器配置 MIPI CSI-2 模块的 PLL 时钟。 章节 1.7.2 详细介绍了如何根据 CX3_PLL_CTRL0（寄存器地址：0x0016）和 CX3_PLL_CTRL1（寄存器地址：0x0018）中的值得到 PLL 时钟。

该寄存器由 *CyU3PMipicsiSetIntfParams()* API 设置。通过使用 *CyU3PMipicsiQueryIntfParams()* API 可以查询它。

位	15	14	13	12	11	10	9	8
名称	PLL PRD				RESERVED			PLL FBD [8]
位	7	6	5	4	3	2	1	0
名称	PLL FBD [7:0]							

寄存器字段	位	说明
PLL PRD	[15:12]	输入分频器： 用于生成 PLL 的输入分频器。有关该字段的详细内容请参考章节 1.7.2 。 范围：0x0 - 0xF。
RESERVED	[11:9]	已保留。固件必须通过读取这些位，修改未保留位和重新写入这些位保留它们的设置。
PLL FBD	[8:0]	反馈分频器： 用于生成 PLL 的反馈分频器。有关该字段的详细内容请参考章节 1.7.2 。 范围：0x000 - 0x1FF。

1.10.9 CX3_PLL_CTRL1（寄存器地址：0x0018）

该寄存器配置 MIPI CSI-2 模块上的 PLL 时钟。章节 1.7.2 详细说明了如何根据 CX3_PLL_CTRL0（寄存器地址：0x0016）和 CX3_PLL_CTRL1（寄存器地址：0x0018）中的值生成 PLL 时钟。

该寄存器由 *CyU3PMipicsiSetIntfParams()* API 设置。通过使用 *CyU3PMipicsiQueryIntfParams()* API 可以查询它。

位	15	14	13	12	11	10	9	8
名称	RESERVED				PLL FRS		RESERVED	
位	7	6	5	4	3	2	1	0
名称	RESERVED			CLOCK ENABLE	RESERVED		PLL ENABLE	

寄存器字段	位	说明
RESERVED	[15:12]	已保留。固件必须通过读取这些位，修改未保留位和重新写入这些位保留它们的设置。
PLL FRS	[11:10]	频率范围选择： 确定 PLL 的频率范围。有关该字段的详细内容请参考章节 1.7.2 。 2'b00: PLL 频率为 500 到 1000 MHz 2'b01: PLL 频率为 250 到 500 MHz 2'b10: PLL 频率为 125 到 250 MHz 2'b11: PLL 频率为 62.5 到 250 MHz
RESERVED	[9:5]	已保留。固件必须通过读取这些位，修改未保留位和重新写入这些位保留它们的设置。
CLOCK ENABLE	[4]	时钟使能： 0: MIPI CSI-2 模块上的时钟被禁用。 1: MIPI CSI-2 模块上的时钟被使能（正常运行）。
RESERVED	[3:2]	已保留。固件必须通过读取这些位，修改未保留位和重新写入这些位保留它们的设置。
PLL ENABLE	[1:0]	使能 PLL 时钟： 进行设置以使能 PLL 时钟。 2'b00: PLL 时钟被禁用 2'b11: PLL 时钟被使能（正常运行） 2b'01- 2b'10: 被保留

1.10.10 CX3_CLK_CTRL (寄存器地址: 0x0020)

该寄存器配置 MIPI CSI-2 模块的接口时钟分频器。

该寄存器由 `CyU3PMipicsiSetIntfParams()` API 设置。通过使用 `CyU3PMipicsiQueryIntfParams()` API 可以查询它。

位	15	14	13	12	11	10	9	8
名称	RESERVED							
位	7	6	5	4	3	2	1	0
名称	RESERVED		CSI RX CLK DIV		MCLK REF DIV		PAR OUT CLK DIV	

寄存器字段	位	说明
RESERVED	[15:6]	已保留。固件必须通过读取这些位，修改未保留位和重新写入这些位保留它们的设置。
CSI RX CLK DIV	[5:4]	CSI RX LP $\leftrightarrow$ HS 切换时钟的时钟分频器： 对 PLL 时钟进行分频可得到该时钟。更多信息，请参见 1.7.3 章节。 2'b00: PLL CLOCK/8 2'b01: PLL CLOCK/4 2'b10: PLL CLOCK/2 2'b00: 被保留 该频率必须位于 66 到 125 MHz 的范围内。
MCLK REF DIV	[3:2]	MCLK 参考时钟的时钟分频器： 对 PLL 时钟进行分频可以生成 MCLK 参考时钟。更多信息，请参见 1.7.5 章节。 2'b00: PLL CLOCK/8 2'b01: PLL CLOCK/4 2'b10: PLL CLOCK/2 2'b00: 保留 该频率必须小于 125 MHz。
PAR OUT CLK DIV	[1:0]	并行输出时钟 (PCLK) 的时钟分频器： 对 PLL 时钟进行分频可得到并行输出时钟 (该时钟用于驱动 GPIF II 接口)。 更多信息，请参见 1.7.4 章节。 2'b00: PLL CLOCK/8 2'b01: PLL CLOCK/4 2'b10: PLL CLOCK/2 2'b00: 保留 此频率不能超过 100 MHz。

1.10.11 CX3_BYTE_COUNT（寄存器地址：0x0022）

该寄存器为 MIPI CSI-2 模块上每一个有效线配置了字节计数。

该寄存器由 *CyU3PMipicsiSetIntfParams()* API 设置。通过使用 *CyU3PMipicsiQueryIntfParams()* API 可以查询它。

位	15	14	13	12	11	10	9	8
名称	BYTE COUNT [15:8]							
TYPE	R/W							
位	7	6	5	4	3	2	1	0
名称	BYTE COUNT [7:0]							
TYPE	R/W							

寄存器字段	位	说明
BYTE COUNT	[15:0]	每线的字节总数： 输入的每一线的字节总数。该值通过下面公式计算得出： 水平像素数量（有效像素）x 每个像素的字节数量。 例如，对于 1920x1080 RGB888 流，该值的计算情况如下： 每个通道的有效像素：1920 每个像素的字节数：3 BYTE COUNT = 1920x3 = 5760 = 0x1680 BYTE COUNT [15:8] = 0x16 BYTE COUNT [7:0] = 0x80

1.10.12 CX3_PHY_TIME_DELAY（寄存器地址：0x0060）

通过该寄存器可以配置 CX3 MIPI CSI-2 模块上的 MIPI CSI-2 接收器 PHY。该设置取决于 1.7.3 中所定义的 CSI_RX_CLK

该寄存器由 *CyU3PMipicsiSetIntfParams()* API 设置。通过使用 *CyU3PMipicsiQueryIntfParams()* API 可以查询它。

位	15	14	13	12	11	10	9	8
名称	TC TERM	RESERVED						
位	7	6	5	4	3	2	1	0
名称	TD TERM	THS SETTLE						

寄存器字段	位	说明
TC TERM	[15]	TC TERM 选择： 1：正常运行；0：不受支持。
RESERVED	[14:8]	已保留。固件必须通过读取这些位，修改未保留位和重新写入这些位保留它们的设置。
TD TERM	[7]	TD TERM 选择： 0：当从 LP 切换到 HS 时，经过 2 x CSI_Rx_Clk 或 3 x CSI_Rx_Clk 时间后，将发生数据通道 HS 终端。 1：从 LP 切换到 HS 时，将立即设置数据通道 HS 终端（优先选择）。
THS SETTLE	[6:0]	THS SETTLE 定时器： 用于控制从 LP 切换到 HS 的延迟。 范围：0x00 - 0x7F。 延迟 = (THS SETTLE + 1) x CSI_RX_CLK

1.11 CX3 MIPI CSI-2 APIs

本章节详细说明了 [EZ USB FX3 SDK](#) 所提供的 API，用于配置和使用 CX3 上的 MIPI CSI-2 模块。欲了解 API 以及它们所使用的枚举和结构类型，请参考 [EZ USB FX3 SDK](#) 中的 EZ-USB FX3/FX3S/CX3 SDK 固件 API 指南。

在 `cyu3mipicsi.a` 库文件中提供了 MIPI CSI-2 模块配置 API；在头文件 `cyu3mipicsi.h` 中提供了功能声明、枚举以及结构类型。

[EZ USB FX3 SDK 1.3](#) 为 CX3 器件支持 BETA 级别。CX3 器件的完整支持仅受 [EZ-USB FX3 SDK 1.3.1](#) 版本（和更高版本）中的 [EZ-USB FX3 SDK](#) 的支持。

1.11.1 CyU3PMipicsiInit()

CyU3PReturnStatus_t CyU3PMipicsiInit (void)

通过该函数可以初始化 CX3 器件上的 MIPI CSI-2 模块，在对 MIPI CSI-2 其他模块进行调用前，要先调用该函数。在调用此函数前，应该初始化 CX3 GPIO 模块、PIB 模块以及 I²C 模块。

这样会使 MIPI CSI-2 模块进入低功耗模式，并且需要调用 [CyU3PMipicsiWakeup\(\)](#) 函数，以使能 MIPI CSI-2 模块上的时钟。

此函数为 MIPI CSI-2 模块提供了内部电源循环。

另外，该函数还会通过将寄存器的 **XSHUTDOWN ENABLE** 和 **XRESET ENABLE** 位设置为 0 来使能 XRESET 和 XSHUTDOWN 信号的输出。

该函数会使 **CX3_CSI_SENSOR_SIG_VAL** 寄存器保持其默认状态（0x0000），从而将 MIPI XRESET 和 XSHUTDOWN 信号设置为驱动低电平。如果需要将这两个信号中的某一个驱动为高电平状态，以执行传感器操作，那么在调用此函数后，必须使用 [CyU3PMipicsiSetSensorControl\(\)](#) API 函数将该信号设置为驱动高电平。

成功执行此函数后将对应调用函数返回 CY_U3P_SUCCESS 值。

1.11.2 CyU3PMipicsiDeInit()

CyU3PReturnStatus_t CyU3PMipicsiDeInit (void)

此函数取消初始化 CX3 器件上的 MIPI-CSI 接口模块，在调用 [CyU3PSysEnterStandbyMode\(\)](#) 函数前，应先调用此函数。取消初始化 I²C 模块或 GPIO 模块前，应该调用此函数。

进行该调用后，CX3 将不再驱动 MIPI XRESET 和 XSHUTDOWN 信号。

成功执行此函数后将针对调用函数返回 CY_U3P_SUCCESS 值。

1.11.3 CyU3PMipicsiReset()

CyU3PReturnStatus_t CyU3PMipicsiReset (CyU3PMipicsiReset_t resetType)

该函数用于复位 CX3 上的 MIPI-CSI 模块。MIPI-CSI 模块提供了两种复位模式 — **硬复位**（为整个模块提供电源循环）和**软复位**（不复位该模块使用的 I²C 通信通道）。通过传送到该函数的 `CyU3PMipicsiReset_t` 类参数可以选择复位模式。

初始化该模块前，在接口上不可调用软复位。但硬复位可随时被调用。

在初始化该模块前，[CyU3PMipicsiInit\(\)](#) 调用在接口上内部执行硬复位。

硬复位是通过将寄存器中的 **XSHUTDOWN ENABLE** 和 **XRESET ENABLE** 位置为 0，以使能 XRESET 和 XSHUTDOWN 信号的输出。该函数使 **CX3_CSI_SENSOR_SIG_VAL** 寄存器保持它的默认状态（0x0000），从而将 MIPI XRESET 和 XSHUTDOWN 信号设置为驱动 0。如果这两个信号中的某一个需要保持驱动 1 状态，

以用于传感器操作，那么在调用 `CyU3PMipicsiReset()`(`CY_U3P_CSI_HARD_RST`)之后，必须使用 `CyU3PMipicsiSetSensorControl()` 函数将该信号设置为驱动 1 状态。

软复位不会改变 XRESET 和 XSHUTDOWN 信号的状态。软复位会将 `CX3_SYSTEM_CTRL` 寄存器的 **RESET** 位设置为 1，等待 CX3 上 5 个定时器计后，才会将 **RESET** 位清除为 0，以恢复正常操作。

成功执行此函数后将针对调用函数返回 `CY_U3P_SUCCESS` 值。

1.11.4 CyU3PCx3DeviceReset()

void CyU3PCx3DeviceReset (CyBool_t isWarmReset, CyBool_t sensorResetHigh)

该函数与 FX3/FX3S 器件的 `CyU3PDeviceReset ()` API 函数相同。它使用 `isWarmReset` 参数来确定器件当前进行的是热复位还是冷复位。

除了复位 CX3 器件外，该函数还会通过内部调用 `CyU3PMipicsiSetSensorControl()` 并根据 `sensorResetHigh` 中所提供的值将 `CY_U3P_CSI_IO_XRES` 信号设置为高电平或低电平来驱动 MIPI XRESET 信号，以便在复位 CX3 器件前已经驱动了 MIPI XRESET 信号，从而复位图像传感器。

此函数不被返回。

1.11.5 CyU3PMipicsiSleep()

CyU3PReturnStatus_t CyU3PMipicsiSleep (void)

通过该函数可以禁用 MIPI CSI-2 模块上的 PLL 时钟，并使它进入低功耗睡眠模式。当模块处于低功耗睡眠模式，图像传感器和 CX3 器件之间不进行数据传输。

此 API 会将 MIPI CSI-2 模块中寄存器的 **SLEEP** 位设置为 1。

将模块成功设置为低功耗睡眠模式后，此函数将被返回 `CY_U3P_SUCCESS`。

1.11.6 CyU3PMipicsiWakeup()

CyU3PReturnStatus_t CyU3PMipicsiWakeup (void)

通过该函数可以使能 MIPI CSI-2 模块上的时钟，以使它从低功耗睡眠模式进入活动模式。

此 API 将 MIPI CSI-2 模块中 `CX3_SYSTEM_CTRL` 寄存器的 **SLEEP** 位清除为 0。

成功执行此函数后将返回 `CY_U3P_SUCCESS`。

1.11.7 CyU3PMipicsiSetSensorControl()

CyU3PReturnStatus_t CyU3PMipicsiSetSensorControl (CyU3PMipicsiSensorIo_t io, CyBool_t value)

通过该函数可以将 XRES 和 XSHUTDOWN 信号从 CX3 驱动到图像传感器内，并且可以将这些信号驱动为高电平或低电平。

此函数可以驱动两个信号中的一个或同时驱动两个信号（驱动为相同的值）。要想将两个信号设置为相同的值，就要将掩码(`CY_U3P_CSI_IO_XRES | CY_U3P_CSI_IO_XSHUTDOWN`)作为 `io` 的值使用。要想将这两个信号设置为独立的值，则需要多次调用此函数（每个信号调用一次）。

此 API 函数通过将 `CX3_CSI_SENSOR_SIG_VAL` 中合适位设置为 1 或 0 来将相应的 MIPI 信号驱动为传感器高电平或低电平。

要想将 XRESET 信号驱动为高电平或低电平，需要将 **XRESET OUTPUT** 位置 1 或清除为 0。同样，要想将 XSHUTDOWN 信号驱动为高电平或低电平，则需要将 **XSHUTDOWN OUTPUT** 位置 1 或清除为 0。

成功执行此函数后将返回 `CY_U3P_SUCCESS`。

1.11.8 CyU3PMipicsiCheckBlockActive()

CyBool_t CyU3PMipicsiCheckBlockActive (void)

通过此函数可以检查 MIPI CSI-2 模块处于活动模式还是低功耗睡眠模式。

如果该模块有效，它将返回 CyTrue；如果该模块无效，则返回 CyFalse。

此函数仅检查一个标志（调用 [CyU3PMipicsiWakeup\(\)](#) 时设置该标志）而不会检查实际的接口寄存器。如果您直接修改了配置寄存器，请检查 [CX3_SYSTEM_CTRL](#) 寄存器的 SLEEP 位的实际值，而不是使用 API 函数。

1.11.9 CyU3PMipicsiSetIntfParams()

CyU3PReturnStatus_t CyU3PMipicsiSetIntfParams (CyU3PMipicsiCfg_t * csiCfg, CyBool_t wakeOnConfigure)

使用此函数可以通过 I²C 接口配置 MIPI CSI-2 模块参数。

此函数采用 [CyU3PMipicsiCfg_t](#) 类的对象并配置 MIPI CSI-2 模块。对接口配置寄存器进行任何修改前，此函数都会通过调用 [CyU3PMipicsiSleep\(\)](#) 来关闭接口时钟。

此函数使用了 [CyU3PMipicsiCfg_t](#) 结构中的值来设置以下各 MIPI CSI-2 模块寄存器，如表 6 中所示。

表 6：用于配置 MIPI CSI-2 模块寄存器的参数

CX3 MIPI CSI-2 模块寄存器	用于配置目的的（各）CyU3PMipicsiCfg_t 参数
CX3_PLL_CTRL0	<i>pllPrd</i> 和 <i>pllFbd</i> 参数
CX3_PLL_CTRL1	<i>pllFrs</i> 参数
CX3_CLK_CTRL （寄存器地址：0x0020）	<i>csiRxClkDiv</i> 、 <i>parClkDiv</i> 和 <i>mClkRefDiv</i> 参数
CX3_MCLK_CTRL	<i>mClkCtl</i> 参数
CX3_BYTE_COUNT （寄存器地址：0x0022）	通过使用 <i>hResolution</i> 和 <i>dataFormat</i> 参数计算得出
CX3_PHY_TIME_DELAY	<i>thsSettle</i> 参数仅在 FX3 SDK 1.3.3 版本中提供
CX3_DATA_FMT	<i>dataFormat</i> 参数
CX3_CONFIG_CTRL	<i>dataFormat</i> 和 <i>numDataLanes</i> 参数

在完成配置或关闭各时钟后，使用参数 *wakeOnConfigure* 可以立刻启动这些时钟。如果这些时钟保持关闭状态，则应该通过调用 [CyU3PMipicsiWakeup\(\)](#) 函数来启动它们。

成功执行此函数后将返回 CY_U3P_SUCCESS。

CX3 配置生成工具属于 EZ-USB 套件（基于 IDE 的 Eclipse）的部分，该套件与 EZ-USB SDK 一起被提供。使用该工具可以生成 [CyU3PMipicsiCfg_t](#) 结构元素，用于根据图像传感器输入和流参数配置 MIPI CSI-2 模块。

在 EZ-USB 套件的帮助菜单和 EZ-USB FX3 SDK 安装路径中文档文件夹内的赛普拉斯 EZ-USB FX3 快速入门指南（[FX3 SDK.pdf](#) 入门）中提供了该配置工具的详细使用说明。

1.11.10 CyU3PMipicsiQueryIntfParams()

CyU3PReturnStatus_t CyU3PMipicsiQueryIntfParams (CyU3PMipicsiCfg_t * csiCfg)

此函数用于回读模块中的 MIPI-CSI 接口参数。通过使用传送到调用函数的 [CyU3PMipicsiCfg_t](#) 类指针，可以将这些参数回读提供给此调用函数。此函数读取由 [CyU3PMipicsiSetIntfParams\(\)](#) 写入的寄存器。在传送到此 API 前，应该初始化由 *csiCfg* 指定的 [CyU3PMipicsiCfg_t](#) 结构对象。

1.11.11 CyU3PMipicsiGetErrors()

**CyU3PReturnStatus_t CyU3PMipicsiGetErrors (CyBool_t clrErrCnts,
CyU3PMipicsiErrorCounts_t * errorCounts)**

通过此函数可以获取 MIPI CSI-2 模块中的 CSI-2 协议和物理层错误。

此函数采用了一个函数，通过该函数可以确定接口上是否清除了错误计数。

通过传送到此函数的 *CyU3PMipicsiErrorCounts_t* 类指针可以检索每个错误类的错误计数。每一类的最大错误计数值可达 0xFF。如果此函数未使用 *clrErrCnts* 清除这些计数，则该计数值将继续记录每次调用的现有错误值。

在传送到此函数前，*errorCounts* 必须被初始化。

1.11.12 CyU3PMipicsiGpifLoad()

**CyU3PReturnStatus_t CyU3PMipicsiGpifLoad (CyU3PMipicsiBusWidth_t busWidth, uint32_t
bufferSize)**

如章节 1.9 中所述，CX3 拥有一个功能固定的 GPIF 接口，用于 MIPI CSI-2 模块中的图像传感器数据采集。

通过此函数可以选择 GPIF 数据总线宽度和 DMA 缓冲区大小的配置情况，以用于 GPIF 传输。在调用此函数之前，应该初始化 PIB 模块。

DMA 缓冲区大小必须是总线宽度的整数倍。例如，如果 GPIF 总线宽度为 24 位宽度，那么 DMA 缓冲区必须为 24 位的整数倍（即 3 个字节）。

另外，DMA 缓冲区大小必须为 16 字节的整数倍，以满足 *CyU3PDmaChannelCreate* API 的要求。

将要传送到该 API 的总线宽度和 DMA 缓冲区大小取决于数据传输的数据格式宽度。

例如，RGB 888 数据类型要求总线宽度为 24 位，YUV 422 要求总线宽度为 16 位，RAW 8 格式要求总线宽度为 8 位。

如果通过 API 函数配置的总线宽度小于选定的数据格式需要的值，则将从 MIPI CSI-2 模块传输的高数据位将被丢失。如果此 API 函数配置的总线宽度大于选定的数据格式所需要的值，则在接收到的数据内未使用的高位将被设置为 0。

1.12 其他参考资料

更多有关 CX3 的信息（包括数据手册和应用笔记）可以在 <http://www.cypress.com/cx3/> 中找到。

更多有关 FX3 的信息，包括数据手册、应用笔记和 EZ-USB FX3 技术参考手册（TRM）可以在 <http://www.cypress.com/fx3/> 中找到。

作为 SDK 部分所提供的 EZ-USB FX3 SDK 软件加载和文档（程序员手册、固件 API 指南和快速入门指南，等）可以在 <http://www.cypress.com/?rID=57990> 中找到。

有关 MIPI CSI-2 规范的详细信息可以在 MIPI 联盟网站 <http://www.mipi.org/specifications/camera-interface> 上找到。

1.13 术语表

API	应用编程接口。一系列的软件程序，包括一个应用与低层服务和函数之间的接口（例如，用于配置和控制 FX3/CX3 器件操作的接口）。 EZ-USB FX3 SDK 提供了一组库，这些库可用于配置和控制 FX3 器件操作的各种 API 函数。
Bayer 滤波器	一种滤色器阵列，通过它可以在光电传感器的正方形网格上布置 RGB 滤色器。滤波格式为 50% 的绿色、25% 的红色和 25% 的蓝色。
CPU 子系统	EZ-USB FX3/CX3 器件拥有一个 32 位的嵌入式 ARM926EJ-S 内核，此内核可提供 200 MIPS 的处理功能。此 ARM 内核与指令和数据缓存、紧密连接存储器（TCM）和 PL192 矢量中断控制器（VIC）相耦合。更多有关信息可以在 EZ-USB FX3 TRM 的 FX3 CPU 子系统章节中找到。
数据通道	用于数据传输的单向 MIPI CSI-2 差分串行接口。每个通道由两个信号组成：Data+ 和 Data-。CX3 可支持多达四个数据通道，每个通道的速度可达 1 千兆/秒。
DMA	直接存储器访问（DMA）子系统执行存储器和外设之间的高带宽数据传输并不需要 CPU 的干预。FX3/CX3 架构包含了一个 DMA 结构，通过它可以控制各种外设接口和系统存储器之间进行的数据传输。更多有关 DMA 子系统的信息可以在 EZ-USB FX3 TRM 的 FX3 DMA 子系统章节中找到。
全局控制器	FX3/CX3 的全局控制器（GCTL）单元包含了一个时钟生成模块、多功能 I/O、电源控制模块和一个复位模块。时钟生成模块为所有器件外设提供了时钟。更多有关信息可以在 EZ-USB FX3 TRM 的 FX3 全局控制器（GCTL）章节中找到。
GPIO II	CX3 GPIO II 接口是 FX3 中可编程接口的通用固定功能的实施。通过 CX3 GPIO II 接口可以传输 MIPI CSI-2 接收器模块中的数据。更多有关 CX3 的固定功能 GPIO II 状态机的信息可以在本技术参考手册补充材料的 1.9 章节内找到。更多有关信息可以在 EZ-USB FX3 TRM 的通用可编程接口 II（GPIO II）章节中找到。
I ² C	飞利浦半导体制定的 I ² C 总线协议，是指“内部集成电路总线”。它允许通过使用两根导线在各个 I ² C 功能器件间进行数据通信。它通过其中一根数据（SDA）线和一个时钟（SCL）信号连续发送信息。可将 FX3/CX3 作为 I ² C 主控制器使用，以支持 100 kHz、400 kHz 和 1 MHz 的操作。更多信息，可在 EZ-USB FX3 TRM 中低带宽外设章节内的 I ² C 接口章节中找到。
JTAG	联合测试行动小组（JTAG）是 IEEE 1149.1 标准测试访问端口和边界扫描结构的普通名称。FX3/CX3 的 JTAG 接口包含一个标准的 5 引脚接口，用于连接 JTAG 调试器。该调试器可通过片上调试电路来调试固件。ARM926EJ-S 内核的这些业界标准调试工具，可用于调试 FX3/CX3 器件。
低带宽外设	FX3/CX3 的串行外设 I ² C、SPI、I ² S 和 UART，包括 GPIO 形成低带宽外设（LBP）。更多有关信息，可以在 EZ-USB FX3 TRM 中的低带宽外设章节内找到。
MCLK	MCLK 是 CX3 的一个可选时钟输出，它可作为图像传感器的输入参考时钟使用。更多有关信息可以在本技术参考手册补充材料的 1.7.5 章节内找到。
存储器子系统	CX3 存储器子系统包含了可作为程序和数据存储器使用的系统 RAM、SRAM 控制器以及一个基于先进高性能总线（AHB）的互连（该互连允许 ARM CPU 和硬件模块访问这些存储器）。通过存储器映射 I/O（MMIO）互连可以对各种外设模块中的寄存器进行访问。更多有关信息可以在 EZ-USB FX3 TRM 的存储器和系统互连章节中找到。
MIPI CSI-2	摄像头串行接口 2（CSI-2）的移动工业处理器接口（MIPI）联盟标准是 MIPI 联盟所定义的串行摄像头接口。它定义了摄像头发送器和接收器间标准的数据传送和控制接口。数据传送接口是一个带有数据及时钟信号的单向差分串行接口。有关 MIPI CSI-2 规范的详细信息可以在 MIPI 联盟网站 http://www.mipi.org/specifications/camera-interface 上找到。
MJPEG 格式	Motion-JPEG 或 MJPEG 是一种视频格式，它通过使用 JPEG 静止图像压缩算法分别对每个视频帧进行解压。这些帧的序列表示的是源视频。
像素	一个像素表示的是屏幕上一个图像的最小可控制元素。每个像素都是原始图像的一个采样；一般情况下，采样越多，则重现的原始图像越准确。像素数量通常定义的是图像传感器的分辨率。
RAW 格式	RAW 格式数据提供了图像传感器中的直接、未处理输出。CX3 MIPI CSI-2 接收器接口支持 RAW8、RAW10、RAW12 及 RAW14 格式的数据，其中数值表示每个图像传感器像素的数据位数。通过滤色器阵列（如 Bayer 滤波器）可以获取 RAW 格式输出中的 RGB 颜色信息。

RGB 格式	它是一种颜色格式，通过使用红色、绿色和蓝色（R、G、B）的成分术语来定义颜色空间。CX3 MIPI CSI-2 接收器接口支持 RGB888、RGB666 和 RGB565 流，在这里数值表示的是某个数据流中各颜色成分的位数量（例如，RGB 565 拥有五位红数据、六位绿数据和五位蓝数据）。
保留位	用户不用修改 RESERVED 寄存器位。 将 RESERVED 位写入到某个寄存器时，需要确保它们的值在写操作期间不会发生改变（首选方法是在进行写操作前先进行读操作，并且在写操作期间保持 RESERVED 位上的值）。
SDK（软件开发套件）	EZ-USB FX3 SDK（软件开发套件）是一个功能齐全的固件方案，通过它，客户可以使用 FX3/FX3S/CX3 器件的各种特性，并且可以在较短时间内编译出一个自定义的设计。SDK 提供一个固件框架，该框架由一组驱动器和方便易用的 API 组成并且适用于这些器件上的各种硬件模块。各驱动器概括了 FX3 器件架构的复杂性并且为客户提供了易于使用的 API，用于配置和控制 FX3 器件操作。
套接字	DMA 套接字是一个 FX3 器件结构，它被映射到器件的输入或输出路径的一端。更多有关信息可以在 EZ-USB FX3 TRM 的 FX3 DMA 子系统章节中找到。
USB	更多有关 FX3/CX3 的 USB 子系统的信息，可以在 EZ-USB FX3 TRM 中的 USB 章节内找到。CX3 器件仅作为 USB 外设器件使用，FX3 所支持的 USB 主机、OTG 和充电器的特性不受该器件的支持。
YUV 格式	它是一种颜色格式，它使用亮度（Y）和色度（UV）成分术语来定义颜色空间。CX3 MIPI CSI-2 接收器接口支持 YUV422 8 和 10 位颜色格式。
XRESET	CX3 是一个用于复位图像传感器的 CX3 输出信号。
XSHUTDOWN	CX3 是一个用于控制图像传感器功耗模式的输出信号。

修订记录



文档修订记录

文档标题: EZ-USB® CX3 技术参考手册

文档编号: 001-92471

版本	ECN 编号	变更人	修订说明
**	4377176	LISA	本文档版本号为 Rev**, 译自英文版 001-91492 Rev**。

PSoC 是赛普拉斯半导体公司的注册商标。此处引用的所有其他商标或注册商标归其各自所有者所有。



赛普拉斯半导体公司
198 Champion Court
San Jose, CA 95134-1709
电话 : 408-943-2600
传真 : 408-943-4730
网站 : www.cypress.com

©赛普拉斯半导体公司, 2014。此处所包含的信息可能会随时更改, 恕不另行通知。除赛普拉斯产品内嵌的电路外, 赛普拉斯半导体公司不对任何其他电路的使用承担任何责任。也不会根据专利权或其他权利以明示或暗示的方式授予任何许可。除非与赛普拉斯签订明确的书面协议, 否则赛普拉斯产品不保证能够用于或适用于医疗、生命支持、救生、关键控制或安全应用领域。此外, 对于合理预计会发生运行异常和故障并对用户造成严重伤害的生命支持系统, 赛普拉斯将不批准将其产品用作此类系统的关键组件。若将赛普拉斯产品用于生命支持系统, 则表示制造商将承担因此类使用而招致的所有风险, 并确保赛普拉斯免于因此而受到任何指控。

该源代码(软件和/或固件)均归赛普拉斯半导体公司(赛普拉斯)所有, 并受全球专利法规(美国和美国以外的专利法规)、美国版权法以及国际条约规定的保护和约束。赛普拉斯据此向获许可者授予适用于个人的、非独占性、不可转让的许可, 用以复制、使用、修改、创建赛普拉斯源代码的派生作品、编译赛普拉斯源代码和派生作品, 并且其目的只能是创建自定义软件和/或固件, 以支持获许可者仅将其获得的产品依照适用协议规定的方式与赛普拉斯集成电路配合使用。除上述指定用途外, 未经赛普拉斯的明确书面许可, 不得对此类源代码进行任何复制、修改、转换、编译或演示。

免责声明: 赛普拉斯不针对该材料提供任何类型的明示或暗示保证, 包括(但不限于)针对特定用途的适销性和适用性的暗示保证。赛普拉斯保留在不另行通知的情况下对此处所述材料进行更改的权利。赛普拉斯不对此处所述之任何产品或电路的应用或使用承担任何责任。对于合理预计可能发生运转异常和故障, 并对用户造成严重伤害的生命支持系统, 赛普拉斯不授权将其产品用作此类系统的关键组件。若将赛普拉斯产品用于生命支持系统, 则表示制造商将承担因此类使用而导致的所有风险, 并确保赛普拉斯免于因此而受到任何指控。

产品使用可能受适用于赛普拉斯软件许可协议的限制。