Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

THIS SPEC IS OBSOLETE

Spec No: 001-58069

Spec Title: AN58069 - IMPLEMENTING AN 8-BIT PARALLEL MPEG2-TS INTERFACE USING SLAVE FIFO MODE IN FX2LP

Replaced by: NONE

AN58069

Implementing an 8-Bit Parallel MPEG2-TS Interface Using Slave FIFO Mode in FX2LP

Author: Praveen Kumar Murugesan
Associated Project: Yes
Associated Part Family: CY7C68013/14/15/16
Related Application Notes: AN66806, AN15456, AN63787

More code examples? We heard you.

To access a variety of FX2LP code examples, please visit our USB High-Speed Code Examples webpage.

Are you looking for USB 3.0 peripheral controllers?

To access USB 3.0 product family, please visit our USB 3.0 Product Family webpage.

This application note explains how to implement an 8-bit parallel MPEG2-TS interface using the Slave FIFO mode. The example code uses EZ-USB FX2LP™ (CY7C68013/14/15/16) at the receiver end and a data generator as the source for the data stream. The hardware connections and example code are included with this application note. In addition, this application note describes a design example that uses this interface.

1 Introduction

The EZ-USB FX2LP is an excellent solution if you want to a high-performance high-speed USB to a design. In applications similar to a TV dongle, an MPEG2-TS to USB Bridge is critical. The EZ-USB FX2LP not only takes care of glueless logic, but also makes it easier for the designer to complete the design. This application note addresses the hardware connections and example firmware required to implement the MPEG2-TS interface using the Slave FIFO mode.

Note: Familiarity with the examples and documentation on the EZ-USB FX2LP development kit and Chapter 9 (Slave FIFOs) of the EZ-USB FX2LP Technical Reference Manual is useful in understanding this application. For complete list of FX2LP application notes, click here. For complete list of USB High-Speed Code Examples, click here. Refer to this link for TV Dongle Reference Design using FX2LP.

2 MPEG2-TS Interface

MPEG2-TS interface is now adopted as the standard encoding and delivery interface for most of the compressed digital video broadcasting technologies. MPEG2-TS interface was designed for error-prone links that do not offer support to carry structured data. It uses packets of small size and provides many features for data link layers such as packet identification (PID), synchronization (Sync Byte), timing (clock references and timestamps), multiplexing, and sequencing information (CC).

The MPEG2-TS packet is a 188-byte packet, consisting of a 4-byte header. The remaining 184 bytes are used for payload as shown in Figure 1. The small packet length is suitable for high error mediums because the error then affects less data, although the header represents a higher overhead for such a packet size.

Figure 1. MPEG2-TS Packet

Header (4 bytes)	Optional variable length adaptation field	Payload (184 bytes)

Figure 2. MPEG2-TS Header

Sync Byte	PID	CC

The header, shown in Figure 2, contains a sync byte used for random access to the stream. It also contains a program ID (PID), which allows identification of all packets belonging to the same data stream. Alternatively, it provides a means to multiplex data streams within transport streams. Finally, the Continuity Counter field (CC) provides a mechanism to detect missing packets by incrementing each packet belonging to the same PID by one.

2.1 MPEG2-TS Interface Signals

- MPEG_CLK: This CLK provides the reference for the parallel data stream.
- MPEG_D[7:0]: This is the 8-bit parallel data of the MPEG2-TS.
- MPEG_Valid: This signal is asserted when the data in the MPEG_D [7:0] is valid.
- MPEG_Sync: This signal is asserted for every first byte of the 188 byte packet.

2.2 FX2LP Slave FIFO Interface Signals

- IFCLK: This pin is not needed in asynchronous mode. The IFCLK pin can be configured as either an input (default) or an output interface clock.
- FD [15:0]: 16-bit data bus.
- FIFOADR [2:0]: These pins select which of the two FIFOs is connected to the FD [15:0] bus, or whether the command interface is selected.
- SLCS: Slave chip select.
- SLOE and SLRD: In synchronous mode, the FIFO pointer is incremented on each rising edge of IFCLK while SLRD is asserted. In asynchronous mode, the FIFO pointer is incremented on each asserted-to-deasserted transition of SLRD. SLOE is a data bus driver enable. When SLOE is asserted, the FX2LP firmware drives the data bus.
- FLAGA/B: FLAGx pins report the status of the FIFO. They can be configured to be Full Flag, Empty Flag, or Programmable Flag.
- SLWR: In synchronous mode, data on the FD bus is written to the FIFO (and the FIFO pointer is incremented) on each rising edge of IFCLK while SLWR is asserted. In asynchronous mode, data on the FD bus is written to the FIFO (and the FIFO pointer is incremented) on each asserted-to-deasserted transition of SLWR.

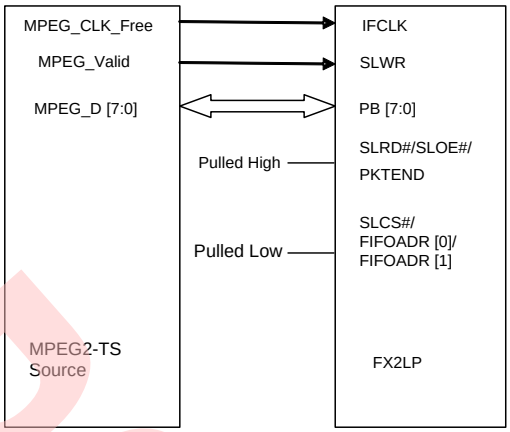
2.3 Hardware Interface Connection

Two types of connections can be made: free running and gated.

2.3.1 Free Running MPEG CLK

In this case, the MPEG_CLK is free running. The CLK is output from the source chip even when the data stream is not available. In this mode, MPEG_CLK from the source is connected to IFCLK in FX2LP. The interface is made to operate in synchronous mode. MPEG_Valid from the source is connected to SLWR in FX2LP. EP2 FIFO is used in this example, so FIFOADR [1:0] is hardwired to “00”. In case of any other end points, the corresponding value should be hardwired.

Figure 3. Hardware Interconnection

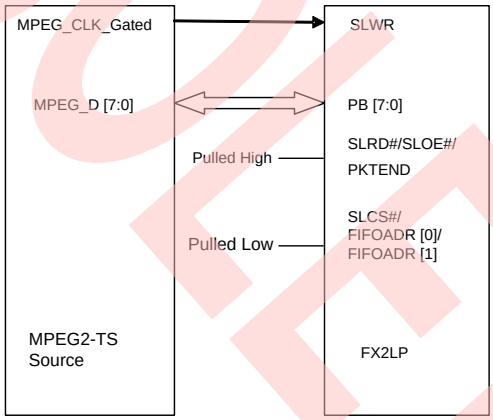


2.3.2 Gated MPEG CLK

If the data source has the option to gate the MPEG_CLK internally using the MPEG_Valid, then the following changes are needed.

There is no CLK needed for this mode. The interface can be made to operate in asynchronous mode. So, MPEG_CLK is connected to the SLWR signal directly.

Figure 4. Hardware Interconnection



2.4 Example Firmware

Example code is available with this application note to ease the firmware development effort.

Two configurations are available in the example code.

MPEG_Free: This configuration is for a free running-CLK mode.

```
IFCONFIG = 0x03;
IFCLK source is external from the MPEG2-TS source and is configured in Slave FIFO
mode.
(IFCONFIG = 0x13, to invert the polarity of the MPEG_CLK)
FIFOPINPOLAR = 0x04;
SLWR is set to be active high and the rest of the FIFO signals are in their default
active low condition.
(FIFOPINPOLAR =0x00; if SLWR is also required to be active low)
EP2CFG = 0xE0;
EP2 is configured as 4 X 512 byte buffer in BULK mode.
EP2FIFOCFG = 0x08;
EP2 FIFO is configured in 8-bit AUTOIN mode.
EP2AUTOINLENH = 0x02;
```

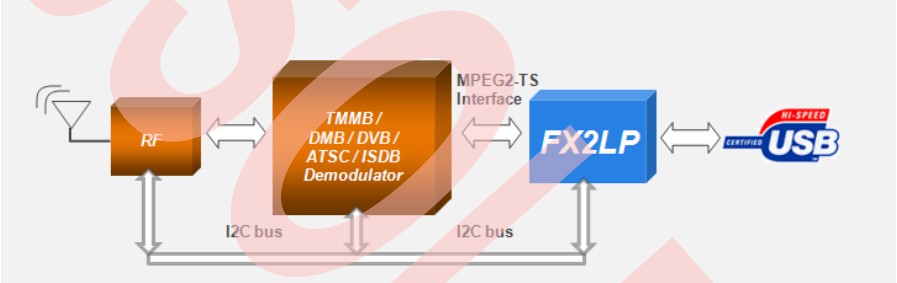
```
EP2AUTOINLENL = 0x00;
AUTO Commit every 512 bytes.
```

MPEG_Gated: This configuration is for gated CLK mode.

```
IFCLK = 0xCB;
FIFOs operate asynchronously without any CLK. Deassert SLWR to get the data into the
FIFO from the 8-bit data bus.
FIFOPINPOLAR = 0x04;
SLWR is set to be active high and the rest of the FIFO signals are in their default
active low condition.
(FIFPINPOLAR =0x00, for negative edge triggered transfer mode)
EP2CFG = 0xE0;
EP2 is configured as 4 X 512 byte buffer in BULK mode.
EP2FIFOCFG = 0x08;
EP2 FIFO is configured in 8-bit AUTOIN mode.
EP2AUTOINLENH = 0x02;
EP2AUTOINLENL = 0x00;
AUTO Commit every 512 bytes.
```

2.5 Design Example: TV Dongle

Figure 5. TV Dongle Block Diagram



MPEG2-TS to USB Bridges are critical in applications such as PC TV dongles. Figure 5 shows a TV dongle block diagram. The three main components are the tuner, demodulator, and MPEG2-TS to USB Bridge. Here, MPEG2-TS to USB bridge functionality is implemented using the FX2LP. The tuner present in the TV dongle tunes the RF signal to the specific demodulator connected to it. This sends the TV signals in the MPEG2-TS stream to FX2LP, which then sends them to the PC through the USB interface. In the PC, the BDA driver forwards the data from the USB to the media player application where it is played.

2.6 I²C Bus

All MPEG2 devices also require an I²C interface between the MPEG2-TS source and FX2LP (I2C bus connection can be observed in the TV dongle block diagram that is shown in Figure 5). This is used to configure the MPEG2-TS source chip from the drivers in the host PC through USB. FX2LP has a dedicated I²C engine. This engine can be used or two GPIO pins can be bit banged for the I²C operation.

2.7 BDA Driver Development

The media player application, which decodes the MPEG2-TS, requires a BDA driver in the host PC to play the MPEG2-TS stream. The BDA driver development is specific and customized to the MPEG2-TS source used in the design. This must be developed based on the source. Cypress will provide support, if required, during the development phase. File a case at the Cypress customer support portal CRM. A reference design, which uses a demodulator from Legend Silicon, is available at the Cypress website. The reference design package consists of all collaterals required for the design including the hardware Gerber files, firmware hex file, and BDA driver.

3 Summary

MPEG2-TS interface can be easily implemented using the slave FIFO mode of FX2LP. Implementation details and hardware connections are mentioned in this application note. Also a design example is described to explain you the exact usage of MPEG2-TS to USB Bridge.

About the Author

Name: Praveen Kumar Murugesan

Title: Applications Engineer Staff

Document History

Document Title: AN58069 - Implementing an 8-Bit Parallel MPEG2-TS Interface Using Slave FIFO Mode in FX2LP

Document Number: 001-58069

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	2822633	PRKU	12/07/2009	New Application Note.
*A	3174364	RSKV	02/15/2011	Updated MPEG2-TS Interface: Added Design Example: TV Dongle.
*B	3859815	RSKV	01/07/2013	Added Summary. Updated to new template.
*C	4160754	NIKL	10/15/2013	Updated to new template.
*D	5060320	NIKL	01/21/2016	Added AN66806, AN15456, AN63787 under Related Application Notes in page 1. Updated Introduction: Updated description. Updated to new template. Completing Sunset Review.
*E	5250213	NIKL	05/11/2016	Updated attached Associated Project. Updated to new template.
*F	5692286	AESATMP8	04/19/2017	Updated Cypress Logo and Copyright.
*G	6490309	HPPC	02/20/2019	Obsolete document. Completing Sunset Review.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

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