个人资料

windtakers

MIPI-CSI2

2016-10-18 16:33 353人阅读 评论(0) 收藏 举报

MIPI联盟，即移动产业处理器接口（Mobile Industry Processor Interface 简称MIPI）联盟。MIPI（移动产业处理器接口）是MIPI联盟发起的为移动应用处理器制定的开放标准和一个规范。

MIPI是做移动应用处理器的几家巨头公司成立的联盟，旨在定义移动应用处理器的接口标准，其全称为“

Mobile Industry Processor Interface”。现在用的比较多是MIPI框架中的摄像头标准和显示标准，

即MIPI CSI和MIPI DSI。CSI代表 Camera Serial Interface，而DSI代表Display Serial Interface。现在CSI

已经升级到CSI2.0版本，即MIPI CSI2接口。

4 Overview of CSI-2

The CSI-2 specification defines standard data transmission and control interfaces between transmitter and receiver. Data transmission interface (referred as CSI-2) is unidirectional differential serial interface with data and clock signals; the physical layer of this interface is the *MIPI Alliance Specification for D-PHY* [MIPI01]. Figure 1 illustrates connections between CSI-2 transmitter and receiver, which typically are a camera module and a receiver module, part of the mobile phone engine.

The control interface (referred as CCI) is a bi-directional control interface compatible with I2C stand

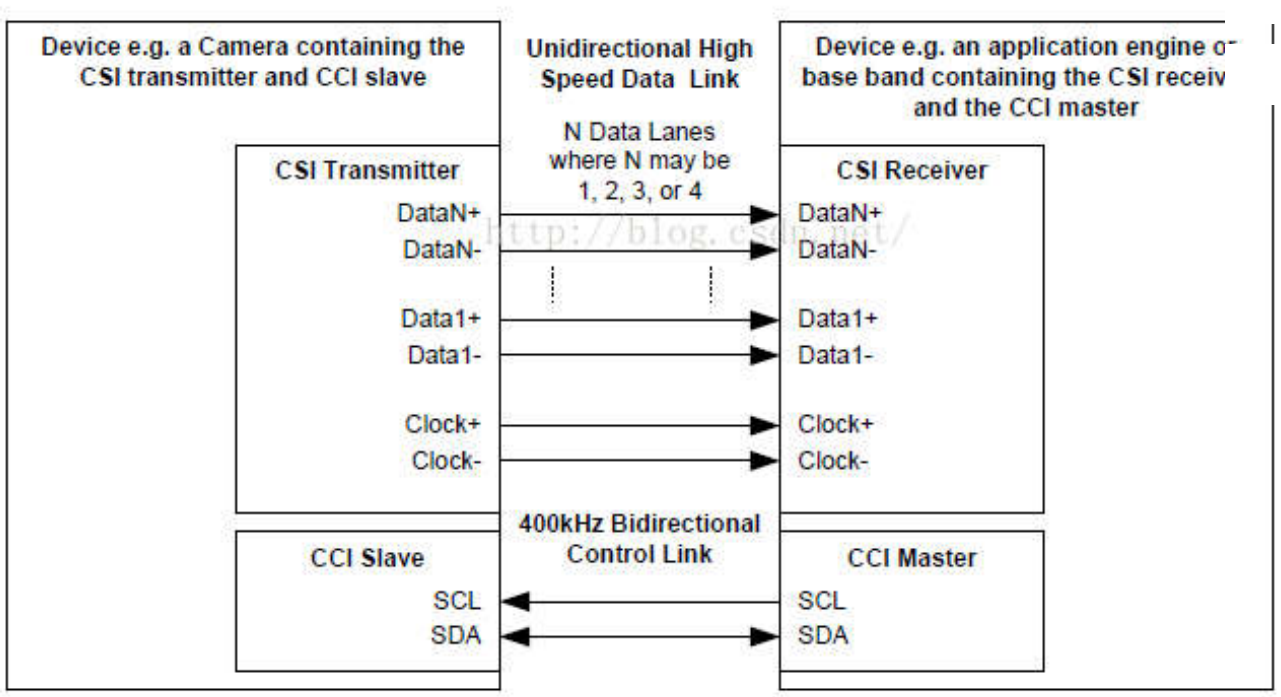

Figure 1 CSI-2 and CCI Transmitter and Receiver Interface

5 CSI-2 Layer Definitions

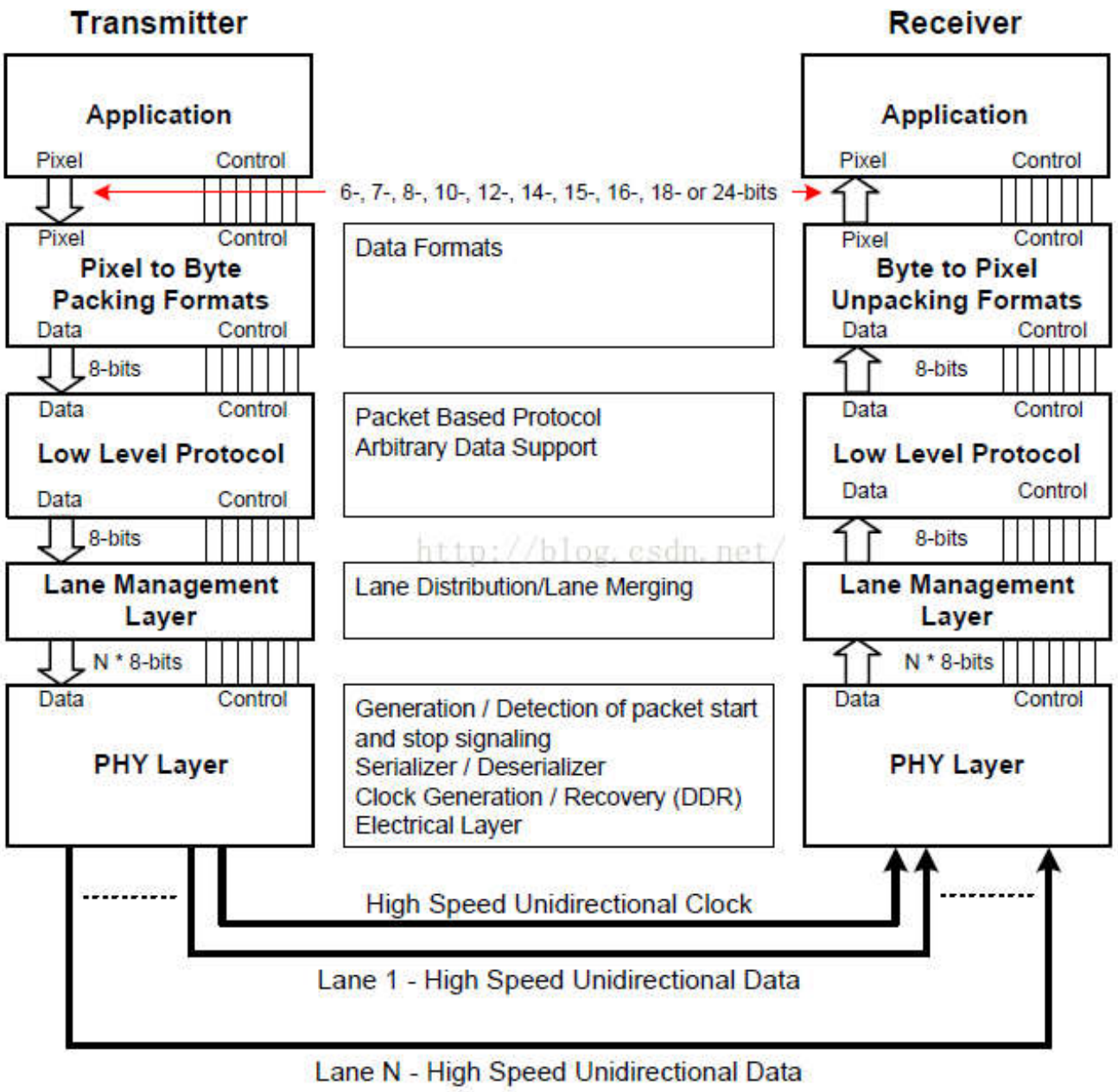

Figure 2 CSI-2 Layer Definitions

Figure 2 defines the conceptual layer structure used in CSI-2. The layers can be characterized as follows:

- **PHY Layer.** The PHY Layer specifies the transmission medium (electrical conductors), the input/output circuitry and the clocking mechanism that captures "ones" and "zeroes" from the serial bit stream. This part of the specification documents the characteristics of the transmission medium, electrical parameters for signaling and the timing relationship between clock and data Lanes.

  The mechanism for signaling Start of Transmission (SoT) and End of Transmission (EoT) is specified as well as other "out of band" information that can be conveyed between transmitting and receiving PHYs. Bit-level and byte-level synchronization mechanisms are included as part of the PHY.

The PHY layer is described in [MIPI01].

- **Protocol Layer.** The Protocol layer is composed of several layers, each with distinct responsibilities. The CSI-2 protocol enables multiple data streams using a single interface on the host processor. The Protocol layer specifies how multiple data streams may be tagged and interleaved so each data stream can be properly reconstructed.
  - **Pixel/Byte Packing/Unpacking Layer.** The CSI-2 supports image applications with varying pixel formats from six to twenty-four bits per pixels. In the transmitter this layer packs pixels from the Application layer into bytes before sending the data to the Low Level Protocol layer. In the receiver this layer unpacks bytes from the Low Level Protocol layer into pixels before sending the data to the Application layer. Eight bits per pixel data is transferred unchanged by this layer.
  - **Low Level Protocol.** The Low Level Protocol (LLP) includes the means of establishing bit-level and byte-level synchronization for serial data transferred between SoT (Start of Transmission) and EoT (End of Transmission) events and for passing data to the next layer. The minimum data granularity of the LLP is one byte. The LLP also includes assignment of bit-value interpretation within the byte, i.e. the "Endian" assignment.
  - **Lane Management.** CSI-2 is Lane-scalable for increased performance. The number of data Lanes may be one, two, three or four depending on the bandwidth requirements of the application. The transmitting side of the interface distributes ("distributor" function) the outgoing data stream to one or more Lanes. On the receiving side, the interface collects bytes from the Lanes and merges ("merger" function) them together into a recombined data stream that restores the original stream sequence.

  Data within the Protocol layer is organized as packets. The transmitting side of the interface appends header and optional error-checking information on to data to be transmitted at the Low Level Protocol layer. On the receiving side, the header is stripped off at the Low Level Protocol layer and interpreted by corresponding logic in the receiver. Error-checking information may be used to test the integrity of incoming data.
  - **Application Layer.** This layer describes higher-level encoding and interpretation of data contained in the data stream. The CSI-2 specification describes the mapping of pixel values to bytes.

The normative sections of the specification only relate to the external part of the Link, e.g. the data and bit patterns that are transferred across the Link. All internal interfaces and layers are purely informative.

D-PHY/M-PHY/C-PHY :

Physical Layer Specifications

The Phy Working Group has developed three specifications for high-speed physical layer designs to support multiple application requirements. The first specification — D-PHY — was developed primarily to support camera and display applications. The second specification — M-PHY® — supports a much broader range of applications, including interfaces for display, camera, audio, video, memory, power management and communication between Baseband to RFIC. While the first two specifications use two-wire interfaces and differential signaling, the third specification — C-PHY — uses 3 phase encoding on a three-wire interface to camera and display applications.

http://mipi.org/specifications/physical-layer#D-PHY Specification

http://diyprojector.info/forum/index.php?app=core&module=attach§ion=attach&attach_id=27737

7 Physical Layer

The CSI-2 uses the [MIPI01] physical layer.

The physical layer for a CSI-2 implementation is composed of between one and four unidirectional data Lanes and one clock Lane. All CSI-2 transmitters and receivers shall support continuous clock behavior on the Clock Lane, and optionally may support non-continuous clock behavior.

For continuous clock behavior the Clock Lane remains in high-speed mode generating active clock signals between the transmission of data packets.

For non-continuous clock behavior the Clock Lane enters the LP-11 state between the transmission of data packets.

The minimum physical layer requirement for a CSI-2 transmitter is

- Data Lane Module: Unidirectional master, HS-TX, LP-TX and a CIL-MFEN function
- Clock Lane Module: Unidirectional master, HS-TX, LP-TX and a CIL-MCNN function

The minimum physical layer requirement for a CSI-2 receiver is

- Data Lane Module: Unidirectional slave, HS-RX, LP-RX, and a CIL-SFEN function
- Clock Lane Module: Unidirectional slave, HS-RX, LP-RX, and a CIL-SCNN function

All CSI-2 implementations shall support forward escape ULPS on all Data Lanes.

9 Low Level Protocol

The Low Level Protocol (LLP) is a byte orientated, packet based protocol that supports the transport of arbitrary data using Short and Long packet formats. For simplicity, all examples in this section are single Lane configurations.

Low Level Protocol Features:

- Transport of arbitrary data (Payload independent)
- 8-bit word size
- Support for up to four interleaved virtual channels on the same link
- Special packets for frame start, frame end, line start and line end information
- Descriptor for the type, pixel depth and format of the Application Specific Payload data
- 16-bit Checksum Code for error detection.

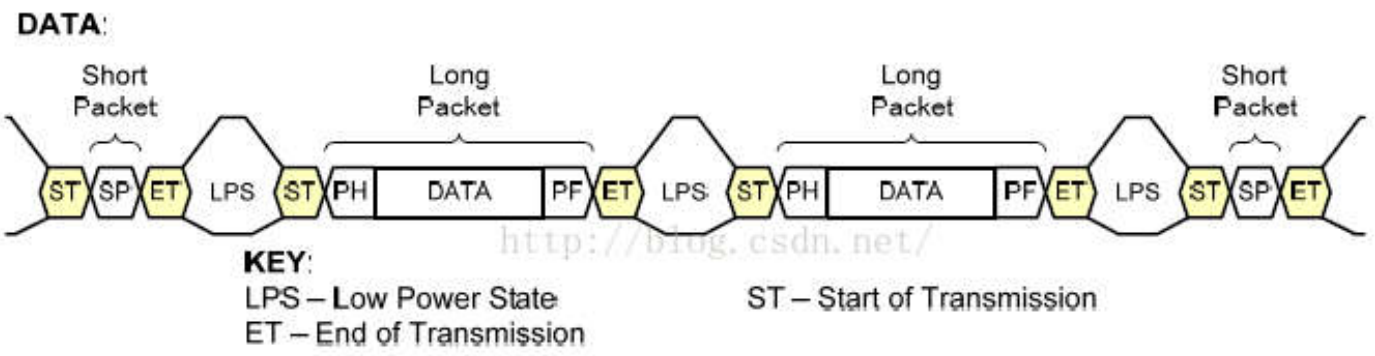

Figure 29 Low Level Protocol Packet Overview

9.1 Low Level Protocol Packet Format

Two packet structures are defined for low-level protocol communication: Long packets and Short packets. For each packet structure exit from the low power state followed by the Start of Transmission (SoT) sequence indicates the start of the packet. The End of Transmission (EoT) sequence followed by the low power state indicates the end of the packet.

9.1.1 Low Level Protocol Long Packet Format

Figure 30 shows the structure of the Low Level Protocol Long Packet. A Long Packet shall be identified by Data Types 0x10 to 0x37. See Table 3 for a description of the Data Types. A Long Packet shall consist of three elements: a 32-bit Packet Header (PH), an application specific Data Payload with a variable number of 8-bit data words and a 16-bit Packet Footer (PF). The Packet Header is further composed of three elements: an 8-bit Data Identifier, a 16-bit Word Count field and an 8-bit ECC. The Packet footer has one element, a 16-bit checksum. See sections 9.2 through 9.5 for further descriptions of the packet elements.

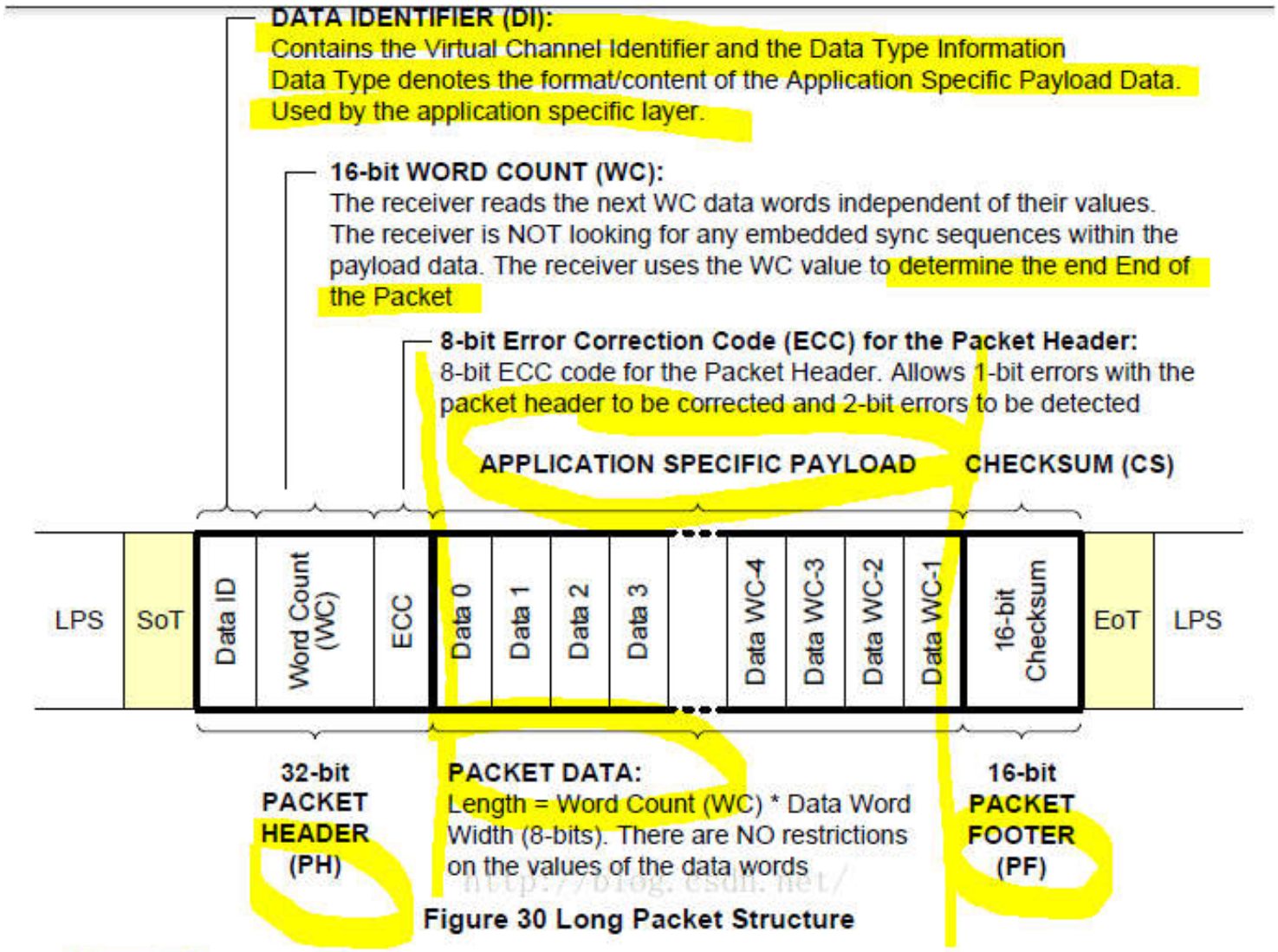

Figure 30 Long Packet Structure

The Data Identifier defines the Virtual Channel for the data and the Data Type for the application specific payload data.

The Word Count defines the number of 8-bit data words in the Data Payload between the end of the Packet Header and the start of the Packet Footer. Neither the Packet Header nor the Packet Footer shall be included in the Word Count.

The Error Correction Code (ECC) byte allows single-bit errors to be corrected and 2-bit errors to be detected in the packet header. This includes both the data identifier value and the word count value.

After the end of the Packet Header the receiver reads the next Word Count * 8-bit data words of the Data Payload. While reading the Data Payload the receiver shall not look for any embedded sync codes. Therefore, there are no limitations on the value of a data word.

Once the receiver has read the Data Payload it reads the checksum in the Packet Footer. In the generic case, the length of the Data Payload shall be a multiple of 8-bit data words. In addition, each data format may impose additional restrictions on the length of the payload data, e.g. multiple of four bytes.

Each byte shall be transmitted least significant bit first. Payload data may be transmitted in any byte order restricted only by data format requirements. Multi-byte elements such as Word Count, Checksum and the Short packet 16-bit Data Field shall be transmitted least significant byte first.

After the EoT sequence the receiver begins looking for the next SoT sequence.

9.1.2 Low Level Protocol Short Packet Format

Figure 31 shows the structure of the Low Level Protocol Short Packet. A Short Packet shall be identified by Data Types 0x00 to 0x0F. See Table 3 for a description of the Data Types. A Short Packet shall contain only a Packet Header; a Packet Footer shall not be present. The Word Count field in the Packet Header shall be replaced by a Short Packet Data Field.

For Frame Synchronization Data Types the Short Packet Data Field shall be the frame number. For Line Synchronization Data Types the Short Packet Data Field shall be the line number. See Table 6 for a description of the Frame and Line synchronization Data Types.

For Generic Short Packet Data Types the content of the Short Packet Data Field shall be user defined.

The Error Correction Code (ECC) byte allows single-bit errors to be corrected and 2-bit errors to be detected in the Short Packet.

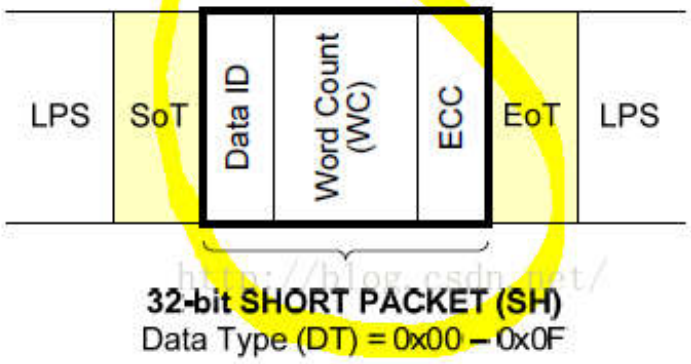

Figure 31 Short Packet Structure

9.2 Data Identifier (DI)

The Data Identifier byte contains the Virtual Channel Identifier (VC) value and the Data Type (DT) value as illustrated in Figure 32. The Virtual Channel Identifier is contained in the two MS bits of the Data Identifier Byte. The Data Type value is contained in the six LS bits of the Data Identifier Byte.

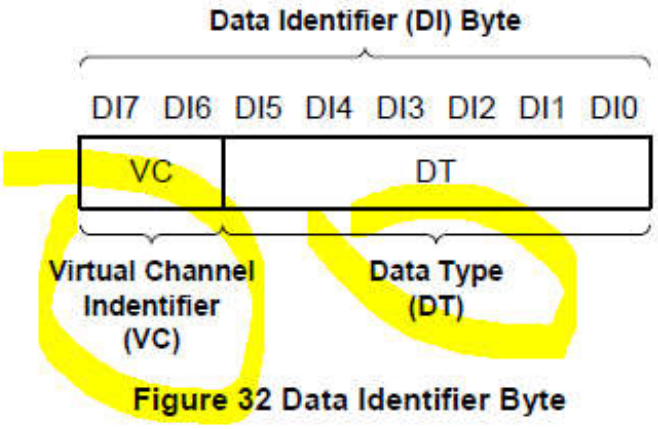

Figure 32 Data Identifier Byte

9.3 Virtual Channel Identifier

The purpose of the Virtual Channel Identifier is to provide separate channels for different data flows that are interleaved in the data stream.

The Virtual channel identifier number is in the top two bits of the Data Identifier Byte. The Receiver will monitor the virtual channel identifier and de-multiplex the interleaved video streams to their appropriate

channel. A maximum of four data streams is supported; valid channel identifiers are 0 to 3. The virtual channel identifiers in the peripherals should be programmable to allow the host processor to control how the data streams are de-multiplexed. The principle of logical channels is presented in the Figure 33.

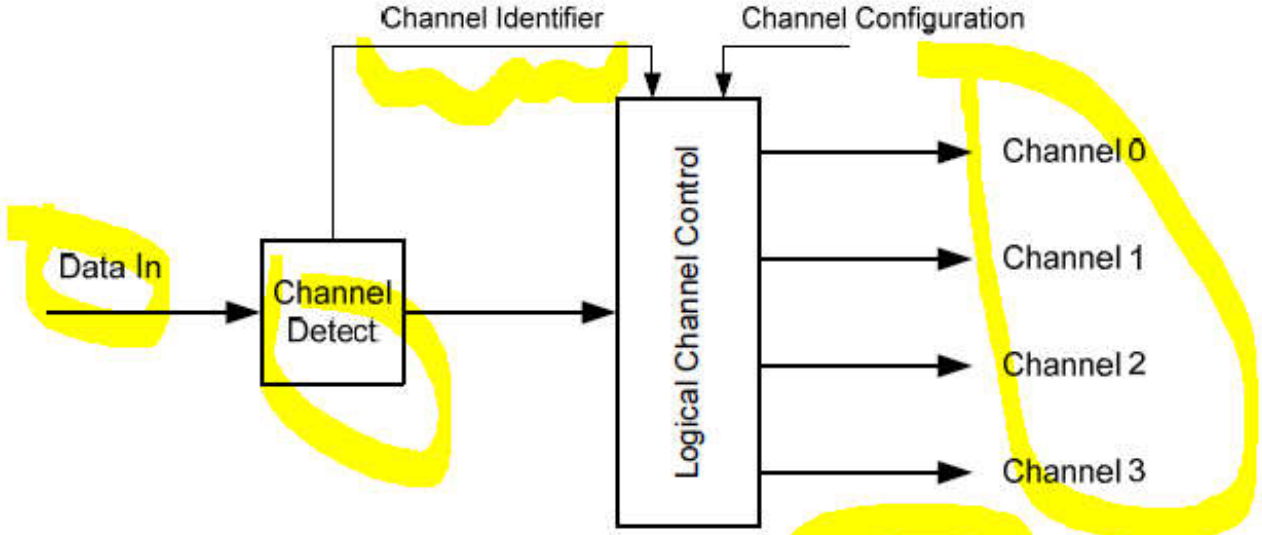

Figure 33 Logical Channel Block Diagram (Receiver)

9.4 Data Type (DT)

The Data Type value specifies the format and content of the payload data. A maximum of sixty-four data types are supported.

There are eight different data type classes as shown in Table 3. Within each class there are up to eight different data type definitions. The first two classes denote short packet data types. The remaining six classes denote long packet data types.

For details on the short packet data type classes refer to section 9.8.

For details on the five long packet data type classes refer to section 11.

Table 3 Data Type Classes

Data Type	Description
0x00 to 0x07	Synchronization Short Packet Data Types
0x08 to 0x0F	Generic Short Packet Data Types
0x10 to 0x17	Generic Long Packet Data Types
0x18 to 0x1F	YUV Data
0x20 to 0x27	RGB Data
0x28 to 0x2F	RAW Data
0x30 to 0x37	User Defined Byte-based Data
0x38 to 0x3F	Reserved

9.8 Synchronization Short Packet Data Type Codes

Short Packet Data Types shall be transmitted using only the Short Packet format. See section 9.1.2 for a format description.

Table 6 Synchronization Short Packet Data Type Codes

Data Type	Description
0x00	Frame Start Code
0x01	Frame End Code
0x02	Line Start Code (Optional)
0x03	Line End Code (Optional)
0x04 to 0x07	Reserved

9.8.1 Frame Synchronization Packets

Each image frame shall begin with a Frame Start (FS) Packet containing the Frame Start Code. The FS Packet shall be followed by one or more long packets containing image data and zero or more short packets containing synchronization codes. Each image frame shall end with a Frame End (FE) Packet containing the Frame End Code. See Table 6 for a description of the synchronization code data types.

For FS and FE synchronization packets the Short Packet Data Field shall contain a 16-bit frame number. This frame number shall be the same for the FS and FE synchronization packets corresponding to a given frame.

The 16-bit frame number, when used, shall be non-zero to distinguish it from the use-case where frame number is inoperative and remains set to zero.

The behavior of the 16-bit frame number shall be as one of the following

- Frame number is always zero – frame number is inoperative.
- Frame number increments by 1 for every FS packet with the same Virtual Channel and is periodically reset to one e.g. 1, 2, 1, 2, 1, 2, 1, 2 or 1, 2, 3, 4, 1, 2, 3, 4

The frame number must be a non-zero value.

9.8.2 Line Synchronization Packets

Line synchronization packets are optional.

For Line Start (LS) and Line End (LE) synchronization packets the Short Packet Data Field shall contain a 16-bit line number. This line number shall be the same for the LS and LE packets corresponding to a given line. Line numbers are logical line numbers and are not necessarily equal to the physical line numbers

The 16-bit line number, when used, shall be non-zero to distinguish it from the case where line number is inoperative and remains set to zero.

The behavior of the 16-bit line number shall be as one of the following:

- Line number is always zero – line number is inoperative.
- Line number increments by one for every LS packet within the same Virtual Channel and the same Data Type. The line number is periodically reset to one for the first LS packet after a FS packet. The intended usage is for progressive scan (non- interlaced) video data streams. The line number must be a non-zero value.
- Line number increments by the same arbitrary step value greater than one for every LS packet within the same Virtual Channel and the same Data Type. The line number is periodically reset to a non-zero arbitrary start value for the first LS packet after a FS packet. The arbitrary start value may be different between successive frames. The intended usage is for interlaced video data streams.

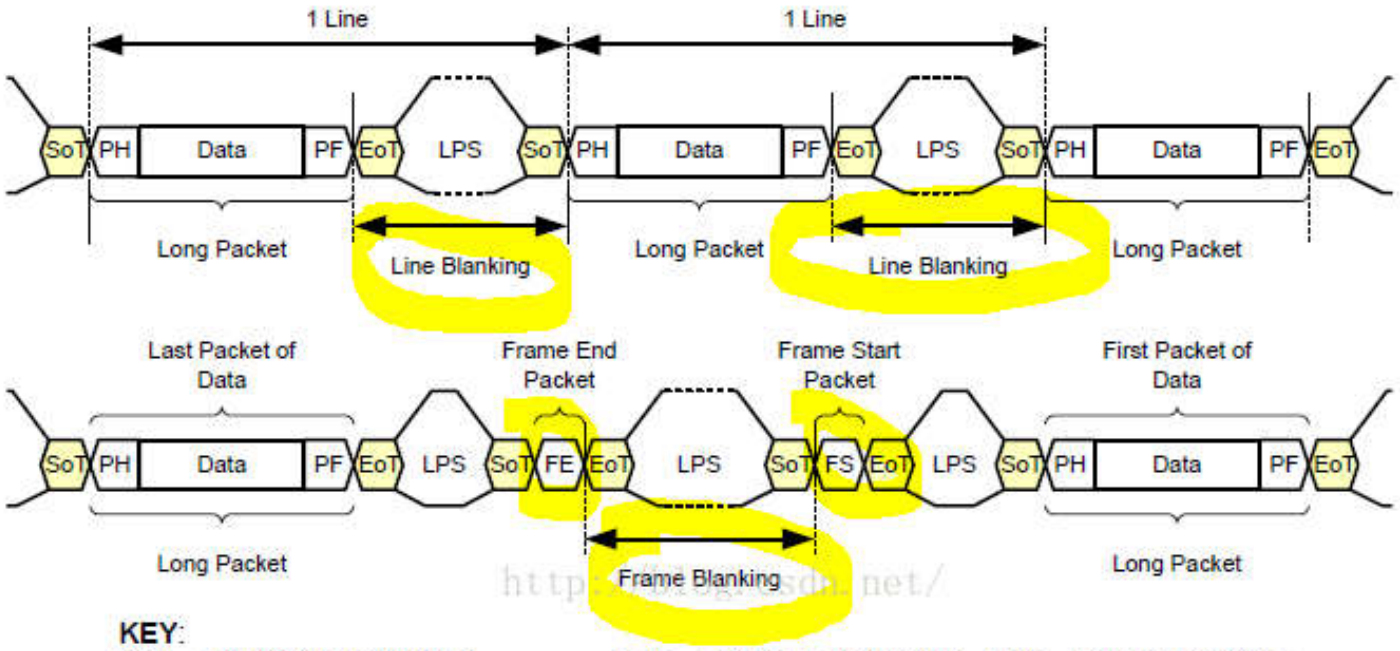

Figure 47 Line and Frame Blanking Definitions

The period between the Packet Footer of one long packet and the Packet Header of the next long packet is called the Line Blanking Period.

The period between the Frame End packet in frame N and the Frame Start packet in frame N+1 is called the Frame Blanking Period (Figure 47).

http://electronix.ru/forum/index.php?act=Attach&type=post&id=67362

http://wenku.baidu.com/link?url=h_rKvKzlk6ERorTsRNOUZ_As0zjsqSYuKgCzm2nBiUOma0_-NGY79QxrUaGS6MB4oqFK6H0Di8Adf4uDefkGB7P_OfWYiLXZtqzEZVD5_GC

http://mipi.org/sites/default/files/files/MIPI%20CSI-2%20Specification%20Brief.pdf

http://mipi.org/specifications/camera-interface

http://baike.baidu.com/link?url=6Hy6JlyrnaR4HSN2TIMi-kKuMKWcUoYK89CKiLUOPgrcwQnvoQAKUd2-P3s4BKg0YkZ_xKDwucksx7DnHLccla

http://blog.csdn.net/eric41050808/article/details/9137623

PHP知识库

6867 关注 | 1085 收录

.NET知识库

3950 关注 | 839 收录

猜你在找

工作流

前端学习路线

进销存

夜大

单身公寓

查看评论

暂无评论

您还没有登录,请[登录]或[注册]

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack
VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery
BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML LBS Unity
Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra CloudStack
FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide
Maemo Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase
Pure Solr Angular Cloud Foundry Redis Scala Django Bootstrap