AN13573
i.MX 8/RT MIPI DSI/CSI-2

Rev. 0 — 21 March 2022

Application Note

Contents

1 Introduction

This application note provides detailed information about the MIPI DSI and CSI-2 interfaces on various i.MX 8 and i.MX RT processors. It covers the implementation differences between processors and provides information to design a CSI-2 and/or DSI system and debug operations.

For detailed physical design reference about various processors, see the Hardware Developers Guides.

This document does not cover detailed software guides either.

1.1 References

- i.MX 8M reference manuals
- i.MX 8M data sheets
- i.MX RT reference manuals
- i.MX RT data sheets
- MIPI specification for D-PHY version 1.2
- MIPI specification for DSI version 1.1
- MIPI specification for CSI-2 version 1.3

1.2 Overview

The Mobile Industry Processor Interface (MIPI) alliance was founded in 2003 by Arm, Nokia ST, and Texas Instruments. Shortly thereafter Intel, Motorola, Samsung, and Philips joined. Today there are over 300 members ranging from IP vendors to chip makers to product developers, and so on. The focus of the organization is to design and promote hardware and software interfaces that simplify the integration of components built into a device.

This document covers the following three specifications:

- The Camera Serial Interface 2 (CSI-2) specification defines a standard data transmission and the control interface between a peripheral device (camera) and a host processor. The specification is focused on the protocols for transferring image data (both still images and video) and basic image sensor control though the Camera Control Interface (CCI). The CSI-2 specification does not cover any advanced imaging system features, such as, image exposure, focusing, processing, or any other common features used in an imaging system.
- The Display Serial Interface (DSI) specification defines the data transfer protocol and the interface between a host processor and peripheral (display).

  Although they do not specify the physical layer, CSI-2 and DSI specifications are low-level layers in the overall system.

- D-PHY specification defines the transmission medium (electrical conductors), the input/output circuitry, and the clocking mechanism that captures **ones** and **zeroes** from the serial bit stream.

Figure 1 shows a typical example of a system design, consisting of a camera and a touch display. They are both connected to an applications processor with separate four-lane MIPI D-PHY interfaces for data transfer and an I^{2}C interface for control.

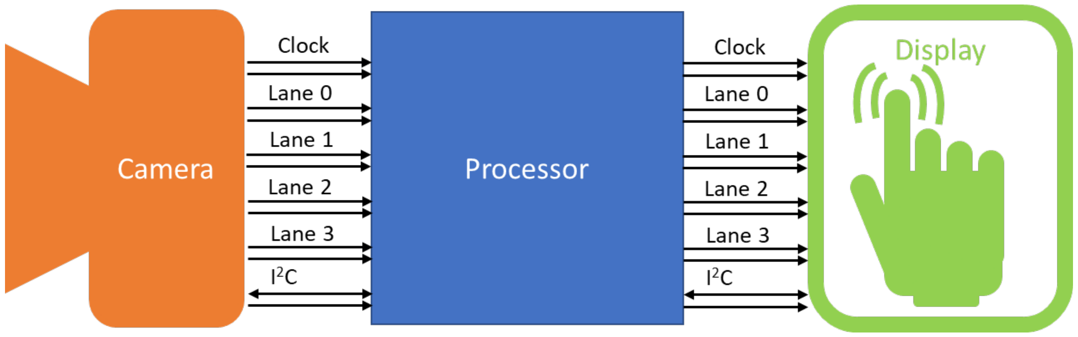


Figure 1. Example system using MIPI CSI-2 and DSI interfaces

NOTE

MIPI DSI, CSI-2, and D-PHY specifications are for chip-to-chip interfacing and designs where the camera and display are part of a fully integrated product like a cell-phone or tablet. Unlike some **hot-pluggable** or **plug and play** high-speed interfaces (such as, HDMI or USB), the MIPI DSI/CSI-2 interfaces are meant for a fixed and permanent connection in a final design.

2 D-PHY

D-PHY is the source-synchronous physical layer used by CSI-2 and DSI protocols for physical and electrical communication. The **D** in D-PHY comes from the Roman numeral for 500. Version 1.0 of the D-PHY specification was written assuming a single lane would communicate on the order of 500 Mbps, and so the **D-PHY** name was born! In the most recent version of the specification, the actual maximum bit rate per lane is up to 9 Gbps. The i.MX 8/RT parts have a maximum of 1.5 Gbps per lane (and some parts even less) due to the specification versions followed and design implementations.

2.1 Link basics

MIPI specifications use a consistent set of terms when talking about pieces required to put together a system based on CSI-2 or DSI.

- **Line**: An interconnect or wire or trace between a single pin on a host processor and a pin on a peripheral device.
- **Lane**: Two lines are required to support differential signaling. D-PHY uses a differential pair for high-speed data and clock transmission. This differential pair is referred to as a **Lane Interconnect**, but mostly shortened to **Lane** in technical documents.
- **Link**: A connection between two devices containing one Clock Lane and at least one Data Lane. A Link consists of at least two PHYs and two Lane Interconnects.

Many data sheets abbreviate the Link information and/or omit the **Link** term entirely. For example, in the i.MX RT500 data sheet, the block diagram shows MIPI-DSI (2-lane). It means that the part contains a clock lane and two data lanes for six pins used in the MIPI DSI interface.

2.2 Modes of Operation

There are four modes of operation: Control, High-Speed, Escape, and Ultra-Low Power State (ULPS). Data Lanes support all four modes but Clock Lanes only support Control, High-Speed, and ULPS mode. During normal operation, a Data Lane is in Control or High-Speed mode. High-Speed Data transmission happens in bursts and starts from and ends at a Stop state (LP-11), which is in Control mode. The lane is only in High-Speed mode during data bursts. Table 1 describes the six physical states that the clock and data lanes can be put into.

Table 1. Lane state descriptions

State Code	Line Voltage Levels		High-Speed	Low-Power	
	Dp-Line	Dn-Line	Burst Mode	Control Mode	Escape Mode
HS-0	HS Low	HS High	Differential-0	N/A[1]	N/A[1]
HS-1	HS High	HS Low	Differential-1	N/A[1]	N/A[1]
LP-00	LP Low	LP Low	N/A	Bridge	Space
LP-01	LP Low	LP High	N/A	HS-Request	Mark-0
LP-10	LP High	LP Low	N/A	LP-Request	Mark-1
LP-11	LP High	LP High	N/A	Stop	N/A[2]

1. During High-Speed transmission, the Low-Power receivers observe LP-00 on the Lines.
2. If LP-11 occurs during Escape mode, the Lane returns to Stop state (Control Mode LP-11).

2.3 I/O signaling

The D-PHY IO pins have two different electrical modes of operation: single ended and differential.

- The single-ended mode is referred to as Low Power (LP) mode. It is used during escape and control modes for system configuration/control between the host processor and the peripheral device. In LP mode, the IO circuits use a traditional CMOS output buffer (with slew rate control) that switches between ground and `VDD_IO`. In LP mode, the two IO circuits, which comprise a lane, can switch independently. They provide four different lane states: LP-00, LP-01, LP-10, and LP-11.
- The differential mode is referred to as High-Speed (HS) mode. It is used only for data transfer. The differential mode only has two states: HS-0 and HS-1. During HS mode, the two IO circuits, which comprise a lane, are always in opposite output polarity. They can NEVER be logic high or low at the same time. Figure 2 illustrates the differences in signaling levels between HS and LP mode.

In normal operation data lanes are constantly switching between low-power control mode and high-speed data transfer mode.

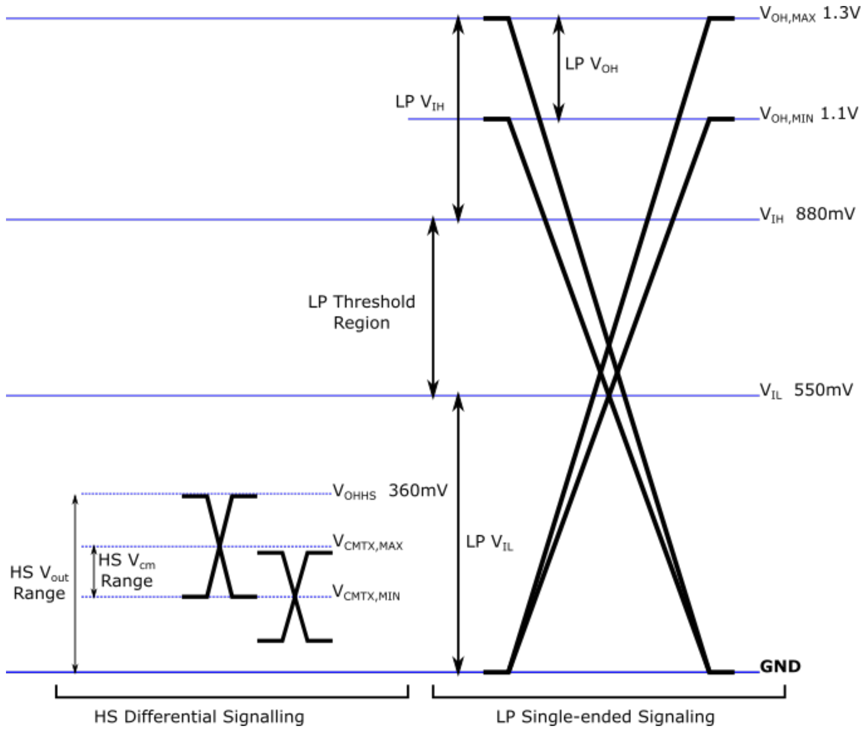


Figure 2. D-PHY signaling levels

Table 2 lists most critical IO signaling voltage levels used in both LP and HS mode. The operating voltage of HS mode ranges below the V_{il} level of 550 mV. If the link transmitter enters HS mode and the receiver does not (for whatever reason), the receiver quickly flags an error because it does not expect to see long periods of logic 0 in LP mode.

Table 2. D-PHY DC electrical specifications

Parameter	Description	Min.	Type	Max.	Unit
V_{OH}	Output high level	1.1	1.2	1.3	V
V_{OL}	Output low level	-50		50	mV
V_{IH}	Input high-level threshold	880			mV
V_{IL}	Input low-level threshold			550	mV
$V_{IL-ULPS}$	Input low-level threshold in ULPS			300	mV
V_{CMTX}	HS transmit static common-mode voltage	150	200	250	mV
$\|V_{OD}\|$[1]	HS transmit differential voltage	140	200	270	mV
V_{OHHS}	HS output high voltage			360	mV

1. The differential output voltage V_{OD} is defined as the difference of the voltages V_{DP} and V_{DN} at the D_P and D_N pins, respectively.

2.4 DDR clocking

Data transfer during HS mode uses a Dual Data Rate (DDR) transfer scheme. The receiver samples the data every time the clock lane signals change state. In a Single Data Rate (SDR) synchronous data bus architecture (for example, SDRAM), the data is sampled on either the rising edge or falling edge but not both. It means that the data rate is equal to the clock speed. The various low-power modes use a single data rate clock structure.

A DDR data bus architecture can transfer data on EVERY clock edge as opposed to only a single edge. It means that for a given clock frequency (Fr), the data rate per lane is 2 × Fr. For example, if the clock frequency is 100 MHz, the data rate is 2 × 100 MHz = 200 MHz. Figure 3 shows the relationship between data and clock for both SDR and DDR clocking scheme.

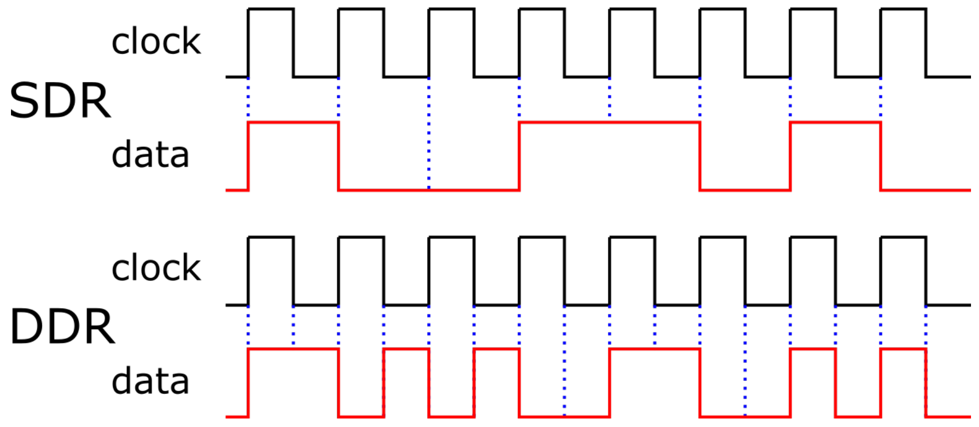


Figure 3. SDR vs. DDR

As HS mode uses differential signaling, the timing values are measured regarding the actual observed crossing of the clock differential signal. Figure 4 shows the timing alignment between data and clock lanes.

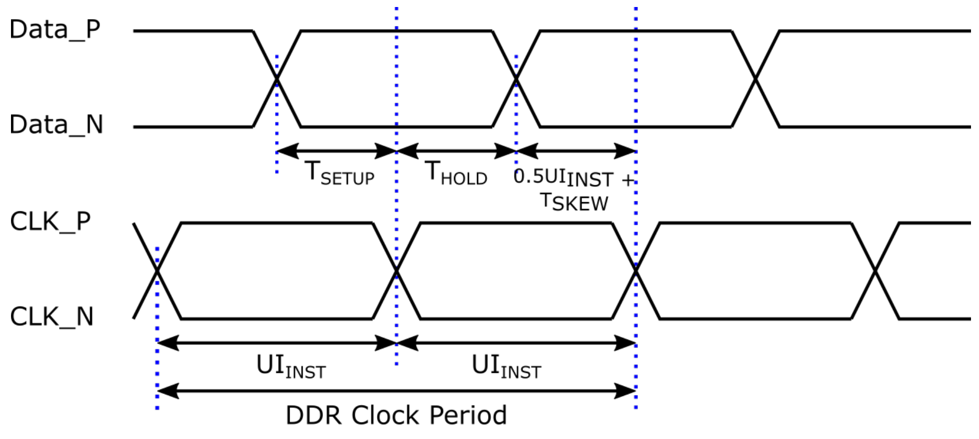


Figure 4. D-PHY DDR clock definition

Unit Interval (UI) is equal to the duration of any HS state on the Clock Lane. The DDR clock period is equal to 2 × UI.

2.5 Mode transitions

To make efficient use of the few pins used in the D-PHY interface, there is a lack of traditional bus handshaking signals (for example, data enable, line select). Therefore, the transition between the various LP and HS modes is handled through a series of handshaking sequences.

2.5.1 Start of HS Transmission

The transition from a stop state (LP-11) into high-speed data transmission mode takes two steps.

1. The clock lane enters high-speed clock mode.
2. The data lane enters high-speed mode.

2.5.1.1 Clock lane HS transition

Table 3 describes the sequence of events for the clock lane, as it transits from a Low-Power state to High-Speed mode.

Table 3. Procedure to initiate High-Speed clock transmission

TX side	RX side
Drives Stop state (LP-11)	Observes Stop state
Drives HS-Req state (LP-01) for time TLPX	Observes transition from LP-11 to LP-01 on the Lines
Drives Bridge state (LP-00) for time T_{CLK}-PREPARE	Observes transition from LP-01 to LP-00 on the Lines. Enables Line Termination after time TCLK-TERM-EN
Enables High-Speed driver and disables Low-Power drivers simultaneously.	Enables HS-RX and waits for timer $T_{CLK\text{-}SETTLE}$ to expire in order to neglect transition effects.
Drives HS-0 for a time $T_{CLK\text{-}ZERO}$	Receives HS-signal
Drives the High-Speed Clock signal for time period TCLK-PRE before any Data Lane starts up	Receives High-Speed Clock signal

2.5.1.2 Data lane HS transition

Table 4 describes the Start-of-Transmission (SoT) procedure for the data lane, as it transits from the Stop state and enters High-Speed mode.

Table 4. Start-of-Transmission sequence

TX side	RX side
Drives Stop state (LP-11)	Observes Stop state
Drives HS-Req state (LP-01) for time TLPX	Observes transition from LP-11 to LP-01 on the Lines
Drives Bridge state (LP-00) for time THS-PREPARE	Observes transition from LP-01 to LP-00 on the Lines, enables Line Termination after time TD-TERM-EN
Enables High-Speed driver and disables Low-Power drivers simultaneously.	
Drives HS-0 for a time THS-ZERO	Enables HS-RX and waits for timer THS-SETTLE to expire in order to neglect transition effects
	Starts looking for Leader-Sequence
Inserts the HS Sync-Sequence ‘00011101’ beginning on a rising Clock edge	
	Synchronizes upon recognition of Leader Sequence `011101`

Table continues on the next page...

Table 4. Start-of-Transmission sequence (continued)

TX side	RX side
Continues to Transmit High-Speed payload data	
	Receives payload data

2.5.1.3 Clock and data SoT relationship

Figure 5 shows the timing relationship between the transmitter/receiver pair for the clock lane and the data lane. Figure 5 only shows a single data lane. Multiple data lanes follow the same sequence and start at the same time.

To ensure proper entry into HS mode, it is critical to configure the timing parameters properly for both the transmitter and the receiver.

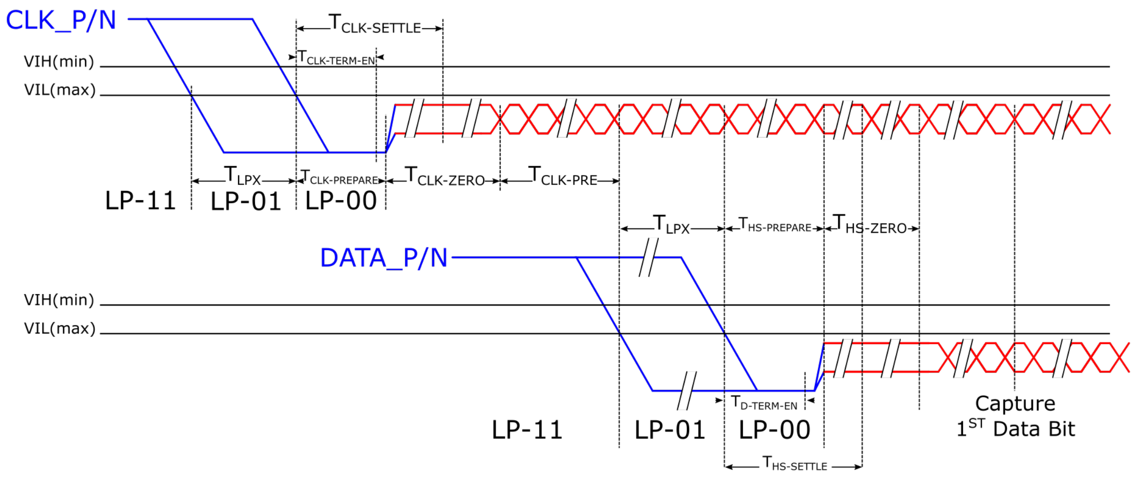


Figure 5. Clock and data lanes entering HS transmission

Some of the timings shown in Figure 5 can be adjusted in DSI/CSI-2/D-PHY configuration registers of the SOC covered in this document. Table 5 lists the D-PHY specs, the parameters, and associated IP blocks that allow for tuning options.

Table 5. D-PHY SoT timing parameters

Parameter	Description	Min.	Max.	Unit	IP
$T_{CLK-PRE}$	Time that the transmitter drives the HS clock before any associated Data Lane begins the transition from LP to HS mode.	8		UI[1]	DSI
$T_{CLK-PREPARE}$	Time that the transmitter drives the Clock Lane LP-00 Line state immediately before the HS-0 Line state starts the HS transmission.	38	95	ns	DSI
$T_{CLK-SETTLE}$	Time interval during which the HS receiver should ignore any Clock Lane HS transitions, starting from the beginning of TCLK-PREPARE.	95	300	ns	N/A[2]
$T_{CLK-PREPARE}$ + $T_{CLK-ZERO}$	$T_{CLK-PREPARE}$ + time that the transmitter drives the HS-0 state prior to starting the Clock.	300		ns	N/A[2]

Table continues on the next page...

Table 5. D-PHY SoT timing parameters (continued)

Parameter	Description	Min.	Max.	Unit	IP
$T_{CLK\text{-}TERM\text{-}EN}$	Time for the Clock Lane receiver to enable the HS line termination, starting from the time point when D_n crosses V_{IL}, MAX.	Time for D_n to reach VTERM-EN		ns	N/A[2]
$T_{CLK\text{-}ZERO}$	Time that the transmitter drives the HS-0 state prior to starting the Clock.				DSI
$T_{D\text{-}TERM\text{-}EN}$	Time for the Data Lane receiver to enable the HS line termination, starting from the time point when D_n crosses V_{IL}, MAX.	Time for D_n to reach VTERM-EN	35 ns + 4 × UI		N/A[2]
$T_{HS\text{-}PREPARE}$	Time that the transmitter drives the Data Lane LP-00 Line state immediately before the HS-0 Line state starting the HS transmission.	40 ns + 4 × UI	85 ns + 6 × UI	ns	DSI
$T_{HS\text{-}PREPARE}$ + $T_{HS\text{-}ZERO}$	THS-PREPARE + time that the transmitter drives the HS-0 state prior to transmitting the Sync sequence.	145 ns + 10 × UI		ns	
$T_{HS\text{-}SETTLE}$	Time interval during which the HS receiver shall ignore any Data Lane HS transitions, starting from the beginning of THS-PREPARE. The HS receiver shall ignore any Data Lane transitions before the minimum value, and the HS receiver shall respond to any Data Lane transitions after the maximum value.	85 ns + 6 × UI	145 ns + 10 × UI	ns	CSI
$T_{HS\text{-}ZERO}$	Time that the transmitter drives the HS-0 state prior to transmitting the Sync sequence.				DSI
T_{LPX}	Transmitted length of any Low-Power state period	50		ns	N/A[2]

1. The MIPI Bit Clock Time Period/2.
2. These parameters must meet the D-PHY specification but are not directly programmable.

2.5.2 End of HS transmission

At the end of a High-Speed Data burst, a data lane leaves High-Speed transmission mode and enters the Stop state with an End-of-Transmission (EoT) procedure. Table 6 describes this sequence of events.

Table 6. End-of-Transmission sequence

TX side	RX side
Completes transmission of payload data	Receives payload data
Toggles differential state immediately after last payload data bit and keeps that state for a time THS-TRAIL	
Disables the HS-TX, enables the LP-TX, and drives Stop state (LP-11) for a time THS-EXIT	Detects the Lines leaving LP-00 state and entering Stop state (LP-11) and disables Termination
	Neglect bits of last period THS-SKIP to hide transition effects
	Detect last transition in valid Data, determine last valid data byte, and skip trailer sequence

2.5.2.1 Clock and data EoT relationship

Figure 6 shows the EoT timing relationship between the transmitter/receiver pair for both the clock lane and the data lanes. Figure 6 only shows a single data lane. Multiple data lanes follow the same sequence and end at the same time. To ensure proper exit from HS mode, it is critical to configure the timing parameters properly for both the transmitter and the receiver.

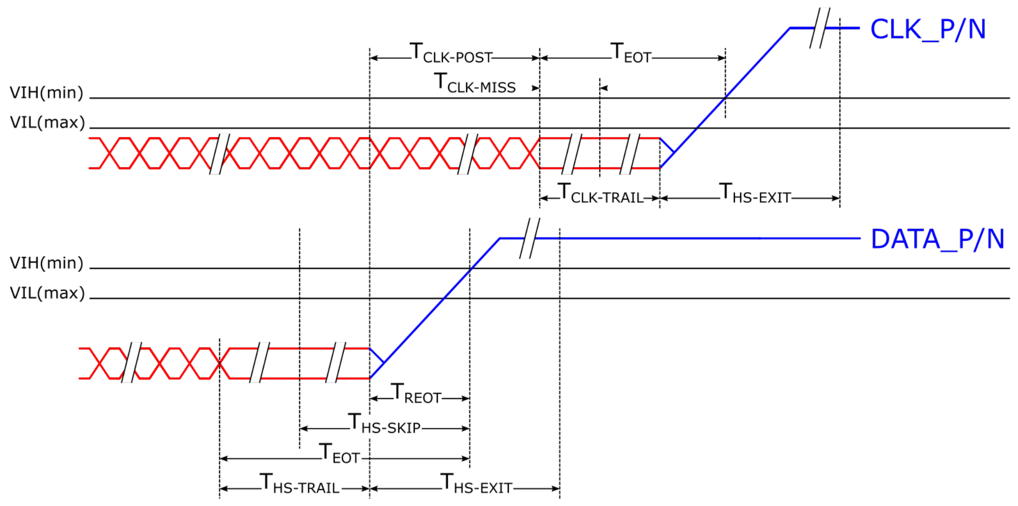


Figure 6. Data and clock lane exiting from HS transmission

Some of the timings shown in Figure 6 can be adjusted in DSI/CSI-2/D-PHY configuration registers of SOC covered in this document. Table 7 lists the D-PHY specs, parameters, and associated IP blocks that allow for tuning options.

Table 7. D-PHY EoT timing parameters

Parameter	Description	Min.	Max.	Unit	IP
$T_{CLK-POST}$	Time that the transmitter continues to send HS clock after the last associated Data Lane has transited to LP mode. Interval is defined as the period from the end of $T_{HS-TRAIL}$ to the beginning of $T_{CLK-TRAIL}$.	60 ns + 52 × UI		ns	DSI
$T_{CLK-MISS}$	Timeout for receiver to detect absence of Clock transitions and disable the Clock Lane HS-RX.		60	ns	N/A[1]
$T_{CLK-TERM-EN}$	Time for the Clock Lane receiver to enable the HS line termination, starting from the time point when D_n crosses VIL, MAX.		38	ns	N/A[1]
$T_{CLK-TRAIL}$	Time that the transmitter drives the HS-0 state after the last payload clock bit of an HS transmission burst.	60		ns	DSI
T_{EOT}	Transmitted time interval from the start of $T_{HS-TRAIL}$ or $T_{CLK-TRAIL}$, to the start of the LP-11 state following an HS burst.		105 ns + n × 12 × UI	ns[2]	DSI
$T_{HS-EXIT}$	Time that the transmitter drives LP-11 following an HS burst.	100		ns	DSI
$T_{D-TERM-EN}$	Time for the Data Lane receiver to enable the HS line termination, starting from the time point when D_n crosses VIL, MAX.		35 ns + 4 × UI	ns	N/A[1]

Table continues on the next page...

Table 7. D-PHY EoT timing parameters (continued)

Parameter	Description	Min.	Max.	Unit	IP
$T_{HS\text{-}SKIP}$	Time interval during which the HS-RX should ignore any transitions on the Data Lane, following as HS burst. The end point of the interval is defined as the beginning of the LP-11 state following the HS burst.	40	55 + 4 × UI	ns	CSI
$T_{HS\text{-}TRAIL}$	Time that the transmitter drives the flipped differential state after last payload data bit of an HS transmission burst	max (n × 8 × UI, 60 ns + n × 4 × UI)		ns[2]	DSI

1. These parameters must meet the D-PHY specification but are not directly programmable.
2. Where n = 1 for Forward-direction HS mode and n = 4 for Reverse-direction HS mode.

2.5.3 Continuous vs Non-Continuous clock

Both the CSI-2 and DSI specifications support a **continuous clock** mode. For continuous clock behavior, the Clock Lane remains in high-speed mode generating active clock signals between HS data packet transmissions. For non-continuous clock behavior, the Clock Lane enters the LP-11 state between HS data packet transmissions. Continuous clock mode allows for higher data rates, because the timing overhead of exiting and reentering HS-mode on the clock lane is eliminated.

2.5.3.1 Non-continuous clock mode transition procedure

Once a data transmission has finished, the clock lane ***can optionally*** switch back to a Low-Power State (LP-11). Table 8 describes the sequence of the events, as the clock lane transitions from a High-Speed mode to a Low-Power State.

Table 8. Procedure to switch Clock Lane to Low-Power mode

Master side	Slave side
Drives High-Speed Clock signal (Toggling HS-0/HS-1)	Receives High-Speed Clock signal (Toggling HS-0/HS-1)
Last Data Lane goes into Low-Power mode	
Continues to drive High-Speed Clock signal for a period TCLK-POST and ends with HS-0 state	
Drives HS-0 for a time TCLK-TRAIL	Detects absence of Clock transitions within a time TCLK-MISS, disables HS-RX then waits for a transition to the Stop state
Disables the HS-TX, enables the LP-TX, and drives Stop state (LP-11) for a time THS-EXIT	
	Detects the Lines transitions to LP-11, disables HS termination, and enters Stop state

2.5.4 Ultra-low power state

From the D-PHY perspective, Ultra-low power state puts both the clock and data lanes into LP-00 state and signals to the peripheral device to enter an ULPS. The actual ULPS implementation is up to the various device makes. The D-PHY specification does not describe what the host or peripheral device compute cores should be doing or how they are operating during this time.

2.5.4.1 Clock lane ULPS entry

Table 9 describes the clock lane ULPS entry sequence.

Table 9. Clock lane ULPS entry and exit sequence

TX	RX	Notes
Start in TX-Stop State (LP-11)	RX-Stop (LP-11)	
Request ULPS by transmitting LP-10 state for T_{LPX}		
Drive TX-ULPS State (LP-00) for desired length of time.		
Exit ULPS by driving LP-10 state for T_{WAKEUP}		

2.5.4.2 Data lane ULPS entry

The data lane ULPS entry sequence is a little more complicated because the data lanes can support bidirectional communication during escape mode with a Link Turnaround procedure.

NOTE

MIPI DSI specification refers to this as a Bus Turnaround procedure (BTA). The CSI-2 specification does not have a BTA procedure instead it relies on an I^2C link for bidirectional communication.

The TX data lane sends a handshake sequence to signal ULPS mode. Table 10 describes the sequence of events.

Table 10. Data lane ULPS entry and exit sequence

TX	RX	Notes
Start in TX-Stop State (LP-11)	RX in stop state (LP-11)	
Request LP by transmitting LP-10 state for TLPX		
Transmit LP-00 for TLPX		
Transmit LP-01 for TLPX		
Transmit LP-00 for TLPX		
Transmit sequence of 8-bit ULPS entry command (00011110)		If the transmitter detects an LP-11 state at any time the sequence aborts and the lane reverts to the Stop State (LP-11)
Transmit LP-00 for desired length of time.		
Exit ULPS by driving LP-10 state for T_{WAKEUP}		

Figure 7 shows the logic states of a data lane as it enters ULPS.

NOTE

ULPS entry command uses **Spaced-One-Hot** encoding.

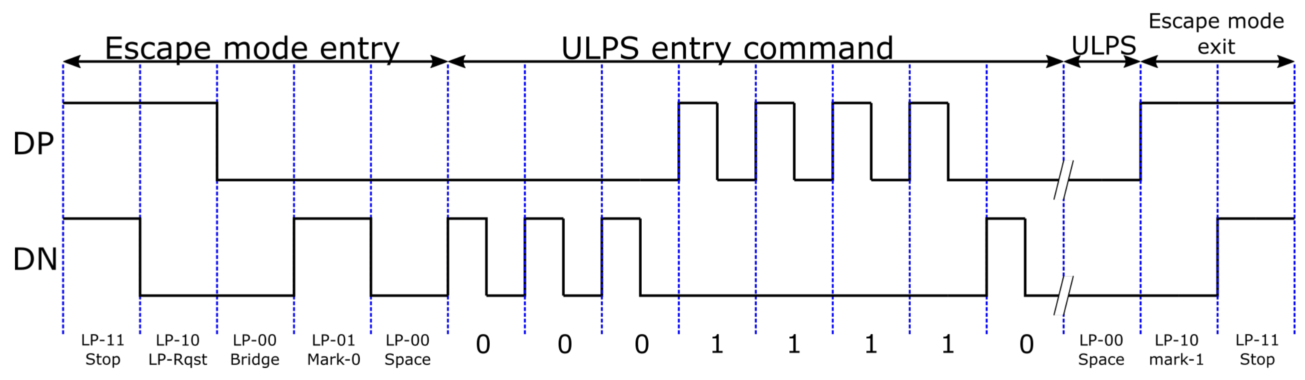


Figure 7. Data lane ULPS entry sequence

Spaced-One-Hot encoding allows for clock recovery at the receiver and therefore the clock lane is not required. The way it works is each Mark state is interleaved with a Space state. Each symbol consists therefore of two parts: a One-Hot phase (Mark-0 or Mark-1) and a Space phase. The TX shall send Mark-0 followed by a Space to transmit a **zero-bit** and it shall send a Mark-1 followed by a Space to transmit a **one bit**. A Mark that is not followed by a Space does not represent a bit. The last phase before exiting Escape mode with a Stop state shall be a Mark-1 state that is not part of the communicated bits, as it is not followed by a Space state.

2.6 Fault detection

If the configurable timing parameters for both the transmitter and receiver are not correctly set up, errors can occur during the SoT and EoT procedures. The D-PHY can detect certain errors, trigger an IRQ, and raise register flags to help with debugging the issue.

2.6.1 SoT errors

The most common errors occur during the Start of Transmission sequence. When certain errors occur, the CSI-2 and DSI IP blocks raise IRQ.

`ErrSotHS` occurs if the high-speed SoT leader sequence is corrupted in such a way that proper synchronization can still be achieved. This error signal is asserted for one cycle of `RxByteClkHS`. This error is considered to be a **soft error** in the leader sequence and confidence in the payload data is reduced.

`ErrSotSync_HS` occurs if the high-speed SoT leader sequence is corrupted in a way that proper synchronization cannot be expected. This error signal is asserted high for HS reception.

If either of these errors occur, it is best to double check the SoT timing parameter setup.

- On the transmitter side (SOC DSI, or Camera), you can shorten the time to enter HS mode by adjusting the $T_{CLK\text{-}PRE}$, $T_{CLK\text{-}PREPARE}$, $T_{HS\text{-}PREPARE}$, $T_{CLK\text{-}ZERO}$, and $T_{HS\text{-}ZERO}$ parameters.
- On the receiver side (SOC CSI-2 or Display), you can increase the wait time before looking for the SoT header by adjusting the $T_{HS\text{-}SETTLE}$ parameter.

3 DSI

This section explains the DSI interface and how a display device connects to a host processor. After reading this section, an engineer can evaluate various displays and decide the suitability and compatibility with any i.MX processor described in this document. A systems engineer can specify the hardware of a full display system. This section does **NOT** describe the details of implementing a full display system in software, such as, image manipulation (GPU/PXP), color correction, drivers, and so on.

3.1 Overview

DSI specifies the interface between a host processor and a peripheral, such as, a display module. It builds on existing MIPI Alliance specifications by adopting pixel formats and command sets specified in the DPI-2, DBI-2, and DCS standards. The DSI block serializes all pixel data, commands, and events. In traditional or legacy interfaces, they are normally conveyed to and from the peripheral on a parallel data bus with additional control signals. A legacy display interface that supports 24 bits per pixel (bpp) color requires a 24-pin data-bus, vsync, hsync, enable, and pixel clock lines for 28 individual signals. The DSI specification can transmit the same data using 4-10 pins depending upon number of data lanes used in the D-PHY.

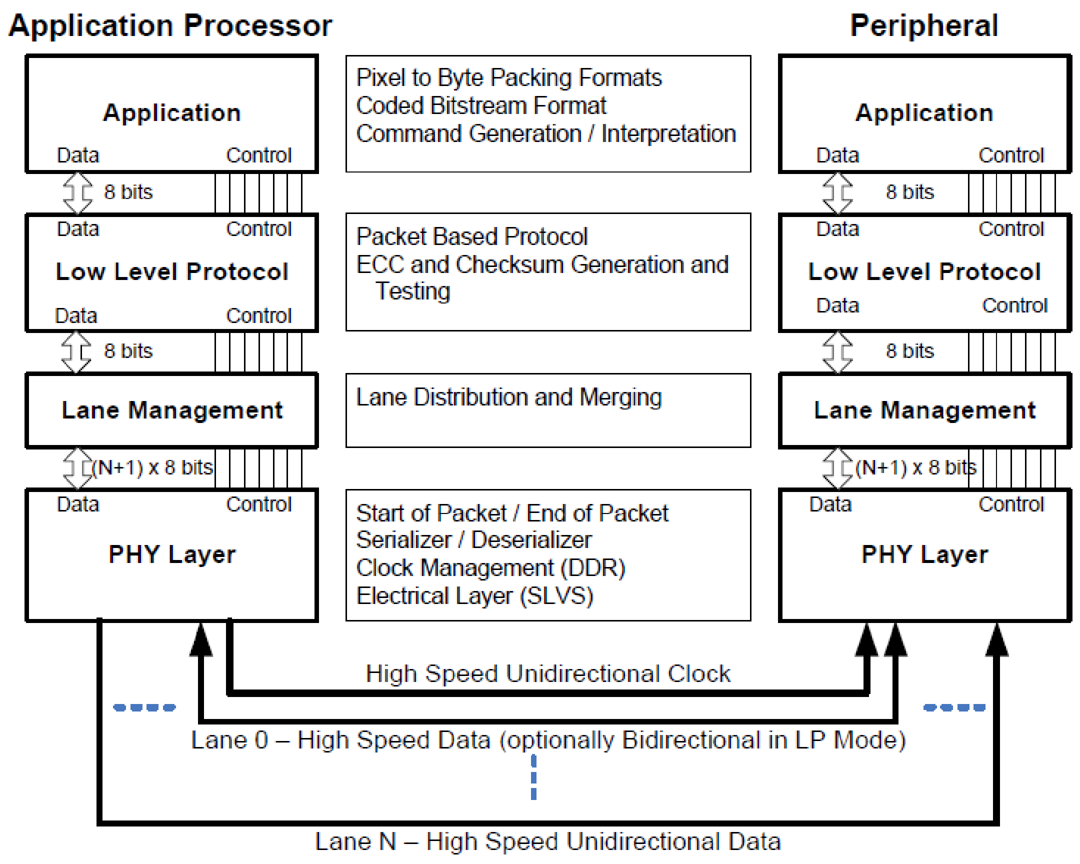

Application Processor
Peripheral
Application
Data
Control
8 bits
Pixel to Byte Packing Formats
Coded Bitstream Format
Command Generation / Interpretation
Low Level Protocol
Packet Based Protocol
ECC and Checksum Generation and Testing
Lane Management
Lane Distribution and Merging
(N+1) x 8 bits
PHY Layer
Start of Packet / End of Packet
Serializer / Deserializer
Clock Management (DDR)
Electrical Layer (SLVS)
High Speed Unidirectional Clock
Lane 0 – High Speed Data (optionally Bidirectional in LP Mode)
Lane N – High Speed Unidirectional Data

Figure 8. DSI layers

3.2 Digital display basics

It helps to have a basic understanding of how digital displays work to understand the data transfer formatting and timing. All modern computer/information displays are raster displays, meaning the displayed image is created by writing one line at a time, pixel by pixel from left to right and from the top of the display to the bottom. This displayed image, or *frame*, is composed of a rectangular array of pixels.

A *Video* stream is composed of a series of frames displayed at a fixed timing interval. This timing interval can be described using Frames Per Second (FPS) or Hz. Traditional movie theater projectors use 24 FPS or 24 Hz as a timing interval. Modern digital displays (smart phones for example) use a frame interval frequency of 60 Hz or more.

The roots of this process come from the old CRT displays where an electron beam was continuously steered across the back of the screen to illuminate a picture. The beam would scan from left to right to generate a single line of the image then it would be turned off and retraced horizontally back to the left before being turned on again at the beginning of the next line. The beam continues this pattern from top to bottom before doing a single vertical retrace. Figure 9 shows the path of the electron beam.

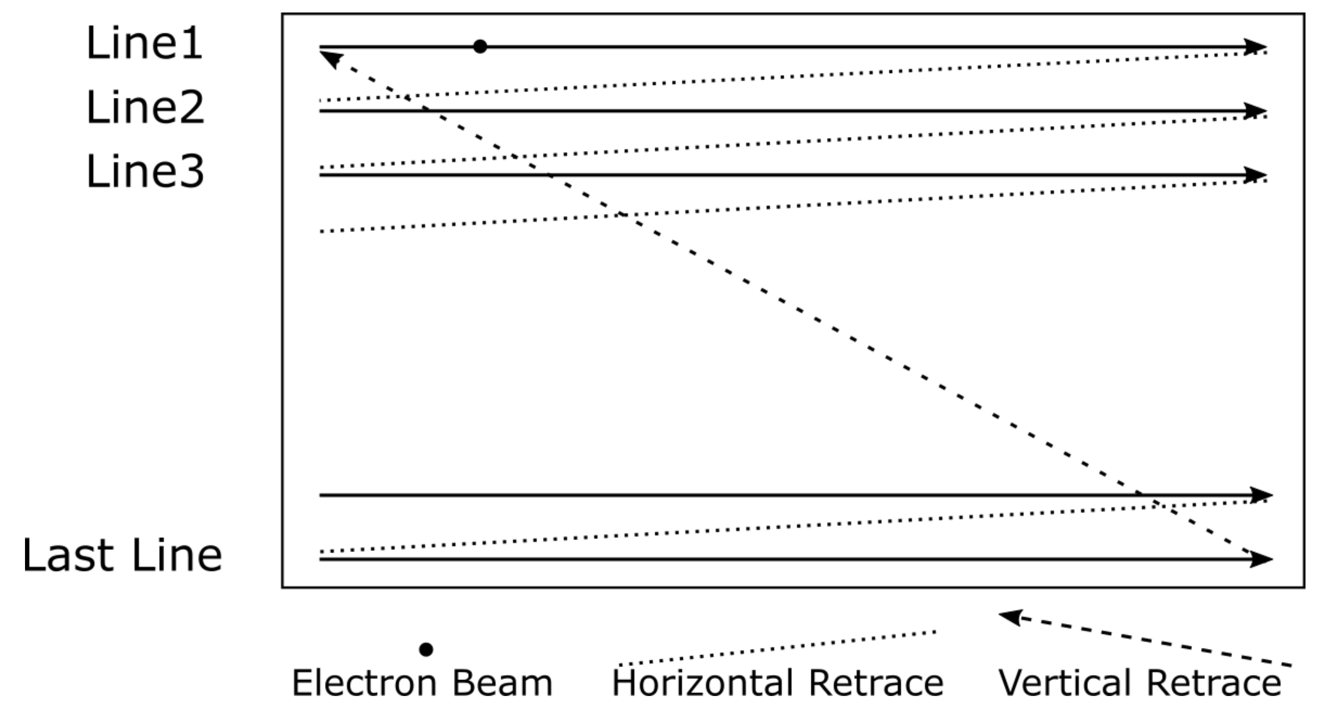


Figure 9. CRT display raster scanning

This horizontal retrace is not instantaneous. A certain amount of time is required to steer the beam from the right back to the left and then to stabilize the beam before the start of the next line. This entire process is known as the horizontal blanking interval. It consists of three distinct parts.

- Horizontal Front Porch (HFP): The time interval required to switch off and turn around the beam once the beam has passed the end of the display line.
- Horizontal Sync Pulse (H_{sync}): The time interval required to steer the beam back across the screen.
- Horizontal Back Porch (HBP): The time interval required to stabilize the beam before it is turned on to illuminate the next line.

The vertical retrace requires a certain amount of time which is called the vertical blanking interval. This interval consists of the following three parts.

- Vertical Front Porch (VFP): The time interval required to turn around the beam once the last line of the image is displayed.
- Vertical Sync Pulse (V_{sync}): The time interval required to steer the beam from the bottom to the top of the screen.
- Vertical Back Porch (VBP): The time interval required to stabilize the beam vertically before starting the first line.

Modern digital displays no longer need time to steer an electron beam around but they still make use of these blanking periods for things like image processing and sending additional data (audio in the case of HDMI, or embedded image data for CSI-2).

3.2.1 Pixels, Clocks, and Synchronization

The electron beam is a useful mental model for how digital display data is transferred through an SOC and across a DSI link. Pixel data is transmitted serially one after another moving left to right and top to bottom. There are gaps in this pixel data for the various blanking intervals. These blanking intervals are what define the image width and position relative to the display.

NOTE

(X,Y) pixel coordinate data is NOT stored or transmitted anywhere.

Figure 10 shows the relationship between all of the syncing signals and the active display area.

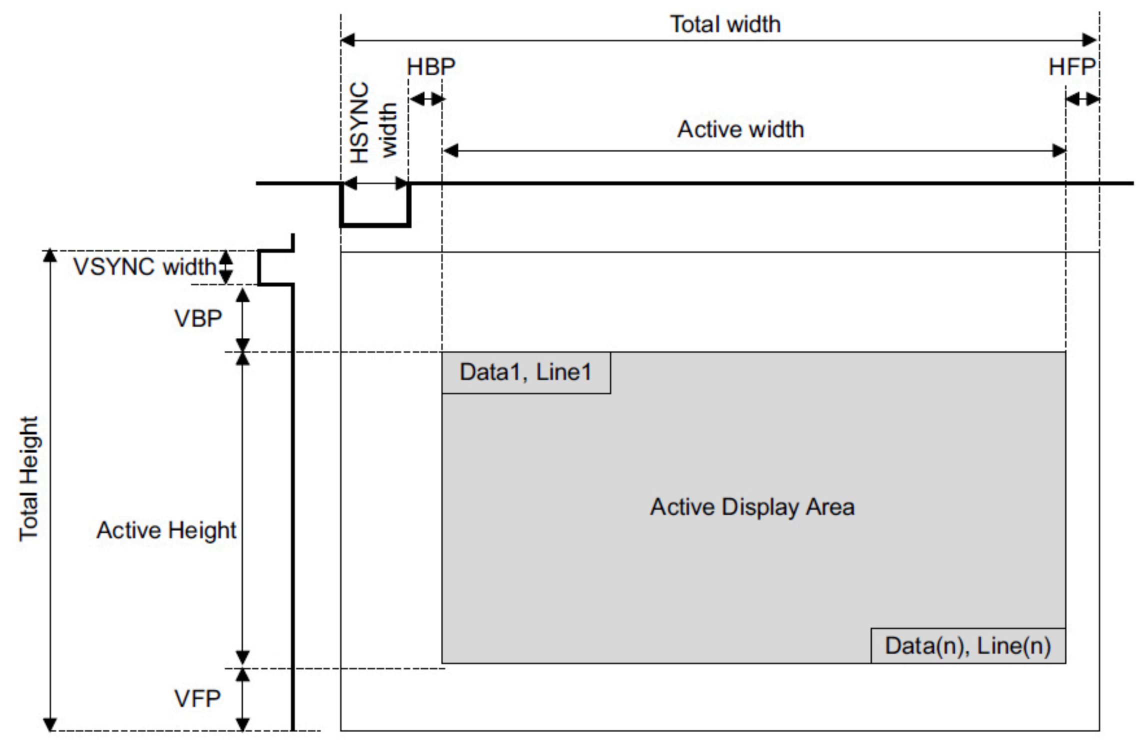


Figure 10. Digital display synchronization relationship

A single horizontal line of image data is signaled by the assertion of an HSYNC pulse. That pulse is followed by the HBP, then the active pixel data and finally the HFP. The HSYNC pulse, HBP, and HFP all use pixels for their unit of measurement.

A VSYNC pulse signals the start of a new frame. That pulse is followed by the VBP then the active image lines and finally the VFP. This process repeats for the next image frame. The VSYNC, VBP, and VFP all use horizontal image lines as their unit of measurement.

A clock must keep the pixel data and synchronization signals in order. This clock is known as a Pixel clock. A single pixel is transferred every clock cycle. It is the primary clock for most image display transport systems.

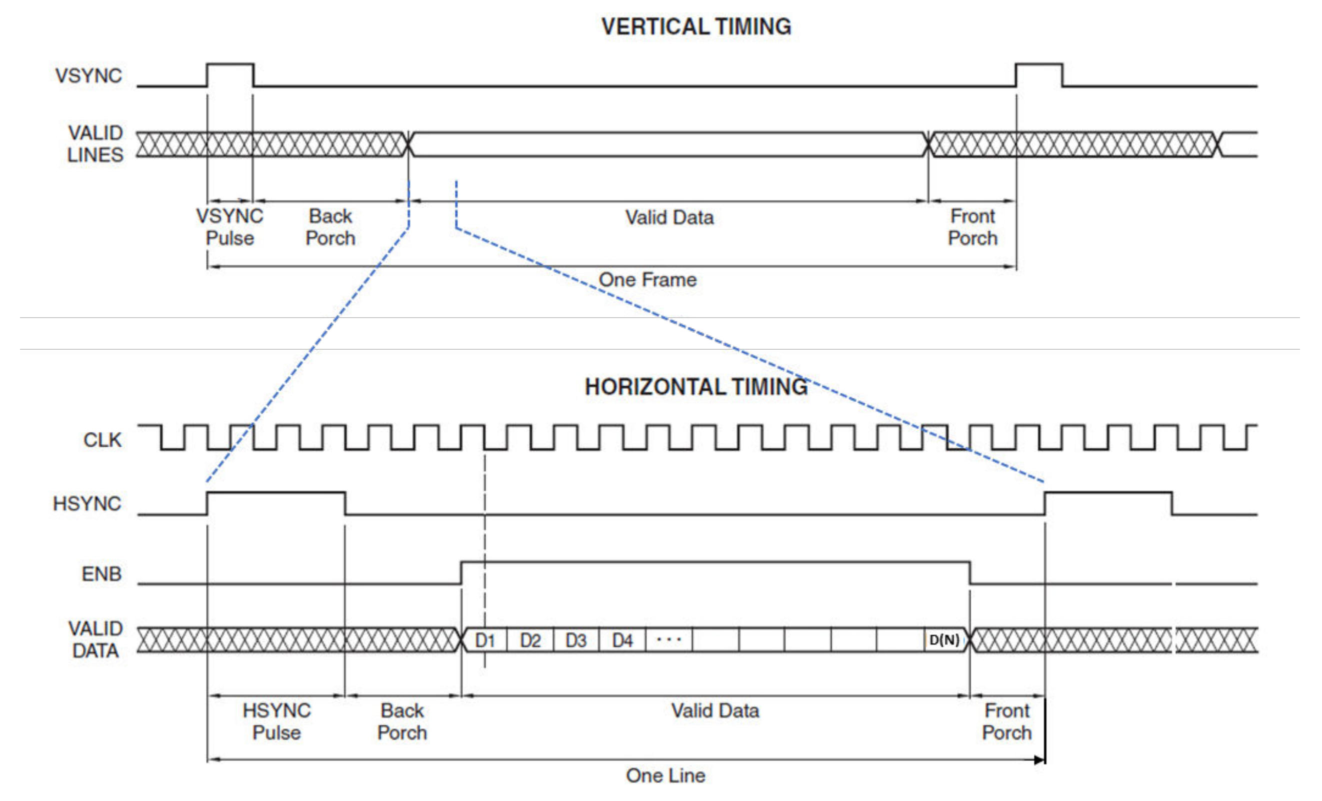


Figure 11. Vertical and horizontal timing relationship

Figure 11 shows the timing relationship between the HSYNC, VSYNC, pixel clock, and data. The data required for a single pixel is transmitted each pixel clock cycle.

The ENB signal is used to signal active pixel data transfers. Without it, the various pixel pipelines could mistake blanking data as pixel data with a 0 value.

3.2.1.1 Example – Pixel clock and resolution relationship

SMPTE 274 M standard defines the resolution and frame rate of what is referred to as **Full HD** TV. The standard defines 1920 horizontal **active** pixels and 1080 **active** lines giving a total of 1920 × 1080 = 2,073,600 active pixels. The standard calls for an HBP of 148 pixels, an HFP of 88 pixels, and an HSYNC width of 44 pixels for a total line width of 2200 pixels. The HD standard also specifies a VBP of 36 lines, a VFP of 4 lines, and a VSYNC width of 5 lines for a total of 1125 vertical lines.

Therefore, a full frame of HD TV data actually contains 2200 × 1125 = 2,475,000 pixels or pixel clocks.

In this example, the pixel clock required for a frame rate of 30 FPS can be calculated by multiplying the total pixels per frame by the desired frame rate. The pixel clock = 2.475 Mpix * 30fps = 74.25 MHz.

3.3 MIPI display pixel interface (DPI-2)

The Display Pixel Interface (DPI-2) is an MIPI Alliance standard for parallel interfaces to display modules without on-panel display controller or frame buffer. These display modules rely on a steady flow of pixel data from host processor to the display, to maintain an image without flicker or other visual artifacts. The DSI specification refers to DPI-2 mode of operation as video mode. This interface is commonly used by host processors and (non-MIPI DSI-compliant) LCD displays to transmit and synchronize video streams. Many SOCs use this interface (or something similar) to transport image data between IP blocks.

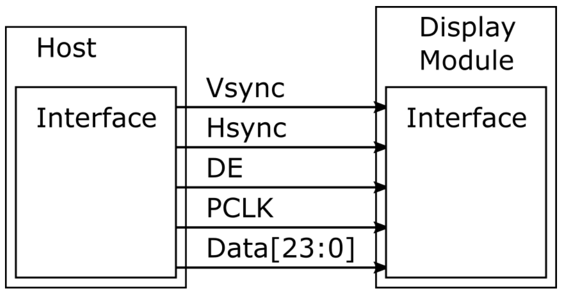


Figure 12. DPI interface signals

Figure 12 shows a generic DPI interface with the critical signals between the host processor and the display.

- **PCLK**: Pixel clock is the main system clock. Pixel data is transferred during one-pixel period. The display module uses the rising edge of PCLK to capture pixel data. This clock runs continuously and is the root clock for all other video timing signals.
- **Data[23:0]**: Each pixel value is transferred from the host processor to the display module with the parallel data bus. It can vary in width from 16 to 24.
- **Hsync**: The Horizontal Sync signals the beginning of each horizontal line of pixels and is defined using numbers of pixel clocks.
- **Vsync**: The Vertical Sync signal indicates the beginning of each frame of the displayed image and is defined using an integer multiple of display lines.
- **DE**: The Data Enable signal indicates when active pixel data is being transferred. This signal is sometimes also referred to as Line Valid (LV) or HREF.

3.4 MIPI display architectures

The MIPI DPI specification defines four different display module architectures. All MIPI-compliant displays follow one of these four architectures which share commonalities. The display driver IC is responsible for translating the incoming image data stream into the required control signal for a display panel. Figure 13 shows a basic display/host system architecture.

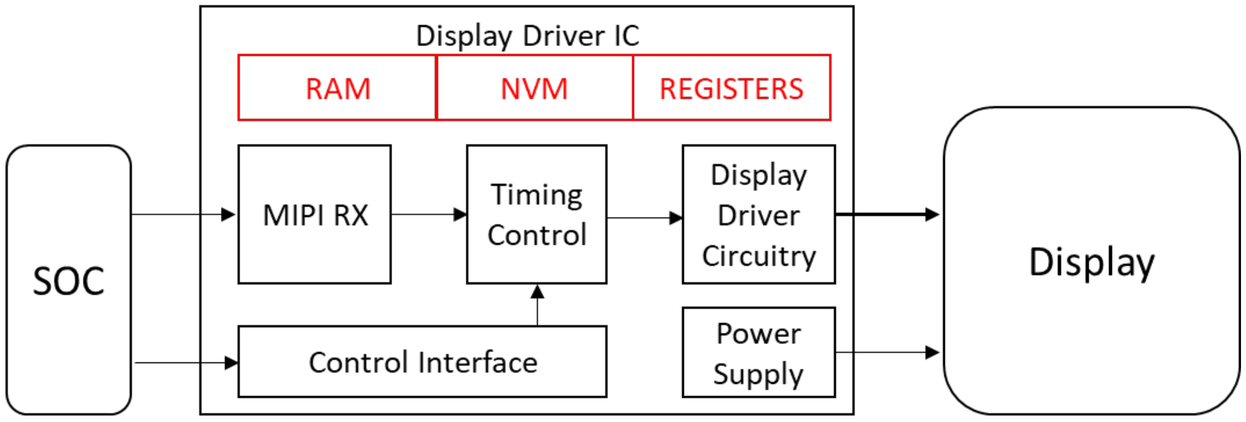


Figure 13. MIPI display architectures

The architecture types differ in the memory configurations. Display types 2, 3, and 4 are most common and supported by SOC. Table 11 lists the differences.

Table 11. MIPI display architecture type differences

Component	Type 1	Type 2	Type 3	Type 4
Full-frame memory	X			
Partial-frame memory		X		
Registers	X	X	X	
Non-volatile memory	X	X	X	
Video Stream Interface		X	X	X

3.5 DSI video mode timing

Figure 14 shows the relationship between the active display area and the horizontal and vertical timing. There are a few more timing parameters required to set up a DSI link between a host and display. Table 12 defines parameters provided by a display manufacturer. Sometimes these values are not in the display data sheet but in the display controller IC data sheet used by a display manufacturer.

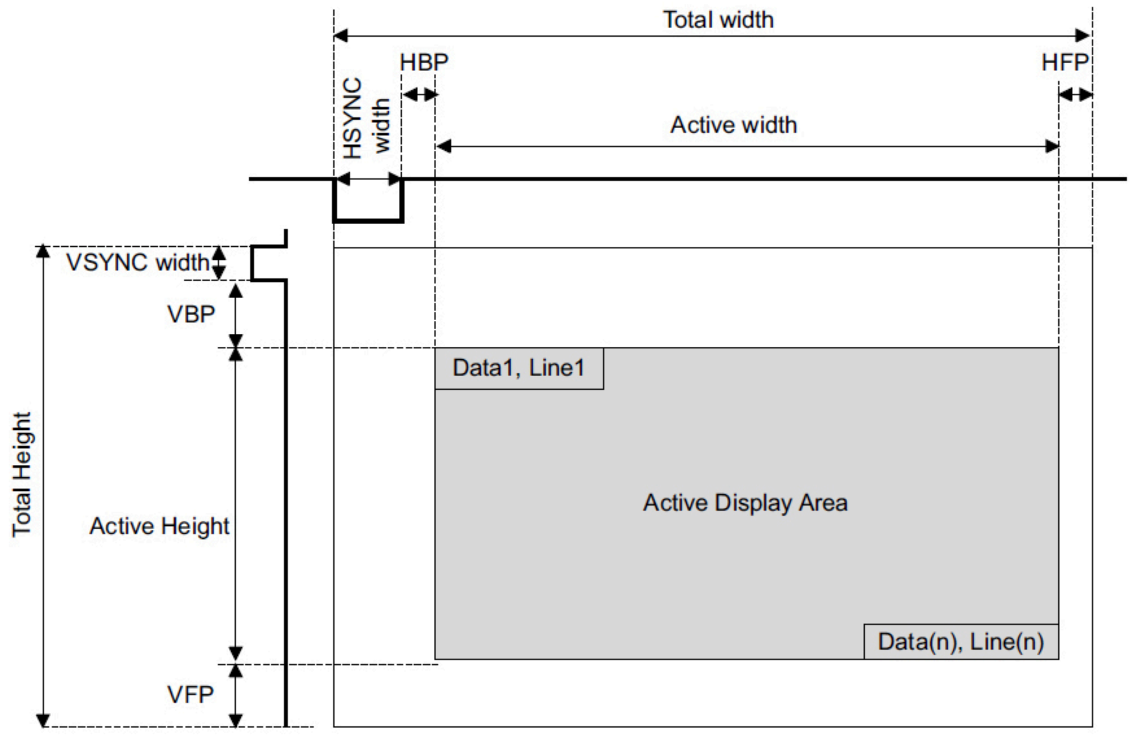


Figure 14. Display timing parameters

Table 12. Required display timing parameters

Parameter	Description	Min.	Max.	Unit	Comments
t_L	Line time			µs	
t_{HSA}	Horizontal sync active			µs	H_{sync} width
t_{HBP}	Horizontal back porch			µs	
t_{HACT}	Time for image data			µs	
HACT	Active pixels per line			Pixels	
t_{HFP}	Horizontal front porch			µs	
VSA	Vertical sync active			lines	V_{sync} width
VBP	Vertical Back Porch			lines	
VACT	Active lines per frame			lines	
VFP	Vertical Front Porch			lines	

3.6 Video packetization

At its core the DSI specification simply takes a parallel image bus data stream and serializes it for transfer over a high-speed D-PHY link. The DSI takes parallel data, signal events, and commands and converts them into packets, which then appends packet-protocol information and headers, and then sends complete bytes to the PHY for high speed transfer across the serial link.

Image data and synchronization information are sent in byte-sized packets using the D-PHY HS Transmission mode. There are two packet types – short or long:

- Short packets are used for most Command Mode commands and associated parameters. Other short packets convey events, such as, H_{sync} and V_{sync} edges. Because they are short packets, they can convey accurate timing information to logic at the peripheral. Short packets are always 4 bytes in length including the ECC. Short packets are also Packet Headers for long packets.
- Long packets are used to transmit large blocks of pixel or other data. Long packets specify the payload length using a two-byte Word Count field. Payloads may be anywhere from 0 to 2^{16} - 1 byte long. Every long packet starts with a four-byte packet header and ends with a two-byte packet footer. If the payload is empty, a long packet can be as short as 6 bytes. Maximum packet length is 65,541 bytes (2^16 - 1 + 6).

Figure 15 illustrates the features and differences between the short and long packets.

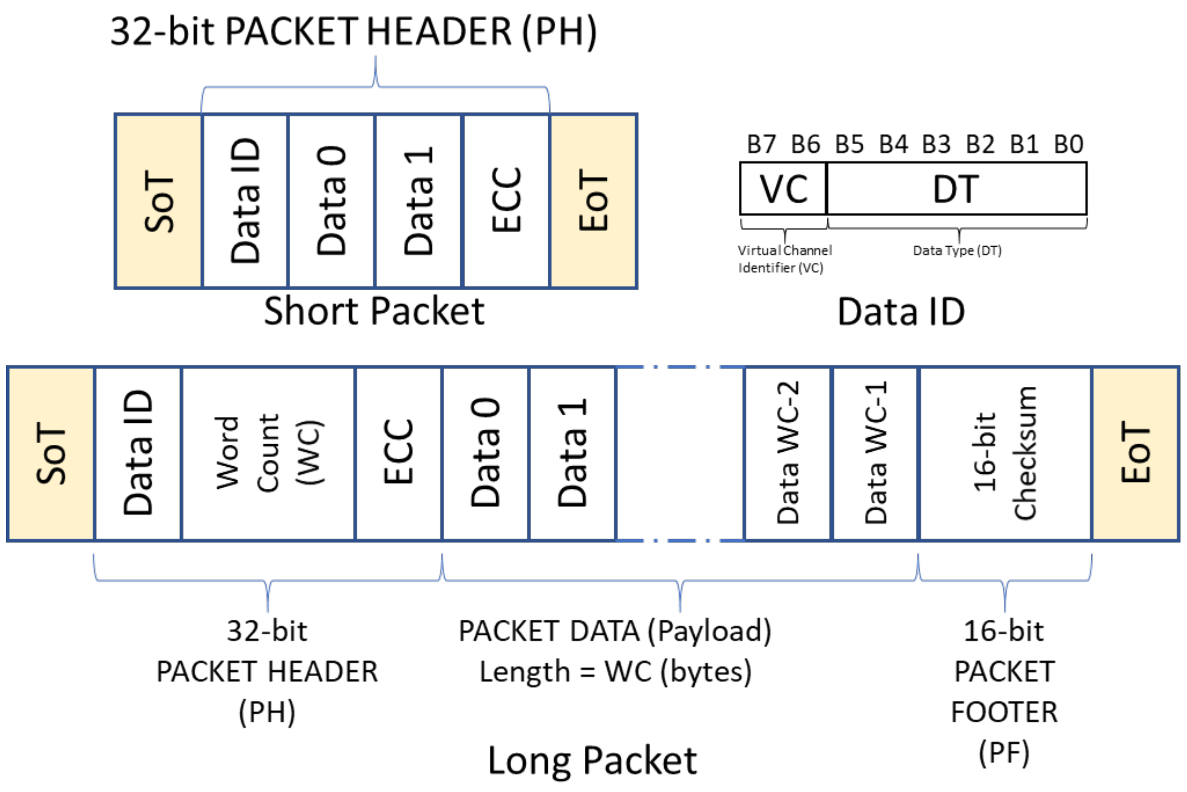


Figure 15. Packet structures

3.6.1 Packet header data types

The 6 bit field Data ID [5:0] in every packet header defines the packet type. Table 13 lists different data types for both short and long packets.

Table 13. Data types

Data type (hex)	Data type (binary)	Description	Packet type
0x01	00 0001	Sync Event, V Sync Start	Short
0x11	01 0001	Sync Event, V Sync End	Short
0x21	10 0001	Sync Event, H Sync Start	Short
0x31	11 0001	Sync Event, H Sync End	Short
0x08	00 1000	End of Transmission packet (EoTp)	Short
0x02	00 0010	Color Mode (CM) Off Command	Short
0x12	01 0010	Color Mode (CM) On Command	Short
0x22	10 0010	Shut Down Peripheral Command	Short

Table continues on the next page...

Table 13. Data types (continued)

Data type (hex)	Data type (binary)	Description	Packet type
0x32	11 0010	Turn On Peripheral Command	Short
0x03	00 0011	Generic Short WRITE, no parameters	Short
0x13	01 0011	Generic Short WRITE, 1 parameter	Short
0x23	10 0011	Generic Short WRITE, 2 parameters	Short
0x04	00 0100	Generic READ, no parameters	Short
0x14	01 0100	Generic READ, 1 parameter	Short
0x24	10 0100	Generic READ, 2 parameters	Short
0x05	00 0101	DCS Short WRITE, no parameters	Short
0x15	01 0101	DCS Short WRITE, 1 parameter	Short
0x06	00 0110	DCS READ, no parameters	Short
0x37	11 0111	Set Maximum Return Packet Size	Short
0x09	00 1001	Null Packet, no data	Long
0x19	01 1001	Blanking Packet, no data	Long
0x29	10 1001	Generic Long Write	Long
0x39	11 1001	DCS Long Write/write_LUT Command Packet	Long
0x0C	00 1100	Loosely Packed Pixel Stream, 20-bit YCbCr, 4:2:2 Format	Long
0x1C	01 1100	Packed Pixel Stream, 24-bit YCbCr, 4:2:2 Format	Long
0x2C	10 1100	Packed Pixel Stream, 16-bit YCbCr, 4:2:2 Format	Long
0x0D	00 1101	Packed Pixel Stream, 30-bit RGB, 10-10-10 Format	Long
0x1D	01 1101	Packed Pixel Stream, 36-bit RGB, 12-12-12 Format	Long
0x3D	11 1101	Packed Pixel Stream, 12-bit YCbCr, 4:2:0 Format	Long
0x0E	00 1110	Packed Pixel Stream, 16-bit RGB, 5-6-5 Format	Long
0x1E	01 1110	Packed Pixel Stream, 18-bit RGB, 6-6-6 Format	Long
0x2E	10 1110	Loosely Packed Pixel Stream, 18-bit RGB, 6-6-6 Format	Long
0x3E	11 1110	Packed Pixel Stream, 24-bit RGB, 8-8-8 Format	Long

Table continues on the next page...

Table 13. Data types (continued)

Data type (hex)	Data type (binary)	Description	Packet type
0xX0 and 0xXF, Unspecified	XX 0000 XX 1111	DO NOT USE All unspecified code are reserved	

The 2 bytes following the Data ID byte in a short packet only contain data when necessary. For example, a DCS Short Write, a Data ID of 0x15 defines one parameter command.

- Data 0 byte contains a DCS write command.
- Data 1 byte contains an optional write parameter.

The shutdown peripheral command defined by the Data ID of 0x22 has no additional data, so Data 0 and Data 1 bytes are empty.

The 2 bytes following the Data ID byte in a long packet always contain the word count. This value ranges from 0 to 65,535.

3.7 Video mode interface timing

Video Mode refers to operation in which transfers from the host processor to the peripheral take the form of a real-time pixel stream. In normal operation, to avoid flicker or other visible artifacts in the displayed image, the display module relies on the host processor to provide image data at sufficient bandwidth.

DSI supports several formats, or packet sequences, for Video Mode data transmission. The timing requirements of the peripheral dictate which format is appropriate. In the following sections, Burst Mode refers to time-compression of the RGB pixel (active video) portion of the transmission. In addition, these terms are used throughout the following sections:

- Non-Burst Mode with Sync Pulses: It enables the peripheral to reconstruct original video timing, including sync pulse widths.
- Non-Burst Mode with Sync Events: It is similar to above, but the accurate reconstruction of sync pulse widths is not required, so a single Sync Event is substituted.
- Burst mode: RGB pixel packets are time-compressed, leaving more time during a scan line for LP mode (saving power) or for multiplexing other transmissions onto the DSI link.

NOTE

For accurate reconstruction of timing, consider packet overhead including Data ID, ECC, and Checksum bytes. This effectively means that the D-PHY link is configured such that the bandwidth is sometimes greater than the minimum required for the video stream.

Figure 16 is a key for the following three figures which show the differences between the three Video Modes. The Blanking or Low-Power Interval (BLLP) is a period during which video packets, such as pixel-stream and sync event packets, are not actively transmitted to the peripheral.

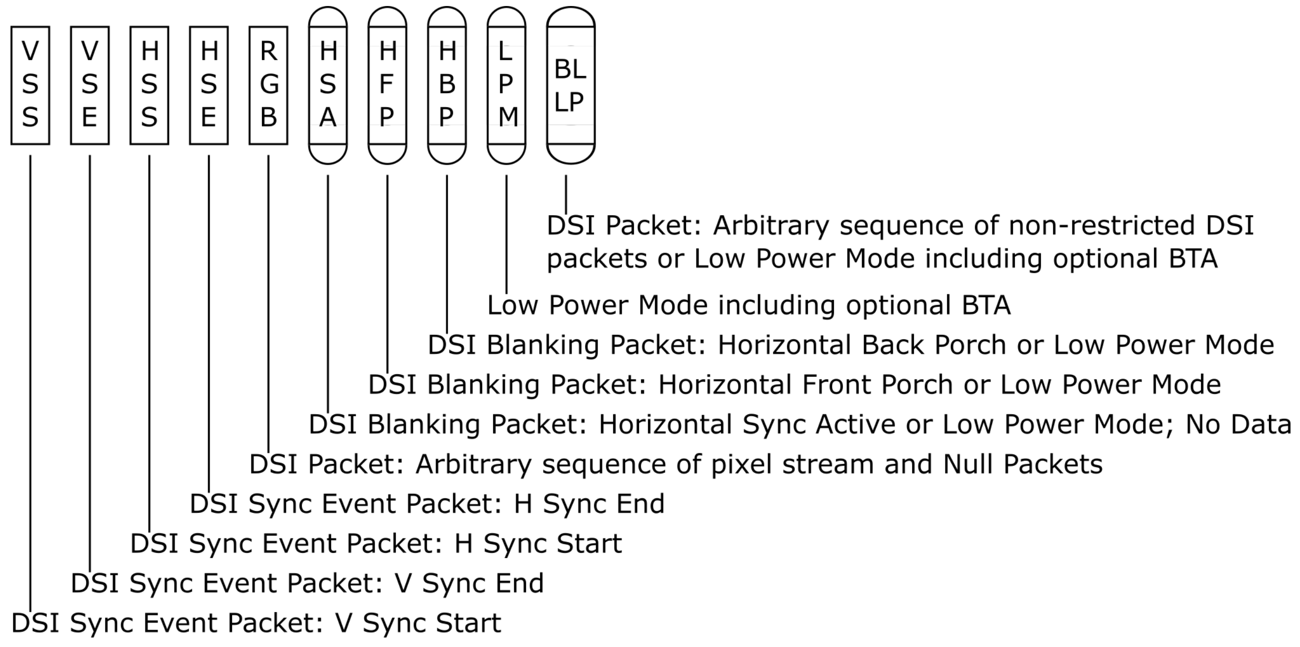

VSS
VSE
HSS
HSE
RGB
HSA
HFP
HBP
LPM
BLLP
DSI Packet: Arbitrary sequence of non-restricted DSI packets or Low Power Mode including optional BTA
Low Power Mode including optional BTA
DSI Blanking Packet: Horizontal Back Porch or Low Power Mode
DSI Blanking Packet: Horizontal Front Porch or Low Power Mode
DSI Blanking Packet: Horizontal Sync Active or Low Power Mode; No Data
DSI Packet: Arbitrary sequence of pixel stream and Null Packets
DSI Sync Event Packet: H Sync End
DSI Sync Event Packet: H Sync Start
DSI Sync Event Packet: V Sync End
DSI Sync Event Packet: V Sync Start

Figure 16. Video mode interface timing legend

3.7.1 Non-burst mode with sync pulses

With this format, the goal is to convey DPI-type timing over the DSI serial Link. It includes matching DPI pixel-transmission rates, and widths of timing events like sync pulses. Accordingly, synchronization periods are defined using packets transmitting both start and end of sync pulses. Figure 17 shows the relationship between the DPI-type timing and the DSI packets.

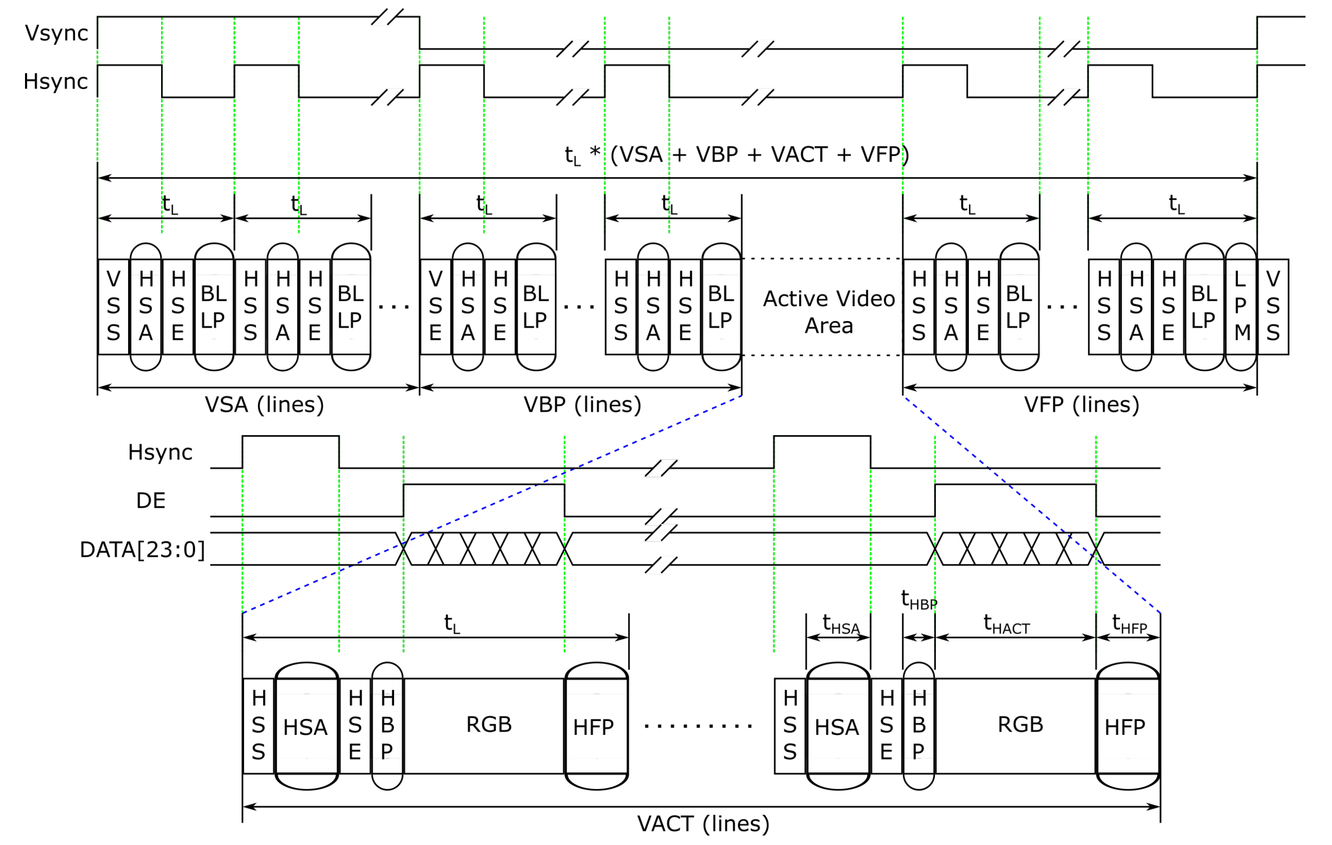


Figure 17. Video mode interface timing: Non-burst transmission with sync start and end

Normally, periods shown as Horizontal Sync Active (HSA), Horizontal Back Porch (HBP) and Horizontal Front Porch (HFP) are filled by Blanking Packets, with lengths (including packet overhead) calculated to match the period specified by the data sheet of the peripheral. Alternatively, if there is sufficient time to transition from HS to LP mode and back again, a timed interval in LP mode may substitute for a Blanking Packet. It saves power. During HSA, HBP and HFP periods, the bus should stay in the LP-11 state.

3.7.2 Non-burst mode with sync events

This mode is a simplification of the previous format described above. Only the start of each synchronization pulse is transmitted. The peripheral may regenerate sync pulses as needed from each Sync Event packet received. Pixels are transmitted at the same rate as they would in a corresponding parallel display interface such as DPI-2. Figure 18 shows the relationship between the DPI-type timing and the DSI packets.

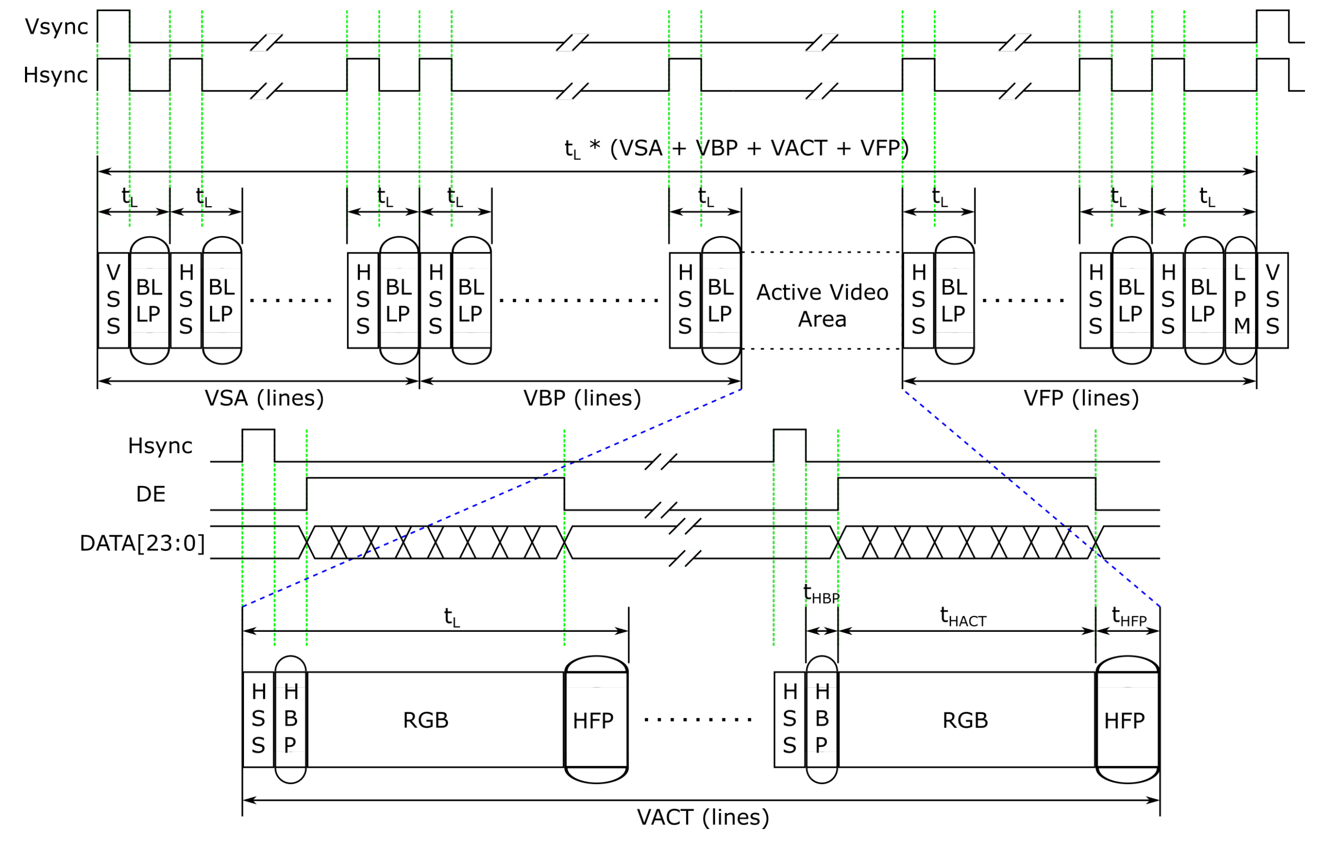


Figure 18. Video mode interface timing: Non-burst transmission with sync events

As with the previous Non-Burst Mode, if there is sufficient time to transition from HS to LP mode and back again, a timed interval in LP mode may substitute for a Blanking Packet. It saves power.

3.7.3 Burst mode

In this mode, blocks of pixel data can be transferred in a shorter time using a time-compressed burst format. This strategy reduces overall DSI power consumption and enables larger blocks of time for other data transmissions over the Link in either direction. Figure 19 shows the relationship between the DPI-type timing and the DSI packets.

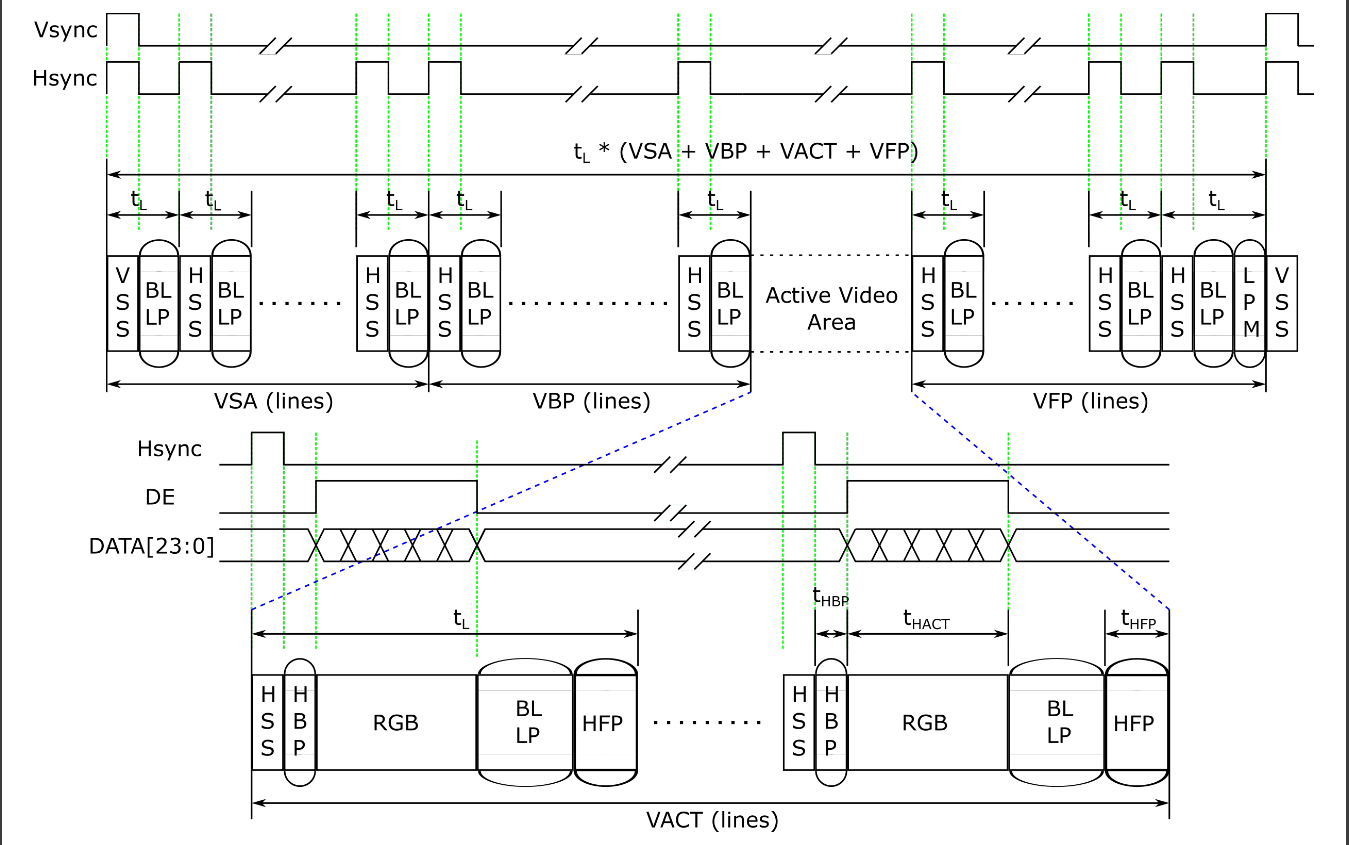


Figure 19. Video mode interface timing: Burst transmission

Similar to the Non-Burst Mode scenario, if there is sufficient time to transition from HS to LP mode and back again, a timed interval in LP mode may substitute for a Blanking Packet. It saves power.

3.7.4 Pixel formats

Since all D-PHY packets are a minimum of 1 byte in length, the DSI packs the various pixel formats into lowest common denominator byte lengths that allow for full transmission of a single pixel of data. This sometimes results in multiple pixels being packed together across multiple bytes. Table 14 lists the pixel bit depth and minimum byte values for the supported formats. The values are used for bandwidth calculations later.

Table 14. DSI pixel packing formats

Data format	Bits per pixel	Pixels per packet (min)	Packet Length (Bytes)[1]
Loosely Packed Pixel Stream, 20-bit YCbCr, 4:2:2	20	2	6
Packed Pixel Stream, 24-bit YCbCr, 4:2:2	24	2	6
Packed Pixel Stream, 16-bit YCbCr, 4:2:2	16	2	4
Packed Pixel Stream, 30-bit RGB, 10-10-10	30	4	15
Packed Pixel Stream, 36-bit RGB, 12-12-12	36	2	9
Packed Pixel Stream, 12-bit YCbCr, 4:2:0	12	2	3

Table continues on the next page...

Table 14. DSI pixel packing formats (continued)

Data format	Bits per pixel	Pixels per packet (min)	Packet Length (Bytes)[1]
Packed Pixel Stream, 16-bit RGB, 5-6-5	16	1	2
Packed Pixel Stream, 18-bit RGB, 6-6-6	18	4	9
Loosely Packed Pixel Stream, 18-bit RGB, 6-6-6	18	1	3
Packed Pixel Stream, 24-bit RGB, 8-8-8 Format	24	1	3

1. This value is the minimum packet length that aligns with an integer number of pixels. The total line width (displayed plus non-displayed pixels) must be a multiple of this value.

3.8 Bandwidth calculations

When evaluating various displays during system design, a few critical parameters must be calculated and compared against the host and display data sheets.

The pixel clock is the main clock for various display processing. The formula to calculate the required pixel clock rate is:

$$\text{Pixel Clock (Hz)} = \text{H}_{TOT} * V_{TOT} * \text{FPS}$$

Equation 1.

Where:

- H_{TOT} = total line width = active pixels + HSYNC + HFP + HBP (pixels)
- V_{TOT} = total frame height = active lines + VSYNC + VFP + VBP (lines)
- FPS = frames per second

The total system bandwidth is the pixel clock times the bits per pixel. It can be calculated using the following equation:

$$\text{Bandwidth (bps)} = \text{Pixel Clock} * \text{Bits Per Pixel}$$

Equation 2.

For the bits per pixel, see Pixel formats.

Data Rate Per Lane (DRPL) is the bandwidth divided by the number of data lanes in the system design. The DSI specification allows minimum one lane and maximum four lanes.

$$\text{Data Rate Per Lane (bps)} = \text{Bandwidth} / \text{number of data lanes}$$

Equation 3.

The MIPI interface is Double Data Rate, so the D-PHY HS Clock output frequency is half that of the Data Rate Per Lane:

$$\text{MIPI D} - \text{PHY Clock Rate (Hz)} = \text{Data Rate Per Lane} / 2$$

Equation 4.

3.9 DSI system clock configuration

The high-level MIPI DSI/D-PHY architecture is common across all i.MX RT and i.MX 8 family of parts. The signal/register names differ slightly but the functionality is common. Figure 20 gives a high-level overview of the clocking system and data interface.

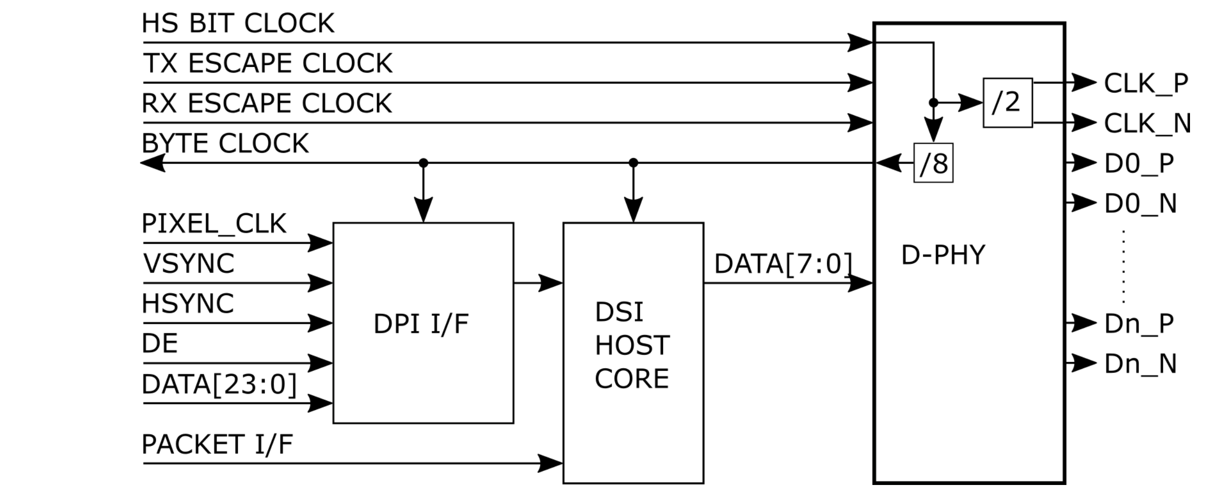


Figure 20. MIPI DSI/D-PHY clock structure

3.9.1 HS bit clock

D-PHY uses the HS bit clock to generate the high-speed DDR clock and byte clock. This clock speed is equal to the Data Rate Per Lane. In most SOCs, a dedicated D-PHY PLL generates this clock but the i.MX RT500 does not have a D-PHY PLL.

3.9.2 Byte clock

D-PHY generates the byte clock and the frequency is 1/8th the data rate of the HS Bit Clock. A D-PHY configured to generate high-speed data at 1 Gbps generates a 125 MHz byte clock. The packet interface of DSI host controller is synchronous to the byte clock.

3.9.3 Escape clock

MIPI D-PHY uses the TX escape clock for state control and low-power data transmission. The frequency of this escape clock ranges between 12 MHz and 20 MHz.

MIPI D-PHY uses the RX escape clock to reverse low-power data reception. The MIPI spec requires this clock frequency to be in the range of 67 % to 150 % of the display LP clock frequency.

3.9.4 Pixel clock

DPI-2 interface uses the Pixel clock. All DPI-2 signals are synchronous to this clock. The DPI bridge module of DSI host controller handles transferring video data received in the **Pixel Clock** domain over to the **Byte Clock** domain. The pixel and byte clock frequencies are related by the following formula:

$$\text{Byte Clock Freq} \;>\; = \;\text{Pixel Clock Freq} * \text{Bits Per Pixel} \,/\, (\,8 * (\text{DPHY\_LANE\_NUM}))$$

Equation 5.

Where:

- Byte Clock Freq = Frequency of Byte Clock = HS Bit Clock / 8
- Pixel Clock Freq = Frequency of the Pixel Clock on the DPI-2 interface.
- Bits Per Pixel = Size of pixels, in bits, on the DPI-2 interface (see Table 14).
- DPHY_LANE_NUM = The number of active MIPI D-PHY data lanes (1-4).

NOTE

If the byte clock frequency does not meet this requirement, the MIPI interface cannot keep up with the video stream on the DPI-2 interface resulting in dropped video lines.

4 DSI examples

The i.MX 8 and i.MX RT family use two different DSI/D-PHY modules based on the underlying process node. For information on the process node and part differences, see Features/Differences between i.MX 8M and i.MX RT parts.

4.1 i.MX RT1170 DSI example

The i.MX RT1170 EVK has an optional display panel that can be purchased separately on the NXP webpage. This display is a 5.5 inch LED TFT panel with a resolution of 720 pixels by 1280 pixels along with a capacitive touch overlay. Figure 21 shows the display connected to the RT1170 EVK.

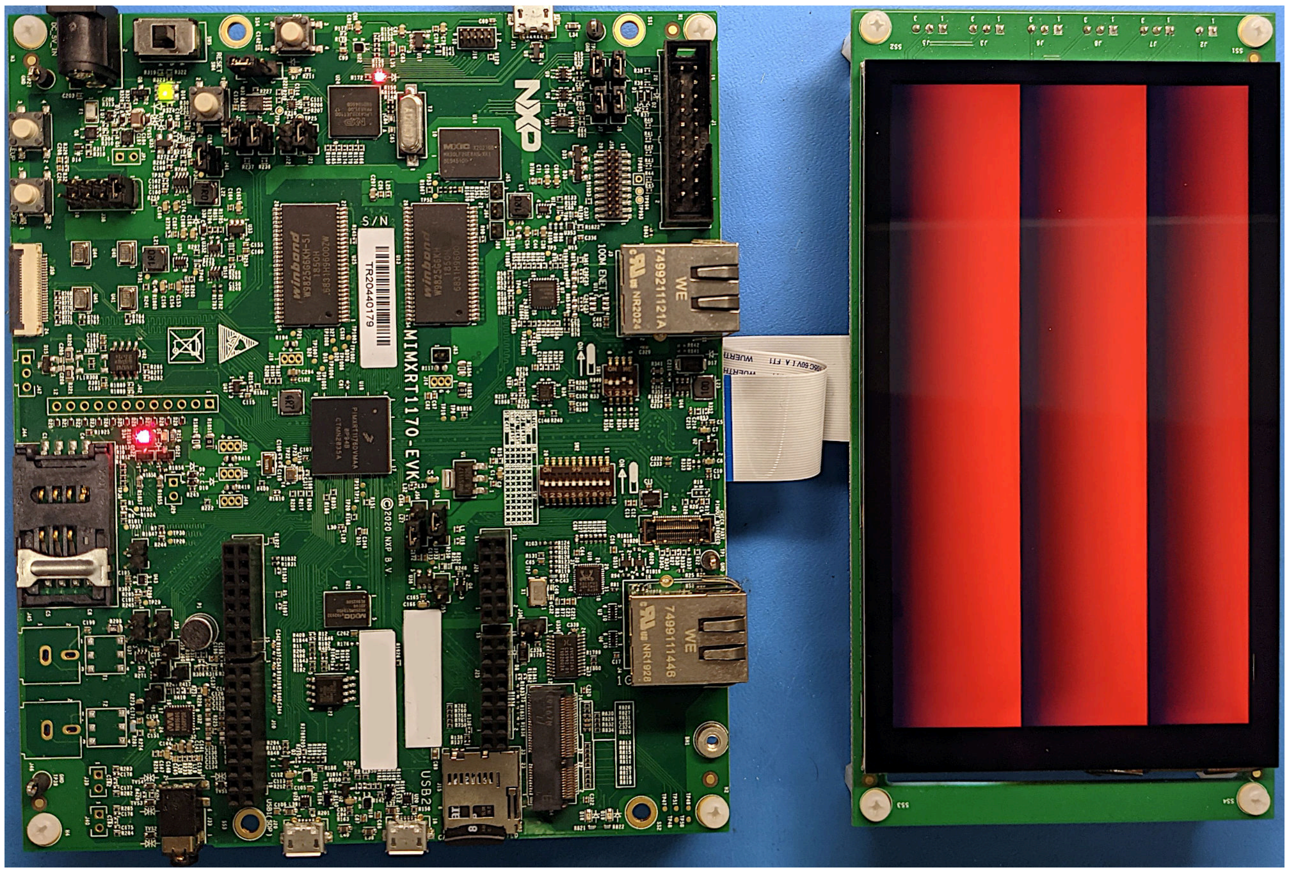


Figure 21. RT1170 with 5.5” TFT display

For this example, we walk through the ELCDIF RGB Demo included with the SDK to understand the required system setup. The ELCDIF RGB project shows how to drive the TFT panel using the eLCDIF driver. If this example runs correctly, a rectangle is moving in the screen, and the color changes every time it reaches the edges of the screen.

4.1.1 Initial system design

The balance between pixel depth, frame rate, and display resolution must be defined before various clocks and control registers can be set. Start by solving the four bandwidth calculations from Bandwidth calculations.

The TFT display uses a Raydium RM68200 display controller IC. The DSI video mode timing parameters from the RM68200 data sheet are shown in Table 15.

Table 15. DSI parameters for RM68200

Parameter	Description	Min.	Type	Max.	Unit	Condition
br_{PHY}	Bit rate total on all Lanes	250		850	Mbps	WXGA
TBR2	Total bit rate			1.7	Gbps	WXGA resolution, 2 lanes
t_L	Line time		12.8		µs	FPS = 60 Hz, VBP = 8, VFP = 8
t_{HSA}	Horizontal sync active				µs	
t_{HBP}	Horizontal back porch	0.5			µs	WXGA
t_{HACT}	Time for image data	9.6		19.2	µs	
HACT	Active pixels per line		800[1]		pixel	WXGA
t_{HFP}	Horizontal front porch	0.5			µs	
VSA	Vertical sync active	1			line	
VBP	Vertical Back Porch		8		line	
VACT	Active lines per frame		1280		line	
VFP	Vertical Front Porch	6	8		line	

1. The RM68200 can support 800 pixels per line but the 5.5in TFT LED display only supports 720.

In this example, the following parameters (pulled from the source code) are used:

- PANEL_HEIGHT = 1280
- PANEL_WIDTH = 720
- HSW = 8 (aka HSYNC or HSA)
- HFP = 32
- HBP = 32
- VSW = 2 (aka VSYNC or VSA)
- VFP = 16
- VBP = 14

Using those parameters and a frame rate of 60 Hz, the required pixel clock can be calculated as below:

- Pixel Clock = (1280 + 2 + 16 + 14) * (720 + 8 + 32 + 32) * 60 FPS = 62,346,240 Hz

The pixel format for this example is RGB888 which uses a bit depth of 24 bpp. The bandwidth for this setup can be calculated as:

- Bandwidth = 62,346,240 * 24 = 1,496,309,760 bps

This system uses a 2-lane MIPI interface, so the data rate per lane is:

- DRPL = 1,496,309,760 bps / 2 = 748,154,880 bps

The MIPI D-PHY HS Bit Clock therefore must be equal to or greater than the DRPL.

- MIPI HS Bit Clock ≥ 748,154,880 Hz

4.1.2 Clock setup

Clock setup is a common source of system problems. Root clock must be chosen. Dividers, multipliers, and/or PLL must be properly configured. This system example requires proper setup of the following clocks:

- Pixel clock
- Escape clocks
- MIPI HS bit clock

4.1.2.1 Pixel clock setup

The pixel clock is set first for the `eLCDIF` module. This module outputs the pixel clock along with the pixel data to the DPI bridge in the DSI module. `CLOCK_ROOT69` controls `ELCDIF_CLK_ROOT`. Inspection of the sample code shows the following register settings:

- CLOCK_ROOT69_CONTROL[MUX] = 4
- CLOCK_ROOT69_CONTROL[DIV] = 9

A mux setting of 4 selects the `SYS_PLL2_CLK` which runs at 528 MHz. The DIV setting of 9 gives an eLCDIF pixel clock = 528 MHz / 9 = 58.6666666667 MHz.

NOTE

58.67 MHz PCLK is less than 60 MHz, so the frame rate is less than 60 Hz.

4.1.2.2 Escape clock setup

`CLOCK_ROOT72` controls the TX escape mode clock root (`MIPI_ESC_CLK_ROOT`). Inspection of the sample code shows the following register settings:

- CLOCK_ROOT72_CONTROL[MUX] = 1
- CLOCK_ROOT72_CONTROL[DIV] = 1

A mux setting of 1 selects the OSC24M which is the 24 MHz oscillator. The DIV setting of 1 results in a divide by 1 so the escape clock root is set to 24 MHz. The RX and TX escape clocks have different requirements. There is a special clock divider just for the TX escape clock. The sample code for the escape clock divider has the following settings:

- EscClockGroupConfig.DIV0 = 1

This results in a TX escape clock of 12 MHz and an RX escape clock of 24 MHz.

NOTE

There can be some confusion surrounding the clock dividers of the i.MX RT family:

1. The Reference Manual shows that the clock root control register (`CLOCK_ROOT0_CONTROL` - `CLOCK_ROOT78_CONTROL`) adds 1 to the DIV selection: Divider-selected clock by DIV + 1.
2. The i.MX RT SDK function `CLOCK_SetRootClock()` *subtracts* 1 from the user-defined DIV. So the user does not have to remember that the divider is actually DIV + 1.
3. The i.MX RT SDK function `CLOCK_SetGroupConfig` does **NOT** subtract 1 from the user-defined DIV. So the user must remember that the actual divider = DIV + 1.

4.1.2.3 HS bit clock setup

The RT1170 D-PHY has an internal PLL which generates the HS Bit Clock. D-PHY automatically generates the byte clock. The byte clock is always 1/8 the frequency of the HS Bit Clock. Recall from the initial system design that the desired HS Bit Clock must be ≥ 748,154,880 Hz.

Since we have packet overhead (Data ID, ECC, Checksum bytes, and so on) to transfer along with the data, setting the HS Bit Clock to exactly 748,154,880 Hz does not give enough system bandwidth. In this example, we use a multiplier value of 1.125 which gives additional bandwidth headroom. The PLL requires a reference clock and three dividers (M, N, and O) to be configured for a required frequency. The D-PHY PLL Clock is configured using the following formula:

$$\mathrm{D-PHY\ PLL\ Clock\ output} = \mathrm{REF\_CLK} * (\mathrm{CM} / (\mathrm{CN} * \mathrm{CO}))$$

Equation 6.

Figure 22 shows the basic operation of the PLL.

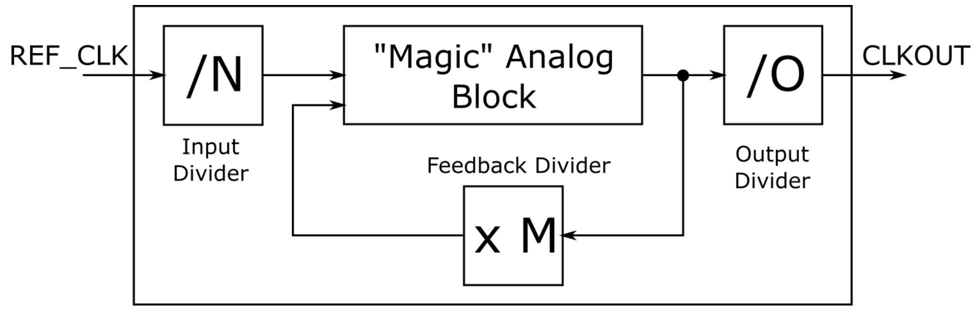


Figure 22. 28FDSOI D-PHY PLL block diagram

`CLOCK_ROOT71` controls the D-PHY PLL reference clock (`REF_CLK`) root (`MIPI_REF_CLK_ROOT`). Inspection of the sample code shows the following register settings:

- CLOCK_ROOT71_CONTROL[MUX] = 1
- CLOCK_ROOT71_CONTROL[DIV] = 1

This setting configures the D-PHY PLL reference root clock to use the 24 MHz oscillator.

Using the 58.667 MHz frequency from the pixel clock setup, 24 bpp, 2 data lanes, and the 1.125 multiplier, we get the following desired PLL frequency:

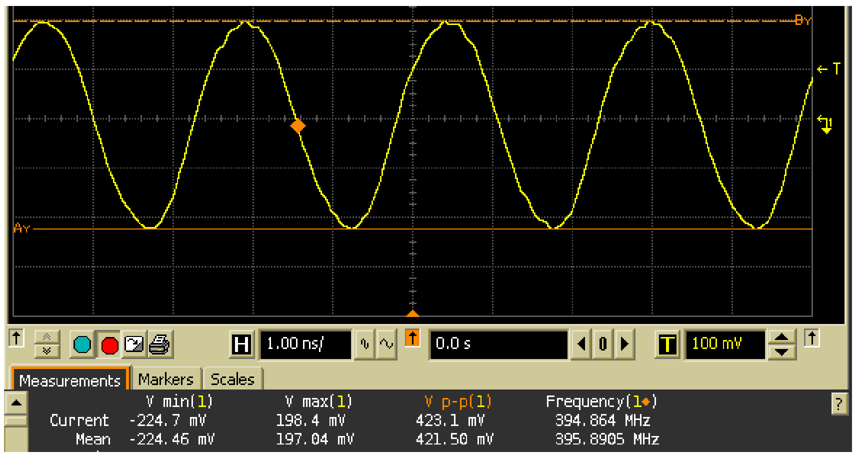
D-PHY Bit CLK = ((58.66667 * 24) / 2) * 1.125 = 792 MHz

When given a desired output frequency and reference clock frequency, the SDK function, `DSI_DphyGetPllDivider` (CN, CM, CO, `refClkFreq_Hz`, `desiredOutFreq_Hz`), attempts to find the correct divider values for CM, CN, and CO. If it cannot find workable values, it returns an error code. The sample code uses the following divider values:

- CN = 0x1F /* = 1 */
- CM = 0xC1 /* = 33 */
- CO = 0x0 /* = 1 */

This code results in the D-PHY PLL Clock output = 24 MHz * (33 / (1 * 1)) = 792 MHz.

Figure 23 is a differential scope plot of the D-PHY HS Transmission clock. Since this is a DDR clocking scheme, the actual frequency is 792 MHz / 2 = 396 MHz.



```
         V min(1)      V max(1)     V p-p(1)     Frequency(1◆)
Current  -224.7 mV     198.4 mV     423.1 mV     394.864 MHz
   Mean  -224.46 mV    197.04 mV    421.50 mV    395.8905 MHz
```

Figure 23. MIPI DSI D-PHY clock output (Differential)

4.1.3 D-PHY setup

As described in Start of HS Transmission, a few D-PHY parameters must be configured to correctly enter and exit HS Transmission mode.

Table 16 lists the registers that directly affect the D-PHY timing parameters for HS data transfer.

Table 16. RT1170 D-PHY configuration registers

Parameter	Register	Description
The following parameters adjust entering HS mode		
$T_{CLK-PRE}$	CFG_T_PRE[NUM_PERIODS]	The D-PHY timing parameter using byte clock periods. The minimum value for this port is 1. $T_{CLK-PRE}$ = NUM_PERIODS × (TxByteClkHS)[1] 0 = Zero clock periods (Not supported) 1 = One clock period (min required) 2 = Two clock periods ... ...
$T_{HS-PREPARE}$	M_PRG_HS_PREPARE	00: $T_{HS-PREPARE}$ = 1 × TxClkEsc Period 01: $T_{HS-PREPARE}$ = 1.5 × TxClkEsc Period 10: $T_{HS-PREPARE}$ = 2 × TxClkEsc Period 11: $T_{HS-PREPARE}$ = 2.5 × TxClkEsc Period
$T_{CLK-PREPARE}$	MC_PRG_HS_PREPARE	0: $T_{CLK-PREPARE}$ = 1 × TxClkEsc Period 1: $T_{CLK-PREPARE}$ = 1.5 × TxClkEsc Period
$T_{HS-ZERO}$	M_PRG_HS_ZERO	$T_{HS-ZERO}$ = (M_PRG_HS_ZERO + 6) × (TxByteClkHS Period)[1]

Table continues on the next page...

Table 16. RT1170 D-PHY configuration registers (continued)

Parameter	Register	Description
$T_{CLK-ZERO}$	MC_PRG_HS_ZERO	$T_{CLK-ZERO}$ = (MC_PRG_HS_ZERO + 3) × (TxByteClkHS Period)[1]
The following parameters adjust exiting HS mode		
$T_{CLK-POST}$	CFG_T_POST	$T_{CLK-POST}$ = NUM_PERIODS × (TxByteClkHS)[1] 0 = Zero clock periods (Not supported) 1 = One clock period (min required) 2 = Two clock periods ... ...
T_{EOT}	CFG_AUTOINSERT_EOTP	Enables the Host Controller to automatically insert an EoTp short packet when switching from HS to LP mode. 0: EoTp is not automatically inserted 1: EoTp is automatically inserted
	CFG_EXTRA_CMDS_AFTER_EOTP	Configures the DSI Host Controller to send extra End Of Transmission Packets after the end of a packet. The value is the number of extra EOTP packets sent. 0 = Zero packets 1 = One packet 2 = Two packets ... ...
$T_{HS-EXIT}$	CFG_TX_GAP	$T_{HS-EXIT}$ = NUM_PERIODS × (TxByteClkHS)[1] 0 = Zero clock periods (Not supported) 1 = One clock period (min required) 2 = Two clock periods ... ...
$T_{HS-TRAIL}$	M_PRG_HS_TRAIL	$T_{HS-TRAIL}$ = (M_PRG_HS_TRAIL) × (TxByteClkHS Period)[1]
$T_{CLK-TRAIL}$	MC_PRG_HS_TRAIL	$T_{CLK-TRAIL}$ = (MC_PRG_HS_TRAIL) × (TxByteClkHS Period)[1]

1. TxByteClkHS = MIPI HS Bit Clock Rate/8

4.1.3.1 HS transmission mode start setup

As shown in Table 5, a few timing parameters must be configured to properly enter HS Transmission mode. Figure 24 is a scope plot showing:

- the MIPI DSI Clock lane using a differential probe
- the MIPI_DSI_DP0 and MIPI_DSI_DN0 signals using a single ended probe

Figure 24 illustrates the timing parameters as configured by the driver code as the D-PHY transitions from LP to HS mode.

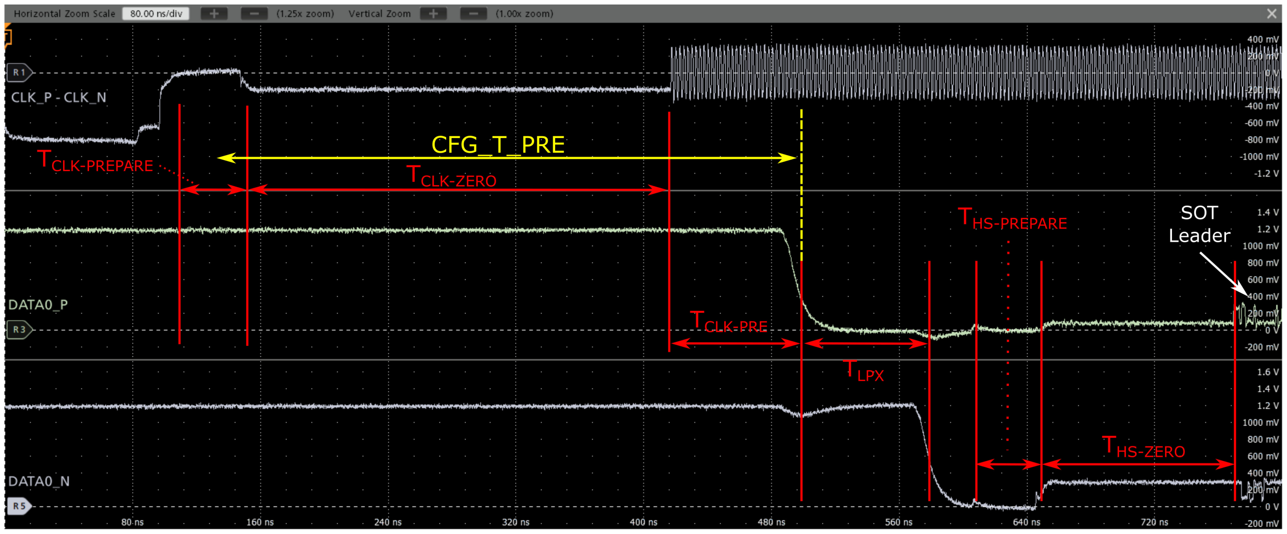


Figure 24. Transition from LP to HS mode

For this example, the configuration registers refer to the following time periods:

- UI is the MIPI HS Bit Clock period which in this example = 1/792 MHz = 1.26263 ns
- TxByteClkHS period = 8 × MIPI HS Bit Clock = 8 × 1.26263 ns = 10.10 ns
- TxClkEsc period = 1 / 12 MHz = 83.33 ns

4.1.3.1.1 TCLK-PRE

The $T_{CLK-PRE}$ parameter is adjusted through the `CFG_T_PRE[NUM_PERIOD]` register. The parameter sets the number of `TxByteClkHS` clock periods that the controller waits after enabling the clock lane for HS operation before enabling the data lanes for HS operation. The MIPI spec calls for a minimum $T_{CLK-PRE}$ of 8 × 1.2626 ns = 10.10 ns. In this example, the $T_{CLK-PRE}$ is about 80 ns.

The `CFG_T_PRE` timer does not start after the $T_{CLK-ZERO}$ timing period. It starts back near the $T_{CLK\_PREPARE}$ time window. Therefore, the `CFT_T_PRE` value is much larger than the required T_{CLK_PRE} value.

- CFG_T_PRE = 0 × 24 = 36
- CFG_T_PRE = 36 × 10.10 ns = 363.6 ns

4.1.3.1.2 HS prepare timer

The $T_{CLK-PREPARE}$ must be between 38 ns and 95 ns. $T_{HS-PREPARE}$ must be between (40 ns + 4 × UI) = 45 ns and (85 ns + 6 × UI) = 92.576 ns.

- MC_PRG_HS_PREPARE[1:0] = 0
- $T_{CLK-PREPARE}$ = 1 × 41.67 ns = 41.67 ns
- M_PRG_HS_PREPARE[1:0] = 0
- $T_{HS-PREPARE}$ = 1 * 41.67 ns = 41.67 ns

4.1.3.1.3 HS zero timer

$T_{CLK\text{-}PREPARE}$ + $T_{CLK\text{-}ZERO}$ must be > 300 ns. Since $T_{CLK\text{-}PREPARE}$ is 41.67 ns, $T_{CLK\text{-}ZERO}$ must be greater than 258.33 ns (300 – 41.67 = 258.33).

- MC_PRG_HS_ZERO = 0 × 17 = 23
- $T_{CLK\text{-}ZERO}$ = (23 + 3) × (10.10 ns) = 262 ns

$T_{HS\text{-}ZERO}$ must be > (105 ns + 6 × UI) = 112.58 ns.

- M_PRG_HS_ZERO = 0x6
- $T_{HS\text{-}ZERO}$ = (6 + 6) × (10.10 ns) = 121.2 ns

4.1.3.2 HS transmission mode exit setup

As described in Table 7, a few timing parameters affect the end of an HS transmission as it transitions back to LP mode. Figure 25 illustrates these parameters as configured by the driver code as the D-PHY transitions from HS to LP mode.

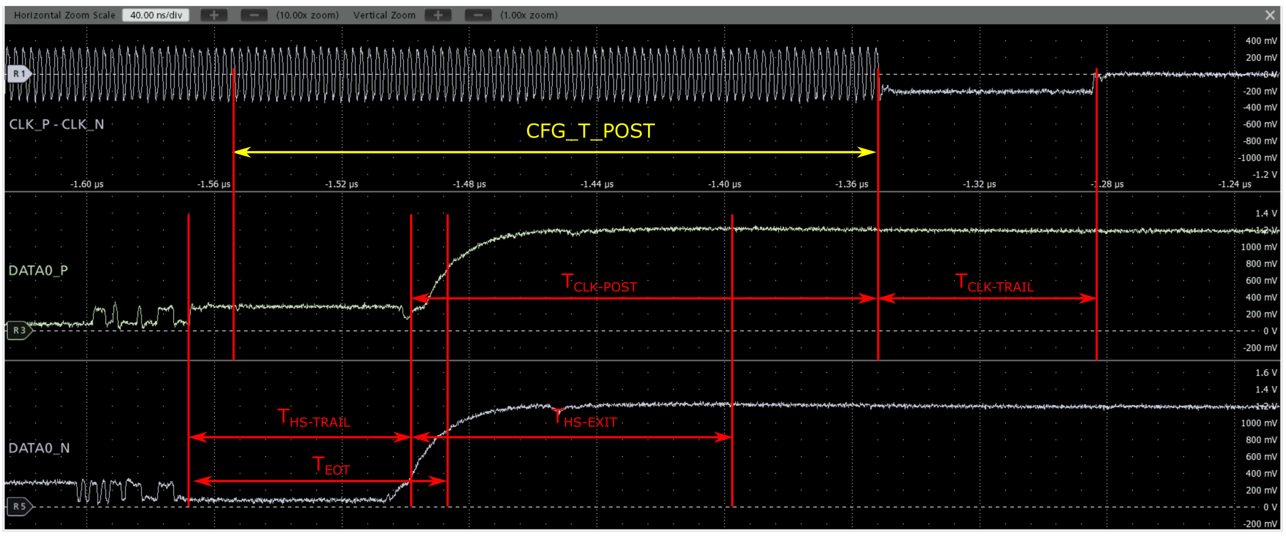


Figure 25. Transition from HS mode to LP mode

4.1.3.2.1 $T_{CLK\text{-}POST}$

The `CFG_T_POST` register is used to adjust the $T_{CLK\text{-}POST}$ parameter in `TxByteClkHS` period multiples. $T_{CLK\text{-}POST}$ must be greater than (60 ns + 52 × UI) = 125.66 ns. In this example, $T_{CLK\text{-}POST}$ is about 148 ns.

- CFG_T_POST[NUM_PERIODS] = 0 × 14 = 20
- CFG_T_POST period (ns) = 20 × 10.10 = 202 ns

4.1.3.2.2 T_{EOT}

T_{EOT} must be less than (105 ns + 12 × UI) = 120.15 ns.

The `CFG_AUTOINSERT_EOTP` and `CFG_EXTRA_CMDS_AFTER_EOTP` registers can be used to adjust this parameter. In this example, CFG_AUTOINSERT_EOTP = 0 so no extra packets are sent and T_{EOT} is about 84 ns.

4.1.3.2.3 $T_{HS\text{-}EXIT}$

The `CFG_TX_GAP` register is used to adjust the $T_{HS\text{-}EXIT}$ parameter in `TxByteClkHS` period multiples. $T_{HS\text{-}EXIT}$ must be greater than 100 ns.

- CFG_TX_GAP[NUM_PERIODS] = 0xA = 10

- $T_{HS\text{-}EXIT}$ = 10 * 10.10 = 101 ns.

4.1.3.2.4 HS trail timer

$T_{CLK\text{-}TRAIL}$ must be larger than 60 ns and $T_{HS\text{-}TRAIL}$ must be configured. It is between (60 ns + 4 × UI) = 65.05 ns and (105 ns + 12 × UI) = 120.15 ns.

- MC_PRG_HS_TRAIL = 0x7
- M_PRG_HS_TRAIL = 0x7
- $T_{CLK\text{-}TRAIL}$ = 7 × 10.10 ns = 70.7 ns
- $T_{HS\text{-}TRAIL}$ = 7 × 10.10 ns = 70.7 ns

4.1.4 DPI-2 video timing recreation over DSI

The DSI module has a DPI-2 Host Bridge Core which:

- accepts a MIPI-compliant DPI-2 set of signals (V_{sync}, H_{sync}, pixel data, and so on)
- creates DSI packets for the video timing events and video data
- transmits those packets via the packet interface of DSI host controller

This module is designed to preserve the video timing by properly spacing the DSI packets that carry the DPI video information and by inserting blanking packets into the packet stream.

To achieve this goal, the DSI host controller must have the HFP, HBP, and HSA configuration registers properly set. The values of these registers reflect the absolute time of the HFP, HBP, and HSA in terms of DSI packet bytes. Figure 26 shows the relationship between the DPI interface with its pixel clock based timing and the DSI interface with its byte-sized packet based timing for a single display line.

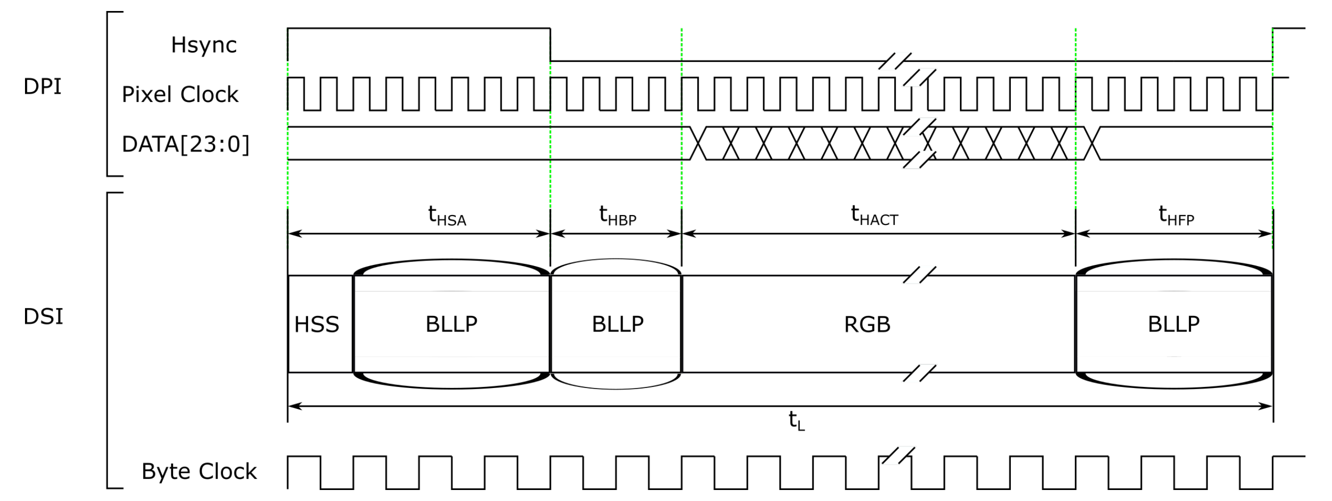


Figure 26. Relationship between DPI video timing and DSI packets – Burst mode

To relate pixel sizes on the DPI interface to DSI packet bytes, use the following relationship:

$$\text{DPI Event (time)} = [\text{Transmit Time}] * [\text{DSI Bytes}]$$

Equation 7.

A **DPI Event** is the length of time required by a DSI parameter: HSA, HFP, or HBP.

$$\text{DPI Event} = \text{DPI\_Event\_Size (pixels)} / \text{DPI\_Pclk\_Frequency (Hz)}$$

Equation 8.

$$\text{Transmit Time} = 1 / (\text{MIPI Byte Clock (Hz)} * \text{Number of Lanes})$$

Equation 9.

To solve `number_of_dsi_bytes`, use the following formula:

$$\text{DSI\_Bytes} = (\text{DPI\_Event\_size} * \text{MIPI Byte Clock} * \text{cfg\_num\_lanes}) / \text{DPI\_Pclk\_Frequency}$$

Equation 10.

Where:

- `DSI_Bytes`: Value to set the HFP, HBP, or HSA registers to.
- `DPI_Event_size`: Size of the video event (HFP, HBP, HSA) in pixels.
- `DPI_Pclk_frequency`: Frequency of pixel clock (Hz).
- MIPI Byte Clock: MIPI data lane byte clock frequency (Hz) in high-speed mode.
- `cfg_num_lanes`: Number of active D-PHY lanes.

4.1.4.1 DPI-2/DSI bridge setup

Table 17. 28 nm DPI-2/DSI bridge configuration registers

Register	Description
PIXEL_PAYLOAD_SIZE[15:0]	Maximum number of pixels to be sent as one DSI packet. Recommended being evenly divisible by the line size (in pixels).
PIXEL_FIFO_SEND_LEVEL[15:0]	To optimize DSI utility, the DPI bridge buffers some DPI pixels before initiating a DSI packet. This configuration port controls the level at which the DPI Host bridge begins sending pixels.
INTERFACE_COLOR_CODING[2:0]	Sets the distribution of RGB bits within the 24-bit data bus, as specified by the DPI specification. 000: 16-bit Configuration 1 001: 16-bit Configuration 2 010: 16-bit Configuration 3 011: 18-bit Configuration 1 100: 18-bit Configuration 2 101: 24-bit 110, 111: Reserved
PIXEL_FORMAT[1:0]	Sets the DSI packet type of the pixels 00: 16 bit 01: 18 bit 10: 18 bit loosely packed

Table continues on the next page...

Table 17. 28 nm DPI-2/DSI bridge configuration registers (continued)

Register	Description
	11: 24 bit
VSYNC_POLARITY[0]	Sets polarity of `dpi_vsync_input` 0: active low 1: active high
HSYNC_POLARITY[0]	Sets polarity of `dpi_hsync_input` 0: active low 1: active high
VIDEO_MODE[1:0]	Select DSI video mode that the host DPI module generates packets for. 00: Non-Burst mode with Sync Pulses 01: Non-Burst mode with Sync Events 10: Burst mode 11: Reserved, not valid
HFP[15:0]	Sets the DSI packet payload size, in bytes, of the horizontal front-porch-blanking packet
HBP[15:0]	Sets the DSI packet payload size, in bytes, of the horizontal back-porch-blanking packet.
HSA[15:0]	Sets the DSI packet payload size, in bytes, of the horizontal sync width filler-blanking packet.
ENABLE_MULT_PKTS[0]	Enable Multiple packets per video line. When enabled, `PIXEL_PAYLOAD_SIZE` must be set to exactly half the size of the video line. 0: Video Line is sent in a single packet 1: Video Line is sent in two packets
VBP[7:0]	Sets the number of lines in the vertical back porch.
VFP[7:0]	Sets the number of lines in the vertical front porch.
BLLP_MODE[0]	Optimize BLLP periods to Low-Power mode when possible. 0: Blanking packets are sent during BLLP periods 1: LP mode is used for BLLP periods
USE_NULL_PKT_BLLP[0]	Selects type of blanking packet to be sent during BLLP. 0: Blanking packet used in BLLP region 1 1: Null packet used in BLLP region
VACTIVE[13:0]	Sets the number of lines in the vertical active area.

Table continues on the next page...

Table 17. 28 nm DPI-2/DSI bridge configuration registers (continued)

Register	Description
VC	Sets the Virtual Channel (VC) of packets to be sent to the receive packet interface. Packets with VC not equal to this value are discarded and the **DSI VC ID Invalid** bit (bit 12) in the DSI error report is set.

The `DSI_SetDpiConfig()` function sets the DPI configuration registers. Below we examine every register setting from Table 17.

-
  ```
  PIXEL_PAYLOAD_SIZE = 720
  ```
  This register is the active horizontal line width (HACT) in pixels.

-
  ```
  PIXEL_FIFO_SEND_LEVEL = 720
  ```
  This register sets the limit at which point the DSI initiates packet transfer.

-
  ```
  INTERFACE_COLOR_CODING = 101
  ```
  This register configures the DPI input for 24-bit RGB pixel format.

-
  ```
  PIXEL_FORMAT = 11
  ```
  This register configures the DSI output packets for 24-bit RGB pixel format.

-
  ```
  VIDEO_MODE = 10
  ```
  This register configures the video mode interface to use burst mode.

-
  ```
  ENABLE_MULT_PKTS = 0x0
  ```
  One whole line of pixels is sent using a single packet.

-
  ```
  VBP = 0xE = 14
  ```
  The Vertical Back Porch is 14 lines.

-
  ```
  VFP = 0x10 = 16
  ```
  The Vertical Front Porch is 16 lines.

-
  ```
  BLLP_MODE = 0x1
  ```
  This register configures the MIPI lanes to enter LP mode for BLLP periods.

-
  ```
  USE_NULL_PKT_BLLP = 0x0
  ```
  With `0x0` during BLLP periods, the DSI sends **blanking packets**, as opposed to Null packets.

-
  ```
  VACTIVE = 0x4ff = 1279
  ```
  This register sets the total number of vertical active lines. Since the first line starts at 0, this number is equal to VACT – 1.

The RT1170 does not support virtual channels but other 28FDSOI parts (such as, the i.MX 8Q) do.

4.1.4.2 Byte length-blanking parameter conversion

Most configuration is fairly straight forward and pulled from the display data sheet but the horizontal blanking parameters do require some manipulation to convert between the pixel time and the DSI-byte time. Recall from earlier in this section the relationship between a DPI and DSI event is governed by the equation:

$$\text{DPI Event} = [\text{Transmit Time}] * [\text{DSI Bytes}]$$

Equation 11.

Using Equation 11, we can create a simple multiplication coefficient to translate DPI bit lengths to DSI byte lengths:

$$\text{UINT32 coff} = (\text{numLanes} * \text{dsiHsBitClkFreq\_Hz}) \;/\; (\text{dpiPixelClkFreq\_Hz} * 8\text{U});$$

Equation 12.

In this example, the coefficient works out to:

*coff = (2 * 792 MHz) / (58.667 MHz * 8) = 3*

Recall the initial horizontal blanking parameters defined by the RM68200 data sheet:

- HSA = 8
- HFP = 32
- HBP = 32

After multiplying the three values by the coefficient (3), we get:

```
            HSA  =  0x18  =  24
            HFP  =  0x60  =  96
            HBP  =  0x60  =  96
```

As shown in Figure 26, HSA is signaled using the HSS short byte packet type followed by a long blanking packet. The width of the two packets is 24 bytes long.

The HFP and HBP are both signaled using long blanking packets 96 bytes in length.

4.1.5 DSI example timing summary

Figure 27 shows the MIPI CLOCK and DATA0 lanes transitioning from LP to HS mode and back for a single line of pixels. The actual frame rate of our example is 58.6667 MHz/1,039,104 pixels = 56.459 Hz.

Since there are 1312 vertical lines, each line takes approximately 1 / (frame rate) / Vtot = 1 / 56.459 Hz / 1312 = 13.5 us of time to transmit the pixel data. The markers in Figure 27 show the timing for exactly one horizontal image line.

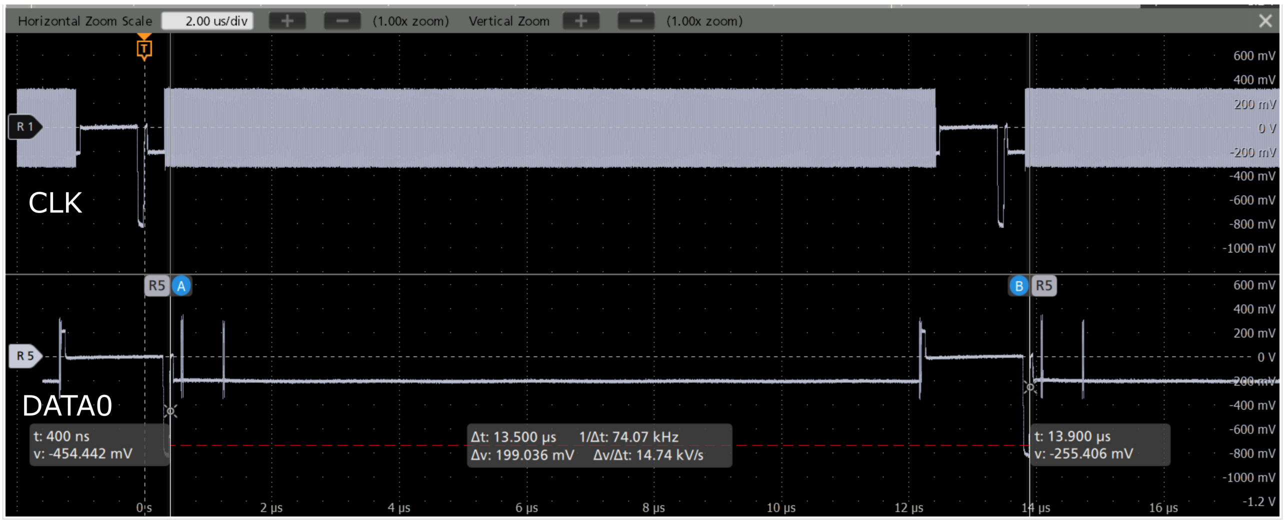


Figure 27. DSI single active line timing

5 CSI-2

This section explains the CSI-2 interface and how a camera or image sensor connects to a processor. After reading this section, an engineer can evaluate various image sensors or camera modules and decide the suitability and compatibility with any i.MX processors described in this document. A system engineer can specify the hardware of a full image system. This section does NOT describe the details of implementing a full image system in software. It does not cover information, such as, ISP calibration, exposure control, white balance, software drivers.

5.1 Overview

The CSI-2 specification follows the same general architecture as the DSI specification. It takes a parallel image data bus and serializes it into byte size data blocks for high speed transmission on a PHY. The biggest difference is the addition of a separate control interface and the support for embedded data in the image frame.

The Camera Control Interface (CCI) is a subset of the I^{2}C protocol, including the minimum combination of obligatory features for I^{2}C slave devices specified in the I^{2}C specification. Therefore, transmitters complying with the CCI specification can also be connected to the system I^{2}C bus. It is important that I^{2}C masters do not try to utilize those I^{2}C features not supported by CCI masters and CCI slaves.

The data transmission 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. Figure 28 shows the various layers comprising the data transmission system between a transmitter (image sensor) and a receiver (host processor).

- Pixel/Byte Packing/Unpacking Layer. The CSI-2 specification supports image applications with varying pixel formats from 6 to 24 bits per pixel. 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. 8 bits per pixel data are transferred unchanged by this layer.
- Low Level Protocol (LLP) includes the means of establishing bit-level and byte-level synchronization for serial data transferred between Start of Transmission (SoT) and End of Transmission (EoT) events and for passing data to the next layer. The minimum data granularity of the LLP is 1 byte. The LLP also includes assignment of bit-value interpretation within the byte, such as, Endian assignment.
- Lane Management. CSI-2 is Lane-scalable for increased performance. Depending on the bandwidth requirements of the application, the number of data Lanes may be one, two, three, or four. The transmitting side of the interface distributes (distributor function) bytes from 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 into a recombined data stream. The stream restores the original stream sequence.

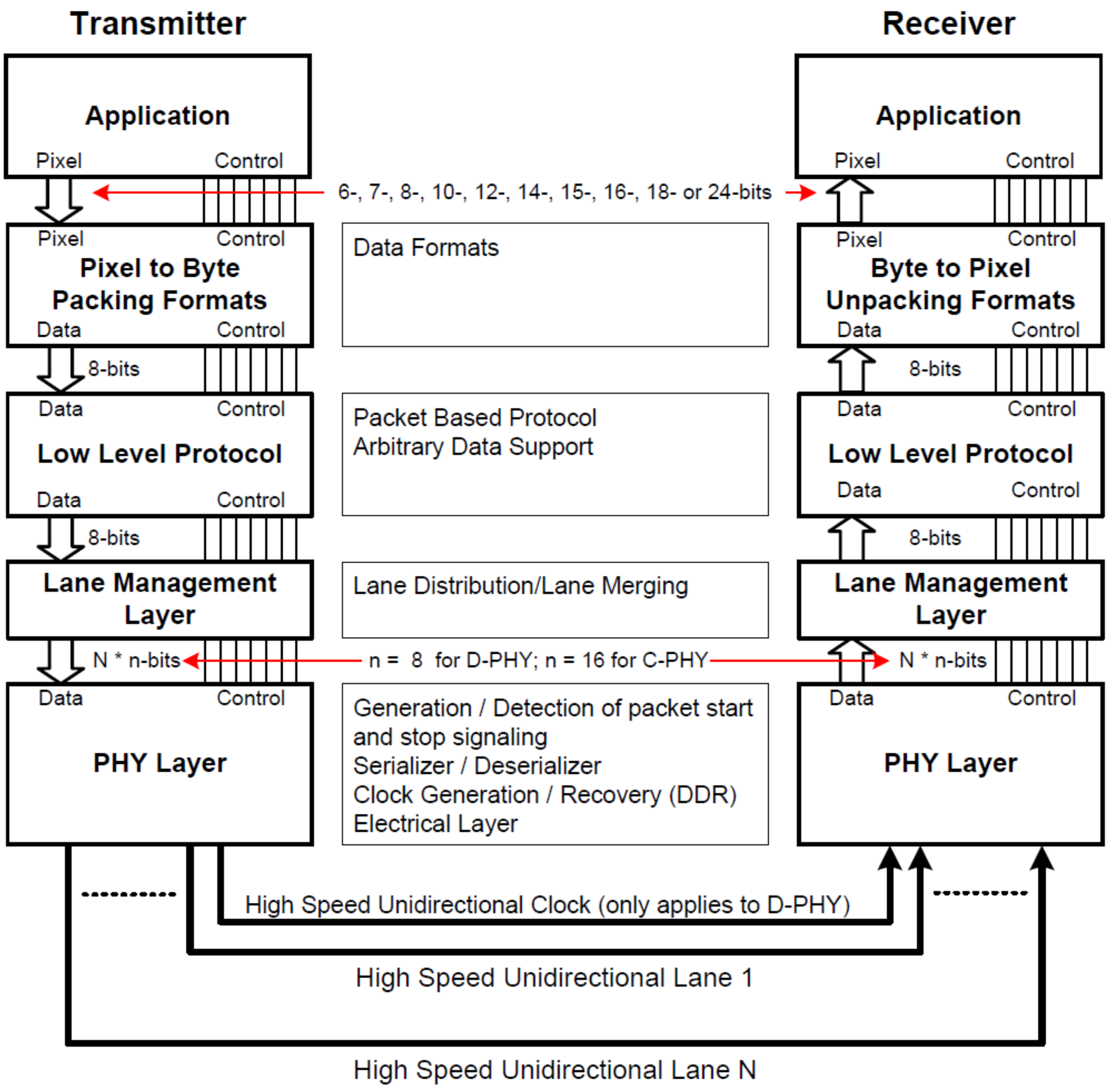

Transmitter
Receiver
Application
Pixel
Control
6-, 7-, 8-, 10-, 12-, 14-, 15-, 16-, 18- or 24-bits
Pixel to Byte Packing Formats
Byte to Pixel Unpacking Formats
Data
8-bits
Data Formats
Low Level Protocol
Packet Based Protocol
Arbitrary Data Support
Lane Management Layer
Lane Distribution/Lane Merging
N * n-bits
n = 8 for D-PHY; n = 16 for C-PHY
PHY Layer
Generation / Detection of packet start and stop signaling
Serializer / Deserializer
Clock Generation / Recovery (DDR)
Electrical Layer
High Speed Unidirectional Clock (only applies to D-PHY)
High Speed Unidirectional Lane 1
High Speed Unidirectional Lane N

Figure 28. CSI-2 layer definitions

5.2 Image sensor basics

It is helpful to have a basic understanding of image sensors when working with a CSI-2 system. This document uses the OV5640 image sensor from OmniVision as an example. This sensor is 5 Megapixels. It supports two MIPI data lanes and is the basis of the OV5640MRFL camera module commonly used by many i.MX 6/7/8/RT EVKs.

An image sensor contains an array of photosensitive elements. Each element converts light into a (RAW) pixel. The OV5640 has a pixel array of 2624 columns by 1964 rows for a total of 5,153,536 pixels. Not all pixels are used for image capture. As shown in Figure 29, the active columns are columns 16 – 2607 and the active lines are lines 14 - 1957, giving a maximum useable image array of 2592 × 1944 = 5,038,848 active pixels.

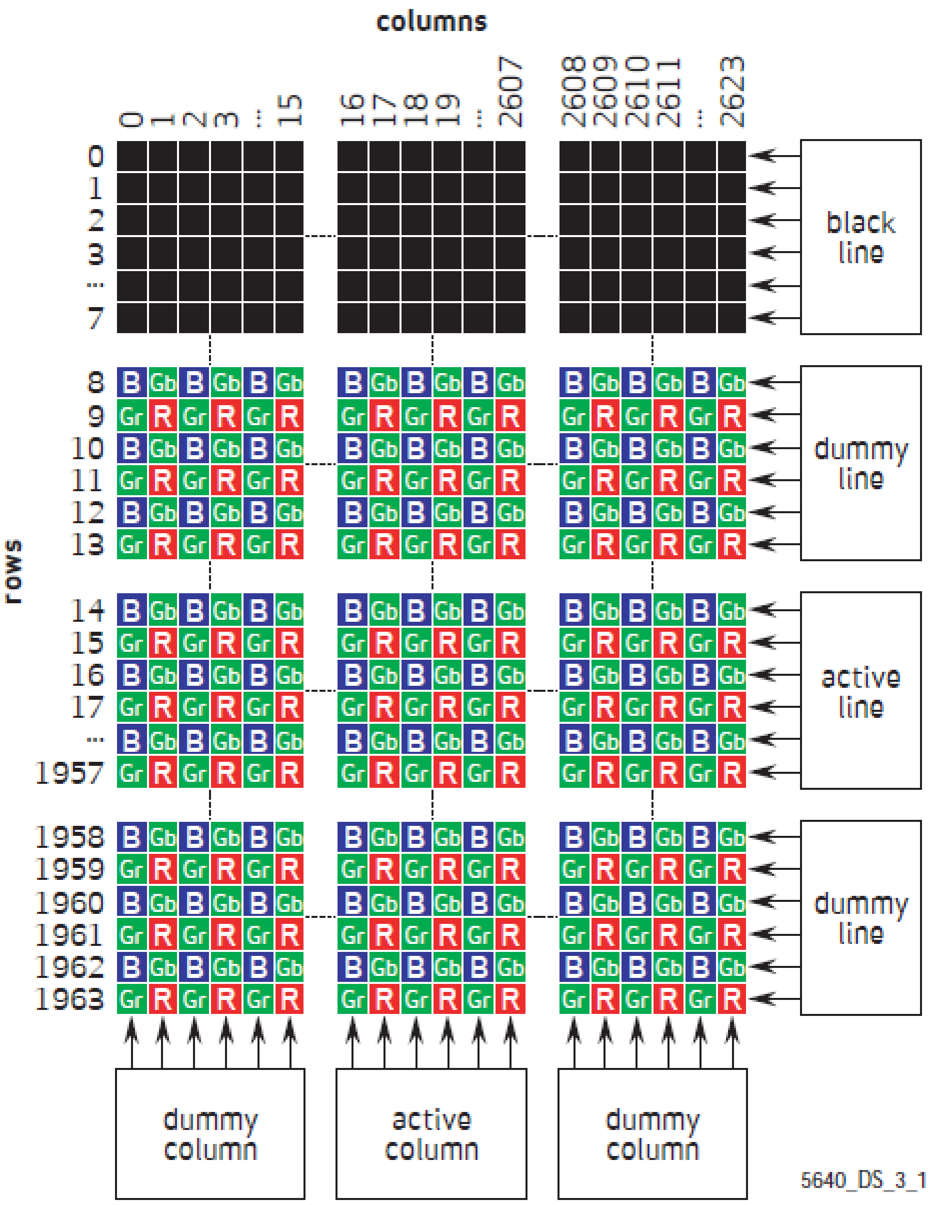


Figure 29. OV5640 image sensor layout

An ADC on the image sensor samples the voltage of each pixel one at time and outputs a digital value onto a data bus. The bus width is equal to the ADC resolution bit-depth. This unprocessed digital value is known as a RAW format. The OV5640 has a 10-bit ADC and a 10-bit wide data bus, so outputs are in a native **RAW10** format. Image sensors can range in resolutions from 6 bits up to 24 bits. The CSI-2 spec only supports a maximum RAW bit depth of 14 bits. ADC reads the value of each pixel across an entire line, moves to the next line, and repeats the process until the entire sensor pixel array or *frame* is read. This process sounds familiar - it is the same raster scan display method as explained in DSI.

A frame of RAW image data goes through a series of post processing steps, such as, demosaicing, color correction, white balance. Before, it is considered useable or pleasing to the eye. Those algorithms are either handled on the image sensor or during post-processing using an ISP. This document does not cover those topics. They do not affect the system specifications, such as, bandwidth, data lanes, and clocks. This document also ignores various image system details, such as, rolling versus mechanical shutter, flash sync, exposure.

5.2.1 Image sensor video timing

Most CMOS image sensors can be programmed to generate a single image (like a still camera) or a video stream. This document focuses on video streams. The video stream sent out by most image sensors follows the same video format described in the DSI section.

The pixel clock time period sets the time interval for the ADC conversion of each pixel. It is the root clock for HSYNC and VSYNC signals. HSYNC signals the start of each line and VSYNC signals the start of each new frame.

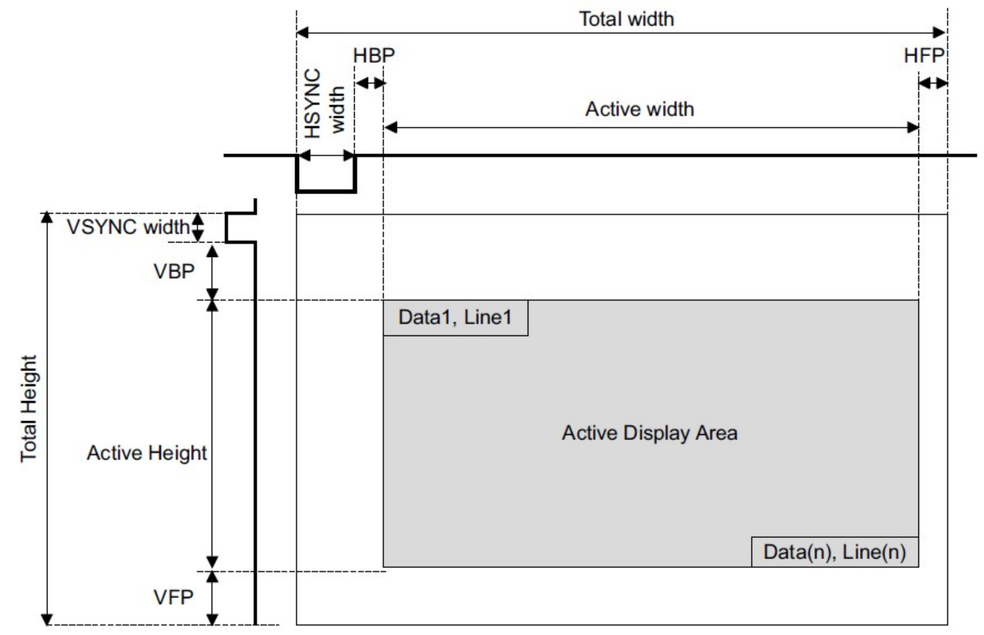


Figure 30. Image sensor video timing

5.3 CSI-2 video timing

The CSI-2 low-level protocol turns the pixel data, VSYNC, and HSYNC pulses into a series of packets. The packets are sent over the physical D-PHY interface while maintaining the critical timing parameters relationship to each other. Figure 31 shows the basic CSI-2 frame structure of a video stream using the RGB888 pixel format.

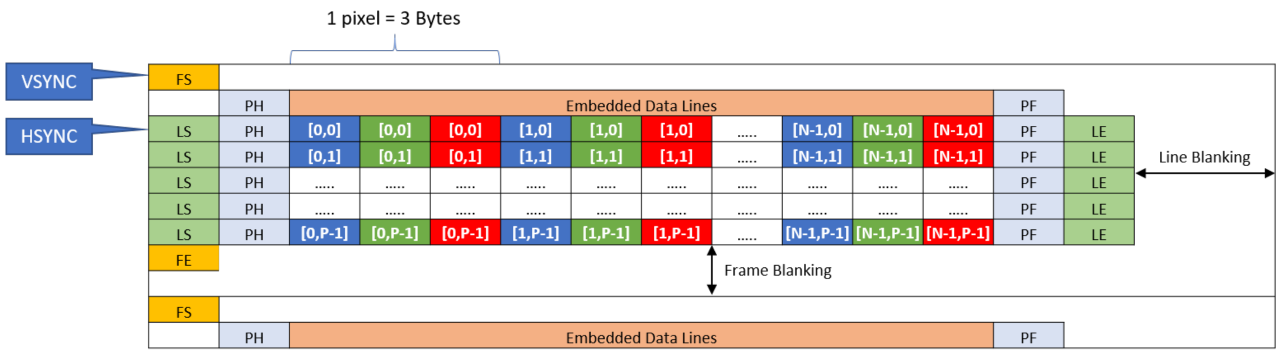
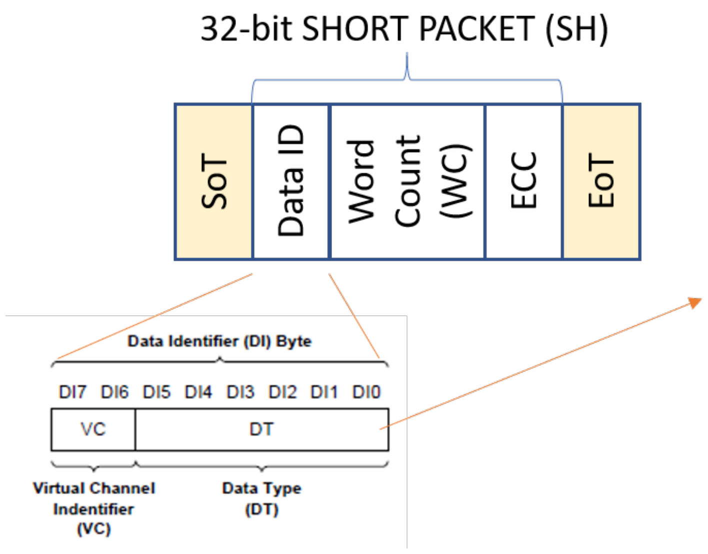


Figure 31. CSI-2 frame structure

The V_{sync} signal is communicated used a frame start (FS) packet. The H_{sync} signal is communicated using a Line Start (LS) packet.

5.3.1 Packet details

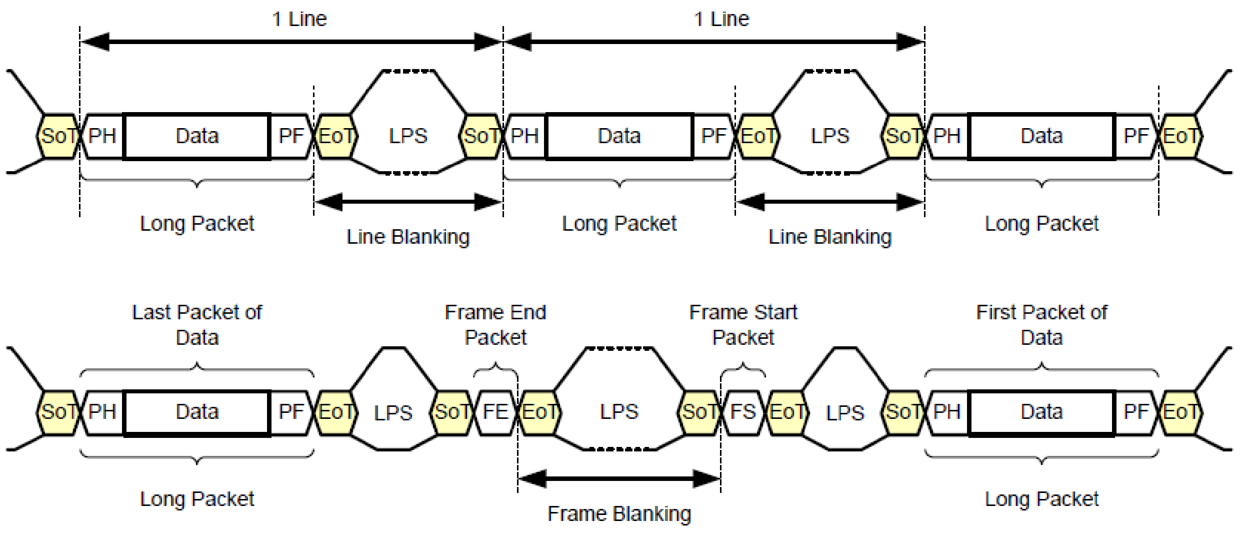
After a Start of Transmission (SOT) process, a short packet is sent. It tells the receiver what to expect next: sync signals, data, virtual channel info, or user-defined data. Figure 32 shows the various packet decode options.


32-bit SHORT PACKET (SH)
SoT
Data ID
Word Count (WC)
ECC
EoT
Data Identifier (DI) Byte
DI7 DI6 DI5 DI4 DI3 DI2 DI1 DI0
VC
DT
Virtual Channel Indentifier (VC)
Data Type (DT)

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
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

Figure 32. Short packet decoder

Figure 33 shows how the packet stream, consisting of HS packets and LP packets, defines the image frame starts, line starts, and data.



KEY:

SoT – Start of Transmission	EoT – End of Transmission	LPS – Low Power State
PH – Packet Header	PF – Packet Footer	
FS – Frame Start	FE – Frame End	
LS – Line Start	LE – Line End	

Figure 33. Line and frame blanking definitions

5.4 Pixel/Byte packing

Old image sensors (pre-dating the CSI-2 standard) use a parallel bus to output the ADC data to an SOC or dedicated image processor. A 10-bit wide data bus plus line, frame, and clock signals are at a minimum 13 signals wide. A flash or other functions requires additional pins. The CSI-2 hardware block in the transmitter takes the image data from the parallel bus and packs it into a serial byte stream which can be transmitted over a DPHY Lane.

The CSI-2 standard supports various pixel formats and bit-lengths. The RAW output from an 8-bit ADC neatly packs into a single pixel per byte. Figure 34 shows an example of a packet stream for one 640-bit wide image line.

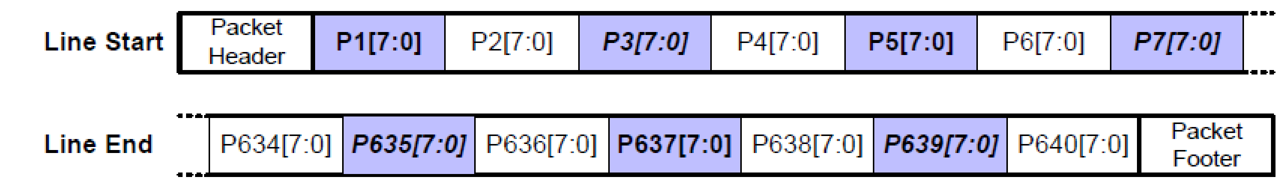


Figure 34. RAW8 packet transmission

The RAW8 pixel/byte packing example is the simplest and easiest format to understand of all the various image data standards covered by CSI-2. Each data byte contains the value of a single pixel.

The RAW10 format from the OV5640 has a 10-bit pixel value but the CSI-2 standard specifies that the minimum data width is 1 byte. Since we cannot transfer partial pixel data, the CSI-2 has various pixel packing standards based on the pixel format used. The transmission of 10-bit RAW pixel data is done by packing four 10-bit pixel values across 5 bytes. The minimum data packet length in this case is always 5 bytes. If the line to be translated does not have a pixel width evenly divisible by five, the extra pixels are zero padded. Figure 35 shows the packet stream for one 640-bit wide image line using RAW10 format.

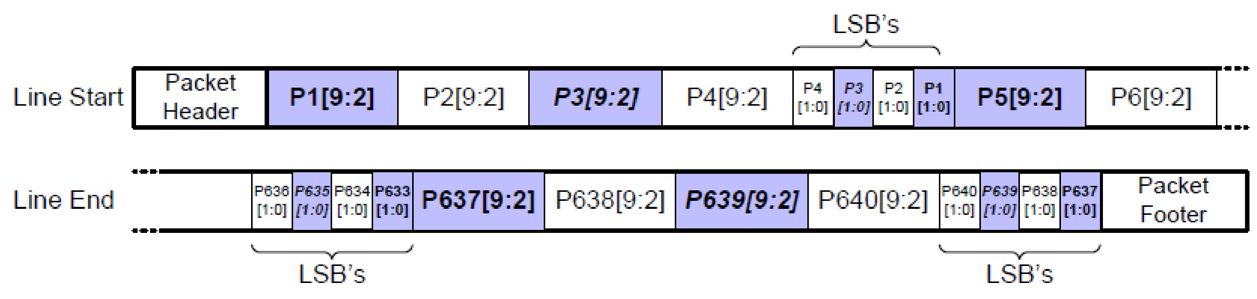


Figure 35. RAW10 packet transmission

Table 18 gives the Bits Per Pixel (bpp) and minimum packet size in bytes for the supported formats. They are the minimum data length sizes for a given format. For example, even though the RAW6 format uses only 6 bits of data to describe each pixel, a minimum of 4 pixels must be sent in a minimum length packet of 3 bytes.

Table 18. CSI-2 supported formats packet data size constraints

Data format	Bits per pixel	Pixels per packet (min)	Packet length (Bytes)	Notes
YUV420 8-bit (legacy)	12	2	3	
YUV420 8-bit	12	2	2/4	Packet Length – Even/Odd lines
YUV420 10-bit	15	4	5/10	Packet Length – Even/Odd lines
YUV422 8-bit	16	2	4	

Table continues on the next page...

Table 18. CSI-2 supported formats packet data size constraints (continued)

Data format	Bits per pixel	Pixels per packet (min)	Packet length (Bytes)	Notes
YUV422 10-bit	20	2	5	YCbCr 4:2:2
RGB888	24	1	3	
RGB666	18	4	9	
RGB565	16	1	2	
RGB555	15	1	2	Padding bit added to the LSB of the green color component.
RGB444	12	1	2	Padding bits added to the LSB of all three color components.
RAW6	6	4	3	
RAW7	7	8	7	
RAW8	8	1	1	
RAW10	10	4	5	
RAW12	12	2	3	
RAW14	12	4	7	

5.5 Bandwidth calculations

To set or evaluate an image capture system, the system bandwidth must be calculated to ensure that the various clocks support the desired system. The basis of all image system design is the Pixel Clock which can be calculated using Equation 13:

$$\text{Pixel Clock (Hz)} = \text{HTOT} * \text{VTOT} * \text{FPS}$$

Equation 13.

Where:

- H_{TOT} = total line width = active pixels + horizontal blanking (pixels)
- V_{TOT} = total frame height = active lines + vertical blanking (lines)
- FPS = frames per second

The total system bandwidth is the pixel clock times the bits per pixel. It can be calculated using Equation 14:

$$\text{Bandwidth (bps)} = \text{Pixel Clock} * \text{Bits Per Pixel}$$

Equation 14.

The output format determines the bits per pixel. Refer to Table 18 for the bit depth of the various pixel formats supported by CSI-2.

The data rate per lane is the bandwidth divided by the number of data lanes in the system design. The CSI-2 specification allows a minimum of 1 lane and a maximum of 4 lanes.

$$\text{Data Rate Per Lane (bps)} = \text{Bandwidth} / \text{number of data lanes}$$

Equation 15.

The MIPI interface is Double Data Rate, so the D-PHY HS Output Clock frequency is half of the data rate per lane:

$$\text{MIPI D} - \text{PHY Clock Rate (Hz)} = \text{Data Rate Per Lane} / 2$$

Equation 16.

5.5.1 OV5640 bandwidth example

See OV5640 for an example of the timing information. Figure 36 shows the relationship between the V_{sync}, H_{sync}, and valid data (HREF) signals. The data sheet uses pixel clocks as the unit of measurement for everything. Do the translation into lines where necessary.

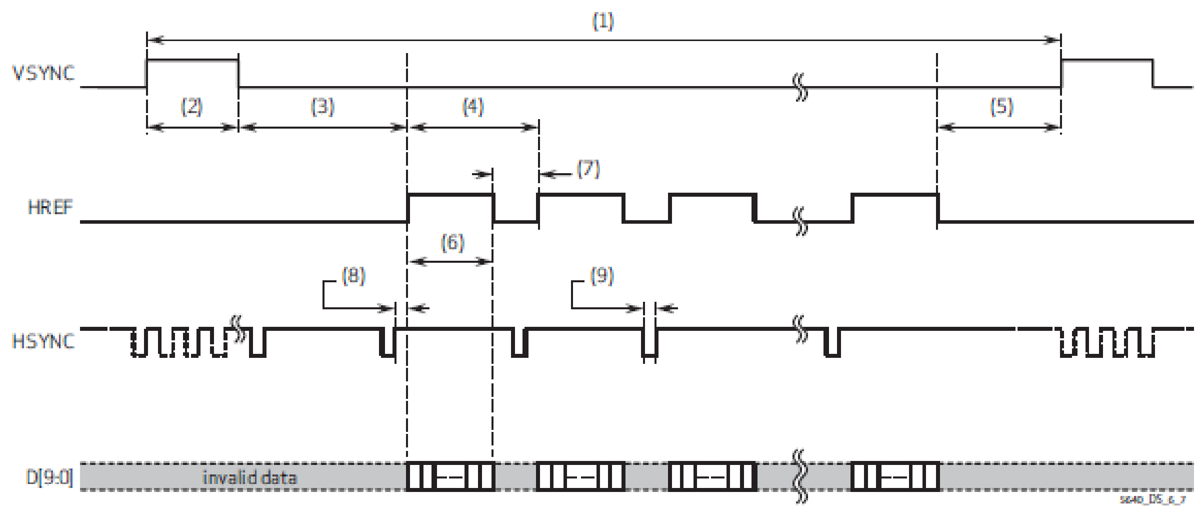


Figure 36. OV5640 timing

Table 19. OV5640 timing values

Mode	1	2	3	4	5	6	7	8	9
		V_{sync}	Vertical Back Porch		Vertical Front Porch	Active Pixels	Horizontal Blanking		
5 Mpix (2592 × 1944)	5596992	5688	48276	2844	14544	2592	252	0	252
720 p (1280 × 720)	5596992	5688	3530644	2844	14544	1280	1564	0	1564

Using the full resolution of 2592 pixels wide and 1944 lines tall, the data sheet gives an H_{sync} pulse width of 252 pixel clocks.

- H_{TOT} = 2592 + 252 = 2844 pixels

The V_{sync} pulse is defined as 5688 pixel clocks, but 5688 / 2844 = 2 lines. The Vertical front porch and back porch timing are 22 lines. If subtracting a single H_{sync} pulse, you get the total vertical porch in clocks.

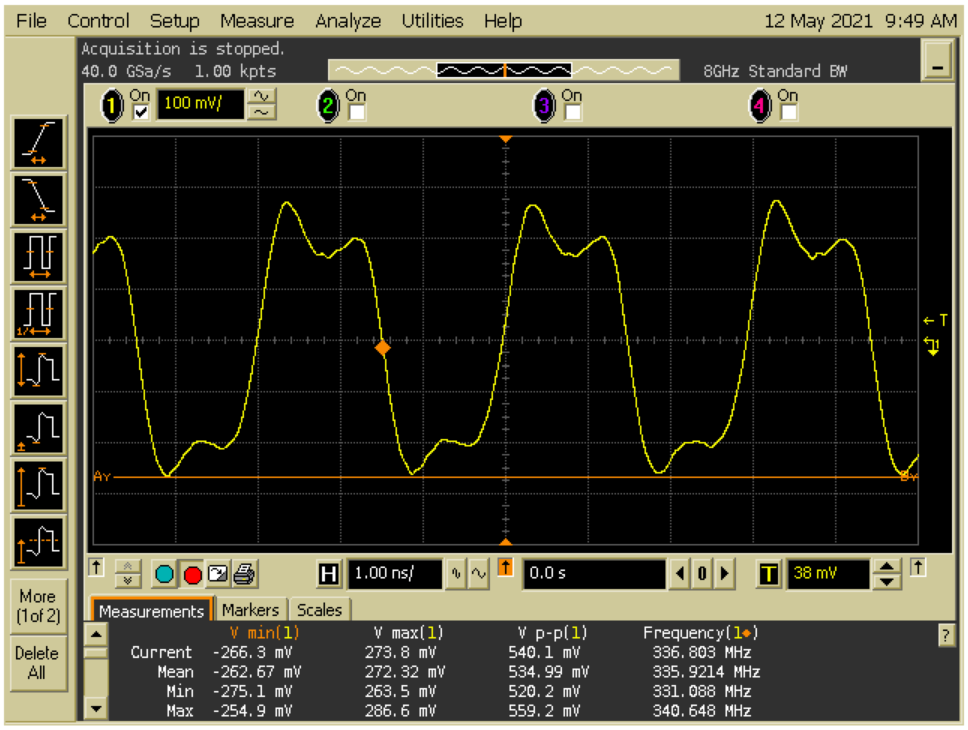
- Vbp + Vfp – Hsync = 48276 + 14544 – 252 = 62,568 pixel clocks
- 62,568 / 2844 = 22 lines

- V_{TOT} = 1944 + 2 + 22 = 1968

The frame rate for the max resolution is 15FPS. The default output format is YUV422 (8-bit) which has a pixel bit depth of 16 bpp. With these values, we can calculate the bandwidth:

- Pixel Clock = 2844 * 1968 * 15 = 83,954,880 Hz
- Bandwidth = 83,954,880 * 16 = 1,343,278,080 bps
- Data Rate Per Lane = 1,343,278,080 bps / 2 = 671,639,040 bps
- MIPI Bit Clock = 671,639,040 bps / 2 = 335.9 MHz

Figure 37 is a scope plot of the OV5640 Camera module MIPI bit clock running at 335.9 MHz.



12 May 2021 9:49 AM

Acquisition is stopped.
40.0 GSa/s 1.00 kpts
8GHz Standard BW

1 On ✓ 100 mV/
2 On
3 On
4 On

H 1.00 ns/
0.0 s
T 38 mV

	V min(1)	V max(1)	V p-p(1)	Frequency(1)
Current	-266.3 mV	273.8 mV	540.1 mV	336.803 MHz
Mean	-262.67 mV	272.32 mV	534.99 mV	335.9214 MHz
Min	-275.1 mV	263.5 mV	520.2 mV	331.088 MHz
Max	-254.9 mV	286.6 mV	559.2 mV	340.648 MHz

Figure 37. OV5640 MIPI bit clock for 5 Mp resolution at 15FPS

5.6 CSI-2 system clock configuration

The high-level MIPI CSI-2/D-PHY clocking architecture is common across all i.MX RT and i.MX 8 family of parts. The signal and register names differ but the functionality is common. Figure 38 gives a high-level overview of the clocking system and data interface.

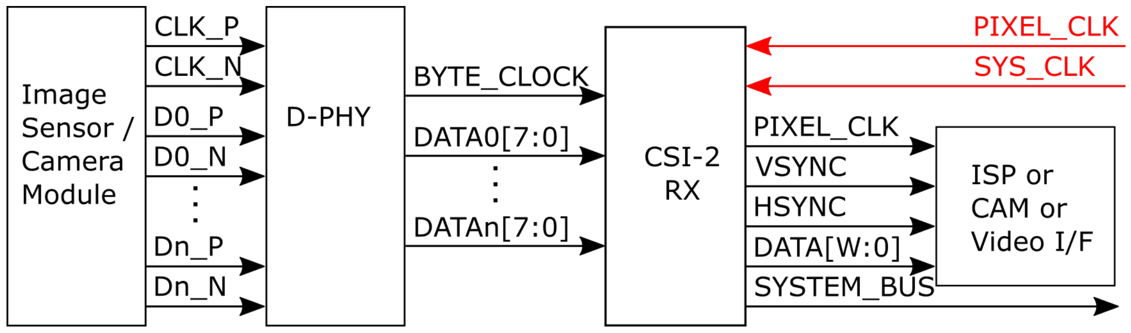


Figure 38. MIPI CSI-/D-PHY clock structure

The clocking relationships and requirements can be confusing, but it is simple if you keep in mind one rule:

- The data bandwidth coming into the part (from the image sensor) must be balanced with the data bandwidth exiting from the CSI-2 block.

5.6.1 Byte clock

D-PHY automatically generates the `BYTE_CLOCK` in Figure 38. `BYTE_CLOCK` is always 1/8 the speed of the HS Bit Clock (which is also the Data Rate Per Lane).

5.6.2 Pixel clock

SOC generates the `PIXEL_CLK`. `PIXEL_CLK` must be set so that image bandwidth exiting the CSI-2 block is greater than or equal to the image bandwidth entering the block. The 28FDSOI parts refer to this clock as the user-interface clock or `CLK_UI`. The 16FF parts refer to this clock as the `I_ACL`K or `ACLK`.

5.6.3 System clock

SOC generates the `SYS_CLK` in Figure 38. `SYS_CLK` is used for command clocking the CSI-2 control interface. The 28FDSOI parts refer to this clock as the CLK. The 16FF parts refer to it as the `I_PCLK` or APB clock.

6 CSI-2 examples

The i.MX 8 and i.MX RT family use two different CSI-2/D-PHY modules based on the underlying process node. For information on the process node and part differences, see Features/Differences between i.MX 8M and i.MX RT parts.

6.1 RT1170 camera example

The i.MX RT1170 EVK ships with an included camera module which uses the OV5640 image sensor from OmniVision. Figure 39 shows the camera module correctly installed with an image being displayed on the 720 × 1280 LED panel described in i.MX RT1170 DSI example.

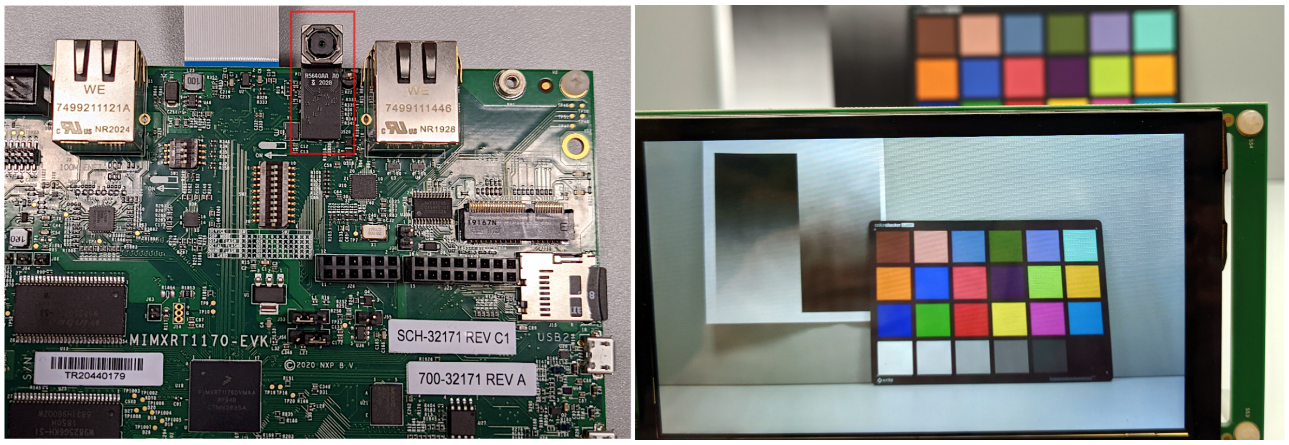


Figure 39. i.MX RT1170 camera installed

In this example, we will examine the `CSI_MIPI_RGB_CM7` demo included with the SDK. In this example, the camera output pixel format is RGB565. The `MIPI_CSI-2` converts it to RGB888 internally and sends to the CSI. In other words, the CSI input data bus width is 24-bit. The CSI saves the frame as 32-bit format XRGB8888. PXP is used to convert the XRGB8888 to RGB565 and output to the LCD panel.

6.1.1 Initial system design

The first thing to do is calculate the image bandwidth required by the OV5640. For this example, the active image size is 1280 × 720 and the frame rate is 30 FPS.

The H_{sync} is 612 pixels which gives an H_{TOT} = 1892.

The V_{sync} is 2 lines and the Vfp + Vbp = 18. Therefore, the V_{TOT} = 720 + 18 + 2 = 740 lines.

With these values and a frame rate of 30 FPS, we can calculate the pixel clock:

- Pixel Clock = 1892 * 740 * 30 = 42,002,400 Hz

The RGB565 format requires 16 bpp, so the bandwidth is:

- Bandwidth = 42,002,400 Hz * 16 bpp = 672,038,400 bps

This system uses a 2-lane MIPI interface, so the data rate per lane is:

- DRPL = 672,038,400 bps / 2 = 336,019,200 bps

The MIPI D-PHY HS clock must be:

- MIPI HS Clock = 336,019,200 / 2 = 168,009,600 Hz

6.1.2 CSI-2 RX clock setup

The RT1170 CSI-2 RX block requires three clocks to be set for camera operation:

- System Clock (CLK)
- Pixel Clock (CLK_UI)
- Escape Clock (CLK_ESC)

6.1.2.1 System clock

The system clock is referred to as simply **CLK** in the 28FDSOI CSI-2 block. This clock must be equal to or faster than the byte clock. The byte clock in this example can be calculated as:

Byte_clock = DRPL / 8 = 336,019,200 / 8 = 42,002,400 Hz

`CLOCK_ROOT73` controls the `CSI2_CLK_ROOT`. Inspection of the sample code shows the following register settings:

- CLOCK_ROOT73_CONTROL[MUX] = 5
- CLOCK_ROOT73_CONTROL[DIV] = 8

A MUX setting of 5 sets the root to the `SYS_PLL3_CLK` which runs at 480 MHz. The DIV setting of 8 gives a clock frequency of 480 MHz / 8 = 60 MHz.

6.1.2.2 Pixel clock

`CLOCK_ROOT75` controls the `CSI2_UI_CLK_ROOT`.

- CLOCK_ROOT75_CONTROL[MUX] = 5
- CLOCK_ROOT75_CONTROL[DIV] = 8

A MUX setting of 5 sets the root to the `SYS_PLL3_CLK` which runs at 480 MHz. The DIV setting of 8 gives a clock frequency of 480 MHz / 8 = 60 MHz.

6.1.2.3 Escape clock

The escape clock must be in the range of 60 – 80 MHz. `CLOCK_ROOT74` controls the `CSI2_ESC_CLK_ROOT`.

- CLOCK_ROOT74_CONTROL[MUX] = 5
- CLOCK_ROOT74_CONTROL[DIV] = 8

A MUX setting of 5 sets the root to the `SYS_PLL3_CLK` which runs at 480 MHz. The DIV setting of 8 gives a clock frequency of 480 / 8 = 60 MHz.

6.1.3 Bandwidth balancing

Using these clock settings, we can check the input and output bandwidth relationships. The Output bandwidth must be greater than or equal to the input bandwidth.

- $$\text{Input bandwidth} \ = \ \text{DRPL} * \text{Number of Lanes}$$
  Equation 17.
- $$\text{Output bandwidth} = \ (\text{Pixel Clock}) * (\text{Image Format bpp}) * (\text{number of pixels per clock cycle})$$
  Equation 18.
- Input bandwidth = 336,019,200 bps * 2 = 672,038,400 bps
- Output bandwidth = 60 MHz * 16 bpp * 1 = 960 Mbps

Since the output bandwidth (960 Mbps) is >= to the input bandwidth (672 Mbps), no image data is lost.

6.1.4 Camera clock setup

A 24 MHz crystal on the RT1170 EVK provides the OV5640 reference clock (XVCLK). Based on the initial system design calculations, we need a pixel clock of 42 MHz and a MIPI Bit Clock of 168 MHz.

Table 20. OV5640 clock setup registers

Register address	Register name	Description	Programmed value
0x3034	SC PLL CONTRL0	Bit[3:0]: MIPI bit mode	0x1A

Table continues on the next page...

Table 20. OV5640 clock setup registers (continued)

Register address	Register name	Description	Programmed value
		0x8: 8-bit mode 0xA: 10-bit mode	
0x3035	SC PLL CONTRL1	Bit[7:4]: System clock divider - Slows down all clocks (SDIV0) Bit[3:0]: Scale divider for MIPI PCLK (MIPI_DIV)	0x21
0x3036	SC PLL CONTRL2	PLL multiplier (4~252): Can be any integer for 4~127 and only even integer for 128~252	0x54
0x3037	SC PLL CONTRL3	Bit[4]: PLL root divider (PLL_RDIV) 0: Bypass 1: Divided by 2 Bit[3:0]: PLL pre-divider: 1, 2, 3, 4, 6, 8	0x13
0x4837	PCLK Period	Period of pixel clock, pclk_div=1, and 1-bit decimal	0x0A
0x3108	SYSTEM ROOT DIVIDER	Bit[5:4]: PCLK root divider (PCLK_DIV) 00: PCLK = pll_clki 01: PCLK = pll_clki/2 10: PCLK = pll_clki/4 11: PCLK = pll_clki/8	0x16

The system clock is based on the required system bandwidth and calculated as follows:

- SYSCLK = XVCLK / PLL Pre-Divider * PLL Multiplier = 24 / 3 * 84 = 672 MHz

The PLL Pre-Divider has undocumented mapping but 3 seems to be commonly used.

The MIPI Bit Clock is derived from the system clock using the following formula:

- MIPI_BIT_CLK = SYSCLK / MIPI_DIV / 2 = 672 / 2 / 2 = 168 MHz

The Pixel Clock is also derived from the system clock with the following formula:

- PIXEL_CLK = SYSCLK / (SDIV0 * PLL_RDIV * BIT_DIV * PCLK_DIV * P_DIV) = 672 / (2 * 2 * 2 * 2 * 1) = 42 MHz

Figure 40 shows the MIPI Bit clock in high-speed mode after being properly set for 1280 × 720 × 30 FPS and 16 bpp. The frequency is 168 MHz as previously calculated.

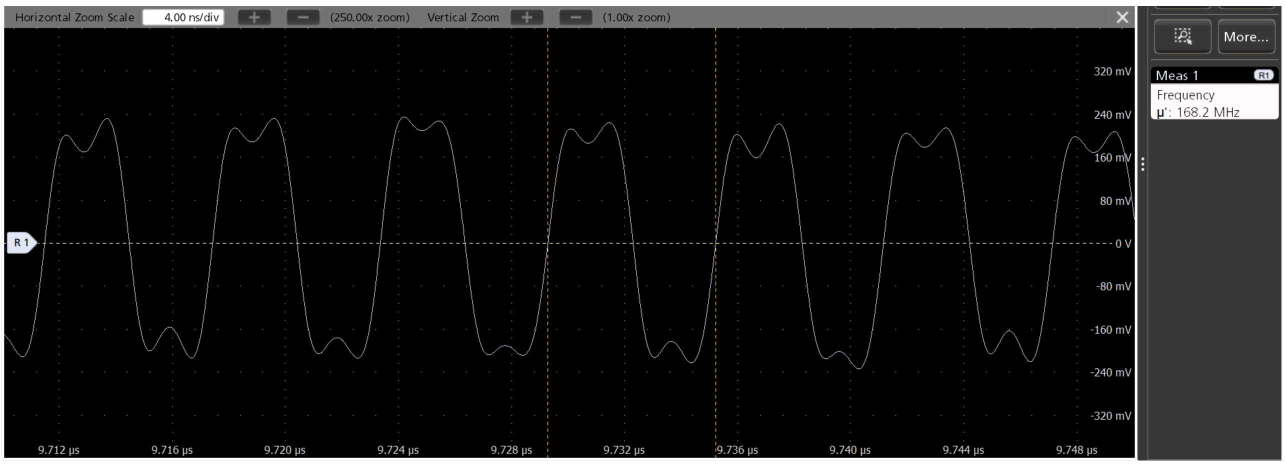


Figure 40. OV5640 MIPI clock for 1280 x 720 @ 30 FPS, 16 bpp - 168 MHz

6.1.5 D-PHY setup

The only D-PHY parameter that needs configuration is the $T_{HS\text{-}SETTLE}$.

In this example, `RxClkInEsc` and `CLK_UI` (pixel clock) are both 60 MHz. The time period for both is 1/60 MHz = 16.67 ns.

The D-PHY specification requires the $T_{HS\text{-}SETTLE}$ to be in the range of **85 ns + 6 * UI** to **145 ns + 10 * UI**. This calculation works out to a range of 185 ns – 311.67 ns.

For the RT1170, the equation for programming $T_{HS\text{-}SETTLE}$ is:

$$\mathrm{T\_HS\_SETTLE} = (\mathrm{PRG\_RXHS\_SETTLE} + 1) * (\mathrm{Tperiod\ of\ RxClkInEsc})$$

Equation 19.

The driver code for the CSI-2 RX block sets the PRG_RXHS_SETTLE = 0x11 = 17.

$T_{HS\text{-}SETTLE}$ = (17 + 1) * 16.67 ns = 300 ns

There is a calculation inaccuracy up to 1 cycle of `RxClkInEsc`. Choose a programming value away from the boundaries.

6.1.6 OV5640 MIPI waveforms

Using the settings from the previous section we can verify some of the timings by looking at a scope plot of the clock and data lanes.

The scope plot in Figure 41 shows the relationship between the clock (shown differentially) and the Data0 lane (P & N shown single-ended) as they transition from LP to HS data transmission.

The THS-SETTLE period of 300ns is high-lighted and is followed by the Start of Tranmission (SoT) header and then the Frame Start (FS) packet.

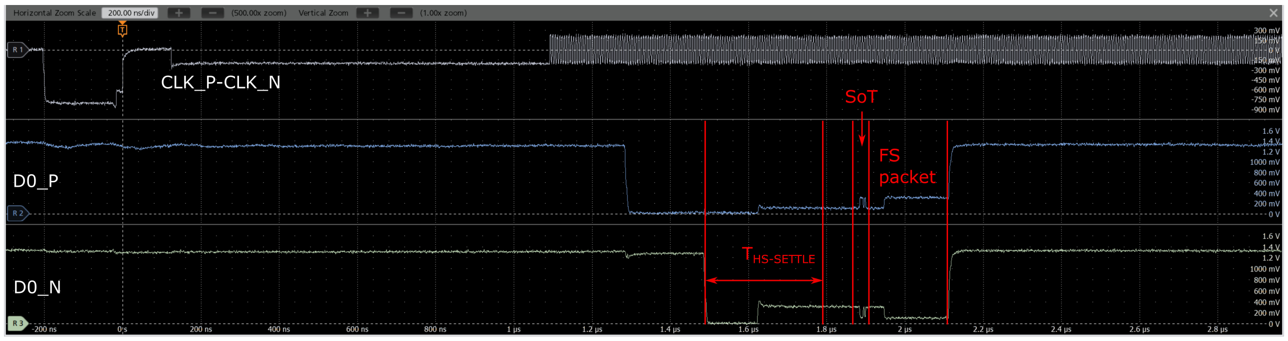


Figure 41. OV5640 startup

Since the Data lanes transition from HS to LP once at the end of the scan line we can easily check the timing using a scope. The scope plot in Figure 42 shows the start of a new frame and the first 13 active image lines. The time required for a single line (tL) can be calculated by dividing the frame rate by the VTOT.

tL = 1 / framerate / VTOT = (1 / 30fps) / 740 lines = 45.045 us

The V_{sync} + VBP = 2 + 7 = 9 lines. Therefore, the total time elapsed between the frame start and active pixel data is:

*9 lines * 45.045us = 405 us*

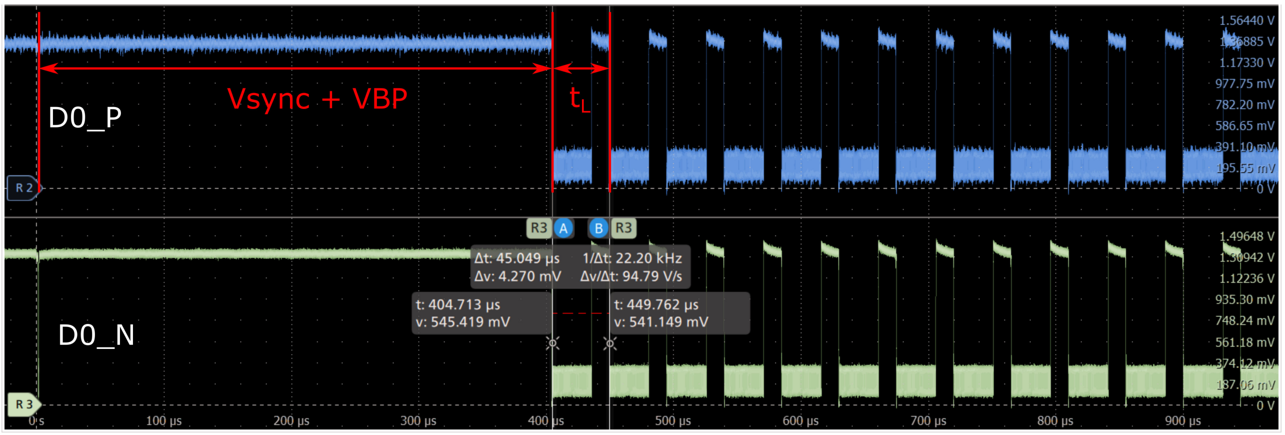


Figure 42. OV5640 frame start timing

7 Features/Differences between i.MX 8M and i.MX RT parts

Table 21. Feature differences

Part	DSI			CSI-2			Process
	Lanes	Max Lane Speed (Gbps)	Max Total Bandwidth (Gbps)	Lanes	Max Lane Speed (Gbps)	Max Total Bandwidth (Gbps)	
i.MX RT1170	2	0.8	1.6	2	1.5	3	28FDSOI
i.MX RT500[1]	2	0.8951	1.79	n/a	n/a	n/a	28FDSOI

Table continues on the next page...

Table 21. Feature differences (continued)

Part	DSI			CSI-2			Process
	Lanes	Max Lane Speed (Gbps)	Max Total Bandwidth (Gbps)	Lanes	Max Lane Speed (Gbps)	Max Total Bandwidth (Gbps)	
i.MX 8	4	1.5	6	4	1.5	6	28FDSOI
i.MX 8M	4	1.5	6	4 (x2)	1.5	6	28FDSOI
i.MX 8M Mini	4	1.5	6	4	1.5	6	16FF
i.MX 8M Nano	4	1.5	6	4	1.5	6	16FF
i.MX 8M Plus	4	1.5	6	4 (x2)	1.5	6	16FF
i.MX 8ULP	4	1.5	6	2	1.5	3	28FDSOI
i.MX 8ULP-CS	4	1.5	6	2	1.5	3	28FDSOI
i.MX 8Q/D/X	4 (x2)	1.05	4.2	4	1.5	6	28FDSOI

1. D-PHY has no PLL or ULPS support.

8 Debug

This section covers commonly encountered bring-up problems and offers tips for debug.

8.1 DSI bring-up checklist

1. Check all supply voltages.
2. Make sure that I^{2}C communication is functional:
   a. Verify ability to send and receive data. One can use Linux I2C commands, such as, `i2cdetect`, `i2cdump`, `i2cget`, `i2cset`.
3. Make sure that display initialization registers are correct.
4. Check MIPI clocks:
   a. MIPI HS bit clock
   b. Pixel clock
   c. Escape clocks
5. Make sure that display and SOC video modes are in sync. For example, **burst** video mode is selected for both the display and the SOC.
6. Check whether blanking parameters are correctly set in the display driver: H_{sync}, HBP, HFP, V_{sync}, VBP, VFP.
7. Check MIPI Low-Power communication: Frequency, IO supply voltages.
8. Check MIPI High-Speed communication: Frequency and differential mode voltages.
9. Check MIPI error registers: `ErrSotHS` and `ErrSotSync_HS`.

8.2 DSI common problems

8.2.1 Discontinuous clock

Some displays do not require a dedicated external crystal/clock. They use the MIPI clock for their system clock. These displays require continuous clock mode from the SOC to provide a proper clock to the display for initialization and/or runtime functionality.

8.2.2 Missing periodic low-power transitions

A common display failure is caused by DSI source configuration of the BLLP behavior. To enable PHY synchronization, the host processor periodically ends HS transmission and drives the Data Lanes to the LP state. This transition takes place at least once per frame. Some common DSI driver examples do not enable LP mode by default during any of BLLP periods in the video stream.

8.2.3 Incorrect DSI packet timing

Most standard DSI interfaces ensure MIPI compliance from a D-PHY physical layer perspective. Problems related to DSI packet order/timing occur during the driver development phase, especially when implementing non-standard video timings or clock rates for custom video solutions. This means that the received timing of video packets, especially synchronization packets, is critical to keeping the system functioning correctly.

8.2.4 $T_{HS-SKIP}$ configuration

The MIPI D-PHY receiver specification requires the sink device to ignore activity on the DSI data lanes at the end of high-speed packet transmission prior to reentering the low-power state (LP-11). The aim is to mask transition effects during the End of Transmission (EoT) sequence. If the $T_{HS-SKIP}$ timing parameter is misconfigured, it may result in data errors in the DSI video stream.

8.2.5 End of Transmission Packets (EoTp)

DSI devices compliant to the MIPI DSI v1.0 specification and later are required to generate End of Transmission Packets (EoTp) following any HS data transmission. The main objective of the EoTp is to enhance the robustness of the DSI interface during transition from HS to LP mode so that the receiver can clearly detect the end of HS transmission even in the presence of non-optimal signaling conditions. To support backward compatibility and interoperability between DSI peripherals, the standard mandates that the transmitter and receiver devices have the optional capability to utilize or not utilize EoTp functionality. This EoTp insertion is optional. If the display being used requires it, enable it on both the 16FF and 28FDSOI DSI blocks.

8.3 CSI-2 Bring Up checklist

1. Check all supply voltages.
2. Make sure that I^{2}C communication is functional:
   - Verify the ability to send and receive data. One can use Linux I2C commands, such as, `i2cdetect`, `i2cdump`, `i2cget`, `i2cset`.
3. Make sure that sensor initialization registers are correct.
4. Check Camera and SOC MIPI clocks:
   a. Camera MIPI HS Bit clock, Pixel Clock, and Escape clock.
   b. SOC Pixel Clock and Escape Clock.
5. Check MIPI low-power communication, low frequency, and non-differential (Scope or MIPI analyzer).
6. Check MIPI high-speed communication, high frequency, and differential (MIPI Analyzer).
7. Check MIPI status register, such as:
   a. ECC and CRC Error Status Register
   b. IRQ Status Register
   c. `ErrSot` HS Status Register

d. `ErrSotSync` HS Status Register

e. `ErrEsc` Status Register

f. `ErrSyncEsc` Status Register

g. `ErrControl` Status Register

8.4 CSI-2 common problems

8.4.1 Pixel format errors

The MIPI CSI-2 specification supports multiple image pixel formats. See Table 18 for the full list. But the application layer and image sensor might be slightly misaligned in the selected formats details. For example, an image sensor might be sending data using the RGB888 format. But the image sensor can send the pixel data 6 different ways, such as, RGB, BGR, GRB. If the application is expecting BGR and the image sensor is sending data using GRB, there are no data transfer errors but the color of the received image goes wrong.

9 Revision history

Rev.	Date	Description
0	21 March 2022	Initial release

Legal information

Trademarks