MIPI DSI/D-PHY/CSI-2 INTERFACES

SHUYANG GUAN
NXP MPU SYSTEM ENGINEER
MAY, 2022

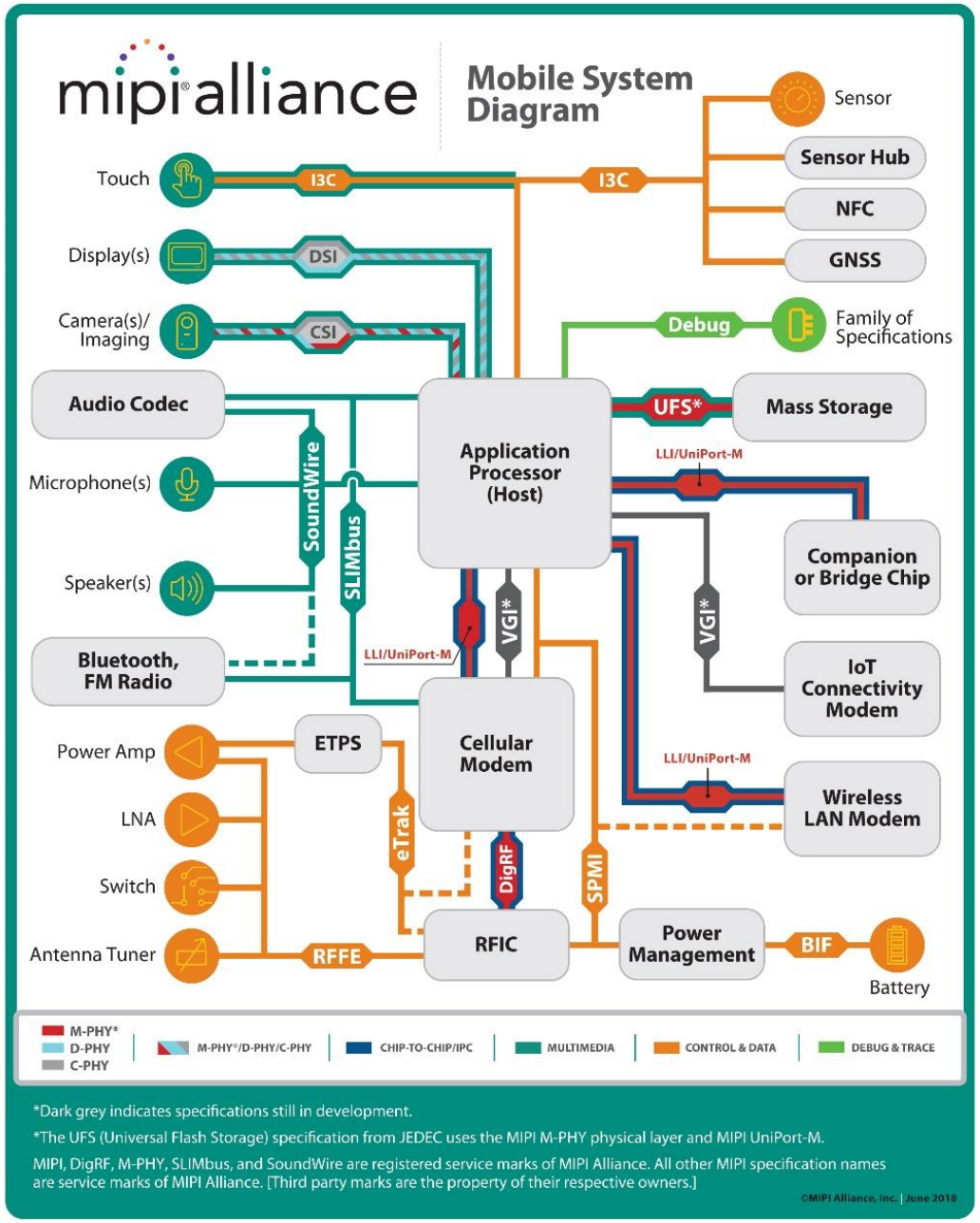
Abstract

- The i.MX 6, 7, 8, RT and future families of parts all use the MIPI DSI and CSI-2 interfaces for Display and Camera integration.

- This presentation will explain:
- MIPI overview
- MIPI DSI protocol
- MIPI D-PHY electrical interface
- guides of how to bring up MIPI DSI panel on i.MX platform
- MIPI CSI-2 protocol
- guides of how to bring up MIPI CSI sensor on i.MX platform

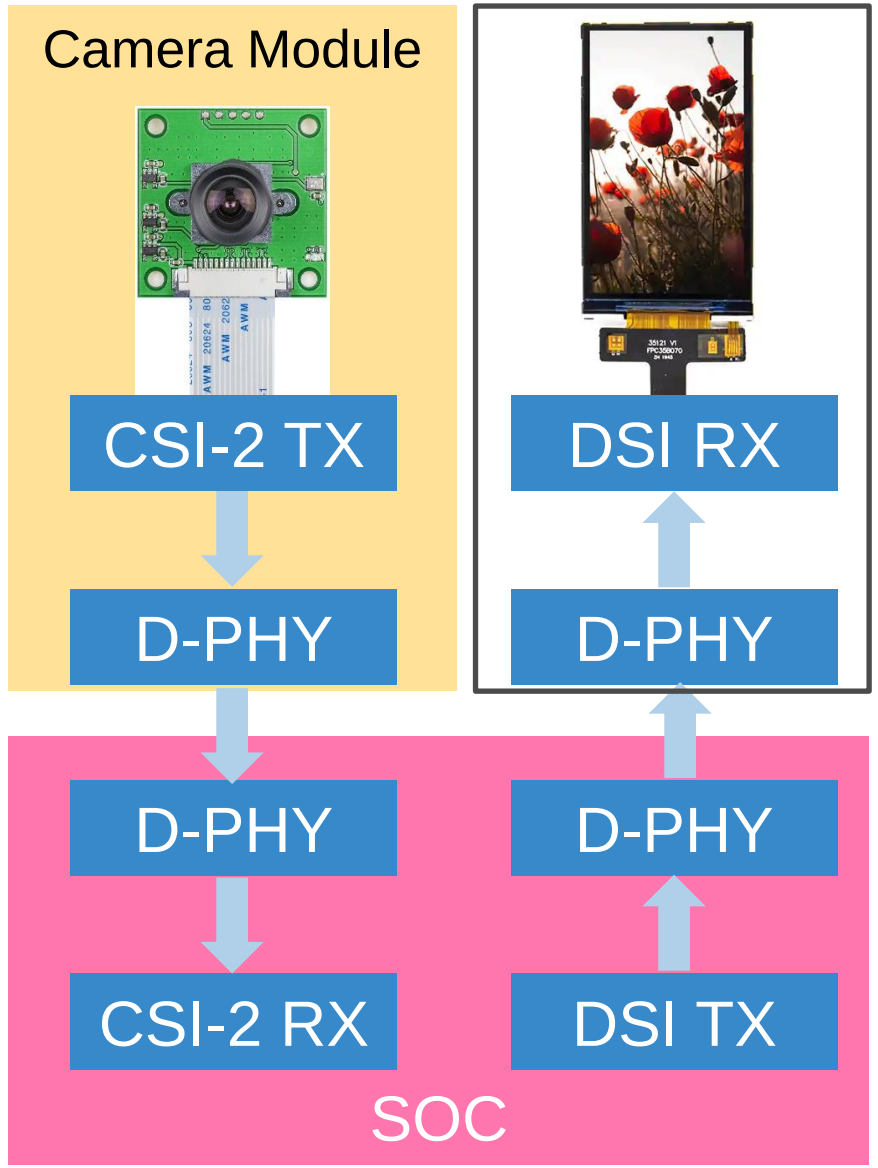
What Is MIPI?!?!

- Mobile Industry Processor Interface - MIPI
- Founded in 2003 by ARM, Nokia, ST, and Texas Instruments. Within a year Intel, Motorola, Samsung and Philips joined.
- Currently over 300 members.
- Creates hardware and software interface standards to simplify the integration of various components.
- MIPI alliance: https://www.mipi.org/specifications/



MIPI Interfaces for Display and Camera

- We use MIPI DSI, CSI-2, and D-PHY
- Display Serial Interface (DSI)
  - Specifies the data transfer protocol from an SoC to a display
- Camera Serial Interface [rev] 2 (CSI-2)
  - Specifies the data transfer protocol and basic image sensor control between camera and SoC
- Both *can* use the D-PHY specification for electrical transmission – but not mandatory (alternative PHY's – C-PHY, A-PHY, etc.…)



MIPI D-PHY Misconceptions/Clarifications

What it is:

- Used for production embedded electronics i.e. cell-phone, or tablet.
- Lowers pin counts and increases bandwidth and functionality over legacy imaging bus standards.
- Point to point interface.
- Small silicon area needed for IP block implementation:
  - DSI ~0.8 size of HDMI, ~0.55 size of USB and ~0.5 size of PCIe (from i.MX 8M Plus)

What it is NOT:

- Not designed for hot plug interfaces like HDMI or USB.
- Not “plug and play” – requires configuration at both TX and RX side.
- Not designed for long distances (25-30cm max).

Note: The “D” in D-PHY comes from the Roman numeral for 500 – original spec was 500Mbps/lane.

Why Change?

- Lower pin count - previous display standards (parallel RGB) could use upwards of 30 pins.

  - A legacy display interface that supports 24 bits per pixel (bpp) color would require a 24-pin data-bus, Vsync, Hsync, enable and a pixel clock for a total of 28 individual signals.

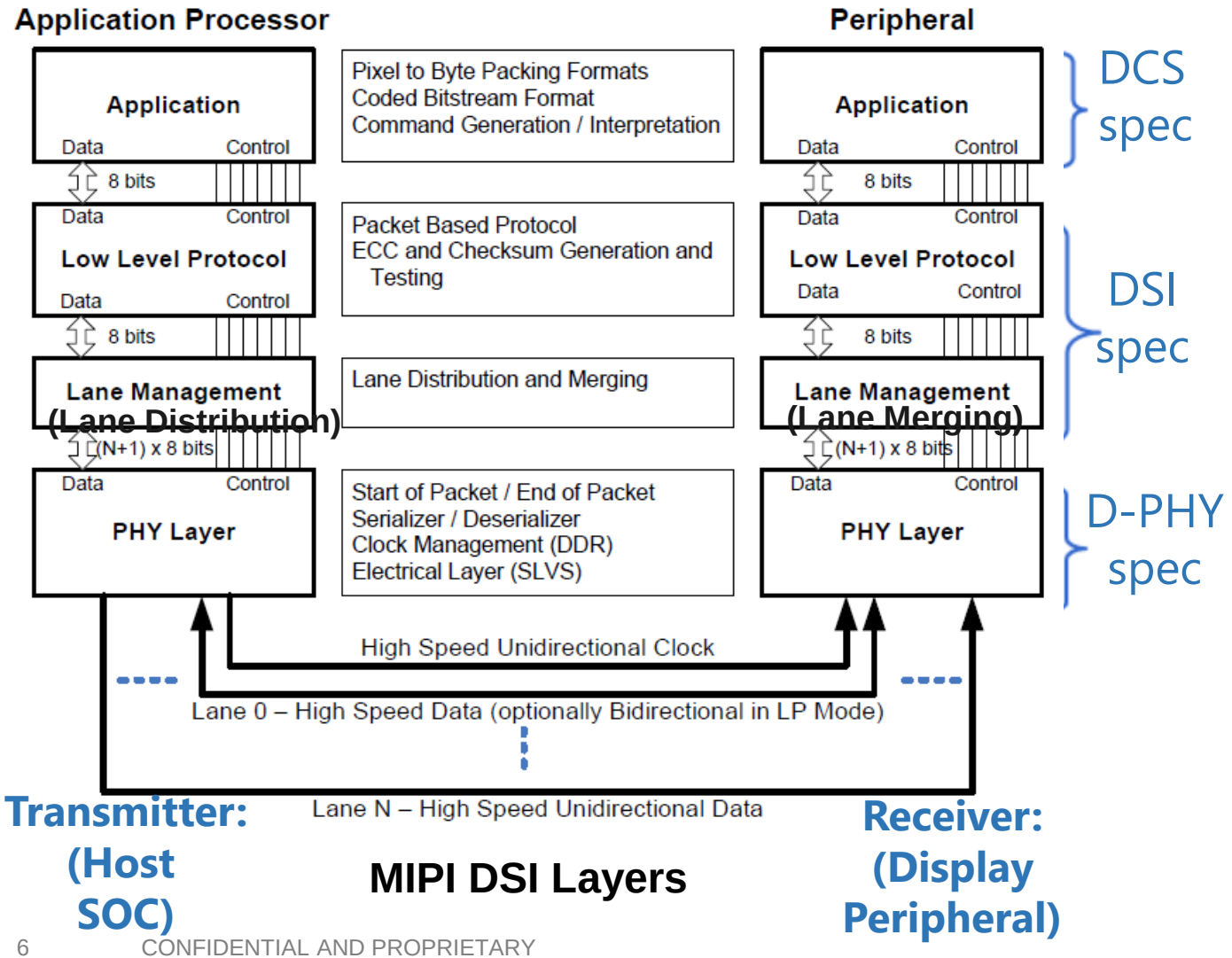
  - A legacy camera interface generally requires signals equal to the ADC bit-depth plus pclk, href, vsync and strobe – which for a 10-bit ADC would = 14 signals.

  - The MIPI D-PHY can transmit the same amount of data (and more) using 4-10 pins depending upon number of data lanes used.

- Higher bandwidth, lower power. Example: 1280x720 @60fps, 24bpp

Interface	Pins	Bus Frequency	Bandwidth	Switching Power
Parallel RGB (24bit)	28	62.35MHz	1.5 Gbps	~ 255mW
DSI 2-lane	6	400MHz	1.6 Gbps	~ 29mW

DSI Layer Definitions



- Application Layer:
  - TX: Pack pixel and commands into bytes;
  - RX: Unpack bytes to pixels.

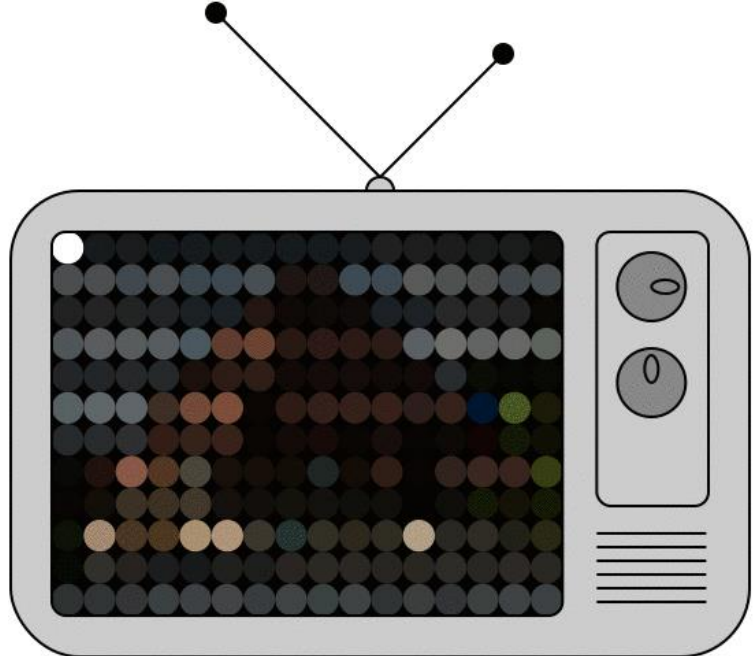
- Low Level Protocol Layer:
  - Specifies how bytes are organized into defined groups called packets.
  - TX: append header, ECC, footers;
  - RX: stripped off header and interpreted by corresponding logic.

- Lane Management Layer:
  - TX: Lane Distribution.
  - RX: Lane merging.

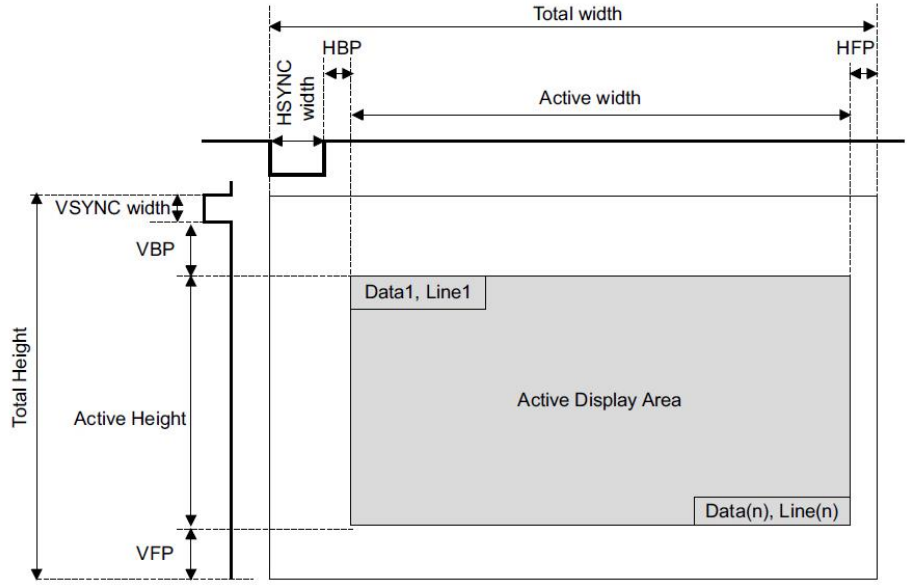
- PHY Layer (D-PHY, C-PHY…):
  - Specifies transmission medium, input/output circuitry and clocking mechanism.
  - Have two modes of data transmission: HS and LP, separated by SoT and EoT sequences.

DSI: Digital Display Basics

- Raster Scan - the displayed image is created by writing one line at a time, pixel by pixel from left to right and from top to bottom.
- Image data is transferred in this same fashion both inside the SoC and over various display links (DSI, HDMI, etc.)
- Roots of this architecture from original CRT displays from 1930's. Blanking area needed for electron beam reset.

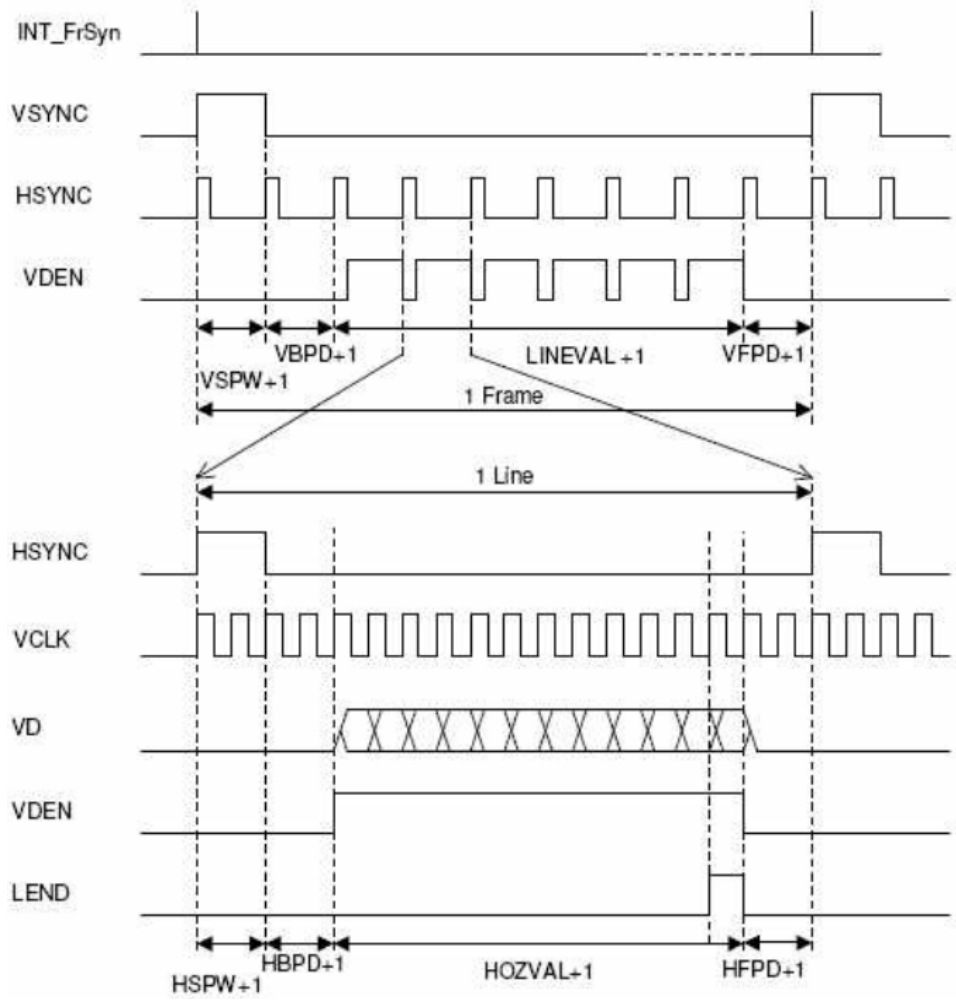


- HSYNC – Horizontal Sync – signals line start
- HBP – Horizontal Back Porch
- HACTIVE
- HFP – Horizontal Front Porch
- VSYNC – Vertical Sync – signals Frame Start
- VBP – Vertical Back Porch
- VACTIVE
- VFP – Vertical Front Porch
- All clocked using Pixel Clock



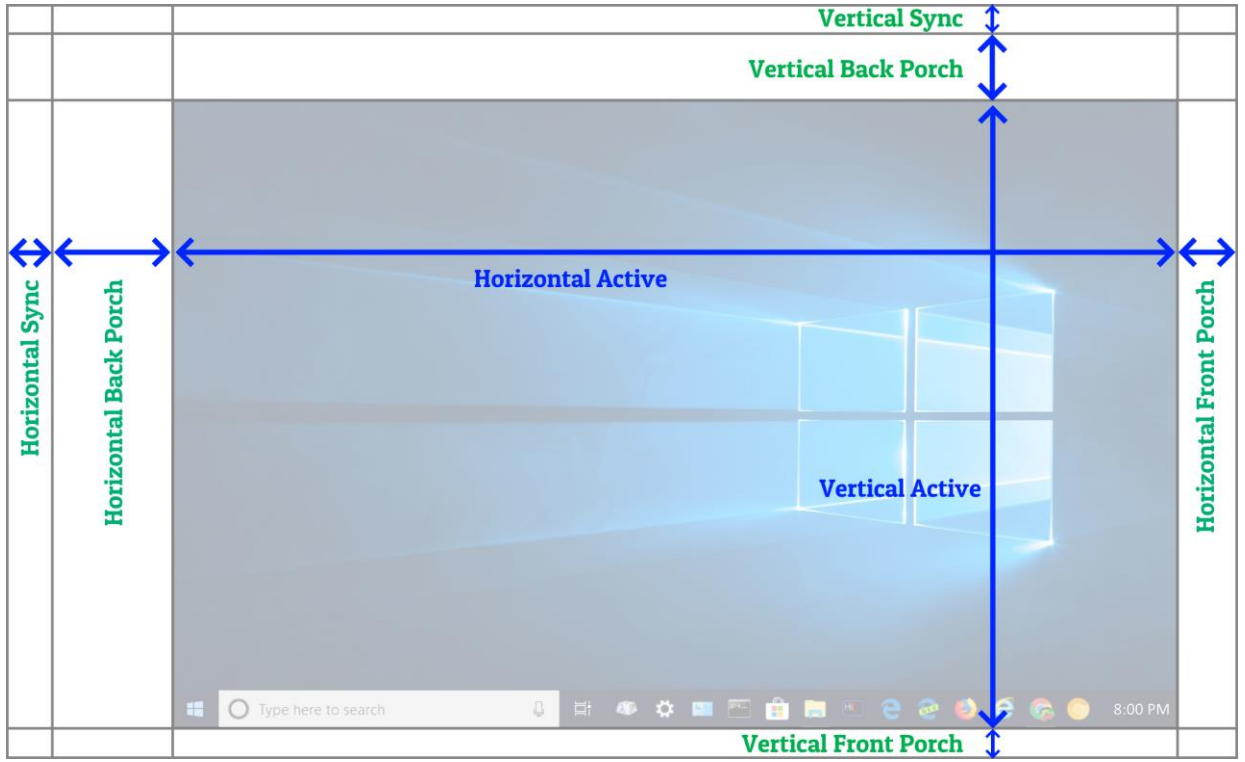
DSI: Digital Display Basics - Timing Sequence

1. Send VSYNC signal, one frame is ready to transmit.
2. Send HSYNC signal, one line is ready to start.
3. After VBP line, start to transmit first valid line.
4. After HBP clk, start to transmit first valid pixel.
5. The RGB data is transmitted each VCLK and confirmed by the VDEN signal, will transmit x-active pixel.
6. After HFP clk, start to transmit next line.
7. Repeat 4-6 to fill y-active line.
8. After VFP line, start to transmit next frame.
9. Repeat 1-8 to provide new frame for LCD.



DSI: Digital Display Basics – HDTV Example

- 1080P@30 HDTV Standard defines 1920 horizontal "active" pixels and 1080 "active" lines giving a total of 1920 x 1080 = 2,073,600 active pixels.
- For a 30Hz frame rate
  - "Horizontal Blanking" specifies:
    - HSYNC width of 44 pixels
    - HBP of 148 pixels
    - HFP of 88 pixels
  - Total line width = 1920 + 44 + 148 + 88 = 2200 pixels.
  - "Vertical Blanking" specifies:
    - VSYNC width = 5 lines
    - VBP = 36 lines
    - VFP = 4 lines
  - Total frame height = 1080 + 5 + 36 + 4 = 1125 vertical lines.
- A full HD TV frame = 2200 x 1125 = 2,475,000 pixels
- Pixel Clock (PCLK) = (Pixels per Frame) * (FPS)
  - PCLK = 2,475,000 pixels * 30 FPS = 74.25MHz



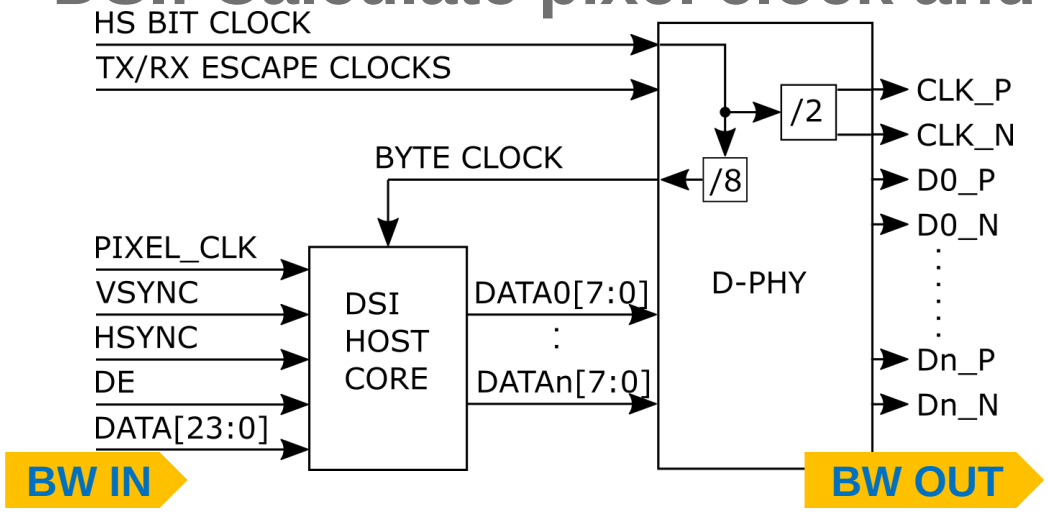
Digital Display Basics – Pixel Formats & Bit Depth

- RGB – 3 separate integers (of varying bit depths) for the values of Red, Green, Blue.
- RAW – 1 integer (of varying bit depth) for the light intensity of a single pixel.
- Color space transform (ITU-R BT.709 standard) defines RGB to Y'CbCr for HDTV.
- Y'CbCr – Y′ is the luma component and C_B and C_R are the blue-difference and red-difference chroma components. Chroma Subsampling allows for 30-50% bandwidth reduction over RGB.
- YUV – Effectively the same as Y'CbCr but used with older analog formats.

Pixel Format	bpp	Notes
24-bit RGB 8-8-8	24	8 bit integer each for the Red, Green and Blue values.
16-bit RGB 5-6-5	16	5 bit integer for the Red and Blue values, and 6-bit integer for Green.
12-bit YCbCr 4:2:0	12	8 bit integers for Y', Cb and Cr. Cb & Cr sampled every 3 pixels.
16-bit YCbCr 4:2:2	16	8 bit integers for Y', Cb and Cr. Cb & Cr sampled every 2 pixels.
RAW8	8	8 bit integer for a single pixel
RAW10	10	10 bit integer for a single pixel.
YUV422 8-bit	16	Same as Y'CbCr 4:2:2

- Critical to know the pixel depth or the **Bits Per Pixel** (bpp) for chosen format.

DSI: Calculate pixel clock and MIPI clock



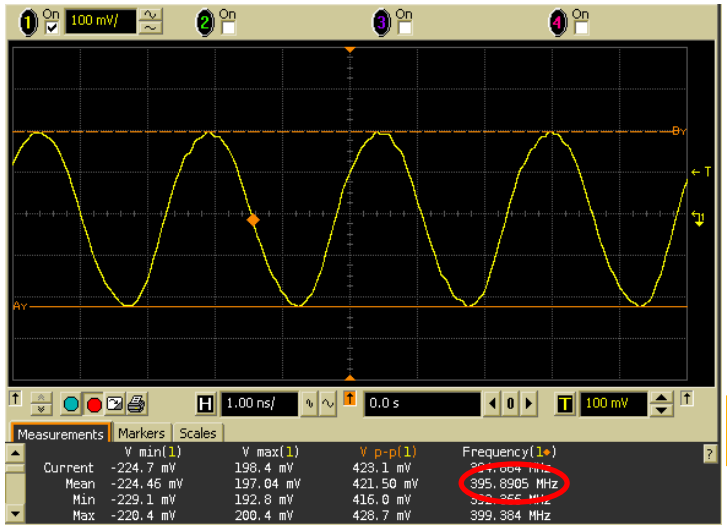
- Pixel Clock: Incoming display data clock.
- HS Bit Clock: Used to generate the HS MIPI clock. Equal to the HS **bit rate** of the data lanes.
- Byte Clock: Is 1/8th the data rate of the HS Bit Clock
- Escape Clock: Used for LP mode control and data transmission. (12MHz - 20MHz)
- BW_OUT (D-PHY) >= BW_IN (Pixel Bus)
- BW_IN = Pixel Clock * Pixel Depth
- BW_OUT = HS Bit Clock * Num. of Data Lanes
- HS Bit Clock >= Pixel Clock * Pixel Depth / Num. of Data Lanes

8MM + rm67191 OLED panel Example:
1080 x 1920, 60FPS, 24bpp

- Pixel Clock = x-active+ HSP + HBP+HFP)×(y-active + VSP + VBP +VFP)×refersh_rate =(1080+2+20+34) ×(1920+10+2+4) ×60 =132MHz
- Number of data lanes = 4
- Pixel Depth = 24bpp

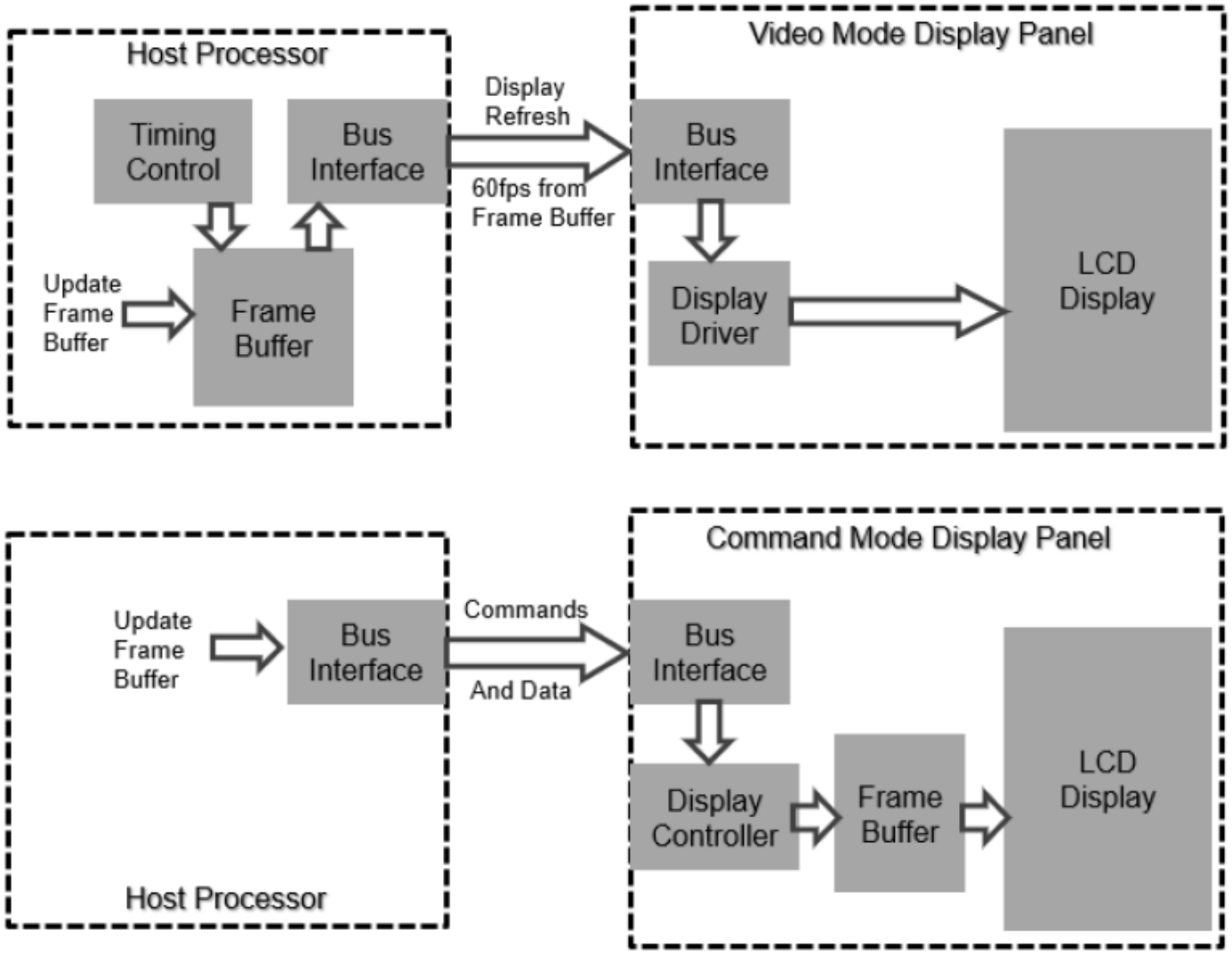
HS Bit Clock needs to be greater than or equal to:
132MHz x 24 / 4 = 792MHz (here we choose equal to)

MIPI_CLK=bitclk/2= 396MHz (clk p/n we measured in oscilloscope, DDR mode)



DSI – Video mode vs command mode

- DSI-compliant LCD support two modes:
- Which mode is used depends on the architecture and capabilities of the LCD.
- Video mode:
  - Transfer by a real-time pixel stream.
  - using High-Speed mode.
  - RGB/DOTCLK interface.
  - non burst/burst mode
- Command mode:
  - transfer by send commands.
  - Display module may include local registers and a frame buffer.
  - S-i80 interface.



DSI – Multiple Packets per Transmission

- At PHY layer, there are two modes of data transmission, High Speed (HS) and Low Power (LPS) transmission.
- A single HS transmission may consist of multiple packets, including Short Packet (SP) and Long Packet (LgP)
- Will always follow below sequence to transfer:
  - SoT Sequence->HS (will send multiple SP and LgP during it) -> EoT Sequence
  - EoT Sequence -> LPS ->SoT Sequence

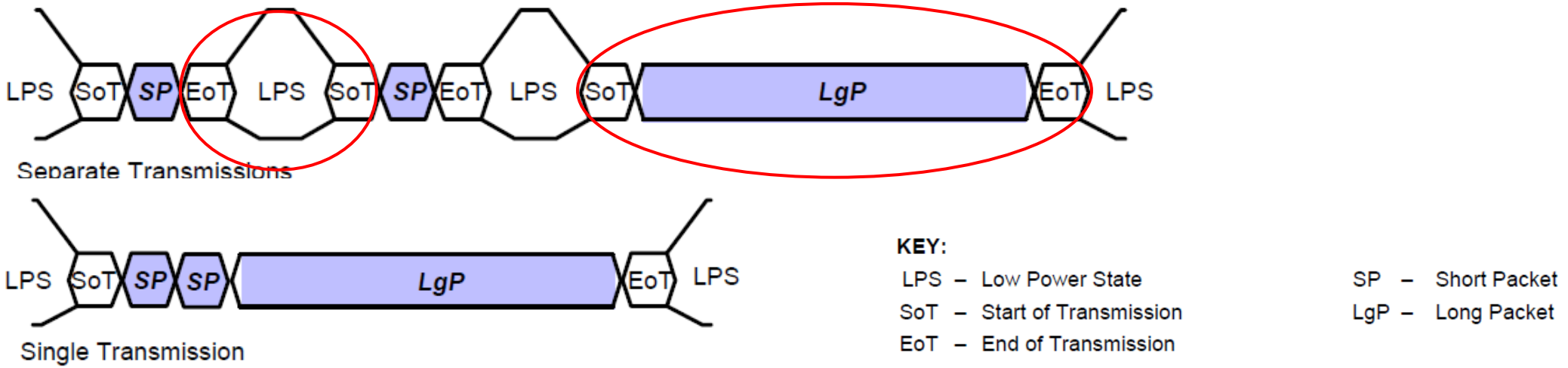


Figure 19 HS Transmission Examples with EoTp Disabled

DSI – Packet Composition

- At Low Level Protocol Layer, bytes are organized into defined groups called packets.
  - TX side: append header, ECC, footers;
  - RX side: stripped off header and interpreted by corresponding logic.
- **Short packets(SP)** (4 bytes) are used for timing critical events like Hsync and Vsync edges or Command Mode commands. Short packet format:
  -packet header: 8bit DI+ 16bit packet data + 8bit ECC
- **Long packets(LgP)** are used to transmit large blocks of pixel or other data. Long packet format:
  - packet header: 8bit DI+16bit WC+8bit ECC;
  - payload: =Length(=WC, Word count)*Data word size(8bit)
  - packet footer: 16bit Checksum

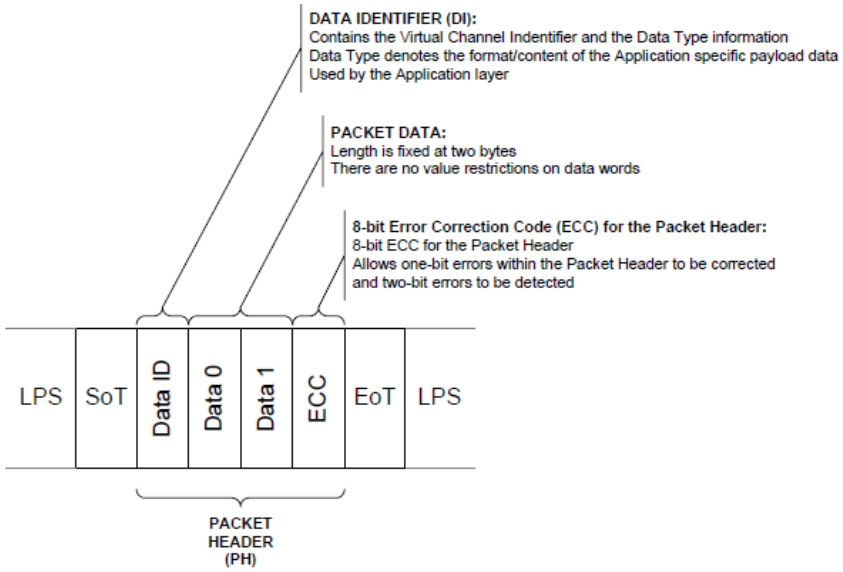


Figure 23 Short Packet Structure

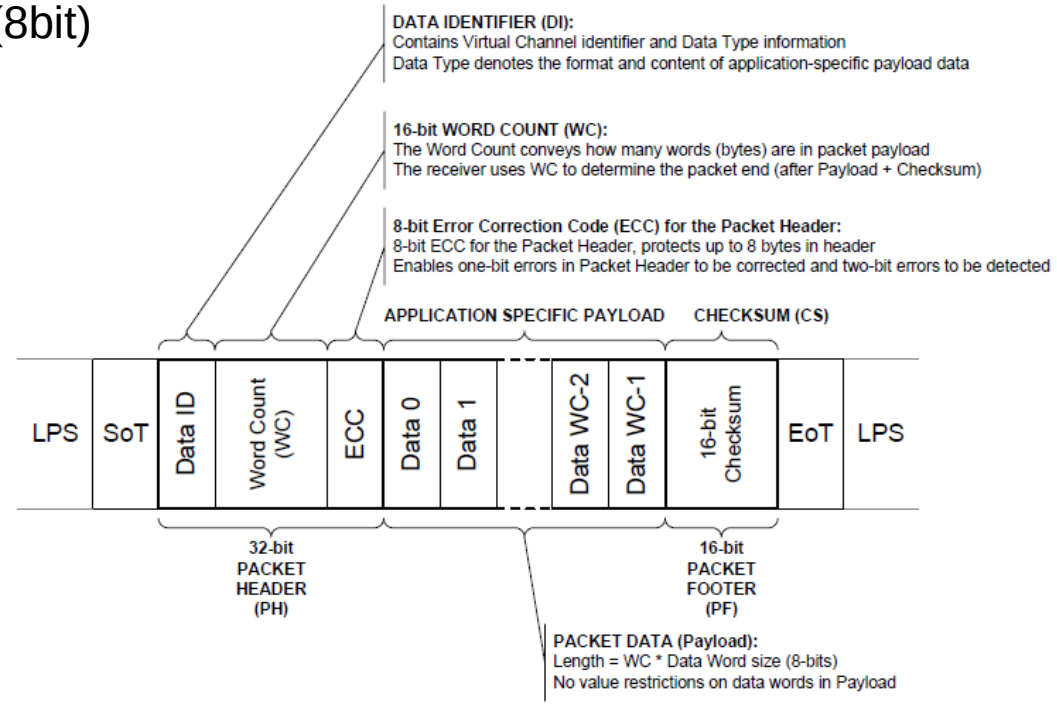
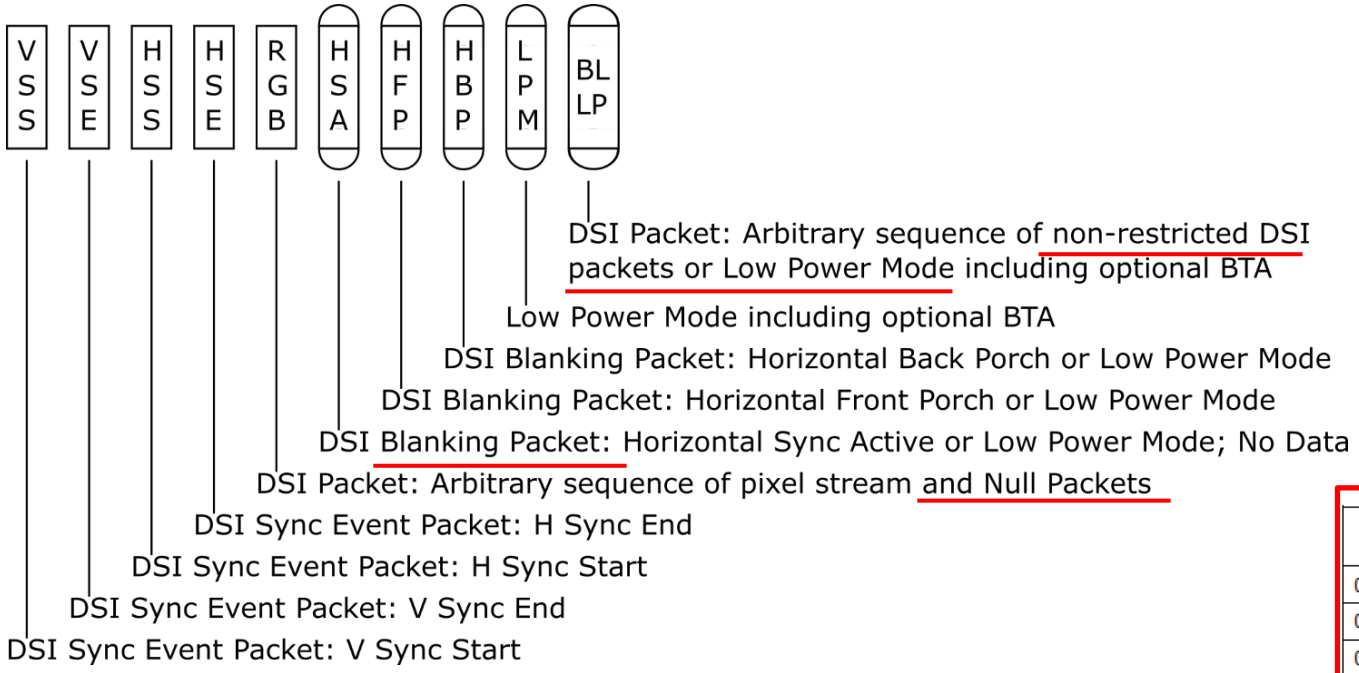


Figure 22 Long Packet Structure

DSI – Video Mode Packetization

- In video mode, it have three packet sequences:
  - Non-Burst Mode with Sync Pulses: enables the peripheral to accurately reconstruct original video timing
  - Non-Burst Mode with Sync Events: not require sync pulse, using Sync Event instead.
  - Burst mode: RGB pixel packets are time-compressed, leaving more for LP mode or other transmissions



- DSI know different packet type from DI(Data Identifier) in packet header, where DI[5:0] specify the Data Type.

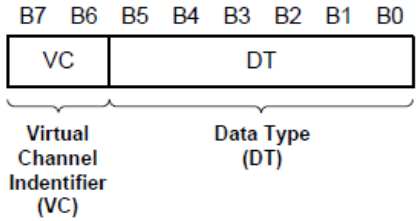
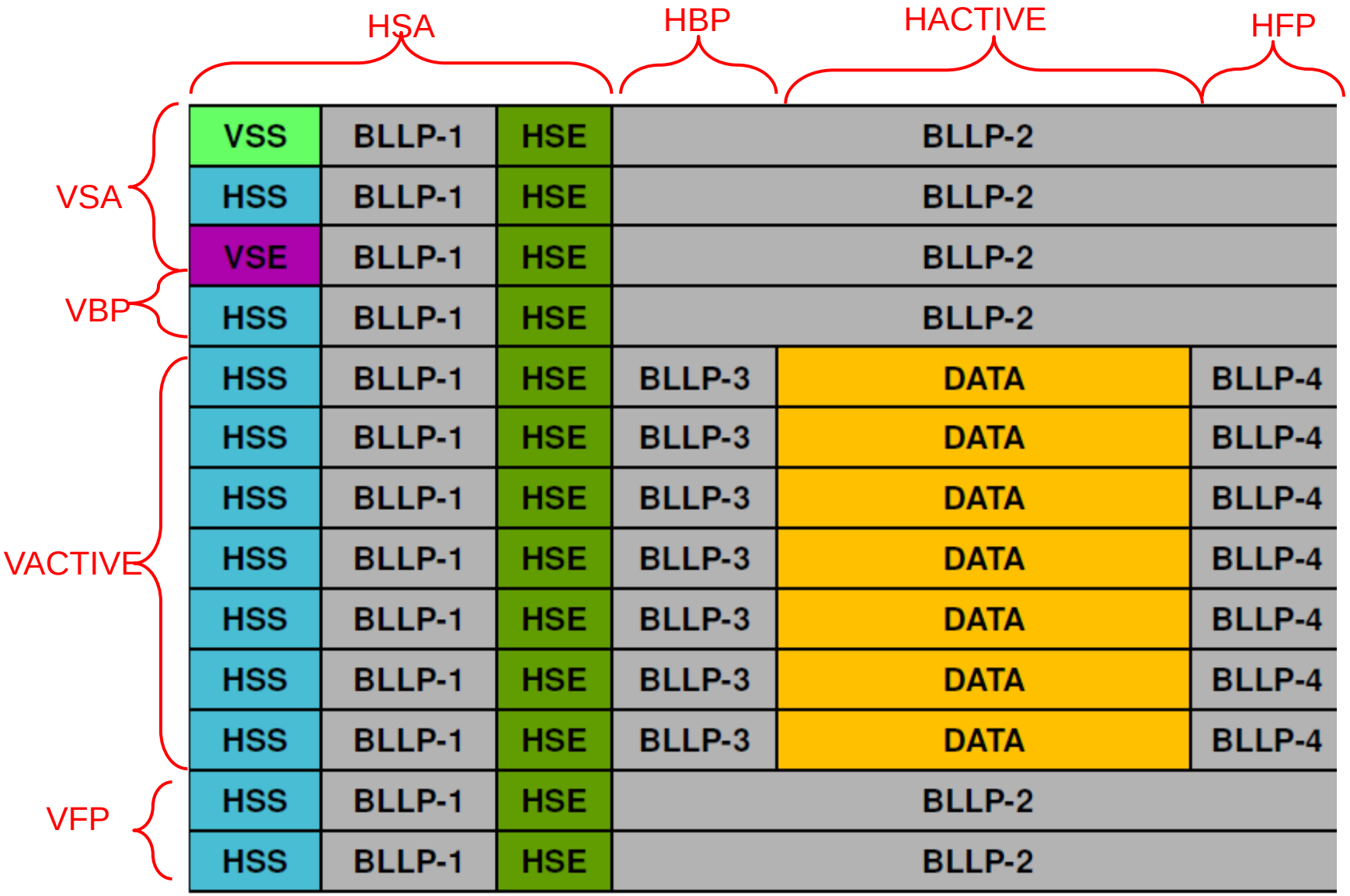


Figure 24 Data Identifier Byte

Table 16 Data Types for Processor-Sourced Packets

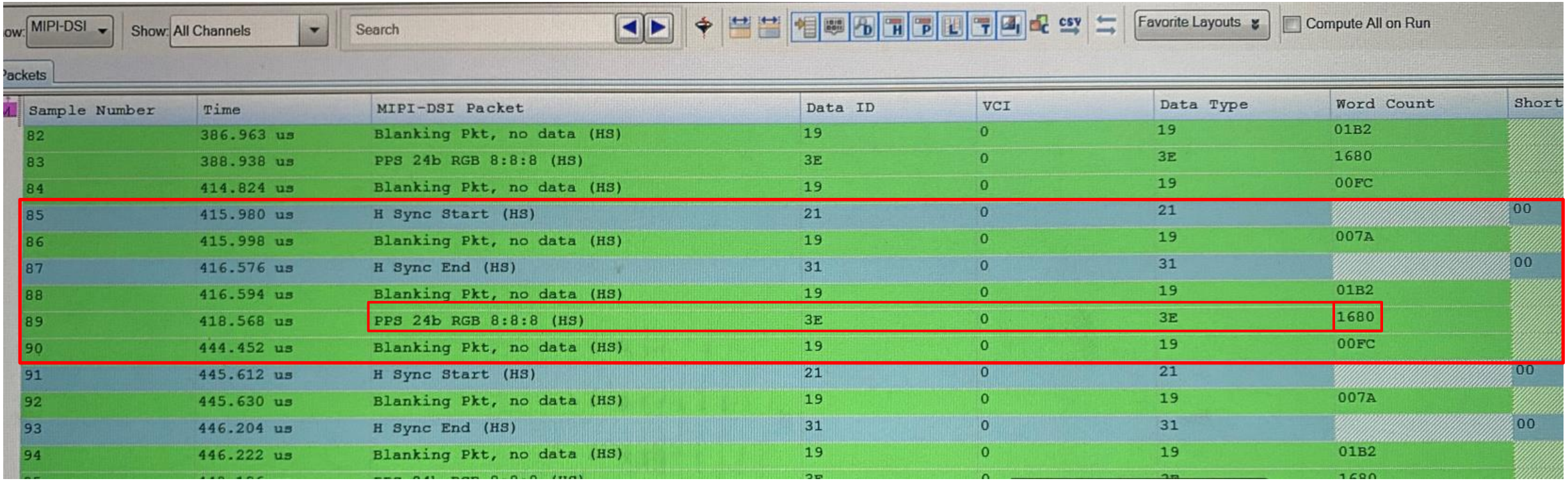
Data Type, hex	Data Type, binary	Description	Packet Size
0x01	00 0001	Sync Event, V Sync Start	Short
0x11	01 0001	Sync Event, V Sync End	Short
0x21	10 0001	Sync Event, H Sync Start	Short
0x31	11 0001	Sync Event, H Sync End	Short
0x19	01 1001	Blanking Packet, no data	Long
0x3E	11 1110	Packed Pixel Stream, 24-bit RGB, 8-8-8 Format	Long

DSI – Video Mode Packetization – eg: Non-Burst Mode with Sync Pulses



- The figure illustrate packet sequence of each frame.
- HSS, HSE appears once in each line for most time.
- VSS,VSE appears once in each frame, will instead HSS.
- BLLP, DATA are short packets.
- VSS,VSE,HSS,HSE are long packets.
- DSI host calculate BLLP-1 length to satisfy period of HSA.
- Calculate BLLP-3 for HBP.
- Calculate BLLP-4 for HFP.
- Calculate BLLP-2 for HBP+ HACTIVE+HFP.
- No HSE and VSE for sync events and burst mode.

DSI – Packet Composition in Analyser

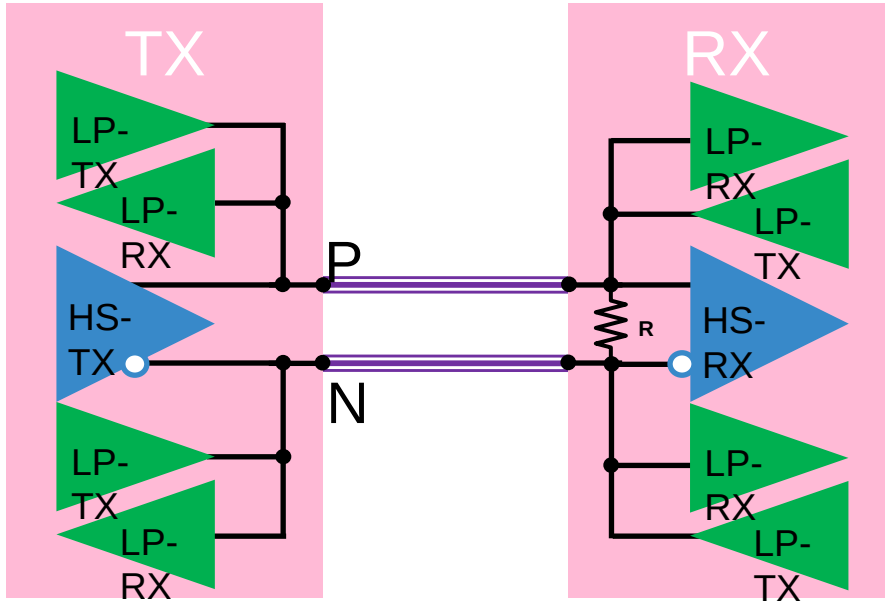


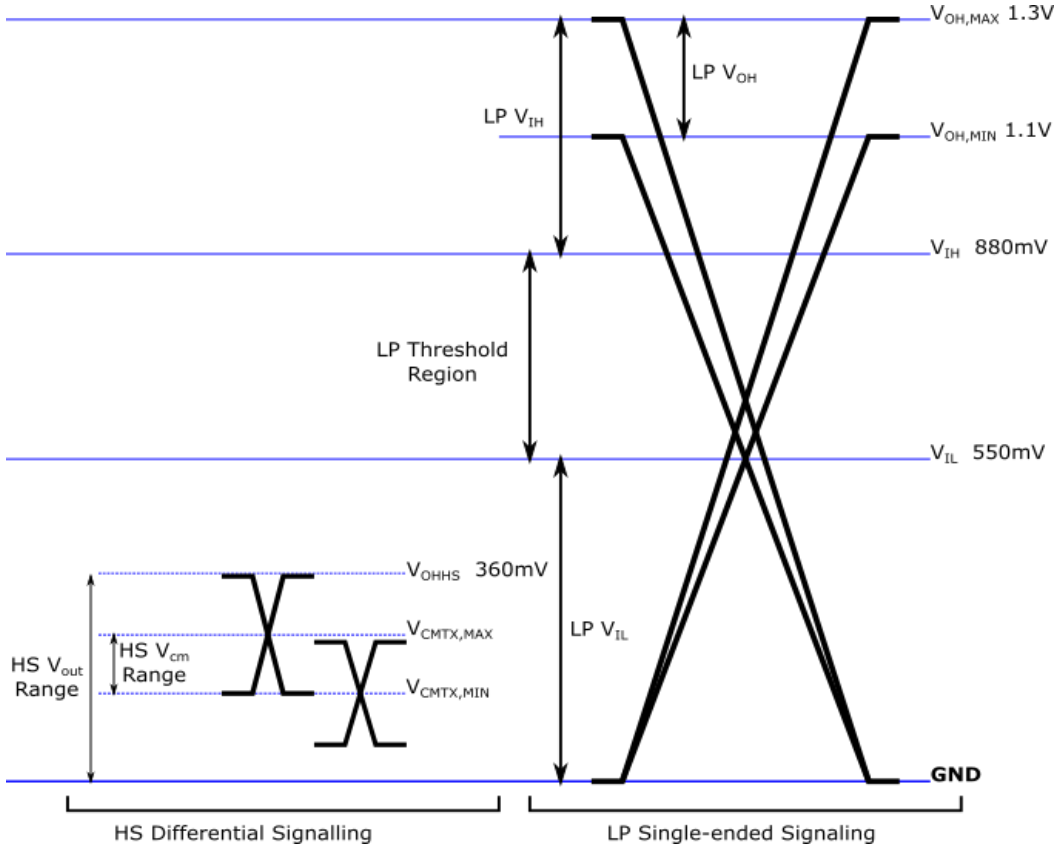
Sample Number	Time	MIPI-DSI Packet	Data ID	VCI	Data Type	Word Count	Short
82	386.963 us	Blanking Pkt, no data (HS)	19	0	19	01B2	
83	388.938 us	PPS 24b RGB 8:8:8 (HS)	3E	0	3E	1680	
84	414.824 us	Blanking Pkt, no data (HS)	19	0	19	00FC	
85	415.980 us	H Sync Start (HS)	21	0	21		00
86	415.998 us	Blanking Pkt, no data (HS)	19	0	19	007A	
87	416.576 us	H Sync End (HS)	31	0	31		00
88	416.594 us	Blanking Pkt, no data (HS)	19	0	19	01B2	
89	418.568 us	PPS 24b RGB 8:8:8 (HS)	3E	0	3E	1680	
90	444.452 us	Blanking Pkt, no data (HS)	19	0	19	00FC	
91	445.612 us	H Sync Start (HS)	21	0	21		00
92	445.630 us	Blanking Pkt, no data (HS)	19	0	19	007A	
93	446.204 us	H Sync End (HS)	31	0	31		00
94	446.222 us	Blanking Pkt, no data (HS)	19	0	19	01B2	

- In red block, the six packets stands for one line including valid pixel:
  - Consist of HSS, BLLP-1, HSE, BLLP-3, PPS data, BLLP-4 packet
- Here i.MX93 output DSI signal, 1920x1080@30fps, RGB888 format (24bpp=3bytes/pixel), 3lane, HSA=48, HBP=148, HACT=1920, HFP=88,VSA=5, VBP=36, VACT=1080, VFP=4
- For each 24bit packet pixel stream(pps), it stands for one line's valid data, its Word Count=h-active*bpp/data_word_size(1byte) =1920 pixel * 3bypte/pixel / 1 byte=5760=0x1680

D-PHY IO Architecture

- D-PHY could be configured as Master (data source) or Slave (data sink).
- Bi-directional data transmission in Low-Power mode at the Master Data Lane 0 only.
- Two modes of data transmission:
  - High Speed: 80~1.5Gbps (spec v1.2), 100~300mV
  - Low Power: <10Mbps, 0~1.2V
- Single Lane IO Block Diagram Below:

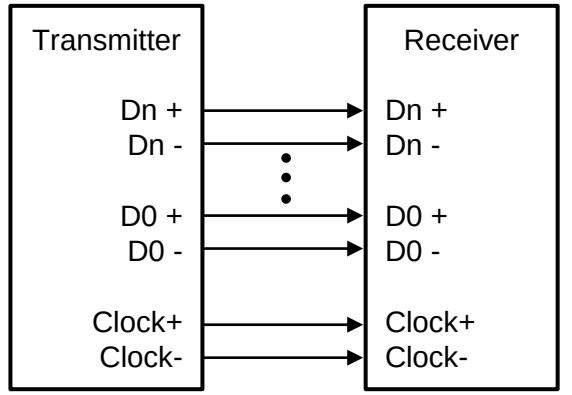




Parameter Description	Min	Typ	Max	Units
LP Output High	1.1	1.2	1.3	V
LP Input low level threshold (Vil)			550	mV
HS Common Mode Voltage	150	200	250	mV
HS TX Differential Voltage	140	200	270	mV
HS Output High Voltage			360	mV

D-PHY Lanes & States

- Dedicated Clock lane and configurable number (1-4) of data lanes.
- Dual Data Rate (DDR) Clock scheme used for HS mode.
- During normal operation lanes are either in Low Power or High-Speed mode. They'll constantly switch back and forth between modes to save power.
- Define six different "lane states".



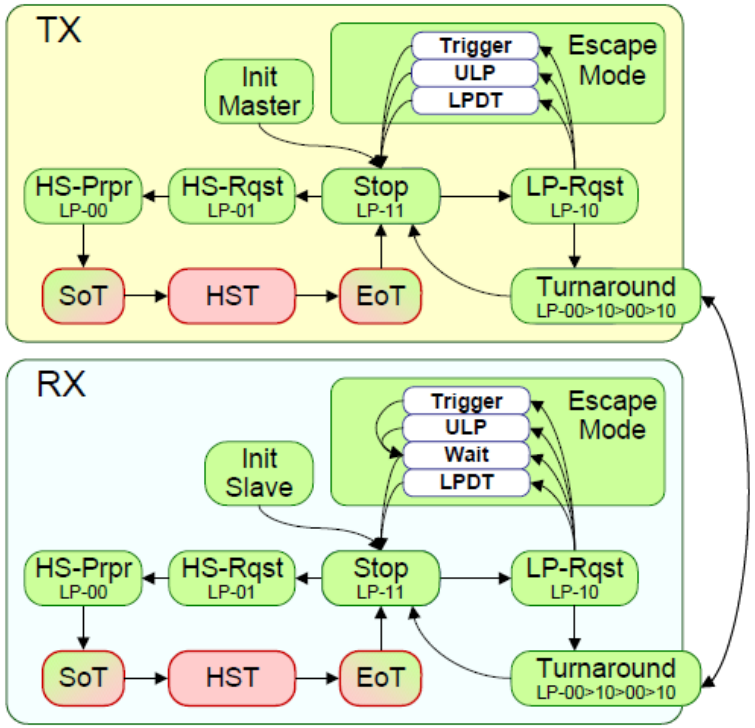


Figure 24 Data Lane Module State Diagram

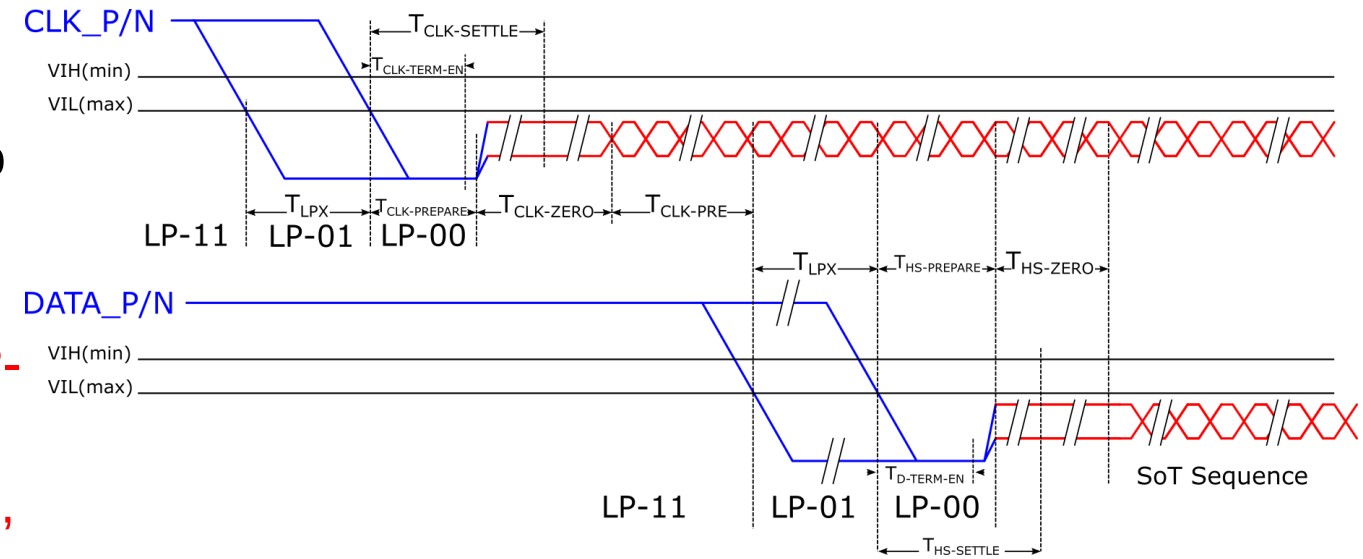
State Code	Line Voltage Levels		High-Speed	Low-Power	
	Dp-Line	Dn-Line	Burst Mode	Control Mode	Escape Mode
HS-0	HS Low	HS High	Differential-0	N/A [1]	N/A [1]
HS-1	HS High	HS Low	Differential-1	N/A [1]	N/A [1]
LP-00	LP Low	LP Low	N/A	Bridge	Space
LP-01	LP Low	LP High	N/A	HS-Request	Mark-0
LP-10	LP High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

D-PHY Transition from LP to HS Mode

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

1. The clock lane enters HS mode through a specific timing sequence (LP-11, LP-01, LP-00).
2. The data lane then enters HS mode (LP-11, LP-01, LP-00) and a transmits a “Start-of-Transmission” (SoT) bit sequence.

- Critical to properly configure the timing parameters for both the TX and the RX to ensure proper entry into HS mode.
- Most of the configurable parameters are on the transmitter side. (such as Tclk-prepare, Tclk-zero, Tclk-pre, Ths-prepare, Ths-zero, Ths-settle)

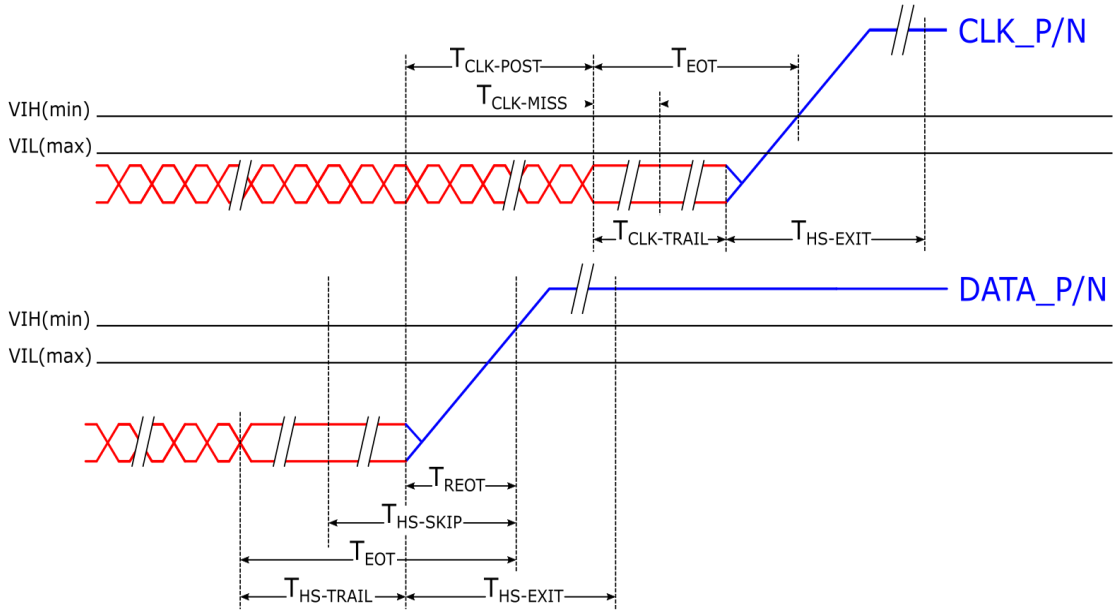


TX Side	RX Side
LP-01 for time T_{LPX}	Observes transition from LP-11 to LP-01
LP-00 for time $T_{HS/CLK-PREPARE}$	Observes transition from LP-01 to LP-00. Enables Termination
Switch from LP to HS driver	Enable HS RX and wait $T_{HS/CLK-SETTLE}$
Drives HS-0 for $T_{HS/CLK-ZERO}$	
Drives HS clock for $T_{CLK-PRE}$	
Data Lane drives SoT sequence (00011101)	

Start-of-Transmission Sequence

D-PHY Transition from HS to LP Mode

- At the end of a High-Speed Data Burst, lane(s) leave HS mode and enter the Stop State (LP-11) by means of an End-of-Transmission (EoT) procedure.
- The clock lane can optionally switch back to a Low Power State. Staying in HS Mode is known as ***continuous clock*** mode.
- Continuous clock mode allows for higher data rates since the timing overhead of exiting and re-entering HS-mode on the clock lane is eliminated



Clock Lane Transition Procedure

TX Side	RX Side
HS Clock continues for $T_{CLK-POST}$	
Clock drives HS-0 for $T_{CLK-TRAIL}$	Detects absence of clock within a time $T_{CLK-MISS}$
Switch from HS to LP and drive LP-11 for $T_{HS-EXIT}$	
	Detects transition LP-11, disables HS termination and enters stop state.

Data Lane Transition Procedure

TX Side	RX Side
Data drives opposite bit immediately after last data packet for $T_{HS-TRAIL}$	
Switch from HS to LP and drive LP-11 for $T_{HS-EXIT}$	Detects transition to LP-11, disables HS termination.
	Neglects bits of last period during $T_{HS-SKIP}$

Bring up MIPI panel – Find Some Materials

- IP spec of display controller/interface of i.MX side, usually get from Reference Manual .
- Datasheet/spec/schematic of the peripherals when porting the panel/camera, search from Internet or ask from vendor.
- MIPI spec when porting MIPI panel/camera or check some details
  https://www.mipi.org/specifications
- Display background: Including RGB mode timing, calculate pixel clock/MIPI clock, framebuffer, LCD/OLED drive theory…
- For i.MX6/7/8, refer Linux DRM Documentation:
  https://www.kernel.org/doc/html/latest/gpu/drm-kms.html

Bring up MIPI panel – on i.MX8

- Refer panel-raydium-rm67191.c/panel-simple.c + dts to write the driver, find below info:
- General Specification/Datasheet -- Get from vendor/website:

  Hactive, Vactive, Lane num
- Timing: Hactive, HFP, HBP, Hsync, Vactive, VFP, VBP, Vsync
- Power up sequence: VCC, RST
- Backlight: OLED no need backlight,

  LCD need backlight IC, usually PWM dimming
- Init cmds -- get from vendor: Generic scope, DCS scope
- Check clocks: MIPI bit clock, pixel clock , escape clock
- DSI related software settings
  - Depends on panel requirement
  - Make sure that SoC and panel video modes are consistent.
  - Video mode/Command mode, burst/non-burst mode, continuous/non-continuous clock, EoTp enabled/disabled, BLLP mode…

DE mode

Parameter	Symbol	Value			Unit
		Min.	Typ.	Max.	
DCLK frequency @Frame rate=60hz	fclk	40.8	51.2	67.2	Mhz
Horizontal display area	thd		1024		DCLK
HSYNC period time	th	1114	1344	1400	DCLK
HSYNC blanking	thb+thfp	90	320	376	DCLK
Vertical display area	Tvd		600		H
VSYNC period time	Tv	610	635	800	H
VSYNC blanking	Tvb+Tvfp	10	35	200	H

Example: Input Timing Table getting from Rocktech RK070CU05H-CTG spec

5	RESET	I	Global reset pin. Active low to enter reset state.

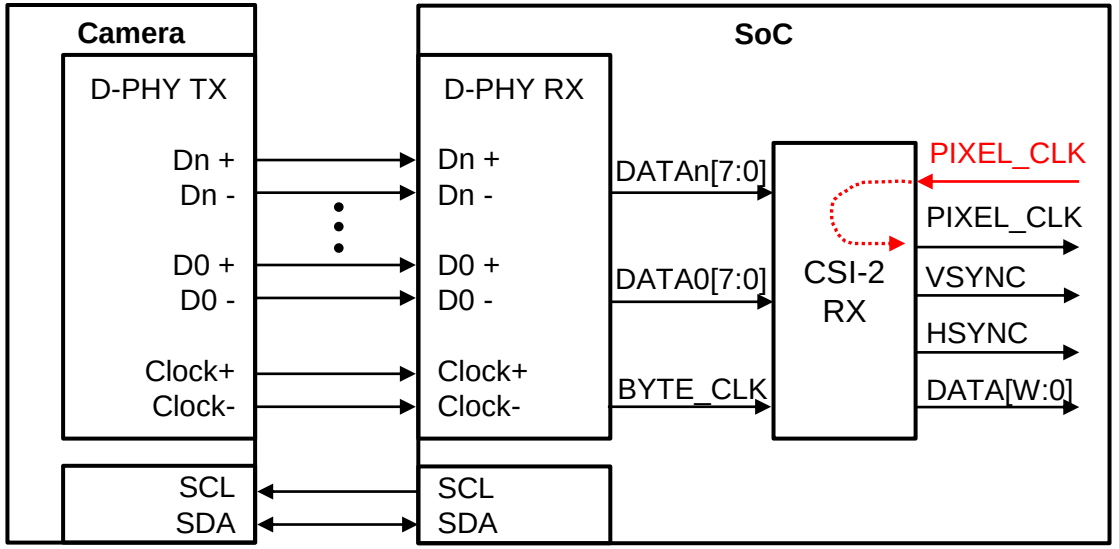
Example: Pin Description getting from Rocktech RK070CU05H-CTG spec

Test & Debug MIPI panel – on i.MX8

- Consider whole system work, HW & SW
  ----identify the problem is HW or SW, ensure the HW is good then debug the SW..
- Check framebuffer data using memtool or debugger (ARM DS-5, Trace32)
- Clock: lcdif pixel clock, dsi phy ref clock, dsi core clock, dsi esc tx clk.
  - Make sure all clock could be generated properly
  - check clock tree: cat /sys/kernel/debug/clk/clk_summary
- Do some test: Using Gstreamer, unit_tests, modetest. Such as:
  - systemctl stop weston*
  - modetest -M mxsfb-drm -s 34:1080x1920-60.02 (34 is connector id) plays color bar
- Check some logs: Enable DRM debug: drm.debug=0x1f in mmcargs
- Check display controller and interface registers, especially MIPI D-PHY
- Oscilloscope
- Using MIPI analyser

Camera Serial Interface Version 2 (CSI-2)

- Think of it like DSI but with an additional I2C link which allows for more advanced command and control capabilities of a camera module.
- CSI-2 **transmitter** turns the pixel data, VSYNC and HSYNC pulses into a series of byte-sized packets to be sent over the physical D-PHY interface while maintaining the critical timing parameters relationship to each other.
- The **receiver** rebuilds the video datastream – Vsync, Hsync, pixel data bus.
- Supports different pixel formats (YUV, RAW, RGB…)



- BYTE_CLOCK: generated automatically by the D-PHY and is always 1/8 the speed of the HS bit clock.
- PIXEL_CLK: generated by the SOC and needs to be set so that image bandwidth exiting the CSI-2 block is *greater than or equal* to the image bandwidth entering the block.

CSI-2 Frame Format

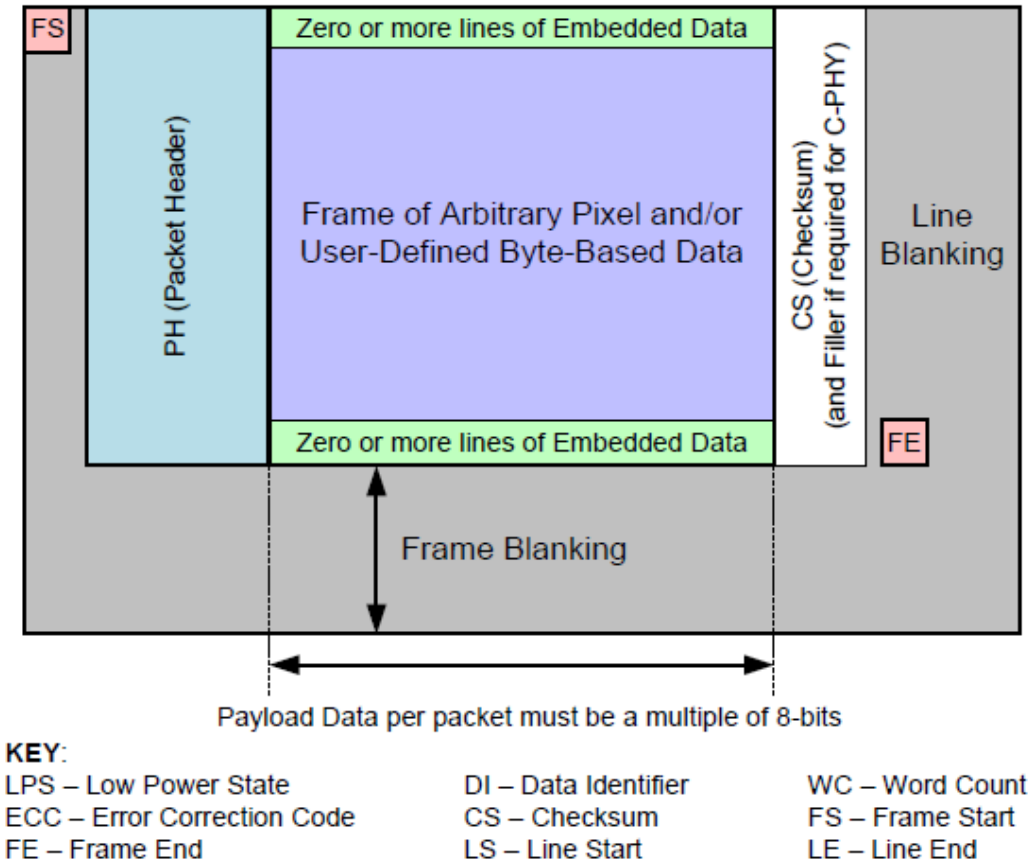
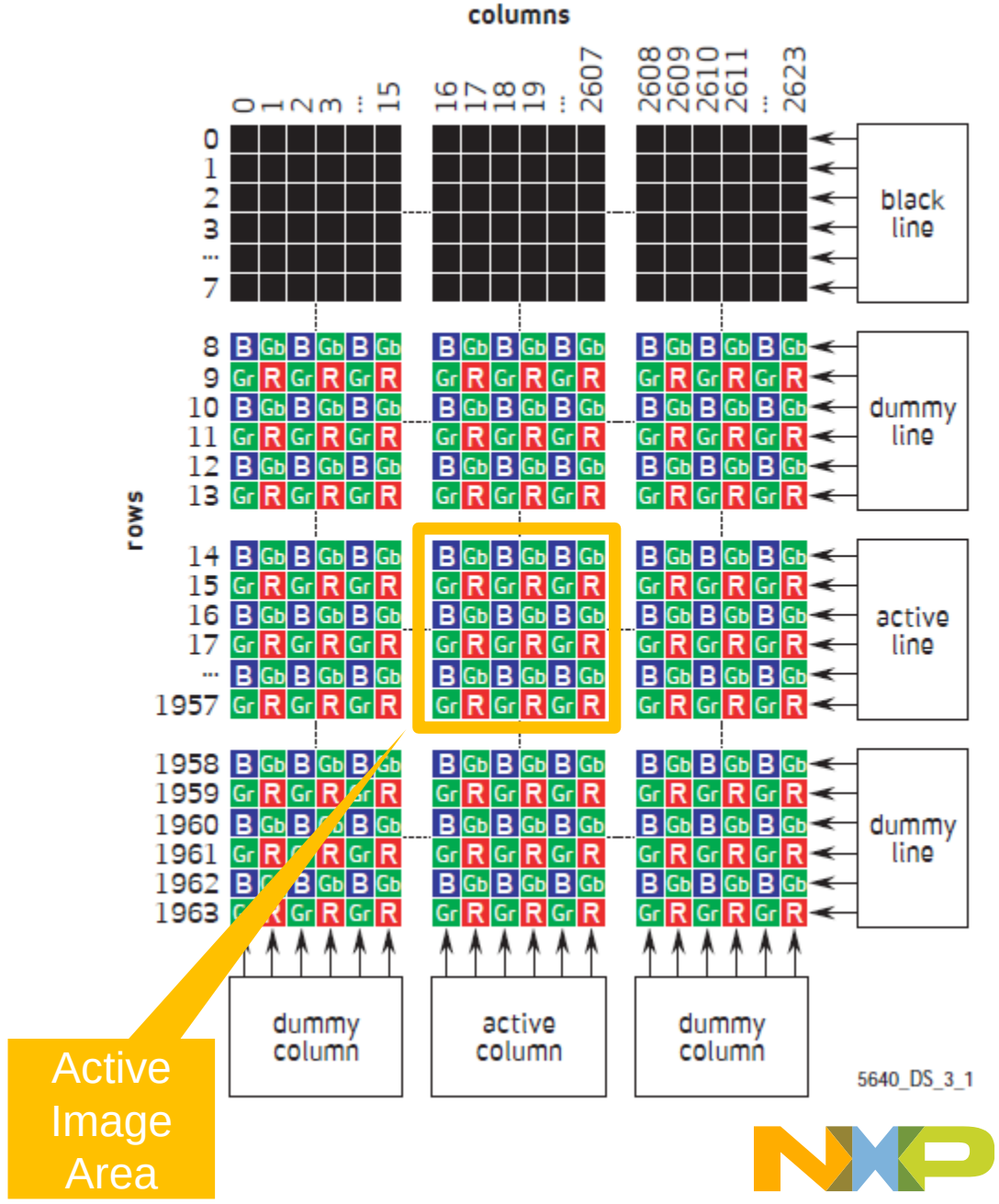


Figure 108 Frame Structure with Embedded Data at the Beginning and End of the Frame

- Similar in concept to DSI frames but different terminology
- Line Blanking=Horizontal Blanking
- Frame Blanking=Vertical Blanking
- Support for embedded data – used for image sensor information (gain, exposure time, etc…)

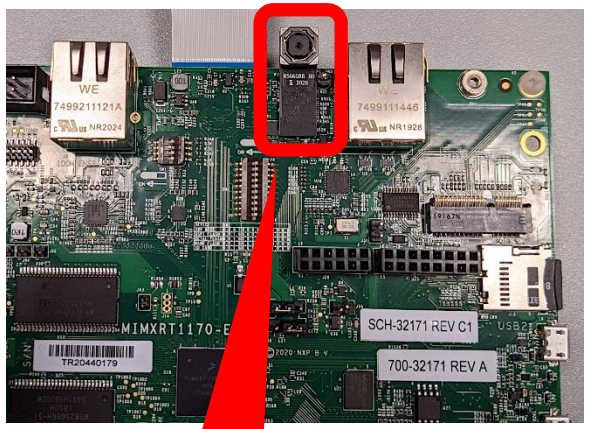
CSI-2: Image Sensor Basics

- An image sensor contains an array of photosensitive elements – pixels.
- The color filters are arranged in a Bayer pattern. The primary color BG/GR array is arranged in line-alternating fashion.
- The OV5640 has a pixel array of 2624 columns by 1964 rows for a total of 5,153,536 pixels. Of the 5,153,536 pixels, 5,038,848 (2592x1944) are active pixels and can be output. Active columns are 16 – 2607 and active lines are lines 14 – 1957. The other pixels are used for black level calibration and interpolation.
- Image data is transferred just like raster displays – a stream of pixel data left to right, top to bottom and with no (X,Y) coordinate data.

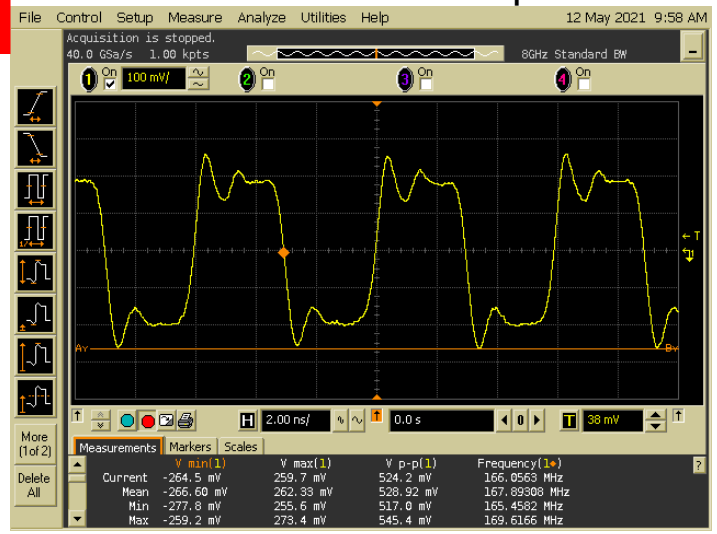


CSI-2 Example: RT1170 + OV5640

- Need to configure both the CSI-2 RX clocks and the camera clocks.
- RX has only one tunable parameter: $T_{HS-SETTLE}$
- 1280 x 720 resolution at 30FPS
- OV5640 Hsync is 612 pixels which gives an HTOT = 1892.
- OV5640 Vsync is 2 lines and the Vfp + Vbp = 18. So the VTOT = 720 + 18 + 2 = 740 lines.
- OV5640 Pixel Clock = 1892 * 740 * 30 = 42MHz
- The RGB565 format requires 16bpp so the bandwidth = 42MHz * 16 bpp = 672Mbps
- This system uses a 2-lane MIPI interface so the data rate per lane is = 672 Mps / 2 = 336Mbps
- The MIPI D-PHY HS Bit Clock therefore needs to be: 336Mbps / 2 = 168MHz

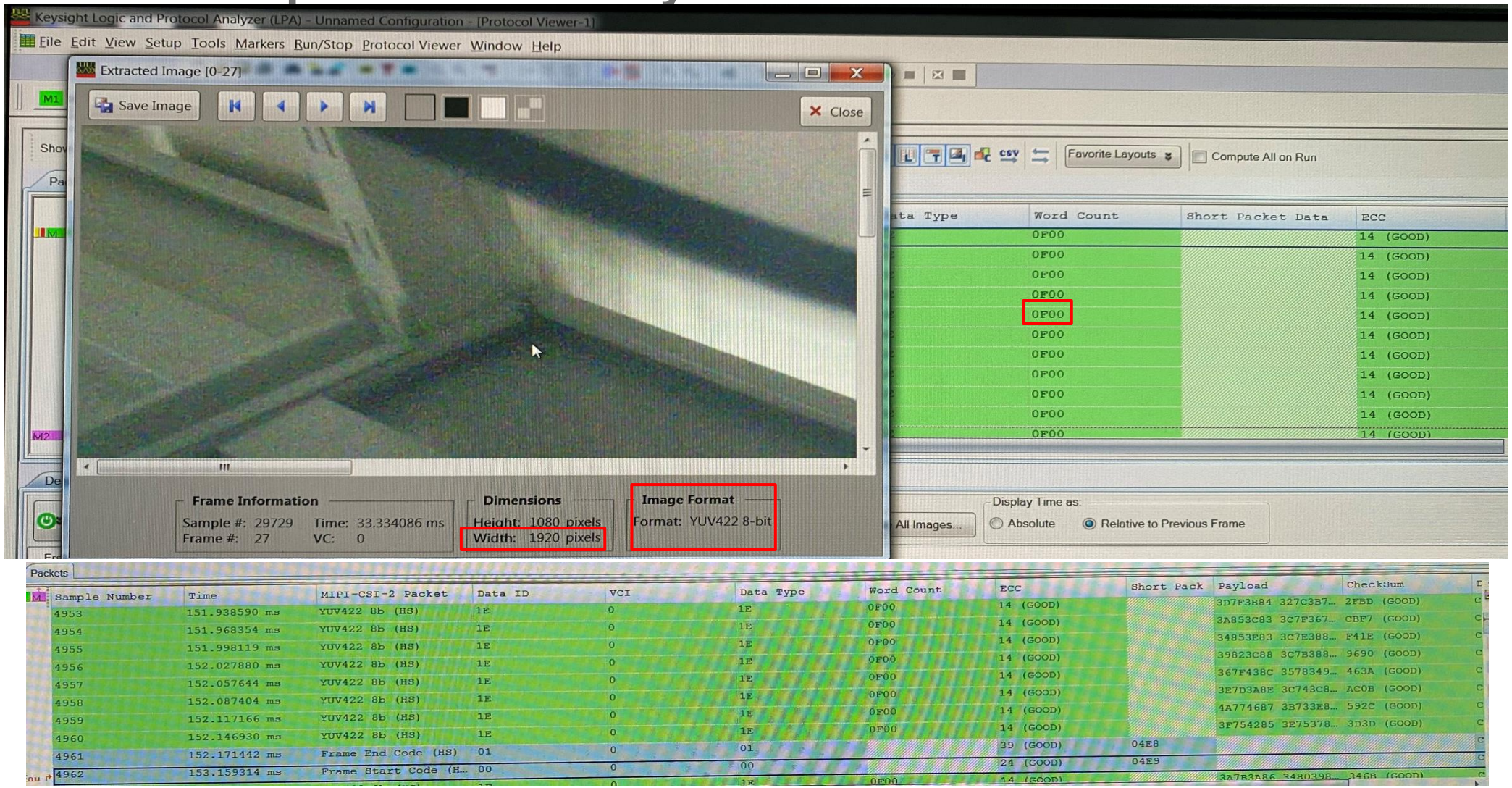


5.5” LCD displaying OV5640 camera input



OV5640 D-PHY Clock = 168MHz

CSI-2: Packet Composition in Analyser



- Here ov5640 send 1920x1080, YUV422 format (16bpp=2bytes/pixel) on i.MX8MQ, MIPI analyser is slave,
- Each frame will generate [1 short packet (frame start),1080 long packet, 1 short packet (frame end)] ,,, repeat
  For each long packet, it stands for one line data,
  Word Count=x-active*bpp/data_word_size(1byte) =1920 pixel * 2bypte/pixel / 1 byte=3840=0xF00

Bring up MIPI sensor

- Ensure the hardware design/connection is good
- Schematic + dts:
  - VDD supply voltage, RESET PWDN
- MCLK ---- Where to get?
  - Got from crystal on adapter board, or from i.MX EVK.
  - SYS_CLK, PCLK, and MIPI_CLK are made from camera's PLL block based on MCLK.
- Power up sequence
  - Take care timing constraints
- Register setting: (Getting from vendor):
  - Format, resolution, framerate, MIPI lane numbers
- Valid sensor I2C communication is correct using Linux I2C commands
- Tune RX_HS_SETTLE for MIPI CSI PHY
- After starting capture, check MIPI CSI PHY register to ensure no error.
- Could watch the interrupt:

  Start of Frame, End of Frame, DMA transfer done…

```
// Frame Rate       : 30fps
// Global Reeset    : Off
// Global Hold      : On
// Fixed Frame      : On
// Event Clock      : 100Mhz
// MIPI Lane Speed : 1Gbps
// Num Of MIPI Lane: 4
// Data Type        : RAW8
// Payload Size     : 7936 Bytes
// Num of Payload   : 20
```

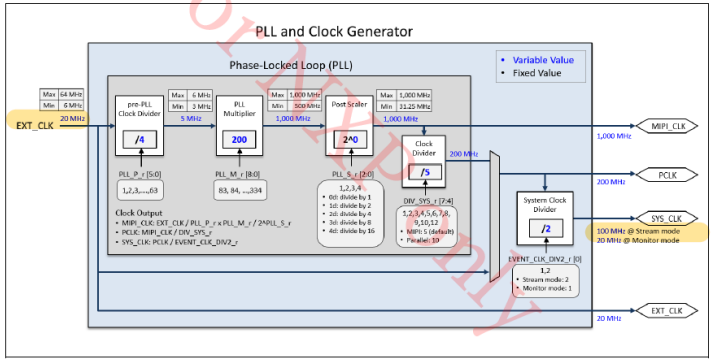


Figure 27 **Clock and Clock Generator Structure and Default Setting for MIPI Interface**

NOTE: All the setting values of registers are represented by decimal unit.
Use the following formula to calculate MIPI_CLK, PCLK, and SYS_CLK.

Clock Output
MIPI_CLK = (External Clock / PLL_P_r) X (PLL_M_r / 2^PLL_S_r)
PCLK = MIPI_CLK / DIV_SYS_r
SYS_CLK= PCLK / EVENT_CLK_DIV2_r

CSI2 RX IRQ status
[0] – crc error
[1] – one bit ecc error
[2] – two bit ecc error
[3] – ULPS status change
[4] – DPHY ErrSotHS has occurred
[5] – DPHY ErrSotSync_HS has occurred
[6] – DPHY ErrEsc has occurred
[7] – DPHY ErrSyncEsc has occurred
[8] – DPHY ErrControl has occurred

Test & Debug MIPI sensor – on i.MX8

- Test methods:
  - GStreamer pipeline
  - v4l2-ctl
  - unit_tests/V4L2/mx6s_v4l2_capture.out
- Preview the captured data: (Choose width, height, image type)
- - YUV player (preview YUV data), ImageJ (preview RAW data)
- Oscilloscope
- Check framebuffer data using memtool or debugger (ARM DS-5, Trace32)
  - Could play a color bar and then check the memory data.
- Debug the MIPI sensor – Using MIPI analyser
  - For MIPI CSI, camera sensor is master, MIPI analyser is slave.
  - Check if camera outputs well, couldn't verify MIPI CSI RX and CSI bridge on i.MX is correct.
  - After stream on, choose lane num, get stabile bit rate.
  - Could get Data type, word count, Short Packet Data, ECC, Payload, CheckSum…
  - Could save .csv file

CSI-2: Samsung DVS sensor on 8MQ/8MM Example

- Basic info:
  - 640x248
  - 30fps
  - 4 lanes
  - RAW8 (need format conversion)

- Several meet issues and solutions:

- D-PHY Start-of-Transmission Synchronization Error:
  Run test procedure fail, all zeros in framebuffer
  Tune(decrease) hs_settle for MIPI CSI PHY

- Doesn’t save full framebuffer:
  Report base address switching Change Err
  Modify register to fix the issue.

```
// Frame Rate      : 30fps
// Global Reeset   : Off
// Global Hold     : On
// Fixed Frame     : On
// Event Clock     : 100Mhz
// MIPI Lane Speed : 1Gbps
// Num Of MIPI Lane: 4
// Data Type       : RAW8
// Payload Size    : 7936 Bytes
// Num of Payload  : 20
```

SECURE CONNECTIONS
FOR A SMARTER WORLD