Embedded Systems
Hardware Design

IO and interconnect fabrics in context of System of Chips (SoCs) and System in Package (SiP)

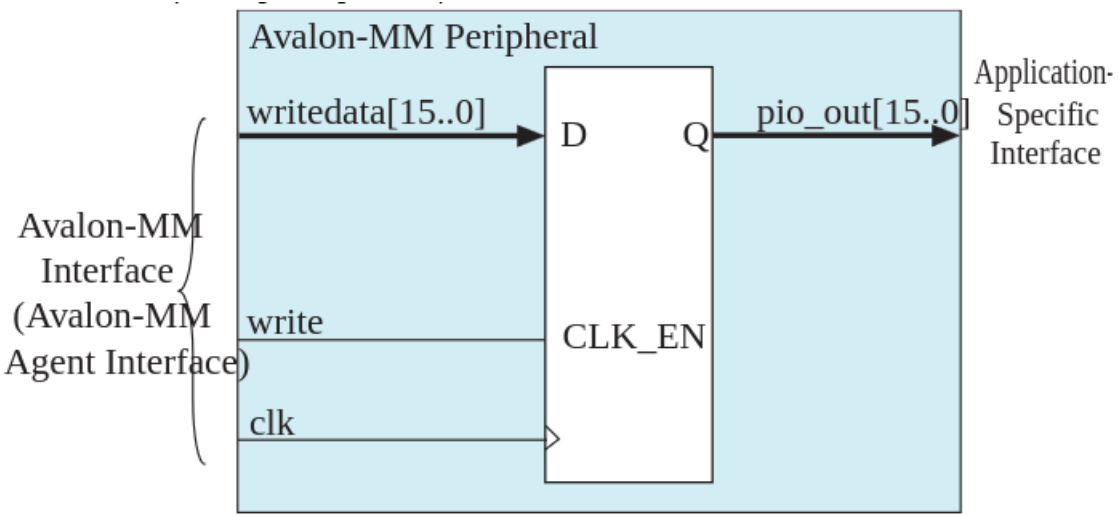
Avalon

The Avalon bus is a simple bus architecture designed for connecting on-dchip processors and peripherals together into a system-on-a-programmable chip (SOPC). The Avalon bus is an interface that specifies the port connections between master and slave components, and specifies the timing by which these components communicate.

- Simplicity - Provide an easy-to-understand protocol with a short learning curve.
- Optimized resource utilization for bus logic - Conserve logic elements
- (LEs) inside the Programmable Logic Device (PLD).
- Synchronous operation - Integrate well with other user logic that coexists on the same PLD, while avoiding complex timing analysis issues.

Avalon – IP Core example

Avalon-MM Peripherial doesn't require implementations of all avilable Avalon signals. IP Core that controls LED require only three input signals: writedata, write and clk.



Source Avalon® Interface Specifications

Avalon – port signals

Table 2. Avalon Slave Port Signals (Part 1 of 2)

Signal Type	Width	Direction	Required	Description
`clk`	1	in	no	Global clock signal for the system module and Avalon bus module. All bus transactions are synchronous to `clk`. Only asynchronous slave ports can omit `clk`.
`reset`	1	in	no	Global reset signal. Implementation is peripheral-specific.
`chipselect`	1	in	yes	Chip select signal to the slave. The slave port should ignore all other Avalon signal inputs unless `chipselect` is asserted.
`address`	1 - 32	in	no	Address lines from the Avalon bus module.
`begintransfer`	1	in	no	Asserted during the first bus cycle of each new Avalon bus transfer. Usage is peripheral-specific.
`byteenable`	0, 2, 4	in	no	Byte-enable signals to enable specific byte lane(s) during transfers to memories of width greater than 8 bits. Implementation is peripheral-specific.
`read`	1	in	no	Read request signal to slave. Not required if the slave never outputs data to a master. If used, `readdata` must also be used.
`readdata`	1 – 32	out	no	Data lines to the Avalon bus module for read transfers. Not required if the slave never outputs data to a master. If used, `read` signal must also be used.
`write`	1	in	no	Write request signal to slave. Not required if the slave never receives data from a master. If used, `writedata` must also be used.
`writedata`	1 – 32	in	no	Data lines from the Avalon bus module for write transfers. Not required if the slave never receives data from a master. If used, `write` signal must also be used.

Source https://eclipse.umbc.edu/robucci/cmpeRSD/discussions/discussion10/

Avalon – port signals

readdatavalid	1	out	no	Used only by slaves with variable latency. Marks the rising clock edge when the slave asserts valid readdata.
waitrequest	1	out	no	Used to stall the Avalon bus module when slave port is not able to respond immediately.
readyfordata	1	out	no	Signal for streaming transfers. Indicates that the streaming slave can receive data.
dataavailable	1	out	no	Signal for streaming transfers. Indicates that the streaming slave has data available.
endofpacket	1	out	no	Signal for streaming transfers. May be used to indicate an “end of packet” condition to the master port. Implementation is peripheral-specific.
irq	1	out	no	Interrupt request. Slave asserts irq when it needs to be serviced by a master.
resetrequest	1	out	no	A reset signal allowing a peripheral to reset the entire system module.

Source https://eclipse.umbc.edu/robucci/cmpeRSD/discussions/discussion10/

AMBA AXI

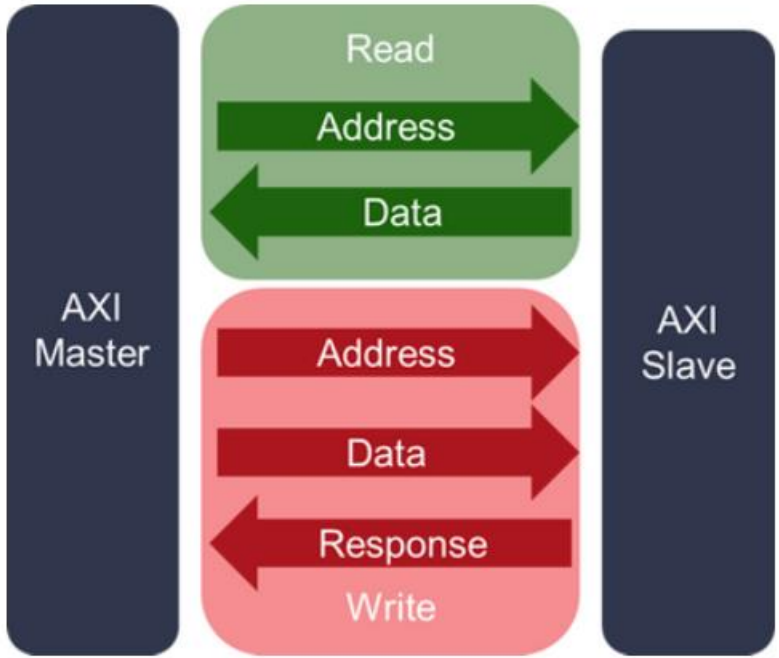
The AMBA AXI protocol supports high-performance, high-frequency system designs.

The AXI protocol:

- is suitable for high-bandwidth and low-latency designs
- provides high-frequency operation without using complex bridges
- meets the interface requirements of a wide range of components
- is suitable for memory controllers with high initial access latency
- provides flexibility in the implementation of interconnect architectures
- is backward-compatible with existing AHB and APB interfaces.

AXI4-Lite

AXI4-Lite is a basic AXI communication protocol. It is often used for simple, low-throughput memory-mapped communication (for example, to and from control and status registers).



Source https://support.xilinx.com/s/article/1137153?language=en_US

PCI Express

PCI Express is a high performance, general purpose I/O interconnect defined for a wide variety of future computing and communication platforms. Key PCI attributes, such as its usage model, load-store architecture, and software interfaces, are maintained, whereas its parallel bus implementation is replaced by a highly scalable, fully serial interface. PCI Express takes advantage of recent advances in point-to-point interconnects, Switch-based technology, and packetized protocol to deliver new levels of performance and features. Power Management, Quality Of Service (QoS), Hot-Plug/Hot-Swap support, Data Integrity, and Error Handling are among some of the advanced features supported by PCI Express

Source PCI Express® Base Specification Revision 4.0 Version 0.3

PCI Express link performance

Version	Introduced	Line code		Transfer rate per lane[i][ii]	Throughput[i][iii]				
					x1	x2	x4	x8	x16
1.0	2003	NRZ	8b/10b	2.5 GT/s	0.250 GB/s	0.500 GB/s	1.000 GB/s	2.000 GB/s	4.000 GB/s
2.0	2007			5.0 GT/s	0.500 GB/s	1.000 GB/s	2.000 GB/s	4.000 GB/s	8.000 GB/s
3.0	2010		128b/130b	8.0 GT/s	0.985 GB/s	1.969 GB/s	3.938 GB/s	7.877 GB/s	15.754 GB/s
4.0	2017			16.0 GT/s	1.969 GB/s	3.938 GB/s	7.877 GB/s	15.754 GB/s	31.508 GB/s
5.0	2019			32.0 GT/s	3.938 GB/s	7.877 GB/s	15.754 GB/s	31.508 GB/s	63.015 GB/s
6.0	2022	PAM-4 FEC	1b/1b 242B/256B FLIT	64.0 GT/s 32.0 GBd	7.563 GB/s	15.125 GB/s	30.250 GB/s	60.500 GB/s	121.000 GB/s
7.0	2025 (planned)			128.0 GT/s 64.0 GBd	15.125 GB/s	30.250 GB/s	60.500 GB/s	121.000 GB/s	242.000 GB/s

Notes

i. ^ [a] [b] In each direction (each lane is a dual simplex channel).

ii. ^ Transfer rate refers to the encoded serial bit rate; 2.5 GT/s means 2.5 Gbit/s serial data rate.

iii. ^ Throughput indicates the unencoded bandwidth (without 8b/10b, 128b/130b, or 242B/256B encoding overhead). The PCIe 1.0 transfer rate of 2.5 GT/s per lane means a 2.5 Gbit/s serial bit rate corresponding to a throughput of 2.0 Gbit/s or 250 MB/s prior to 8b/10b encoding.

Source https://en.wikipedia.org/wiki/PCI_Express

PCIE in embedded systems

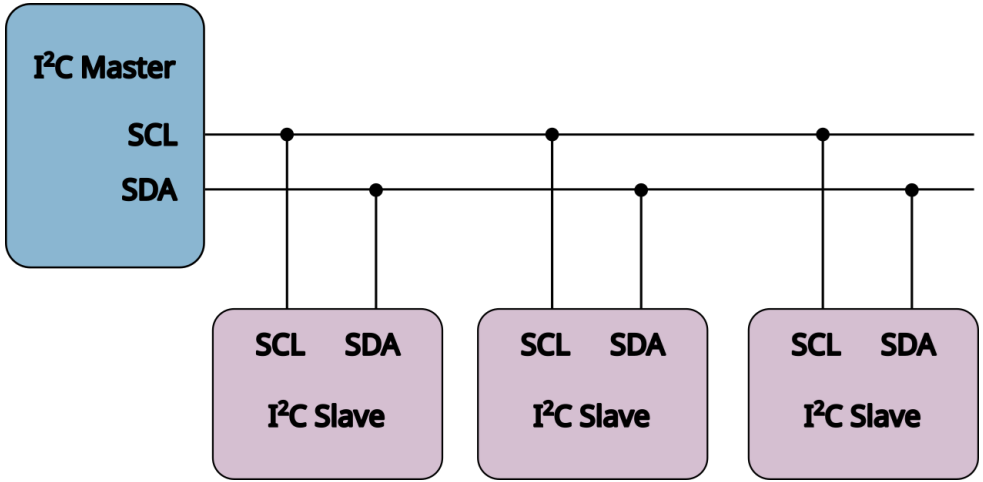
Not only PCs can use PCIE but also some embedded systems boards are equipped with PCIE. RPI 5 has PCIE 2.0 x1.

So they can use external graphics cards(https://www.youtube.com/watch?v=BLg-1w2QayU).

Source https://www.cnx-software.com/2024/02/23/52pi-p02-pcie-expansion-board-raspberry-pi-5-pcie-x1-slot/

Inter-Integrated Circuit

I²C is protocol allows connection of a wide variety of peripherals without the need for separate addressing or chip enable signals. Additionally, a microcontroller that includes an I2C interface is more successful in the marketplace due to the wide variety of existing peripheral devices available.



Source UM10204 I2C-bus specification and user manual

Inter-Integrated Circuit

Protocol features:

- two wires
- half duplex
- speeds up to 3.4Mbps
- max capacitance load 400 pF

Inter-Integrated Circuit

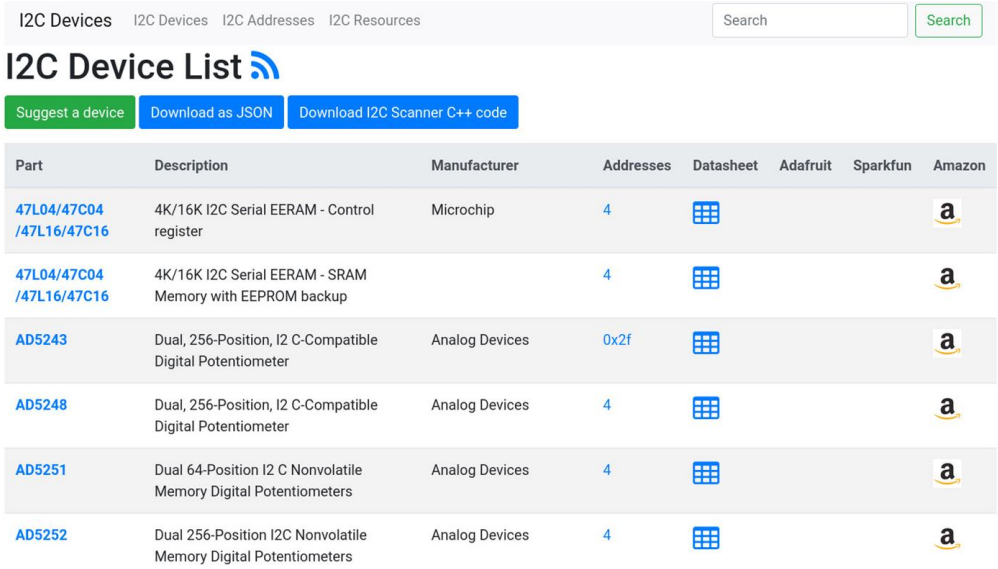
Usage:

- sesnors(temperature, humidity, pressure, etc.)
- memories(EEPROMs, RAMs, etc.)
- RTCs
- LCD controllers
- ADC/DAC
- etc.

Inter-Integrated Circuit

List of available devices can be found here https://i2cdevices.org/devices

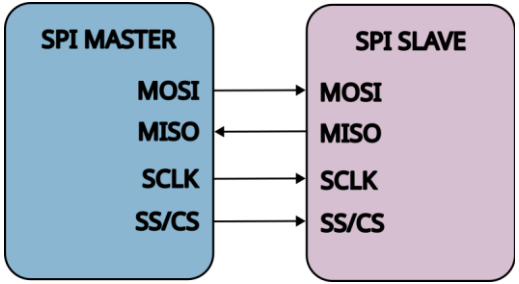
When you need at least few devices it is important to choose devices with different address.



Part	Description	Manufacturer	Addresses	Datasheet	Adafruit	Sparkfun	Amazon
47L04/47C04 /47L16/47C16	4K/16K I2C Serial EERAM - Control register	Microchip	4				
47L04/47C04 /47L16/47C16	4K/16K I2C Serial EERAM - SRAM Memory with EEPROM backup		4				
AD5243	Dual, 256-Position, I2 C-Compatible Digital Potentiometer	Analog Devices	0x2f				
AD5248	Dual, 256-Position, I2 C-Compatible Digital Potentiometer	Analog Devices	4				
AD5251	Dual 64-Position I2 C Nonvolatile Memory Digital Potentiometers	Analog Devices	4				
AD5252	Dual 256-Position I2C Nonvolatile Memory Digital Potentiometers	Analog Devices	4				

Serial Peripheral Interface (SPI)

SPI uses four wire, remember that devices need additional two wires being power supplies.



Serial Peripheral Interface (SPI)

Protocol features:

- four wires
- full duplex
- faster than Inter-Integrated Circuit

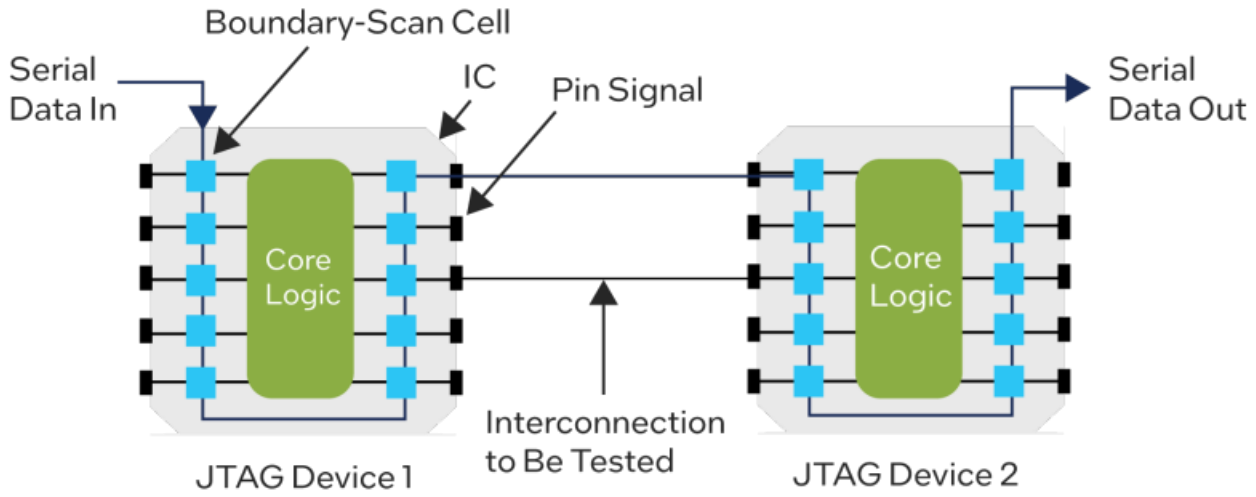
Serial Peripheral Interface (SPI)

Usage:

- sesnors(temperature, humidity, pressure, etc.)
- memories(EEPROMs, RAMs, etc.)
- RTCs
- LCD controllers
- ADC/DAC
- displays
- etc.

Joint Test Action Group (JTAG)

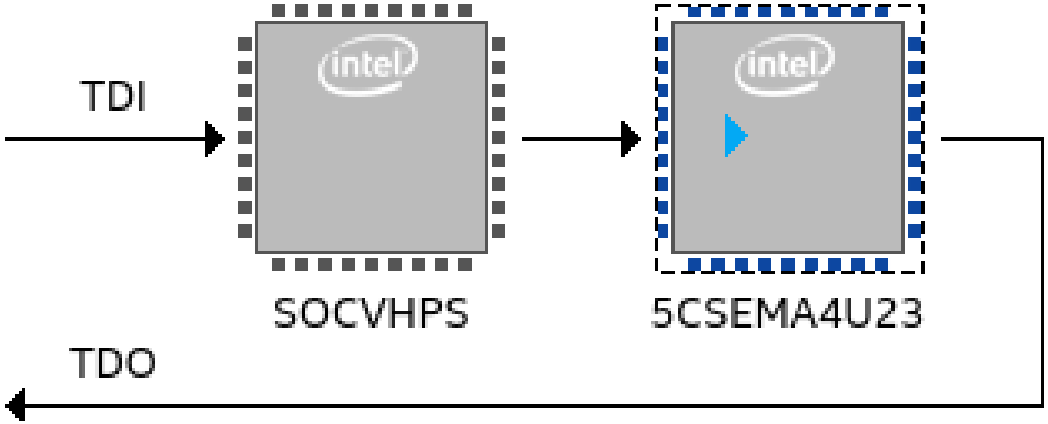
In the 1980s, the Joint Test Action Group (JTAG) developed a specification for boundary-scan testing that was later standardized as the IEEE Std. 1149.1 specification. This boundary-scan test (BST) architecture offers the capability to efficiently test components on PCBs with tight lead spacing.



Source https://cdrdv2-public.intel.com/654590/an039.pdf

JTAG programmer

JTAG programmers are often used to program and debug FPGAs. They also allow you to detect devices on the board. Apart from the FPGA system, they can detect e.g. FLASH memories.



JTAG / boundary scan used to testing

Boundary scan cells (see above) can operate in two modes. In their functional mode they have no effect on the operation of the device – this is the mode in which they operate when the board is running normally. In their test mode they disconnect the functional core of the device from the pins. By putting boundary scan cells into test mode they can be used to control the values being driven from an enabled device onto a net and also be used to monitor the value of that net.

Disconnecting the control of the pins from the functionality of the enabled device makes boundary scan test development significantly easier than traditional functional test as no device configuration or booting is required to use the pins. By providing a mechanism to control and monitor all the enabled signals on a device from a four-pin TAP, JTAG significantly reduces the physical access required to test a board.

Source https://www.xjtag.com/about-jtag/what-is-jtag/

Universal Serial Bus (USB)

Universal Serial Bus (USB) is an industry standard that allows data exchange and delivery of power between many types of electronics. It specifies its architecture, in particular its physical interface, and communication protocols for data transfer and power delivery to and from hosts, such as personal computers, to and from peripheral devices, e.g. displays, keyboards, and mass storage devices, and to and from intermediate hubs, which multiply the number of a host's ports.

Source https://en.wikipedia.org/wiki/USB

USB standards

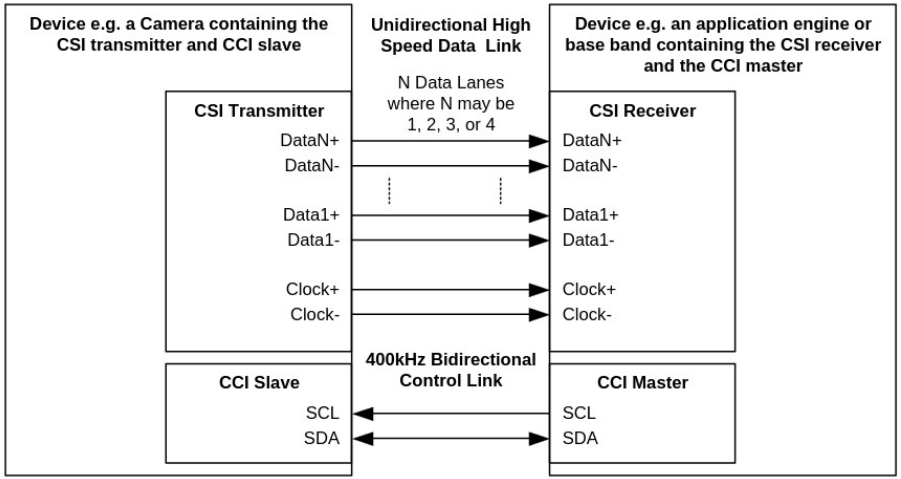
- USB 1.0(Low-speed) – standard from 1996, signaling rate 1.5Mbit/s
- USB 1.1(Full-speed) - standard from 1998, signaling rate 12Mbit/s
- USB 2.0(High-speed) - standard from 2001, signaling rate 480Mbit/s
- USB 3.0(SuperSpeed (Gen 1)) - standard from 2008, signaling rate 5Gbit/s
- USB 3.1(SuperSpeed+ (Gen 2)) - standard from 2013, signaling rate 10Gbit/s
- USB 3.2(SuperSpeed USB 20 Gbps (USB 3.2 Gen 2×2)) - standard from 2017, signaling rate 20Gbit/s
- USB4(USB4 40 Gbps (USB4 Gen 3×2)) - standard from 2019, signaling rate 40Gbit/s
- USB4 v2.0(USB4 80 Gbps (USB4 Gen 4)) - standard from 2022, signaling rate 80Gbit/s

Source https://en.wikipedia.org/wiki/USB

Camera Serial Interface (CSI)

The Camera Serial Interface 2 specification defines an interface between a peripheral device (camera) and a host processor (baseband, application engine).

CSI-2 provides the mobile industry a standard, robust, scalable, low-power, high-speed, cost-effective interface that supports a wide range of imaging solutions for mobile devices.



Sourece DRAFT MIPI Alliance Specification for Camera Serial Interface 2 (CSI-2)

MIPI CSI-2 Features part 1

Feature	Specification Version						
	v1.1	v1.2	v1.3	v2.0	v2.1	v3.0	v4.0
Basic image frame transmission using short frame synchronization packets, long data packets, at least one virtual channel, and a MIPI PHY physical layer	●	●	●	●	●	●	●
MIPI D-PHY version reference*/bit rate per lane	v1.1/1.5 Gbps	v1.2/2.5 Gbps	v1.2/2.5 Gbps	v2.1/4.5 Gbps	v2.1/4.5 Gbps	v2.5/4.5 Gbps	v3.0/9.0 Gbps
MIPI C-PHY version reference*/bit rate per lane	–	–	v1.0/3.9 Gbps	v1.2/8.0 Gbps	v1.2/8.0 Gbps	v2.0/13.7 Gbps	v2.1/13.7 Gbps
MIPI A-PHY version reference* (using PAL/CSI-2 v1.0)	–	–	–	–	–	–	v1.1/16 Gbps
Limit on number of data lanes per D- or C-PHY serial link	4	None	None	None	None	None	None
I2C Fast-mode Camera Control Interface (CCI)	●	●	●	●	●	●	●
CCI support for I2C Fast-mode Plus	–	–	–	–	●	●	●
CCI support for I3C SDR and HDR-DDR modes	–	–	–	–	●	●	●
Short Packet Data Types Synchronization: Frame Start and Frame End, Line Start and Line End; Generic User-Defined (8 data types)	●	●	●	●	●	●	●
Long Packet Data Types Pixels: YUV{422,420}; RGB{888,666,565,555,444}; RAW{6,7,8,10,12,14} User-Defined, Byte-Based Data (8 data types) Null, Blanking, and Embedded Data	●	●	●	●	●	●	●

Source https://www.mipi.org/specifications/csi-2

MIPI CSI-2 Features part 2

Additional RAW pixel data types	–	–	–	16, 20	16, 20	16, 20, 24	16, 20, 24, 28
Generic, user-defined, long packets (4 data types)	–	–	–	–	●	●	●
DPCM compression: 10-{8,7,6}-10 and 12-{8,7,6}-12	●	●	●	●	●	●	●
DPCM compression: 12-10-12	–	–	–	●	●	●	●
Virtual channel and data type long packet interleaving	●	●	●	●	●	●	●
Maximum number of virtual channels (using D-/C-PHY)	4/-	4/-	4/4	16/32	16/32	16/32	16/32
Latency Reduction and Transport Efficiency (LRTE)	–	–	–	●	●	●	●
LRTE with End-of-Transmission short packets (EoTp)	–	–	–	–	–	●	●
Long packet data scrambling	–	–	–	●	●	●	●
Smart Region of Interest (SROI)	–	–	–	–	–	●	●
Universal Serial Link (USL)	–	–	–	–	–	●	●
Always-On Sentinel Conduit (AOSC)	–	–	–	–	–	–	●
Multi-Pixel Compression (MPC)	–	–	–	–	–	–	●
Functional safety support (via CSE v1.0 specification)	–	–	–	–	–	–	●

Source https://www.mipi.org/specifications/csi-2