Implementing Flexible USB Type-C Control Using FPGA Technology

A Lattice Semiconductor White Paper

November 2014

Lattice Semiconductor
5555 Northeast Moore Ct.
Hillsboro, Oregon 97124 USA
Telephone: (503) 268-8000
www.latticesemi.com

Introducing USB Type-C

20 years after the first Universal Serial Bus (USB 1.0) helped bring interoperability to a fragmented electronics industry, the Type-C interface specification updates USB to meet the needs of 21st-century electronics and promises to once again change the way computers, consumer electronics and mobile devices connect and interact with each other. The slim, rugged and reversible Type-C connector extends the capabilities introduced by the USB 3.1 SuperSpeed+ specification by using two channels to provide a total bandwidth of 20Gbps, opening the door to reducing data transfer time for large files such as HD movies and 3D images by up to half. Thanks to USB Type-C's high-power capabilities it can supply up to 100W to allow fast battery charging and the provision of power to larger devices such as laptops, monitors and TVs. Type-C also introduces several other unique features, including new video modes which enable the USB connector and cable to carry video streams in formats such as DP, VGA and HDMI.

As noted Type-C interfaces bring dramatic benefits to consumers. However, in order to realize this potential designers must implement Type-C's Power Delivery (PD) protocol, detect cable orientation and switch high speed signals, as needed and, if desired, provide support for Vendor Defined Messaging (VDM). This whitepaper will show you how low-cost FPGA devices can be used in combination with off-the-shelf USB devices to develop products which take full advantage of the interface's speed, power and versatility, and how they can be brought to market in a rapid manner.

Type-C in a Nutshell

A New Robust Connector

The Type-C USB interface derives its name from the Type-C connector, selected by the USB Implementers Forum (USB-IF) as a sturdier, easier-to-use alternative to the notoriously-fragile Micro-B connector used in many of today's mobile devices. (Figure 1).

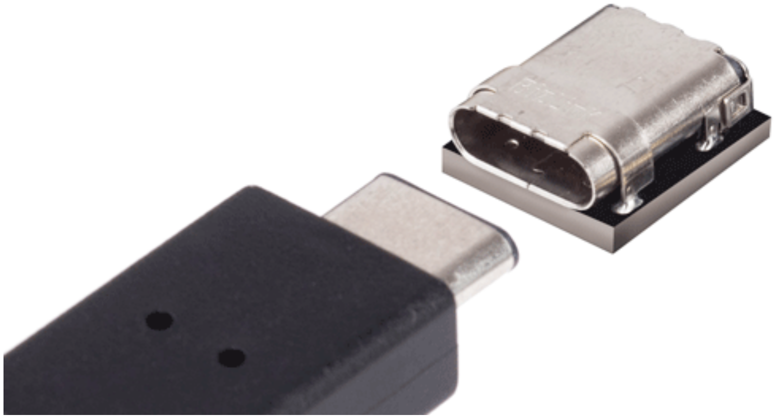


Figure 1: - The Type-C connector

The non-polarized 24-pin connector's mechanical design reflects the lessons learned from the Micro-B's checkered service history and is rated for 10,000 mate/de-mate cycles. In addition, users no longer need worry about "which end goes up" because the Type-C connector is non-polarized, allowing it to be mated in either orientation. And, unlike the majority of other USB cables, Type-C cables use the same male connector on both ends.

Increased Data Channels

The Type-C cable includes two sets of Tx/Rx signal paths which support the 10Gbps USB 3.1 SuperSpeed+ standard providing a total bandwidth of 20Gbps, providing the possibility of a 2x improvement in data transfers and download times. A separate set of pins is reserved for legacy applications requiring a USB 2.0 connection.

Enhanced Power Delivery

The cable also includes a Configuration Channel (CC) which handles the discovery, configuration and management of Type-C's advanced Power Delivery functions, capable of supplying a peripheral or mobile device with up to 100W.

More Flexible USB Topologies

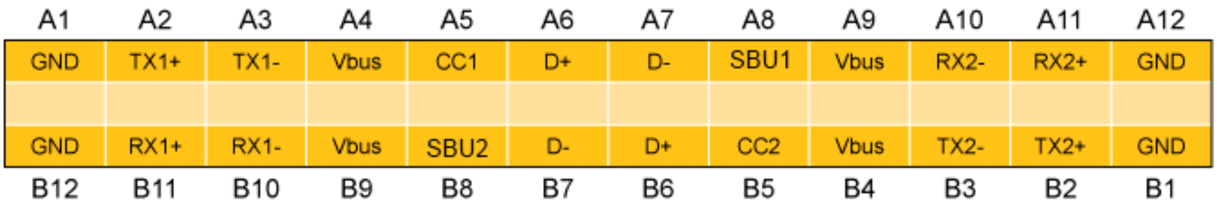
In addition to the "*Upstream-Facing Port (UFP)*" and "*Downstream-Facing Port (DFP)*" ports defined in recent versions of the USB standard the Type-C specification defines the *"Dual Role Port (DRP)*." This new term refers to a USB data port that can operate as either a DFP or a UFP. A DRP may be permanently configured as either a DFP or UFP, or may be changed dynamically between the two port states. *Note: this term should not be confused with the terminology in the USB Power Delivery specification where "dual-role port" refers to power roles*.

Vendor Defined Messages (VDM)

The standard also incorporates capabilities to allow extension of the connection for non USB applications. Structured VDMs allow the host to discover and configure alternate modes of connected devices that enable signal pairs to be reassigned to non-USB uses such as PCIe or DisplayPort. Unstructured VDMs allow vendors to pass vendor specific information and configuration control data.

Type-C Signals

The Type-C connector's 24 pins are arranged within a symmetrical shell which allows it to mate in either a "normal" or an "inverted" orientation. While convenient for the user, it does mean that only some of the connections it makes are "symmetric" i.e. orientation-independent while the others require some level of accommodation from the USB devices. (Figure 2).



A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
GND	TX1+	TX1-	Vbus	CC1	D+	D-	SBU1	Vbus	RX2-	RX2+	GND
GND	RX1+	RX1-	Vbus	SBU2	D-	D+	CC2	Vbus	TX2-	TX2+	GND
B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1

Figure 2: Pinout and primary signal assignments for the USB Type-C connector.

Type-C's symmetric connections include:

D+/D-: These pins provide a signal path for USB2 signals when a USB3 interface isn't available.

Vbus/GND: These pins can be used to deliver up to 100W of power to upstream-facing ports or, in some cases, to support peer-to-peer power transfers.

The remainder of the connections are "*asymmetric,*" i.e. they won't function properly when the connector is mated in its inverted configuration unless the ports make either an electrical or logical correction. Type-C's asymmetric connections include:

Tx1/2 Rx1/2: Provide for up to 2 channels of SuperSpeed+ data links providing a bandwidth of up to 20Gbps in both directions.

CC1/CC2: Configuration Channel signals used in the discovery, configuration and management of connections. Note that only one of these signals is used as a configuration channel. The other is repurposed as a power supply for the USB logic in the upstream facing port.

SBU1 & 2: The Side Band Use signals are intended to carry non-USB signals. They are used in Analog Audio mode and may be used by Alternate modes.

Implementation Challenges for Type-C Applications

At present, adding a Type-C USB interface to a new design requires developers to supplement the capabilities of the chips they intend to use in their systems because neither the PHY devices, MCUs nor Application Processors (APs) used in today's systems include the hardware needed to support several critical functions required to unlock the power of USB Type-C interfaces. These essential building blocks include Cable Detect (CD), SuperSpeed+ switch control, Power Delivery (PD) negotiation and Vendor-Defined Messaging (VDM).

Challenge #1: Providing Type-C Cable Detect (CD) and Power Delivery (PD) PHY functions. The Type-C interface adds a number of PHY layer concerns that must be implemented by most Type-C devices. The majority of Type-C devices need to implement a Cable Detect (CD) function in order to determine if they are connected to a DFP or UFP and the orientation of the cable. This mechanism relies on applying pull-up and pull-down resistors to the CC1 and CC2 lines. To implement CD a device must be able to sense a variety of voltage points associated with these pull-up and pull-down resistors. Any CD solution needs the ability to measure these analog voltages

If it is desired to take advantage of the Power Delivery (PD) communications to negotiate higher power operation, swap port roles or utilize VDMs then the PD PHY layer must also be implemented. The PD communication occurs over one of the CC lines and is specified in the USB Power Delivery specification. It is a half-duplex scheme that transmits 4b5b encoded data using Bi-phase Mark Coding (BMC) that simplifies receiver design. BMC can be viewed as a form of Manchester encoding. In addition the data uses Cyclic Redundancy Checking (CRC) algorithms to guard against data errors.

Although it is possible to implement the PD PHY using a general purpose microcontroller, a logic-based implementation may be preferable in situations where low power consumption is required.

Challenge #2: Implementing Power Delivery (PD) negotiation without a main system processor. If it is desired for the UFP to take advantage of the higher power possible with Type-C, it initiates the request for a power contract whereupon the DFP can either agree to the request or counteroffer with values it can supply. Once an agreement has been made a power delivery contract is in place. In many instances it may not be desirable to have a system processor that can perform these functions. First in some situations, such as smart chargers, there may not be a system processor. Second, it may be desired to negotiate a power contract when a device's battery is dead; to enable rapid charging for instance. Third, in some situations (such as sharing power between a laptop and mobile phone) it may be desirable to keep the main processor in sleep mode.

Although there are number of methods for providing this negotiation function, integrating this capability into one of the other chips already in use will allow for the smallest footprint and lowest cost implementation.

Challenge #3 Supporting Structured and Unstructured Vendor Defined Messages. As noted previously structured Vendor Defined Messages can be used to negotiate the use of a number of standardized alternate modes that extend the functionality of USB Type-C. Designers need to provide both capabilities for negotiating these modes as defined in the USB PD specification and for controlling the high-speed switches required to feed the appropriate signals onto the data pairs within the USB connector. Unstructured Vendor Defined Messages allow manufacturers to implement non standardized functions. This could include custom use of unused signal channels for functions such as GPIO aggregation between a dock and a dockable device or the transmission of data for challenge and response authentication mechanisms. Designers must implement the communication capability along with required processing, switch control and other hardware.

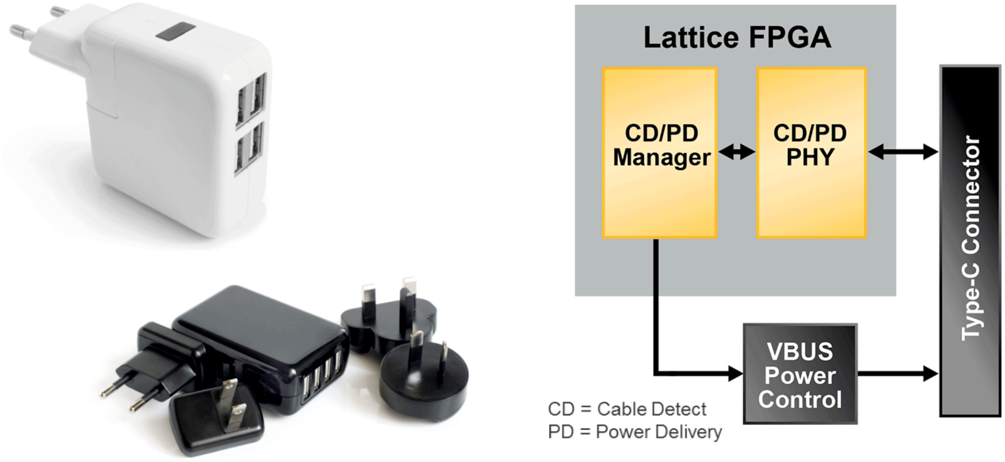
Again a variety of methods can be used to implement the negotiation and control mechanisms associated with VDM. However, the challenge is implementing these in an integrated manner to achieve the lowest cost and footprint.

Use Case Scenarios

This final section provides examples that indicate how FPGA technology can be used to immediately implement USB Type-C interfaces that make use of the compelling advantages that the standard provides. They do it in a manner that uses integration to drive small size, uses logic-based implementations to provide excellent power and preserves flexibility for implementing additional changes as needed.

Scenario 1: An FPGA-based PD for a Smart Charger

As Type-C USB interfaces make their first appearances in tablets, smart phones and other mobile devices, they will require chargers that can take advantage of PD negotiations to supply the advertised voltages and current levels to the devices they are charging and to negotiate a Power Contract to deliver the combination which most closely meet their requirements. Once the Power Contract is established, the PD must pass its voltage and current requirements to the Power Management Integrated Circuit (PMIC) within the charger to enable the provisioning of the negotiated current and voltage.



It is important to note that the design must also include Cable Detect (CD) to select the correct set of CC lines for the PD to communicate over. But since chargers and power supplies do not need access to USB's High-Speed or SuperSpeed data streams, there is no need to include the switch control logic these signal paths.

Figure 3: Using FPGAs to Implement CD/PD Functions for Chargers

Lattice FPGA technology is able to integrate the CD, PD PHY and PD contract negotiation functions into a single device housed within an easy to manufacture QFN package. Flexible IO technology is used to implement the required analog functions. A logic-based encoding, decoding and CRC section enables BMC communication at low power. The PD management functions are partitioned across logic and an embedded processor to provide the best combination of power consumption and low cost implementation. In addition unstructured VDMs may be passed to the embedded processor for the implementation of functions such as authentication if needed.

Scenario #2: An FPGA-Based "CD/PD-Lite" for a Mobile Device

The smart phones, tablets and other mobile devices which will use a Type-C interface as their primary I/O and power connection. The "CD/PD Lite" solution shown in Figure 4 provides CD and PD PHY functionality in a small 2.5x2.5mm package at cost levels appropriate for the highest volume applications.

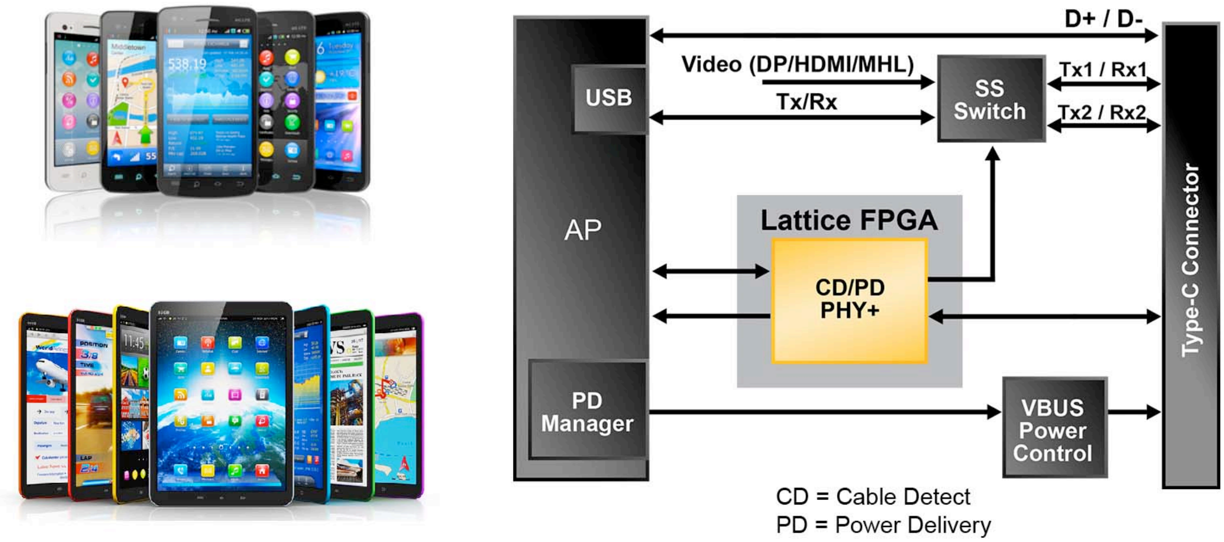


Figure 4 – CD/PD Lite for Mobile Devices

PD Negotiation functions are handled by the AP in order to achieve absolute lowest cost. Since mobile devices routinely exchange data over their USB ports, the CD function in this design provides control signals to the SS switch.

Scenario #3 - CD/PD for Device/Host

As noted previously, many products will want the ability to conduct PD and VDM negotiations without involving the system processors. The CD/PD solution described in

Figure 5 provides this capability. Optionally, when available, a processor can interrogate the PD function to determine status and change its settings from the loaded default values.

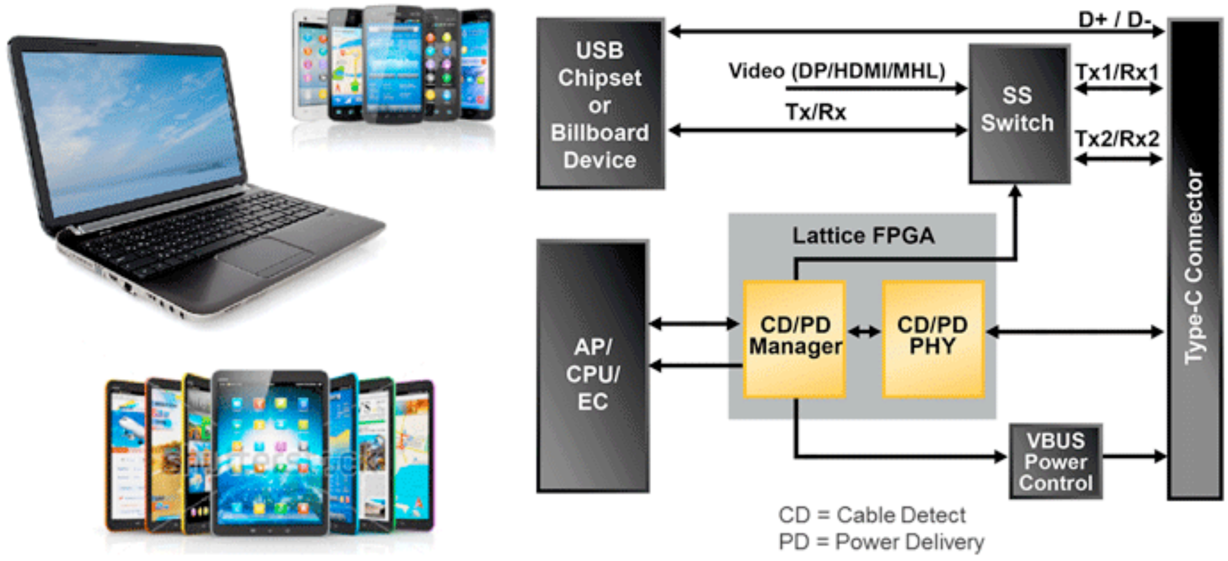


Figure 5 – CD/PD for Mobile Devices

This solution is available in an easy to manufacture QFN packages and a variety of small BGA packages and can operates as a DRP. The solution also provides switch control support for Alternate mode functionality. This allows the USB connector to be used to transport a variety of video standards such as DP, HDMI, MHL and VGA.

###