AN96527

Designing USB Type-C Products Using Cypress's CCG1 Controllers

Authors: John Hyde, Vihang Trivedi

Associated Project: No

Associated Part Family: CYPD11xx

Related Documents: For a complete list, see Related Documents

To get the latest version of this application note, or the associated project file, please visit http://www.cypress.com/go/AN96527.

AN96527 explains technological advances introduced by the USB-IF Type-C Specification and the Power Delivery Specification and how they can be designed into practical products using Cypress CCG1 Controllers. Four application areas are explored: host design, cable design, peripheral design, and power adapter design. Working prototype reference designs are presented for each application area, along with design and debug tools.

Contents

1 Introduction

In August 2014, the USB Implementers' Forum (USB-IF) introduced two separate but complementary specifications. The USB Power Delivery (PD) Specification Revision 2.0, Version 1.1 details how a USB link can be transformed from a 4.5-W power source (900 mA at 5 V) to a 100-W power source or consumer (up to 5 A at 20 V). The USB Type-C Cable and Connector Specification Revision 1.1 details a reversible, slim connector system with high-speed USB 2.0 signals and two SuperSpeed lanes at up to 10 Gbps that can be repurposed to support Alternate Modes. Power Delivery can be implemented independent of Type-C, and Type-C implementations need not include Power Delivery.

This application note explains a range of USB solutions that can be implemented with both Power Delivery and Type-C within the context of several reference designs: host, cable, client, power adapter, and docking station.

This application note first looks at Power Delivery (PD) and how it is architected and implemented. All USB products, at all speeds, can take advantage of Power Delivery as a power provider, power consumer, or sometimes both, including dynamically changing roles. Up to 100 W can be distributed on a Type-C cable, meaning that even a tablet or notebook can be bus-powered.

Most Type-C cables will implement all of the Type-C and Power Delivery features and will therefore need to be electronically marked, making a cable an "electronically marked cable assembly" (EMCA). This document covers an EMCA design implemented as a PCB that uses two Cypress CCG1 devices, one at each end of the cable. It also describes a practical EMCA design that can be built inside a Type-C plug.

The application note then looks at Alternate Modes, which allow the USB differential pairs defined by the Type-C Specification to be repurposed for other serial protocols such as DisplayPort, PCIe, Mobile High-Definition Link, and eSATA. This reduces the diversity of connectors on a USB product, because USB Type-C connectors can replace all previous USB connectors, power connector, and connectors for other uses.

Finally, this application note examines a docking station design that performs power aggregation and sharing among multiple Type-C (and a few Type-A) ports.

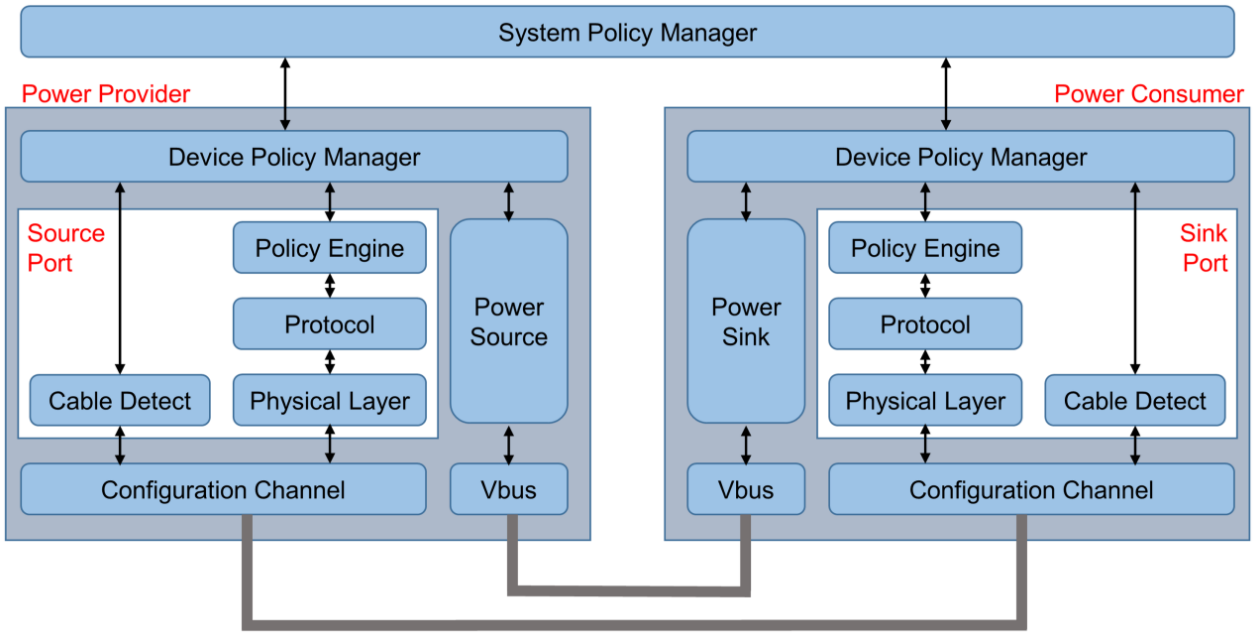
Note: This application note references the Cypress CCG1 (Type-**C** **C**ontroller **G**eneration **1**), a Cypress USB Type-C port controller compliant with the latest USB Type-C and PD standards. This device provides a programmable hardware and firmware solution with field upgradability to adapt to changes in Type-C spec. CCG1 devices integrate Type-C Transceivers. For power delivery applications, it integrates the Over Current Protection (OCP) and Over Voltage Protection (OVP) features. It provides a complete solution to add USB Type-C and PD support to power adapters, cables, notebooks, tablets, and monitors.

This application note also references the Cypress CY4501 CCG1 Development Kit and is intended as supplemental information to the CY4501 CCG1 Development Kit Guide. We recommend that you have the kit and this guide available to implement the example designs. Readers without the kit, however, will still learn a great deal from the USB Type-C theory sections and example designs.

2 Implementing Type-C and Power Delivery Systems

Before discussing specific applications, it is beneficial to review some essential theory. Figure 1 shows the USB Power Delivery architecture for a Type-C system. Derived from the PD Specification, Figure 1 shows only the biphase mark code (BMC) operation with the Type-C connector (this is described in the Configuration Channel: Message Passing section). Note that the PD spec includes support for the previous generation solution using the binary frequency shift keying (BFSK) method over Vbus on legacy (Type-A/B) connectors but this is not considered in this application note.

Figure 1. Type-C Power Delivery Architecture



Before the PD Specification, the host computer (a laptop, for example) was always the power provider and the peripheral device was always a power consumer. Charging the battery of the host computer was considered a separate action. The PD Specification greatly improves the USB ecosystem and defines power roles that are independent of "host" and "peripheral" and includes ability for swapping power roles. An example for such versatile behavior is that a powered peripheral such as a monitor can charge the battery of a host such as Notebook or PC. This will dramatically reduce the number of cables used in new USB systems.

The PD Specification defines a power provider as the device that provides power onto Vbus and a power consumer as a device that consumes power from Vbus. Here 'device' is 'host' or 'peripheral' and they take on the role of being a power provider or a power consumer. This role may be swapped under program control. Note that these new power roles are independent of the USB data transfers between a host and a peripheral.

The PD Specification defines a System Policy Manager that would be implemented on the USB Host running as an operating system task. This topic is beyond the scope of this application note; the reference designs discussed in this document operate with no involvement of the host operating system. For more details and insight to System Policy, please refer to the PD Specification.

When a USB Host and USB device are interconnected, they form a USB link pair, and each link partner has a configuration channel (CC) controller. Messages are then logically exchanged among Device Policy Managers within each controller. These messages are physically transferred over the CC and a power delivery (PD) contract is set up between the link pair, and then power is delivered over Vbus.

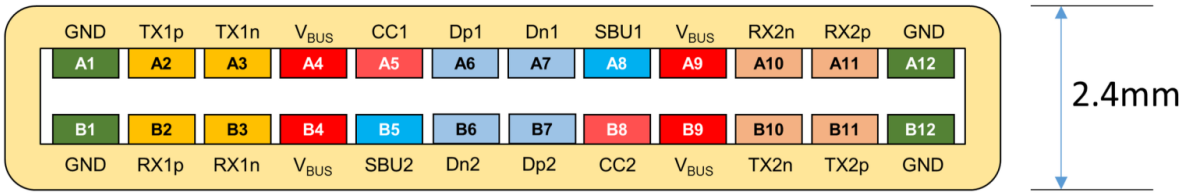
A CCG1 is used to implement the Power Delivery architecture within all of the presented reference designs. The policy protocol layer is implemented in firmware, and the reference designs include solutions for the USB Host (PC, Laptop, Notebook, Smartphone), EMCA (Cable), power adapter, and USB peripheral (Monitor, Hard Disk). The CCG1 device also implements the cable detect mechanism and the messaging physical layer.

The configuration channel is a new signal pair within the Type-C signal definition, so this is described next.

2.1 Type-C Signal Definition

Figure 2 shows the connections of a Type-C receptacle. For reference, the new Type-C signals are shown in bold text below the figure and signal names found in a USB Type-A receptacle are shown in normal text. The Type-C receptacle has two rows of 12 connections arranged symmetrically, both horizontally and vertically to facilitate plugging in a cable right-side up and right-side down. The USB IF did a great deal of simulation and modeling to derive this arrangement of distributed signals and grounds to enable a reliable signal transfer rate of 10 Gbps and beyond.

Figure 2. Type-C Signal Definition of a Receptacle



RX1n RX1p = Differential Pair 1
TX1n TX1p = Differential Pair 2
} SuperSpeed Lane1

RX2n RX2p = Differential Pair 3
TX2n TX2p = Differential Pair 4
} **SuperSpeed Lane2**

SBU1, 2 = Side Band Use (Alternate modes)
Dn, Dp = USB 2.0 High Speed Signals
Vbus = 5.0 **to 20 volts**
Vconn = 5.0 Volts for Cable Configuration IC
CCx = Configuration Channel

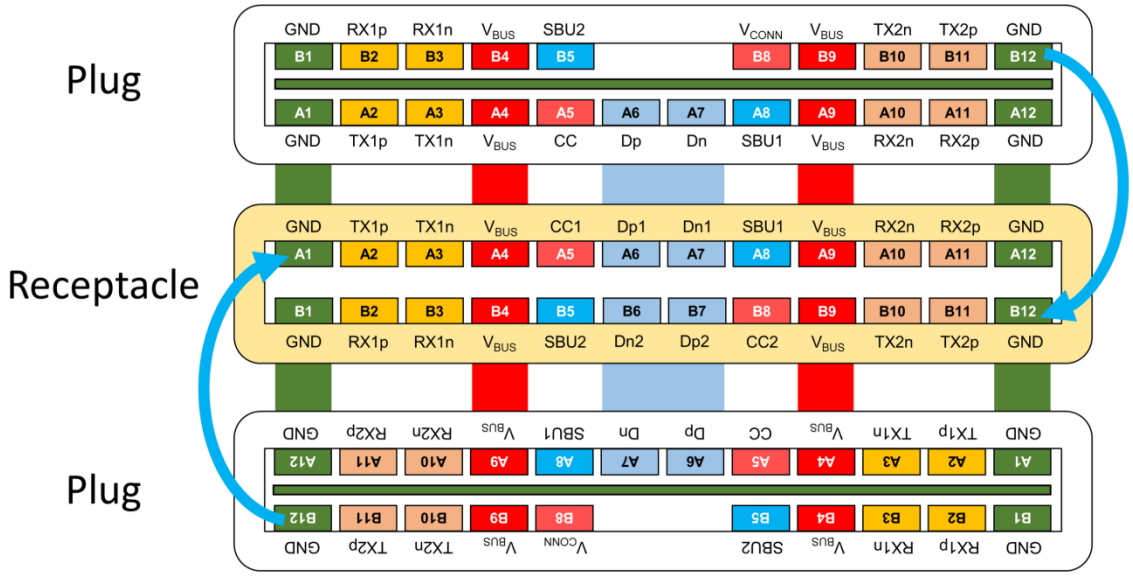
As shown in Figure 2, there are four ground signals and four Vbus signals. Their relative position remains the same when a plug is connected in any orientation.

There are two USB SuperSpeed lanes (four pairs of signals). One or both of these lanes can be repurposed for use in an Alternate Mode to transport other data for other serial protocol buses such as DisplayPort. Alternate Modes are described in the Type-C Alternate Modes section. When an Alternate Mode is active, two additional signals – SideBandUse (SBU1/2) – may also be utilized to carry the control signals.

A USB 2.0 pair of signals, Dp and Dn, is always present, which means that USB data transfers up to high speed (480Mb/s) will always be available even if both USB SuperSpeed lanes have been reassigned for Alternate modes. Two sets of USB 2.0 signals are defined (one in the top row and one in the bottom row) on a Type-C receptacle. It is expected that they are connected together in the PCB that hosts the receptacle. Dp1 should be shorted to Dp2 and Dn1 should be shorted to Dn2. This ensures that the USB 2.0 connection will be in place when a plug is connected in any orientation.

The configuration channel is defined using two signals, Vconn and CC on a plug and CC1 and CC2 on a receptacle, as shown in Figure 3.

Figure 3. Type-C Plug, Receptacle, and Flipped Plug



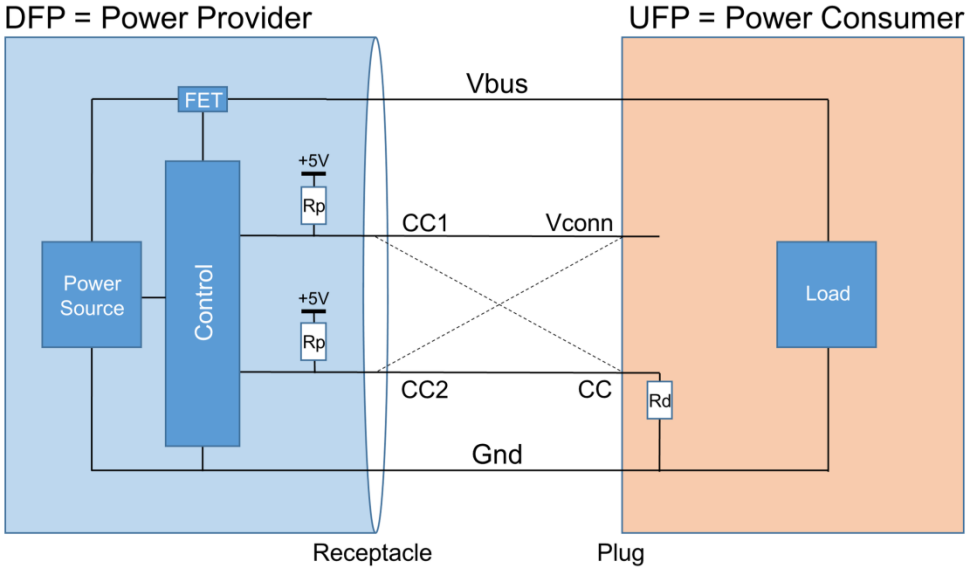
Because the plug can be inserted right-side up or right-side down, Vconn on the plug will be connected to either CC1 or CC2 on the receptacle, and CC on the plug will be connected to either CC2 or CC1 on the receptacle. The first task then is to determine the orientation of the cable.

This application note only discusses the power delivery aspect of the reference designs; it does not discuss anything about the USB data delivery over the Type-C cable. This is an important point to note because the configuration channel is implemented as a totally separate subsystem to the USB data delivery channels; it does not rely on any USB signaling.

The Type-C Specification introduces a variety of new terms and acronyms (see Appendix A: USB and Type-C Acronyms). These will be explained as they are introduced in this application note.

Figure 4 shows the power delivery signals of a downstream facing port (DFP), which would be a host or a hub and is typically a power provider, and a directly connected (no intervening cable) upstream facing port (UFP), which would be a peripheral device such as thumb drive and is typically a power consumer. Only power signals are shown. USB signals will be added later in the application note, and, as explained previously, their operation is independent of the power signals.

Figure 4. Direct Connection of a Power Provider and Power Consumer



2.2 Configuration Channel: Cable Detect

A typical cable connection detection goes through the following sequence:

- By Default, DFP (power provider) sets Vbus to 0 V.
- DFP (power provider) has two pull-up resistors (Rp) connected to CC1 and CC2.
- UFP (power consumer) is connected to Vbus, waiting for power.
- UFP (power consumer) has Vconn open.
- UFP (power consumer) has a pull-down resistor Rd connected to CC.
- On connection, Rd pulls CC1 or CC2 LOW depending on the orientation.
- DFP (power provider) determines whether CC1 or CC2 was pulled LOW and therefore knows the orientation of the cable.
- DFP (power provider) provides 5 V on Vbus.
- UFP (power consumer), on seeing 5 V on Vbus, responds to the enumeration of its USB data channel, if it has one. Most peripherals will have a USB data channel but the PD Specification permits a power adapter to be designed without a USB data channel.

The value of Rp determines the initial power limit of the DFP, as shown in Table 1. A higher current may be negotiated using USB-PD later as described in Power Provider Design. The CC voltage shown is a typical value because Rp and Rd are specified with ±20 percent tolerance components (additionally, the table assumes that Rp is connected to +5 V; the PD Specification actually defines this connection in terms of current flow through Rd and describes other implementation methods).

Table 1. Rp Determines Initial Vbus Current Limit

Rp	Rd	CC Voltage	VBUS Current Limit
56 KΩ	5.1 KΩ	0.42V	Default USB Current
22 KΩ	5.1 KΩ	0.94V	1.5 A @ 5V
10 KΩ	5.1 KΩ	1.67V	3.0 A @ 5V

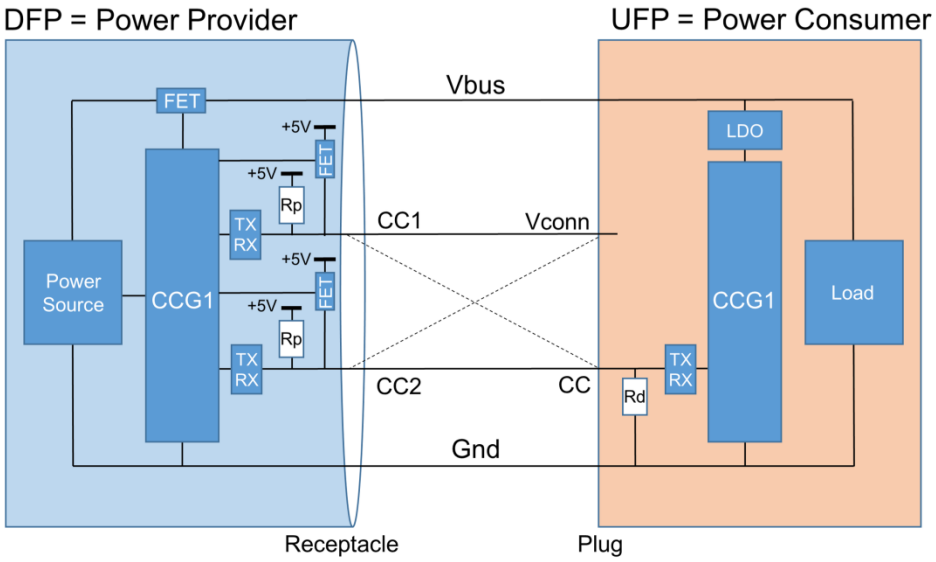
Other resistor values are also valid; refer to Appendix B: Accessories for a description of the accessories.

2.3 Configuration Channel: Messaging Physical Layer

If the UFP peripheral needs only up to 15 W of power (3 A at 5 V), then a PD controller is not needed. If the power provider has an Rp value of 10 kΩ as shown in Table 1, it can source up to 3 A of current. Note that this is twice as much as that provided by the Battery Charging Specification 1.2.

Most new applications will want more power, however. Figure 5 shows the same power source and power consumer of Figure 4 with a configuration channel controller (a CCG1 device) implemented within each link partner so that advanced features such as power delivery can be negotiated.

Figure 5. Type-C Configuration Channel implementation with CCG1

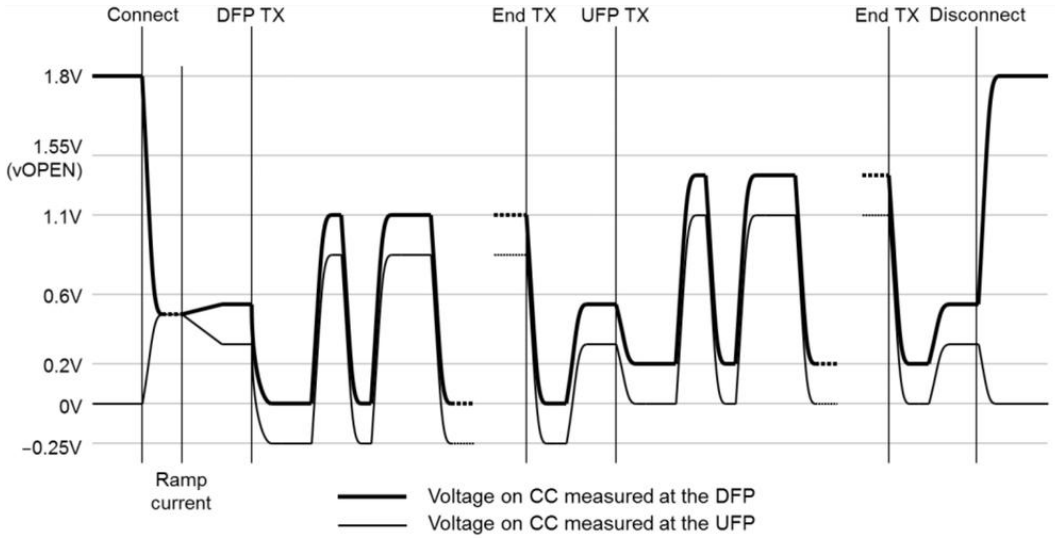


When connected, either CC1 or CC2 will be connected to the plug's CC line (UFP side). The CCG1 (DFP side) will disconnect Rp from the CCx pin, which is not connected to the CC line (UFP side) and instead connects a +5 V Vconn supply to it. This Vconn supply is not used by the example implementation shown in Figure 5. Refer to the Adding a Type-C Cable section to understand how Vconn supply is used.

In the implementation shown in Figure 5, UFP and DFP are directly connected (without a cable). UFP uses a plug, where Vconn and CC are fixed. Another example of a directly connected device is a mouse with a captive cable. It will also have Vconn and CC in a fixed position. Figure 5 highlights TX+RX transceivers that are used to implement the configuration channel; these are actually embedded within the CCG1.

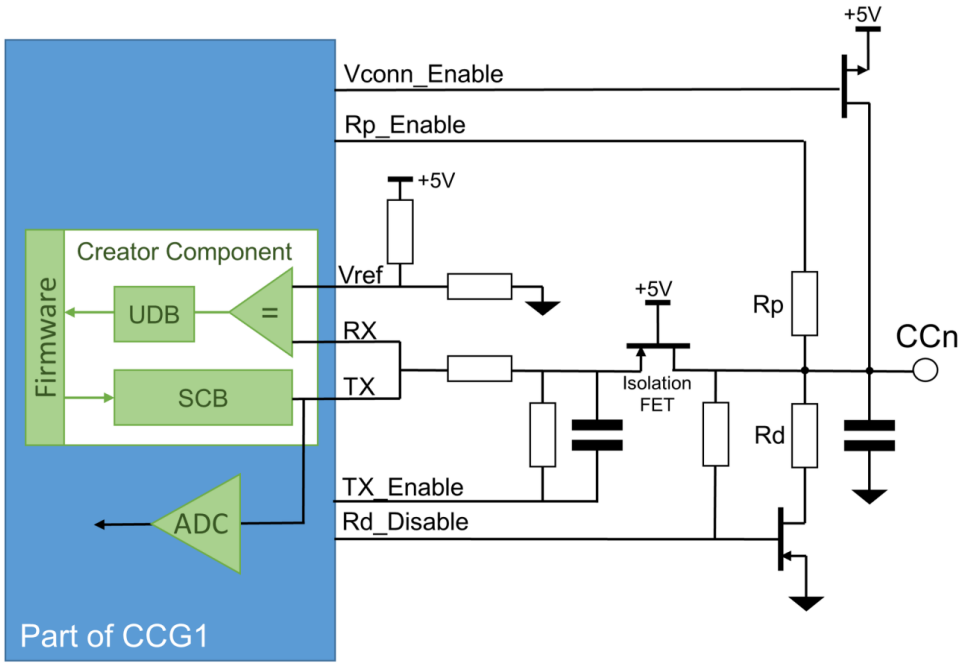
The configuration channel uses half-duplex biphase mark coding (BMC) to transmit and receive messages. BMC is often called "Differential Manchester encoding" and is an extension of Manchester encoding, wherein the output signal toggles on both rising clock edge and falling clock edge if the data bit is HIGH. If the data bit is LOW, the output transitions only every rising edge. Because there are transitions for both LOW and HIGH data bits, BMC is effectively DC-balanced, making it suitable for superimposing on the DC level set by Rp and Rd on the CC line as shown in Figure 6. More details can be found in the Type-C specification from the USB-IF website.

Figure 6. CC Messages (BMC Superimposed on the DC Level of the CC Wire)



CCG1 is based on PSoC® 4, and uses UDB, SCB, and analog blocks and firmware to implement the BMC encoding and decoding scheme. As shown in Figure 7, external resistors Rp and Rd are used to set the correct biasing values (connection detection). If you are not familiar with the PSoC 4 architecture and related terms, refer to the PSoC home page for a wealth of technical documentation, videos, and other training materials.

Figure 7. Detail of One of the Two Configuration Channel Physical Layers on a DRP



2.4 Dual-Role and Dead-Battery Provision

Many USB products (a laptop, for example) operate as a power provider and a power consumer at different times. The Type-C Specification calls this operation a dual-role port (DRP). (Note that this term has no relationship to the USB OTG ability to swap data host and data peripheral roles).

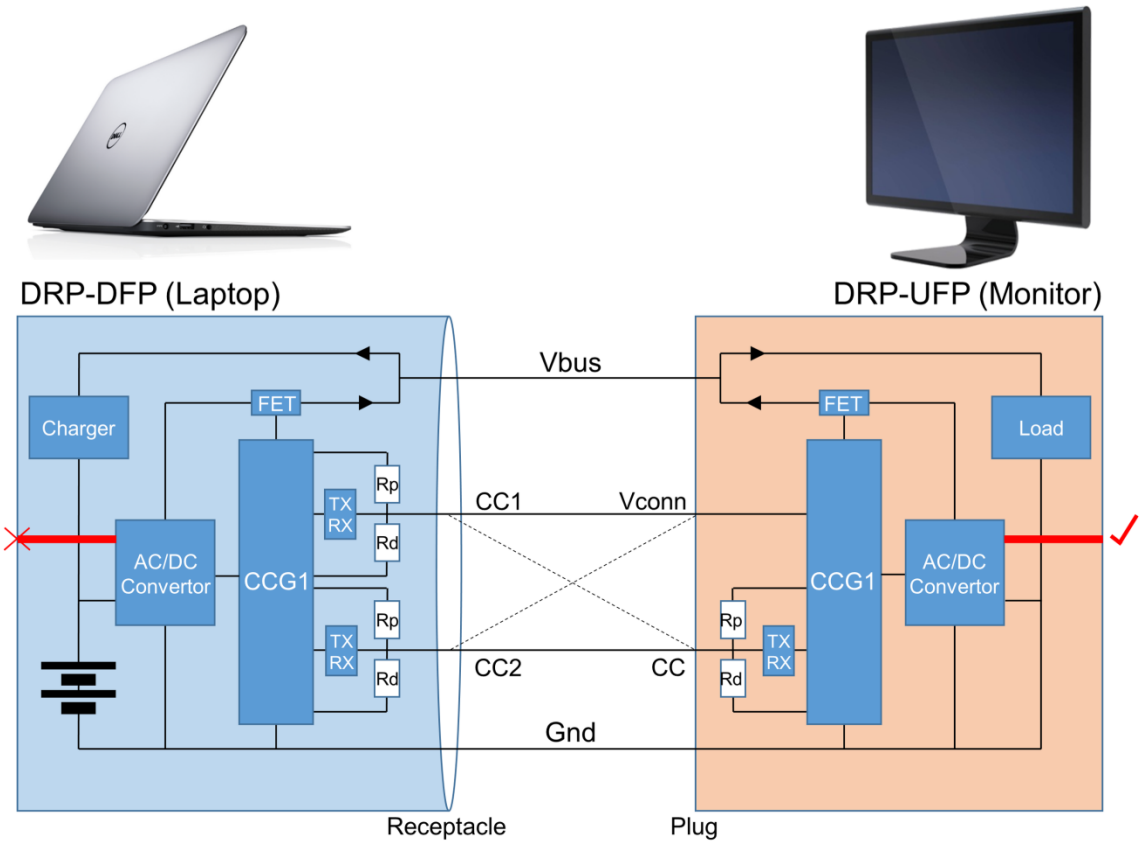
A laptop with DRP capabilities will be a power provider when running from its internal battery and a power consumer when being charged from the Type-C connector. DRP devices have the ability to connect and disconnect either Rp, Rd, or both to and from the CC lines. Upon detecting a Type-C connection, DRP devices will toggle their state between trying to be a DFP (by connecting Rp and disconnecting Rd) and trying to be an UFP (by disconnecting Rp and connecting Rd). When a DRP device (like a laptop) is connected to an UFP device like a mouse, after toggling to the DFP state, the laptop will detect a UFP attached and will settle its power state as a DFP. Then the laptop will provide power to the mouse.

On the other hand, when a DRP device (like a laptop) is connected to a DFP device like a power adapter, after toggling to the UFP state, the laptop will detect a DFP attached and will settle its power state as a UFP. Then the power adapter will provide power to the laptop.

A DFP will have Rp connected to a local +5 V supply, and a UFP will have Rd connected to ground. Consider what happens in Figure 7 if the DRP does not inject power into the system; that is, its local +5-V supply is, in fact, floating. An example is a laptop with a dead battery. A requirement for a power provider is that it should be a DRP and present Rd to ground on both CC lines when it is not providing power. This will make the DRP look like a UFP.

Figure 8 extends Figure 7, with both sides augmented to be DRPs. In this example, consider that a laptop is connected to a powered monitor. The laptop battery is dead, so Rp is floating and Rd is present.

Figure 8. Two Interconnecting DRPs Example – Laptop and Powered Monitor



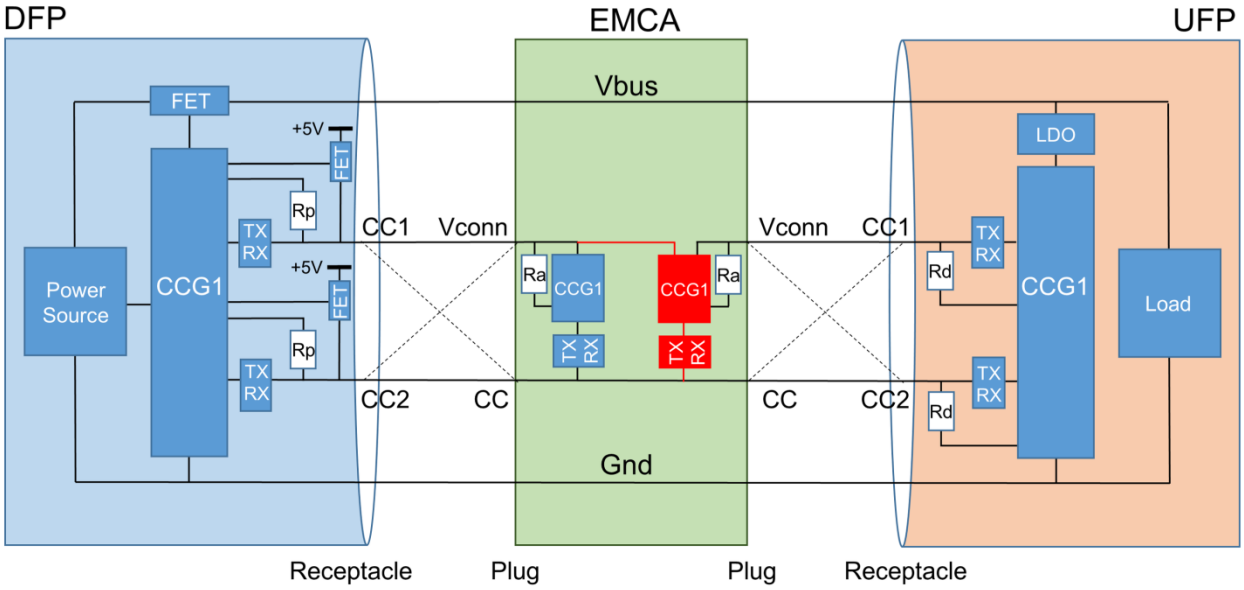
The DRP-UFP (powered monitor in this example) will initially present Rd on its CC line, and the DRP-DFP will not be recognized because the laptop battery is dead and the CC line is not pulled up. The DRP-UFP periodically swaps roles to detect a connection. The DRP-UFP's CCG1 will disconnect Rd and then connect Rp; this CCG1 will then detect the voltage on the CC line and apply 5 V to Vbus with the current rating as defined in Table 1. With the DRP-DFP CCG1 in the laptop now powered, the two CCG1 devices will negotiate a higher Vbus so that the laptop battery can be charged more quickly, as described in Power Provider Design.

2.5 Adding a Type-C Cable

The example in Figure 5 used a directly connected UFP, making the explanation of the configuration channel simpler. Figure 9 shows the more common implementation of interconnecting a DFP and a UFP using an EMCA.

Two types of cables are defined in the USB specifications: a full-featured cable that supports all the Type-C signals shown in Figure 2, and a USB 2.0 cable that does not implement the SuperSpeed or sideband signals. The USB 2.0 Type-C cable, which has fewer wires, would be cheaper and more flexible.

Figure 9. Connecting a DFP and UPF With an EMCA

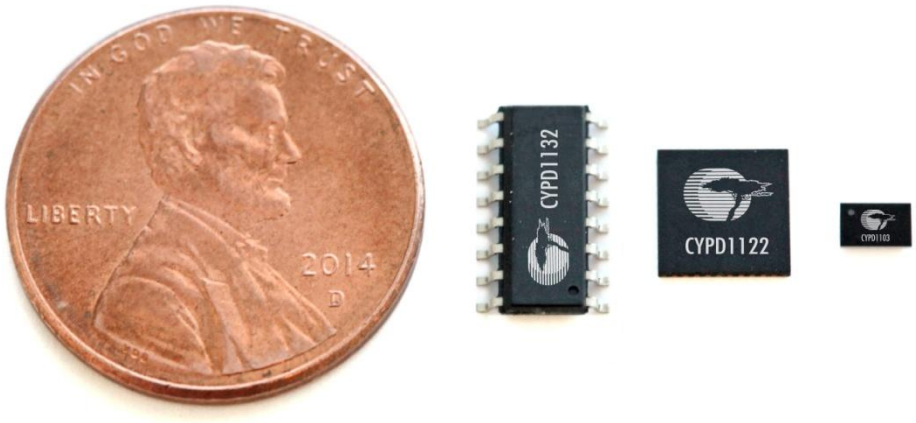


There are two plugs in the EMCA solution shown in Figure 9, which provide four possible ways of interconnecting the DFP and the UFP. This implementation is slightly more complex compared to the direct connection shown in Figure 5. Now the UFP also has to take care of the cable twist. The Vconn signal is terminated at one end of the cable electronics and is not connected to other end of the cable. This Vconn isolation allows the simple Rp and Rd cable detect mechanism on the CC line to continue to operate.

Vconn in the cable plug has a resistor, Ra, connected to ground. This enables the DFP to determine that it is connected to a cable rather than a directly connected device. Therefore, the DFP will supply +5 V on Vconn to power the cable electronics. It is recommended that the DFP disconnects Rp and the EMCA disconnects Ra once the PD negotiation is complete to conserve power. The EMCA reference design described in the CY4501 CCG1 Development Kit Guide allows three different implementations of an EMCA to be evaluated. Please refer to the Design Files on the CY4501 CCG1 Development Kit webpage.

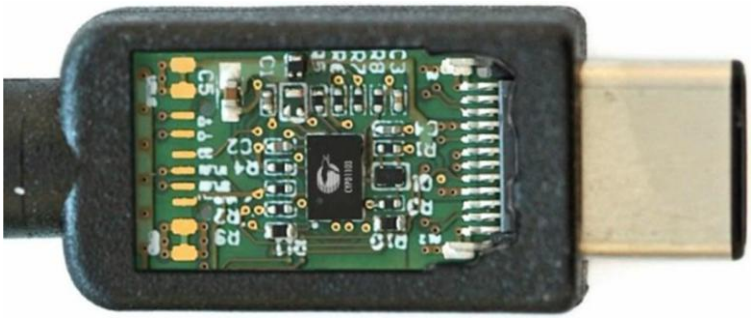
An EMCA needs a configuration channel controller (a CCG1) mounted in the cable. The CCG1 device is available in three packages, as shown in Figure 10, one small enough to fit into the plug of an EMCA cable. Therefore, whether you are building hosts, clients, EMCAs, or power adapters, you can choose the one best suited to your design project.

Figure 10. CCG1 Package Types



The Figure 11 shows the image of a CCG1 cable paddle card. The CCG1 device is powered by Vconn. Vconn is supplied from either or both the ends of the cable (and can be swapped), and the cable vendor has the option of using a single CCG1 or two CCG1 devices, one at each end of the cable. Figure 9 shows the second CCG1 device in red.

Figure 11. Type-C EMCA Plug with CCG1

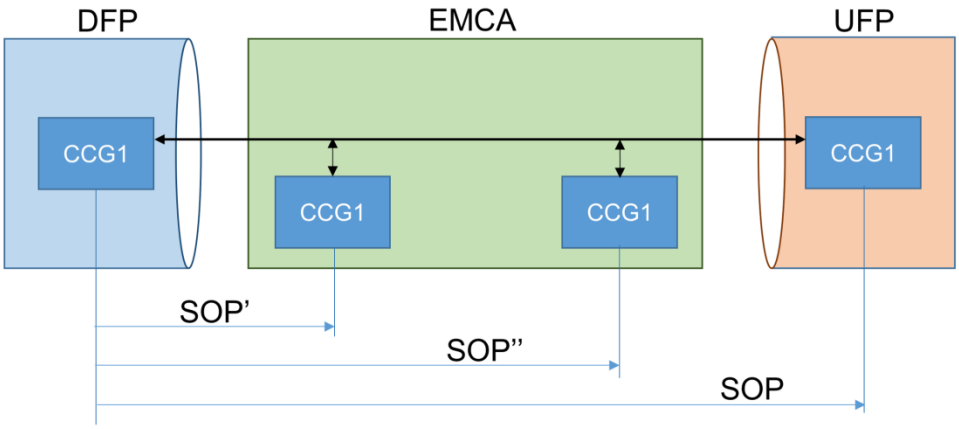


2.6 Configuration Channel: Message Passing

Power Delivery parameters are exchanged between CCG1 devices by passing messages. Up to four CCG1 devices are connected to the CC wire, and the Type-C scheme to uniquely address a destination CCG1 device. The addressing scheme is shown in Figure 12.

Assuming that the DFP is the message source, then the CCG1 device closest to the DFP is called SOP' (SOP prime). The CCG1 device at the other end of the cable, if it exists, is called SOP'' (SOP double prime), and the CCG1 device in the port partner UFP is called SOP. The Type-C Specification also uses SOP* to refer to SOP, SOP' or SOP''. When the UFP is the message source, then the CCG1 device in the cable closest to it is SOP', and the far-end CCG1 device is SOP''. The CCG1 device in the link partner is always SOP.

Figure 12. DFP and UFP connection with EMCA (using 4 CCG1)

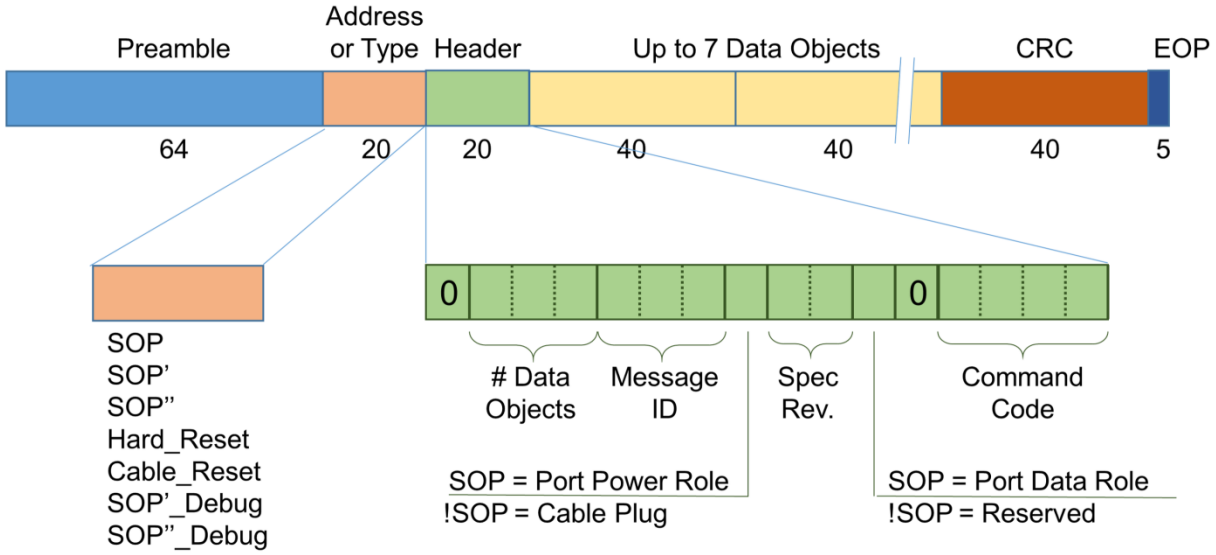


You do not need to know all the details of the messaging because the CCG1 device implements the Power Delivery (and Alternate Modes) messages in firmware, and a high-level Host Processor Interface (HPI) is presented to users who would like to add messages at run time. The reference designs implement the feature set defined by the PD Specification (and the VESA DisplayPort on Type-C Specification). Should the USB specifications change, then a new firmware will be updated and released.

Messages are sent half-duplex and in the format as shown in Figure 13. Also included is a first-level decoder ring that explains some of the fields. The preamble is 64 bits of alternating 0s and 1s, which allow the receiver to sync with the actual data rate of the transmitter (300 ±30 kHz). All other data is transmitted with 4b5b encoding to improve reliability.

The CCG1 device decodes and extracts the data bytes to send to the protocol layer (refer to Figure 1). A 16-bit address/type field (which the PD Specification calls "SOP") is encoded as 20 bits as a result of the 4b5b encoding. This is always followed by a 16-bit header field, also encoded as 20 bits. The header field includes a data object count; if it is 0, then the message type is "control." If it is 1 through 7, then up to seven 32-bit data objects follow, and the message type is called "data." A 40-bit CRC is transmitted; a 5-bit end of packet (EOP) token completes the message. If the calculated CRC is the same as the received CRC, then the CCG1 device passes the decoded message up to its protocol layer, where it is processed (refer to Figure 1).

Figure 13. CC Message Format

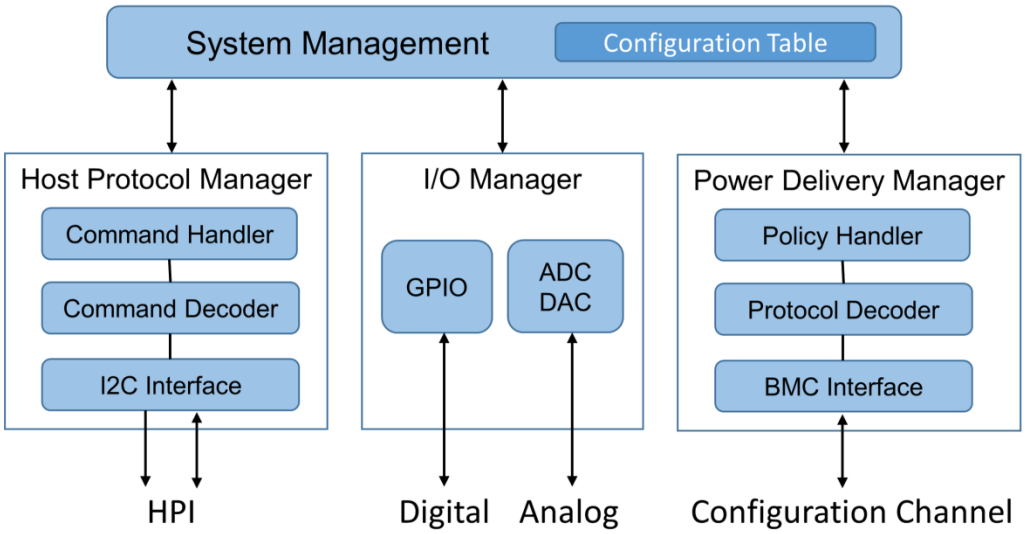


Code	Control Message	Data Message
0000	Reserved	Reserved
0001	GoodCRC	SourceCapabilities
0010	GotoMin	Request
0011	Accept	BIST
0100	Reject	SinkCapabilities
0101	Ping	Reserved
0110	PS_RDY	Reserved
0111	Get_Source_Cap	Reserved
1000	Get_Sink_Cap	Reserved
1001	DR_Swap	Reserved
1010	PR_Swap	Reserved
1011	Vconn_Swap	Reserved
1100	Wait	Reserved
1101	SoftReset	Reserved
1110	Reserved	Reserved
1111	Reserved	VendorDefined

The names of most of the messages are self-explanatory. The CCG1 device also implements the built-in self-test (BIST) message format designed to characterize the BMC data transfer channel. The USB PD specification also includes eye diagrams, which can be used to test compliance.

A block diagram of the architecture of a CCG1 device is shown in Figure 14.

Figure 14. CCG1 Architecture Block Diagram



The CCG1 device monitors digital and analog I/O signals to create configuration channel (CC) messages. The main processor of the target host or client sends and receives the CC messages through the Host Processor Interface (HPI) included in host and client designs. The CCG1 firmware can be configured or updated over either I^2C or the CC channel (depending on the type of application) using a Cypress-supplied flash utility (described in KBA96466).

The HPI interface is a 400-kHz I^2C slave interface with an interrupt line implementing a defined command/response protocol to change configurations, monitor statuses, update firmware, or transparently interact with other CCGx devices using unstructured vendor-defined messages (VDMs).

3 Application Example Overview

Previous sections have covered enough theory to work through a Power Delivery example. The USB data paths are explained in the Adding USB Data Paths section. Figure 15 shows a before-and-after overview of what will be achieved with the first few examples. At the top of the figure is a "today" host computer with three cables attached to it: power, USB, and display. They are connected to various devices. Cypress provides the CY4501 CCG1 Development Kit that can "instantly add USB Type-C" to a today system to demonstrate the features of "tomorrow's" Type-C system. The lower half of the figure shows the same equipment, but now they are connected with a host board, an EMCA board, and a client board. The result is a single Type-C cable serving all three functions of power, USB data and display.

The CY4501 CCG1 Development Kit board designs are comprehensive, containing more features than are needed for each example that will be discussed. This enables you to select features that you need from each design. In the future, it is expected that the host design will be absorbed into the host and the client design will probably be absorbed into a display.

A Chromebox desktop PC is used as an example host, but any SuperSpeed-aware tablet or laptop can also be used.

Figure 15. Goal of the Reference Designs

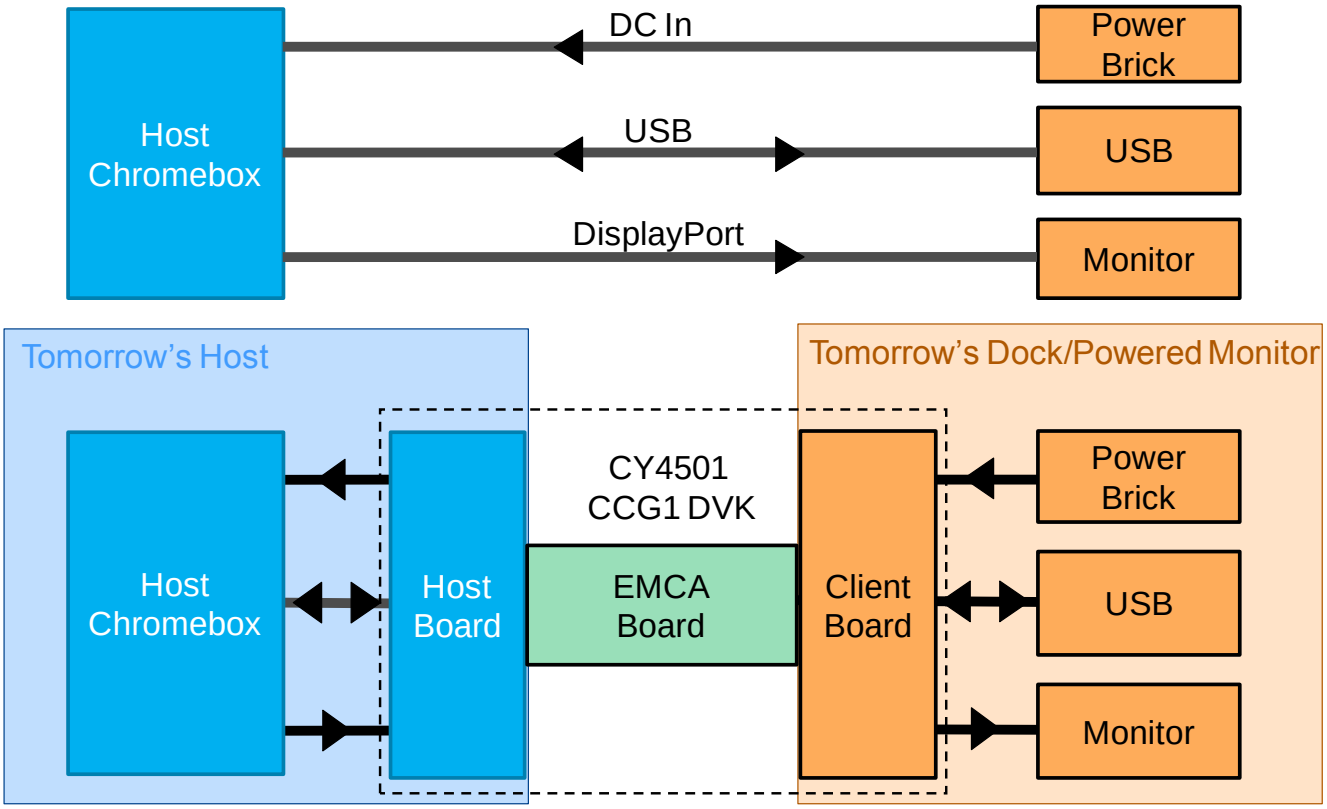


Figure 16 shows a photo of the final system setup including the three CY4501 CCG1 Development Kit boards. A full description of each board is included in the CY4501 CCG1 Development Kit User Guide. The same boards will be used in all the examples in this application note, but not every feature of each board will be used in every example.

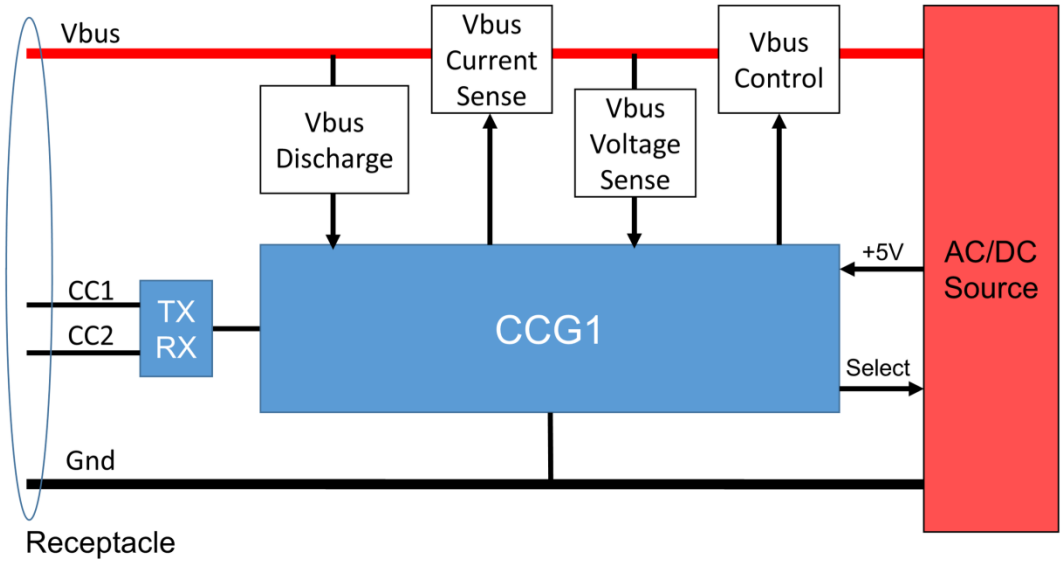
Figure 16. Chromebox PC and Three CY4501 CCG1 Development Kit Boards with Debug Laptop

3.1 Power Provider Design

Many considerations are involved in how 100 W of power, 20 V at 5 A, can be safely delivered via a Type-C connector. Figure 17 shows additional circuit elements that must be considered. The CCG1 device includes capabilities to manage this complete power delivery path.

In this example, the client board will operate as a power adapter. It expects a 24-V input that it will regulate down to 20 V, 13 V, 12 V, or 5 V at up to 5 A as commanded by the CCG1 device. The schematics in the CY4501 CCG1 Development Kit User Guide show details of the connections and part numbers used in the reference design.

Figure 17. Safe Delivery of 100 W Must Include Protection Circuits



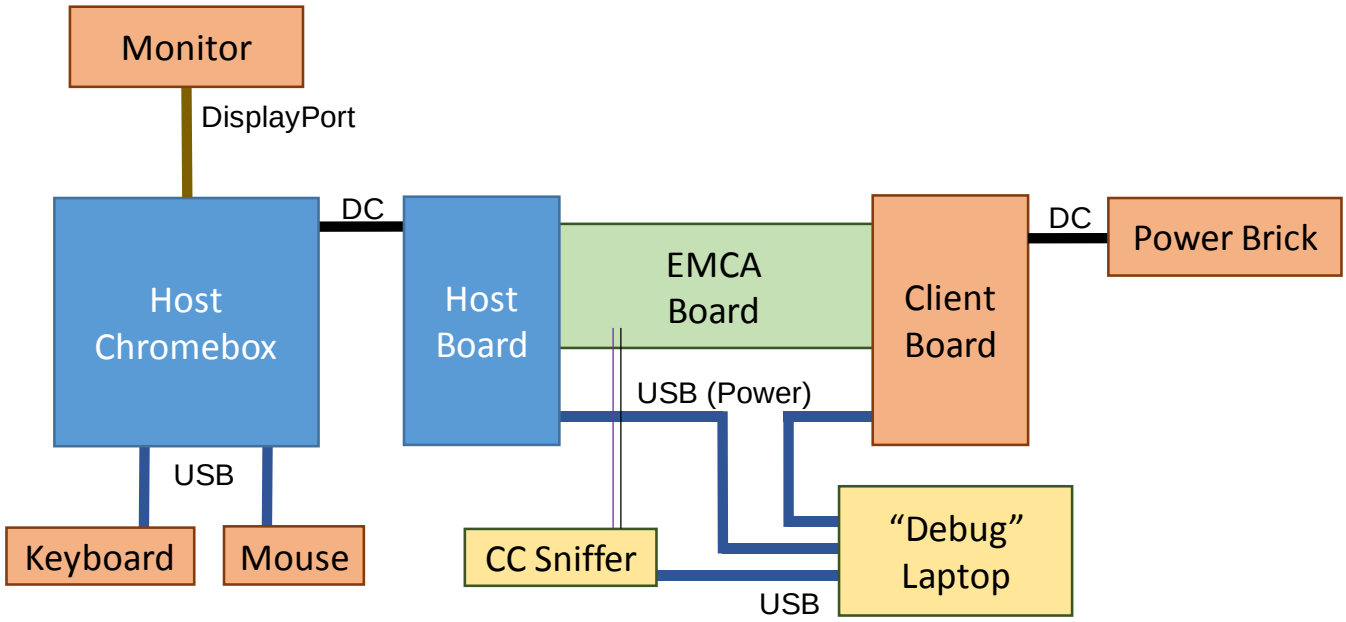
At power on, the CCG1 device waits for a connection, as described in the Configuration Channel: Cable Detect section. On initial connection, the CCG1 device selects Vbus at 5 V from the DC/DC converter and sets its overvoltage detection circuit to +5 V + 20 percent and its overcurrent detection to 3 A + 50 percent (assuming Rp is 10 kΩ). The Vbus discharge block is required by the USB PD specification to safely discharge the power supply when the connection is terminated, thus resetting the Vbus voltage on the Type-C connector to 0 V.

Set up the jumper settings on the CY4501 CCG1 Development Kit boards as described in the CY4501 CCG1 Development Kit User Guide and then connect them to the host and a power brick, as shown in Figure 18. Note that two USB cables are connected to the mini-B connector of the client board and mini-B connector of the host board. They are only used to supply +5 V to each of the boards.

Also connected to the "debug" laptop is the CC sniffer. The sniffer has a single CC probe wire and a ground wire. The easiest place to attach it is on the EMCA board; TP8 = CC and pin 18 on J9 and J10 = ground (you will probably solder a stake pin onto TP8 so that the sniffer is pluggable).

This example uses a spare PC power brick with the DC cable cut to implement Figure 18. These connections are labeled "DC." The power brick provides power to the client board, and the host board provides power to the Chromebox PC.

Figure 18. Hardware Setup for Power Delivery Example



On power up, both the host board and the client board have the default state as DRPs and will periodically switch roles until they decide between themselves as to which is the DFP and which is the UFP. Upon connecting the UFP and DFP with each other, the host board would come up as the DFP. The reason is that the PD Specification allows you to set a bias for a DRP (or a UFP). The host board firmware implements the Try.DFP portion of the Power Delivery state machine, and this bias causes the host board to always be a DFP. Figure 19 shows a partial state machine of two interconnected DRPs (extracted from the Type-C Specification, section 4.5.2).

Figure 19. Two DRPs Resolving DFP and UFP Roles

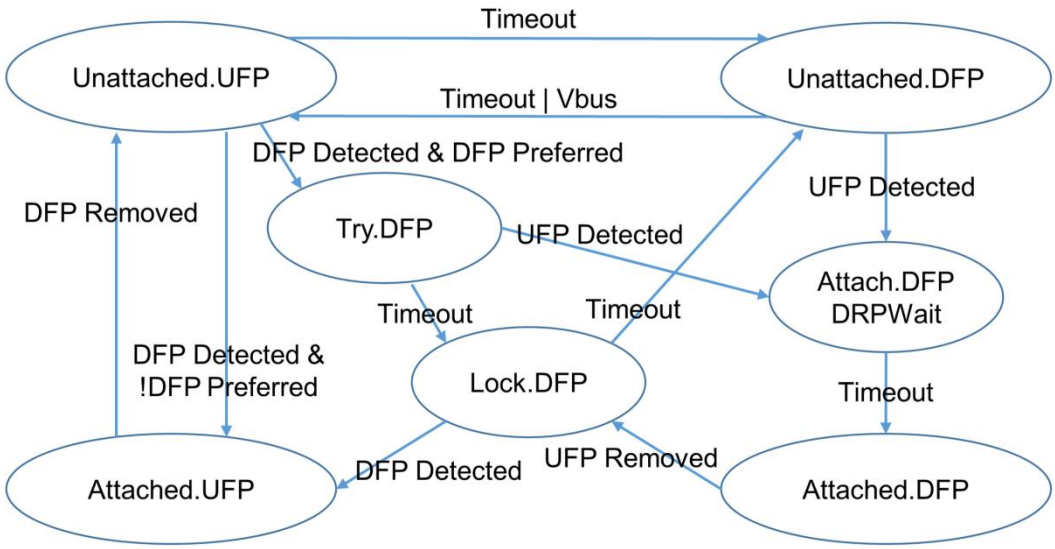


Figure 20 shows the message flow during initial connection. The DRP discovers that an EMCA is attached due to the presence of Ra, so it discovers the characteristics of the cable via the DiscoverACK response message. Note that each command and data message is acknowledged by the receiver, which sends a GoodCRC response back to the initiator. The initiator starts a timer for each message it sends, and if a GoodCRC message is not received within a timeout (the PD Specification lists all of them), then an error recovery is started. The message could be lost due to a failed CRC, for example, and it is the responsibility of the initiator to take corrective action. Only the first GoodCRC message is labeled in the figure. All others are colored purple.

The DFP detects the presence of a UFP due to the Rp resistor, so it sends it a capabilities message. Within this message is the ability to source 5 V at 900 mA, and the UFP makes a request for 500 mA of it. The DFP accepts and then turns on Vbus. The UFP, on detecting Vbus, could start enumeration if it had USB data channels. The comments in the figure explain the flow, and the vertical red line is Vbus.

Figure 20. Flow of CC Messages at Power On

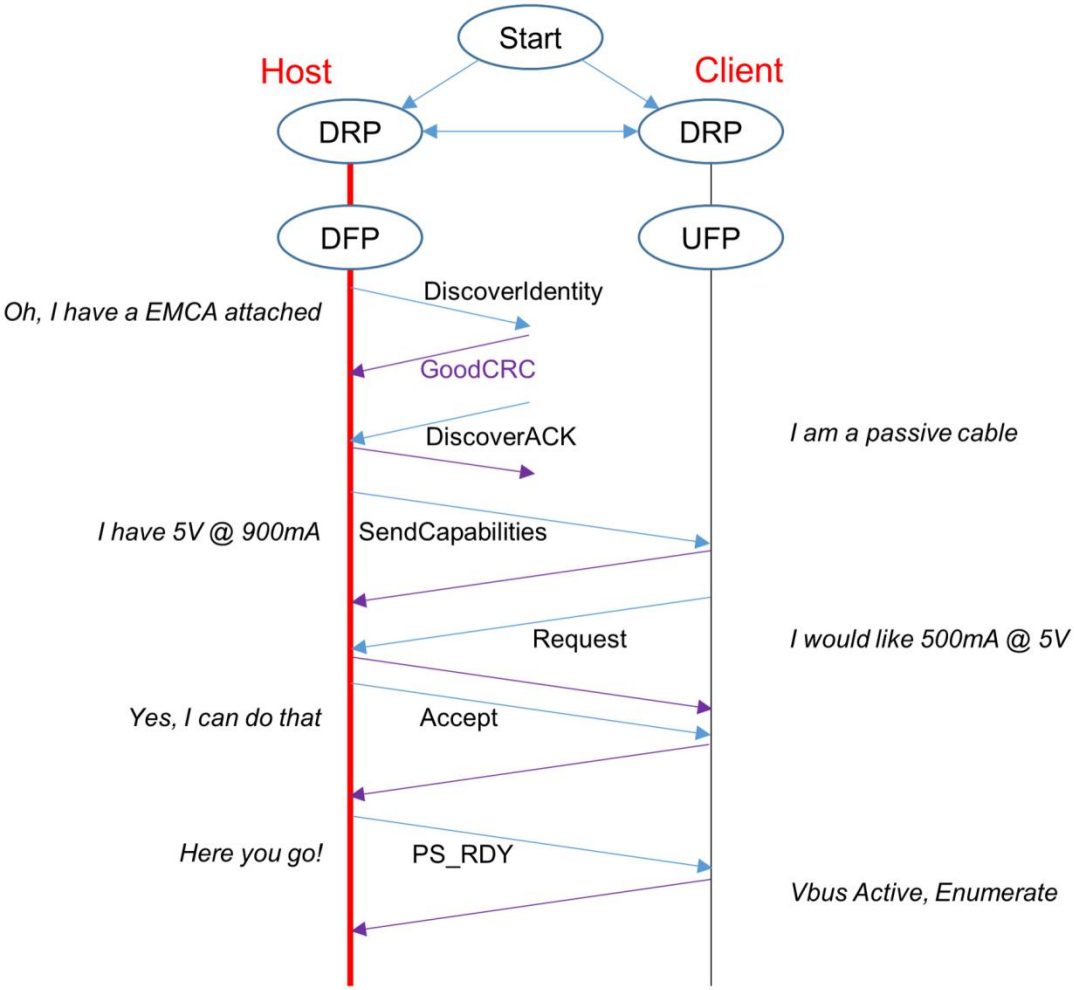
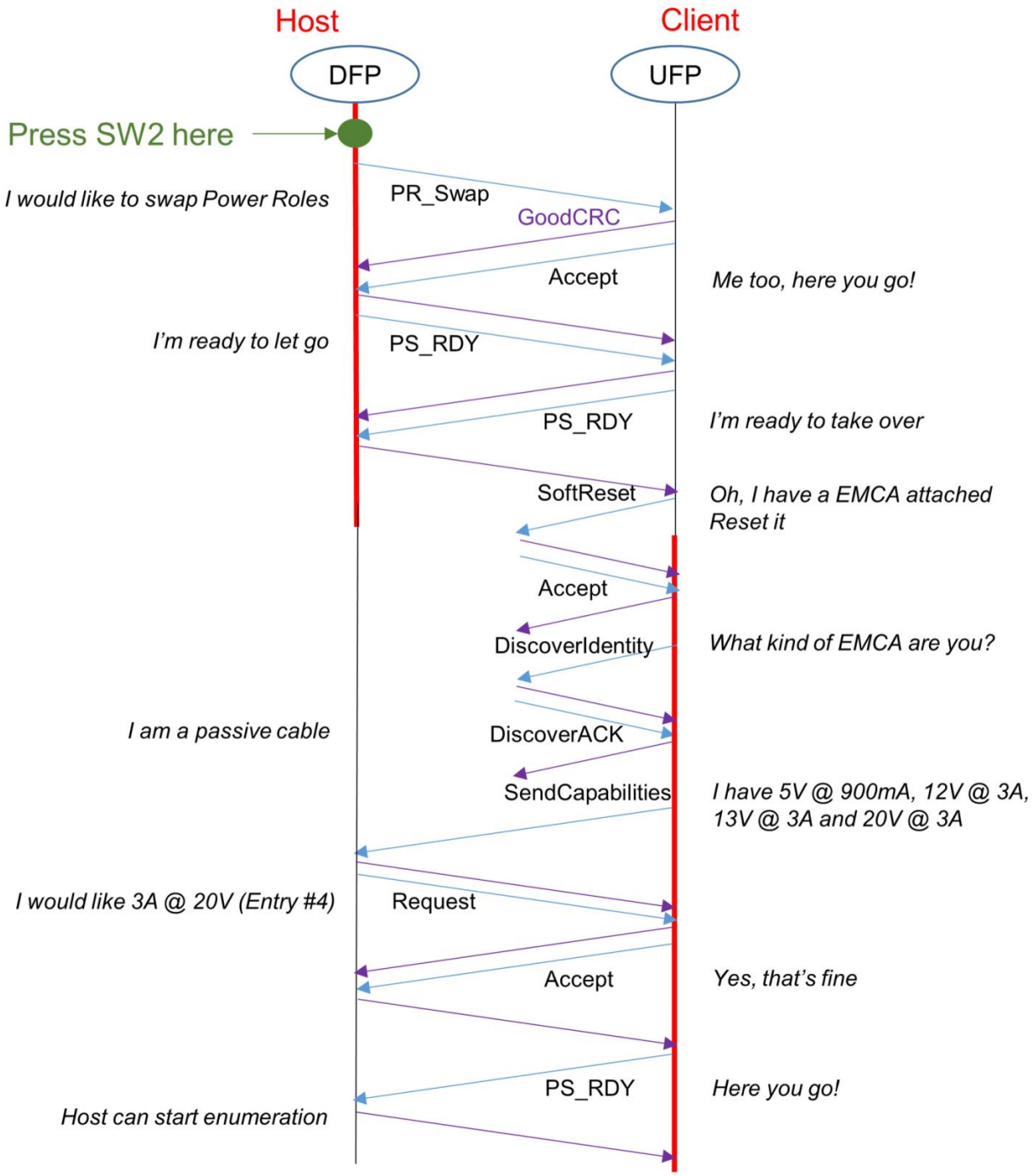


Figure 20 hides some of the complexity that the CCG1 devices are implementing. Each generated message has a MessageID inserted and a CRC added. The conversation between CCG1 devices typically entails a request followed by a response. Each CCG1 device will check that all messages have the correct CRC and that response MessageIDs match request MessageIDs. All actions are also timed and counted, and the PD Specification details a great number of constraints and what to do if any are violated. The CCG1 reference designs implement all the requirements of the PD Specification. The CCG1 firmware will be updated as necessary so that all compliance tests, when available, will be passed.

To demonstrate the power adapter capabilities, press SW2 on the host board to initiate a power role swap (PR_Swap). Figure 21 shows the messages that are exchanged. They are similar to the initial power-on messages except that the client board is the new initiator. The client board has a power brick attached, so it can offer more power options. The host board chooses 3 A at 20 V, and the client board delivers it over Vbus to be available at the host board's DC out connector. The vertical red lines show the transfer of ownership of the Vbus line.

The host computer is now being powered by the power brick via the Type-C cable.

Figure 21. Flow of CC Messages in Power Delivery Example



The PD Specification makes power delivery seem complex, because a specification must cover all conceivable cases. This Type-C–only example demonstrated the essence of power delivery. It is straightforward because all the heavy lifting has been done by the CCG1 firmware.

3.2 Adding USB Data Paths

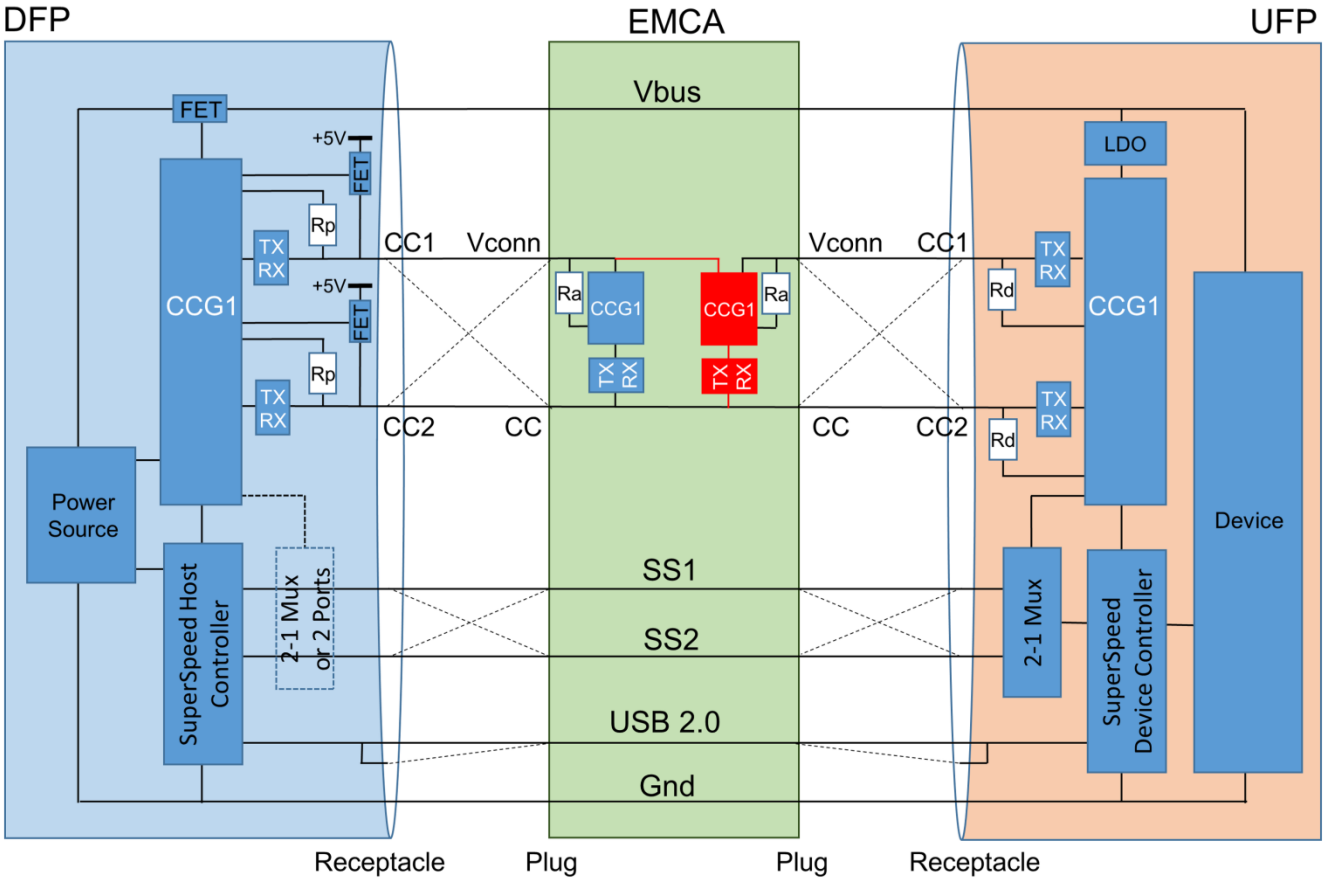
This document has covered only the power delivery aspect of a Type-C design thus far. The reversible connector has implications on the data path too. As shown in Figure 2, the SuperSpeed data lanes can be swapped; however, the USB 2.0 signals, Dp and Dn, are not swapped because Dp1 and Dp2 are connected and Dn1 and Dn2 are connected in the receptacle.

Figure 22 extends Figure 9 to include the USB data paths. A data multiplexer (MUX) must now be included in a UFP so that it can select the SuperSpeed lane, SS1 or SS2, which is carrying the data. The DFP needs the same logical switch, which could be implemented as a MUX or two host ports on the SuperSpeed host controller. The CCG1 devices control these switches.

It is technically possible to have both SuperSpeed lanes active in a Type-C cable. This would be easy for the Host, but the peripheral would need to implement two independent SuperSpeed device controllers.

The schematics of the CY4501 CCG1 Development Kit show details of the 5-Gbps MUXes used in both designs. Data switching circuitry is also included for Alternate Modes, as described in the following section.

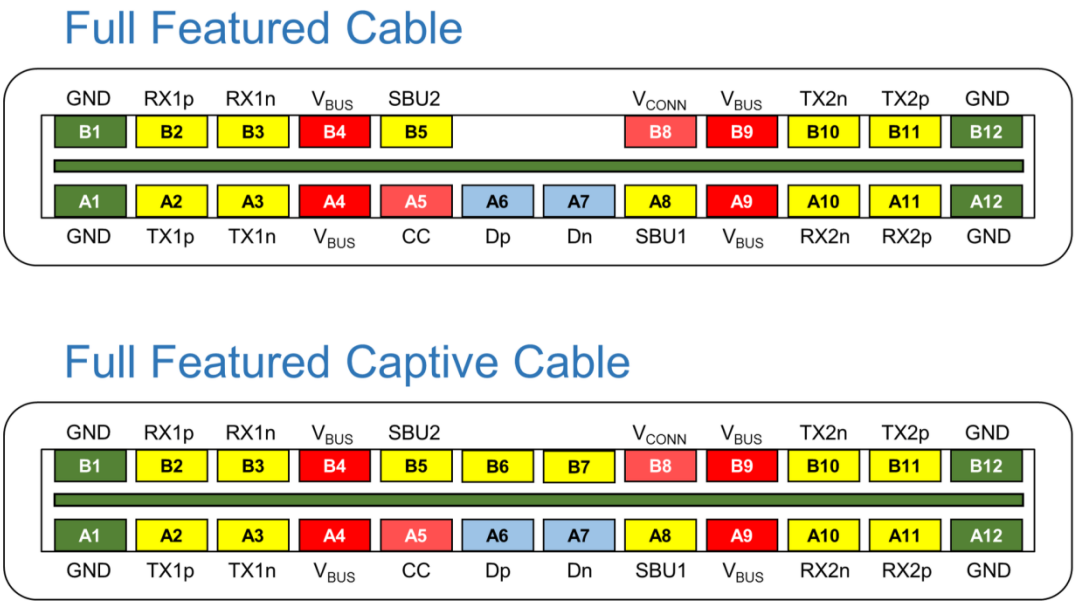
Figure 22. Routing Is Required on the SuperSpeed Data Lanes



3.3 Type-C Alternate Modes

The Type-C Specification designates several signals that can be repurposed within a Type-C interconnect. Figure 23 shows these signals in yellow. The two SuperSpeed lanes (four differential pairs), two sideband signals, and if the peripheral is directly connected (no intervening cable between UFP and DFP), Vconn and the two unallocated plug pins, B6 and B7, can also be used. A dock or a dongle with a captive cable would be examples of directly connected devices.

Figure 23. Up to 9 Signals (4 pairs + 5) Can Be Repurposed



Before the signals can be reassigned, the DFP and UFP must have a conversation to agree on the new mode in which the signals will now operate. Basically, the DFP asks the UFP which modes it supports, and if the DFP also supports this mode, it will enter it. However, the DFP must check that any intervening cable also supports this new mode. Hubs are not required to propagate Alternate Mode signals, so Alternate Mode operation is defined only between a DFP and a connected (via cable or directly) UFP.

If the UFP does not support the mode that the DFP desires, then it must not fail silently. The Type-C Specification uses the USB 2.0 channel, which is always present, as the mechanism to report and resolve Alternate Mode conflicts. A USB device that supports the Billboard class must be present on the UFP attached to the Dp and Dn lines.

A popular example of using a Type-C interconnection in Alternate Mode is DisplayPort. Figure 24 is a block diagram of the host and client boards showing data MUXes that the CCG1 devices control to switch the use of both SuperSpeed pairs and the sideband signals for DisplayPort use. The CY4501 CCG1 Development Kit design hardware supports four lanes of DisplayPort at the host and client end. The design files for these boards can be found here. Other examples of Alternate Mode signals that could be implemented include PCIe, Mobile High-Definition Link, and eSATA.

Figure 24. Block Diagram of Data Path Switching for Alternate Mode Example

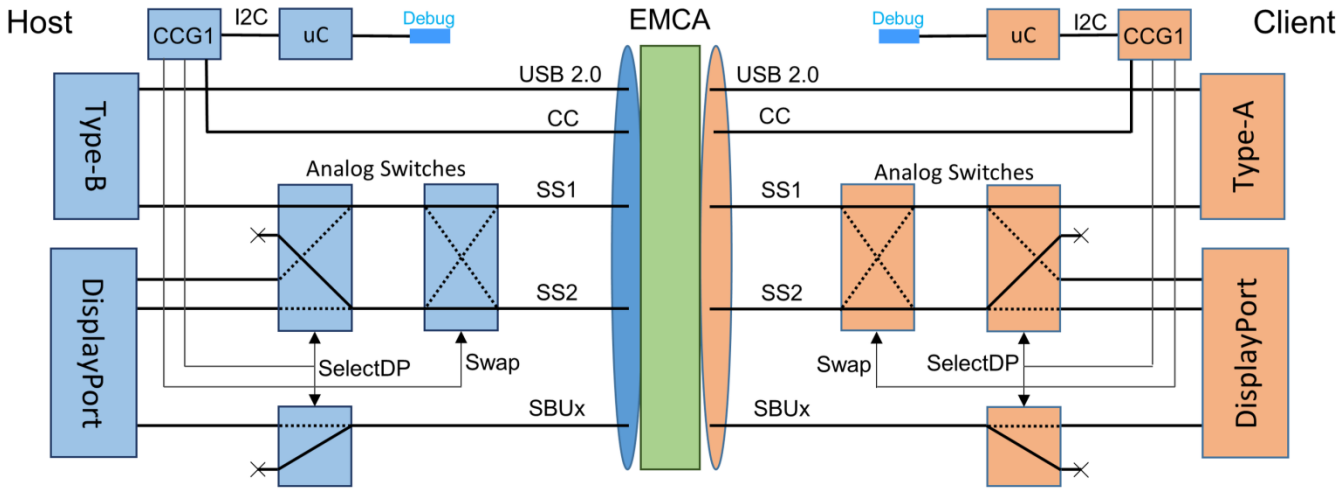
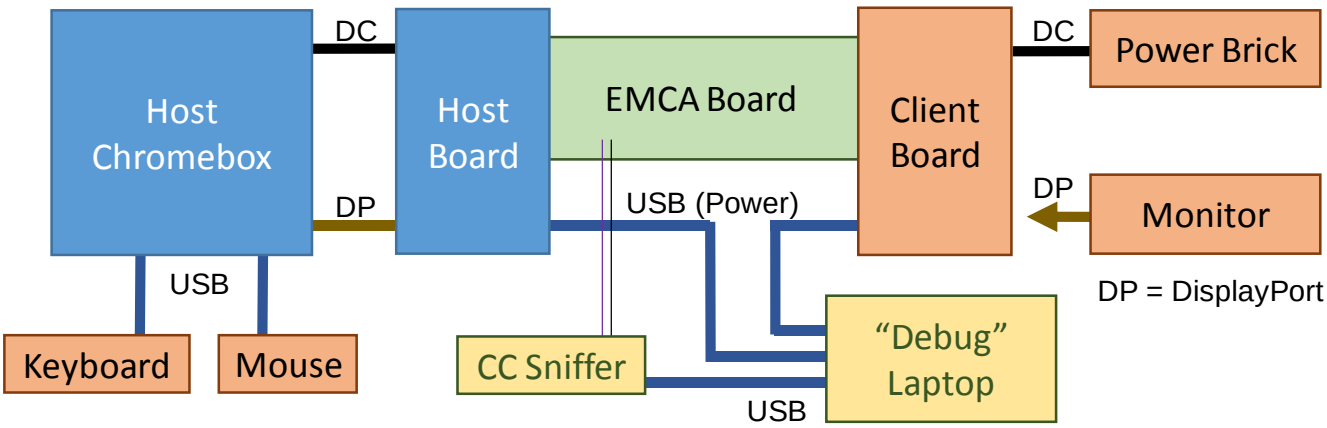


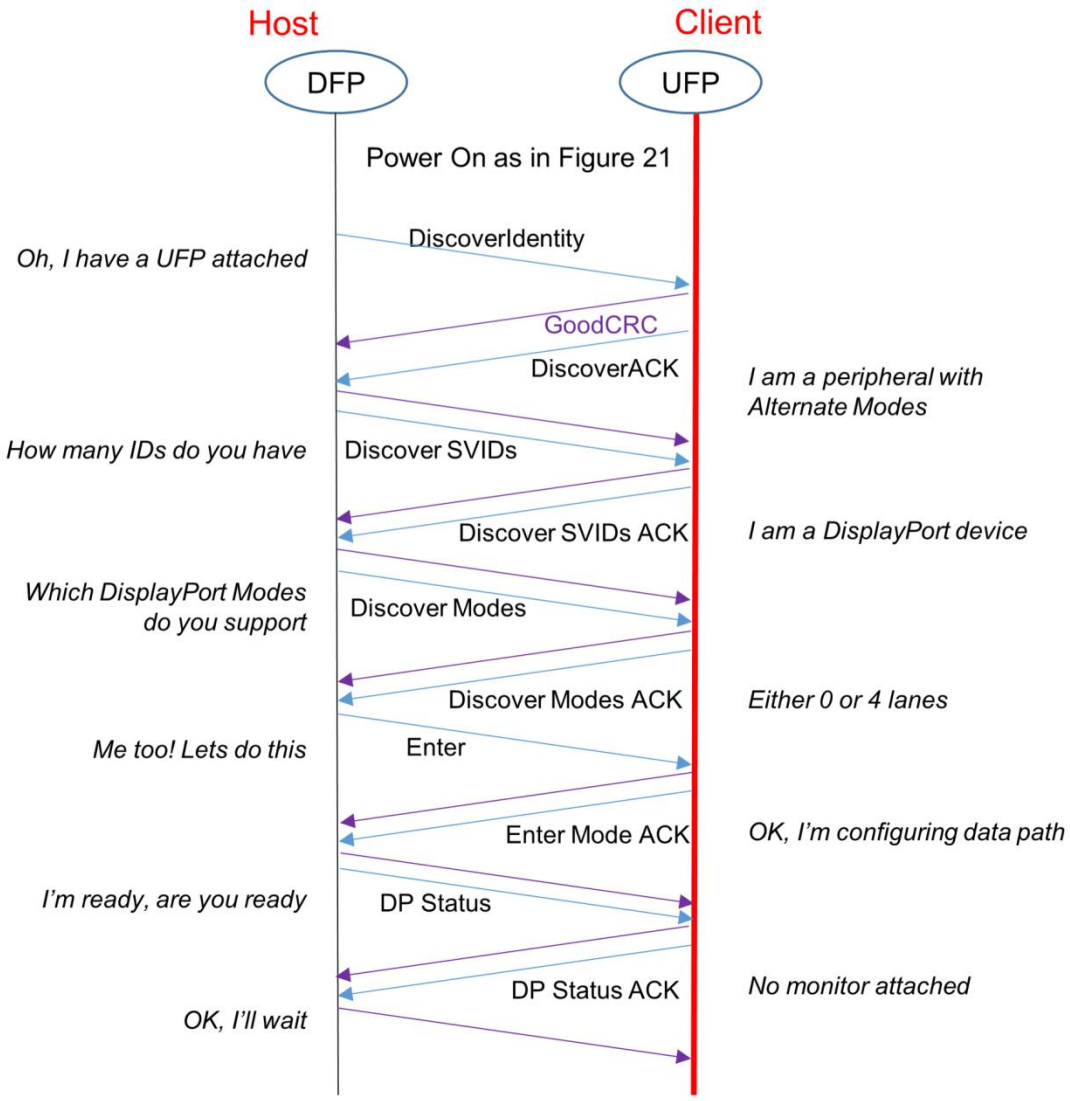
Figure 25 shows a configuration of the hardware for the Alternate Mode Example. The power brick is still connected, and the DisplayPort monitor has been moved from the Chromebox PC. It will be connected to the client board. The Chromebox PC is now connected to the host board with a standard DisplayPort cable. Note that the debug computer connections are still present.

Figure 25. Setup for Alternate Mode Example



The boards power up as before, and the client board provides power to the host board. Figure 26 shows the startup conversation between the host and client, with the GoodCRC messages shown in purple.

Figure 26. CC Messages Exchanged During Alternate Mode Initialization

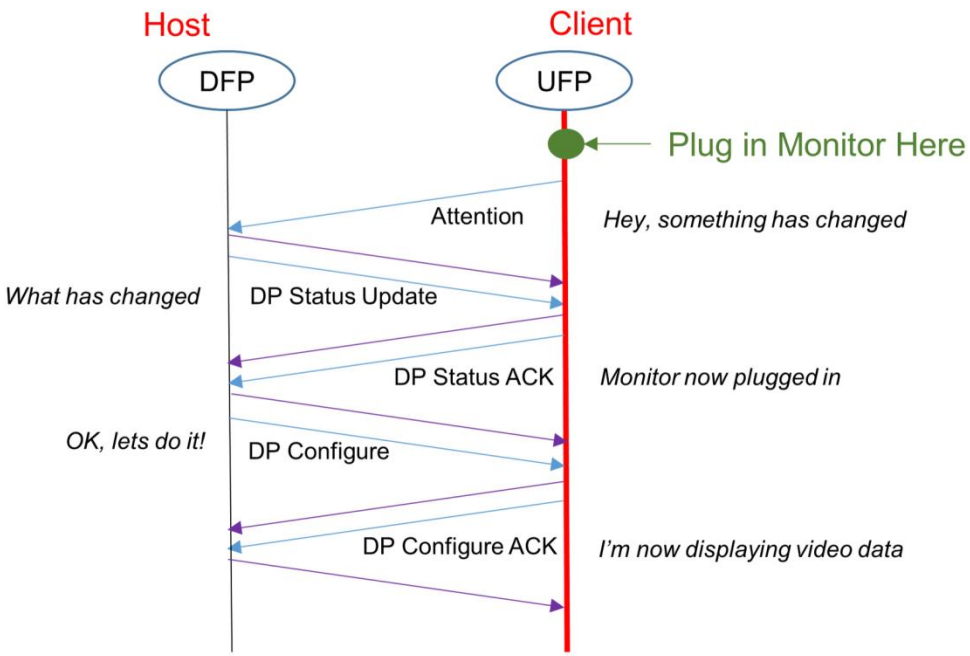


The host board discovers that it has an Alternate Mode device attached, so it discovers its Type-C Specification defined Standard ID (SID) or Vendor ID (VID) (so SVID = SID or VID). The client board reports an SVID of 0xFF01, which is an ID assigned to a DisplayPort connection. This is fully documented in the VESA specification, DisplayPort Alt Mode on USB Type-C Standard, which is available only to VESA members. This example uses key information extracted from the specification for annotation purposes.

The client board reports that it supports four DisplayPort lanes. The host also supports the four-lane mode, so it enters this mode, as does the client (signaled by its positive ACK). The host now checks the DisplayPort status, and the client board reports that no display is attached.

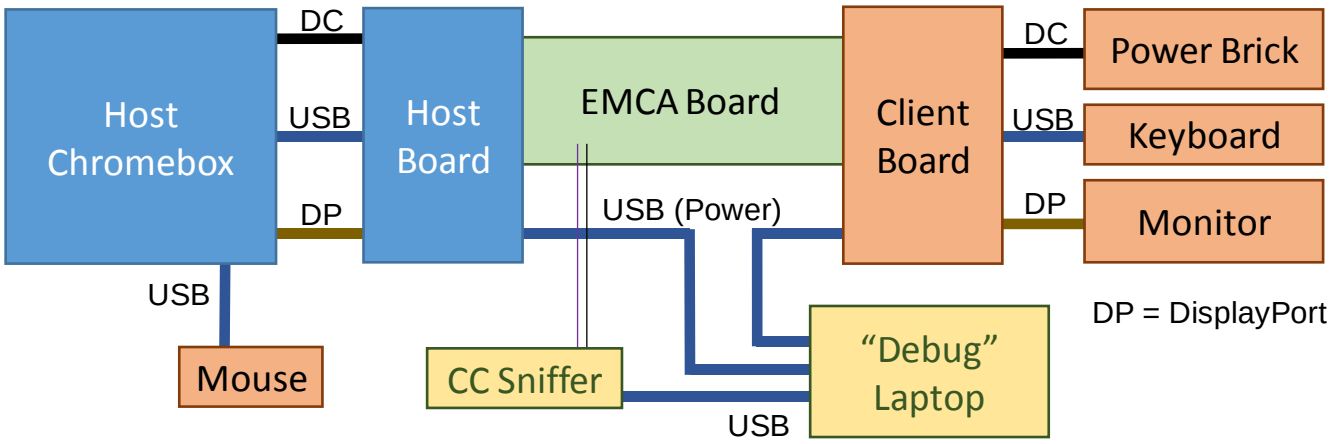
Now, attach the DisplayPort monitor to the client board. The client CCG1 device recognizes the connection, via an I/O pin connected to the Hot Plug Detect (HPD) pin, and initiates an Alternate Mode sequence by sending an Attention message to the host, as shown in Figure 27. The host rechecks the status and, since all is good, it sends a DisplayPort Configure message to the client. Both CCG1 devices switch over their MUXes so that the Chromebox PC display appears on the monitor.

Figure 27. Host and Client Negotiate to Support DisplayPort



As a final experiment, you could disconnect the keyboard from the Chromebox PC and connect it to the USB Type-A connector on the client board, as shown in Figure 28. The Host PC will be connected to keyboard here too as expected.

Figure 28. Setup for DisplayPort plus USB Example

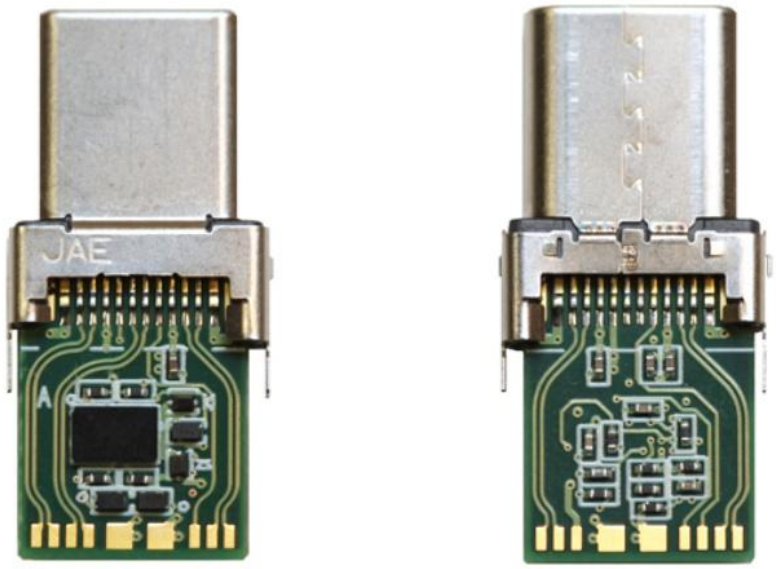


You now have power, display, and USB all running over the single USB Type-C cable.

4 Designing a Type-C EMCA

The CCG1 EMCA board of the CY4501 CCG1 Development Kit implemented a cable as a PCB with many jumpers and probe points so that various solutions using one or two CCG1 devices could be explored. The paddle card design shown in Figure 11 is a full reference design. Schematics are downloadable from the CCG1 EMCA Paddle Card Reference Design web page. Figure 29 shows the front and back of the paddle board without wires or plastics.

Figure 29. Complete CCG1-Based Cable Paddle Board Reference Design



4.1 Hardware Design Guidelines

Within a paddle card design, CCG1 devices are powered from the Vconn supply of the DFP. Consider the following while designing the power system network:

1. Placement of bulk and decoupling capacitors
2. Placement of power and ground planes

Placement of Bulk and Decoupling Capacitors

Place decoupling capacitors for high-frequency noise filtering close to the Vconn, Vddd, and Vccd pins as shown in Reference Schematics. Place the bulk capacitor, which acts as a local power supply to the power pin, close to the Vddd pin of CCG1. The Vbus connections need bulk decoupling close to the connector; a 10-nF bypass capacitor per Vbus pin is recommended.

Make the power trace width the same as the power pad width. To connect the power pins to the power plane, keep vias very close to the power pads. This helps minimize stray inductance and IR drop on the line.

Placement of Power and Ground Planes

Place the power plane close to the ground plane for good planar capacitance. Planar capacitance between the planes acts as a distributed decoupling capacitor for high-frequency noise filtering, thereby reducing electromagnetic radiation.

The USB data lines, both SuperSpeed lanes and USB 2.0 signals, although not connected to the CCG1, are always close to it in a paddle card design. The USB data lines are most critical to achieve good signal quality and reduce emission. Follow the guidelines below while designing a paddle card:

1. Use a high-performance substrate material for paddle cards.
2. Keep USB SuperSpeed traces as short as possible. Ensure that these traces have a nominal differential characteristic impedance of 90 ohms.

3. Match the differential SS pair trace lengths within 0.12 mm (5 mils).
4. Match the high-speed (Dp and Dn) signal trace lengths within 1.25 mm (50 mils).
5. Ensure that the differential pairs have a minimum pair-to-pair separation of 0.5 mm.
6. Adjust the high-speed signal trace lengths near the USB receptacle, if necessary.
7. Make adjustments for SS RX signal trace lengths near the USB receptacle. Make adjustments for SS TX signal trace lengths near the device if necessary.

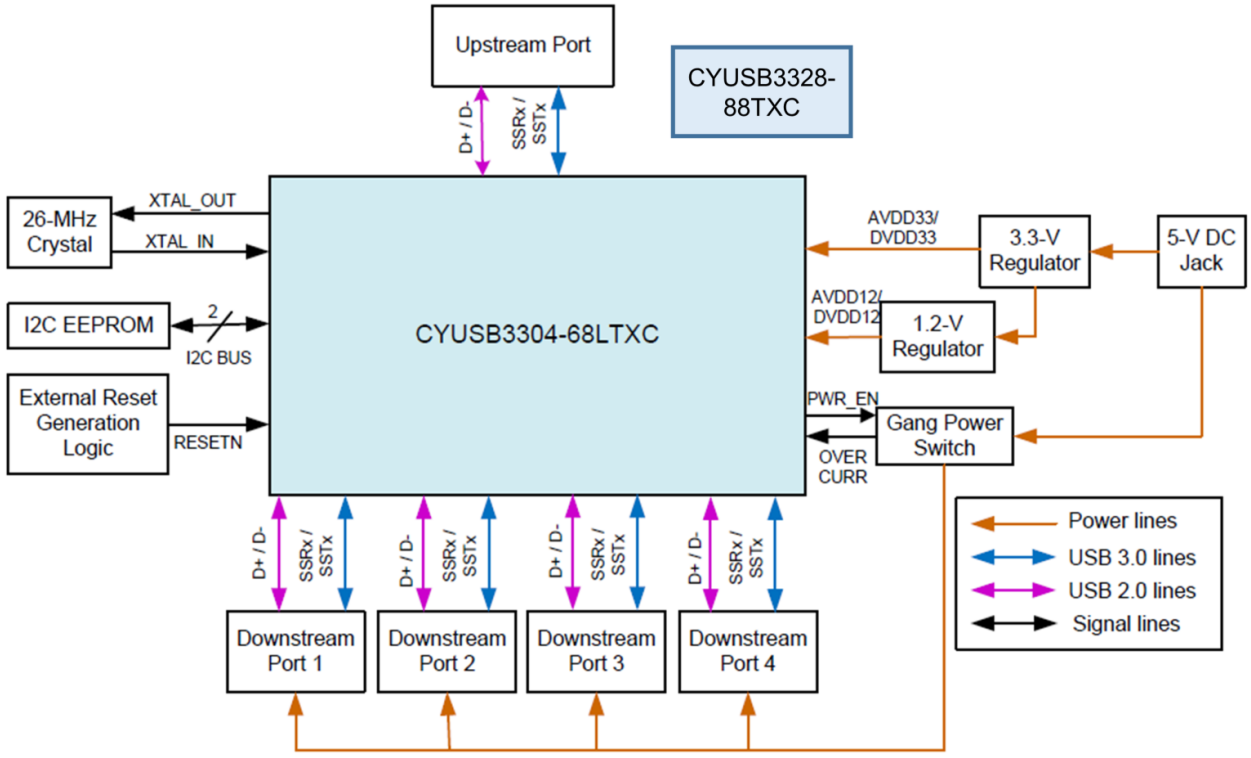
These guidelines should be followed with the layout of host and peripheral CCG1 designs but typically these are less critical because a larger-package CCG1 may be used and it is located further from the USB data lines. In these designs, an I^2C bus is connected to the CCG1 and this should be pulled up with 2.2-kΩ resistors to the CCG1 Vddd supply. The CY4501 CCG1 Development Kit includes reference layout material for these designs.

A compact Type-C to DisplayPort Adapter Reference Design is also available for study.

5 Designing a Docking Station

This section extends the design of the client board and adds a hub and a collection of other devices that would be found in a docking station. Designs using a hub were straightforward prior to the introduction of the PD Specification, as shown in Figure 30. This is a figure from AN94150 – Designing a SuperSpeed Hub Using HX3 with Optimized BOM in which a self-powered hub distributed 900 mA at 5 V to four downstream ports (a total of 18 W). The major design decisions were whether to support ganged or distributed overcurrent detection and which indicator LEDs to include.

Figure 30. Hub Design Before Power Delivery Specification



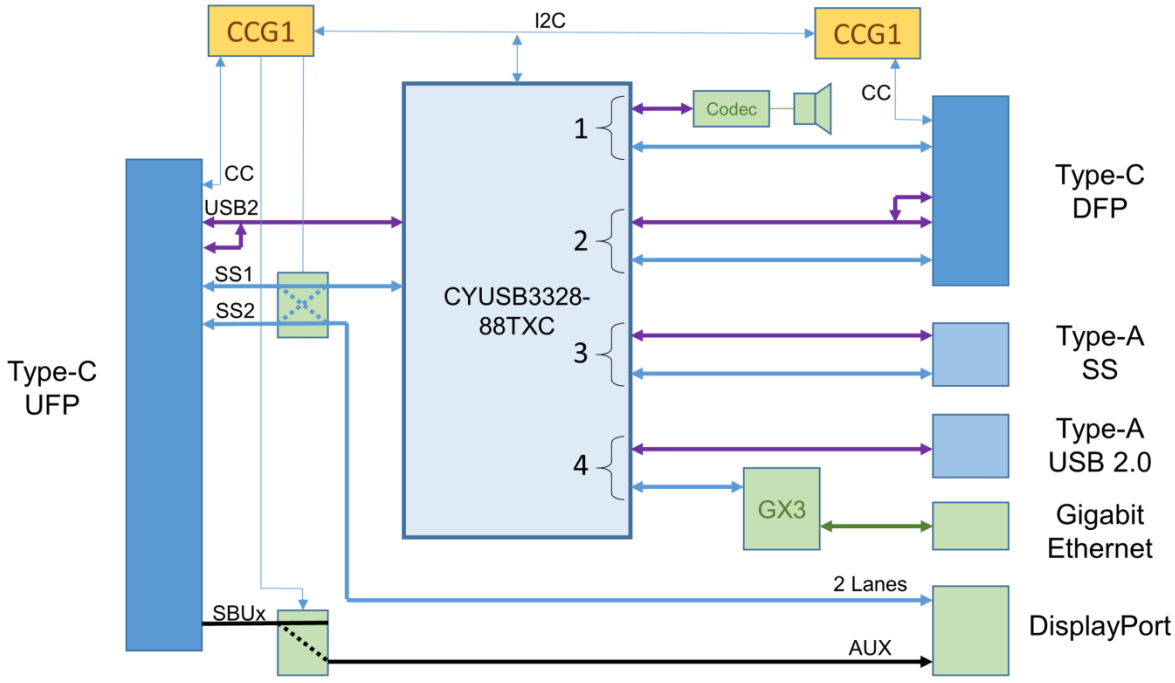
A hub now plays a much more important role because it can manage and distribute up to 100 W of power for each Type-C port. Each port can be a power provider or power consumer, and it is now feasible for a docking station or powered monitor to power and/or charge a laptop or tablet using the same single Type-C cable used for data.

The HX3 also has some unique features that are showcased in this design. A USB 3.1 bus is, in fact, two separate and independent buses, a SuperSpeed bus and a high-speed USB 2.0 bus packaged in a single cable, but only one of these buses is usually active. The HX3 contains additional circuitry that enables both buses to be active if embedded devices are implemented. Figure 31 shows the data paths within a docking station/powered monitor design. Note that the HX3's Shared Link™ technology allows more downstream capability than a typical hub component.

Downstream ports 1 and 2 are both routed to a Type-C connector. Because both SS1 and SS2 are driven, the cost and complexity of a data switch is saved. Also, there is a "spare" USB 2.0 channel, which is used to drive an audio codec. HX3 Downstream Port 3 is connected to a Type-A connector for the attachment of flash drives, backup hard drives, or similar SuperSpeed peripherals. The SuperSpeed lines of the HX3 Downstream Port 4 are connected to a Gigabit Ethernet controller, the GX3. Because the USB 2.0 lines are not needed for this embedded device, they are connected to a Type-A connector for a mouse, keyboard, or wireless dongle that would support both the mouse and keyboard.

Also shown is a two-lane DisplayPort Alternate Mode device, implemented on SuperSpeed lane 2. A switch would not be needed if the dock was a directly connected device and the host knew how to send DisplayPort data on lane 2 (that is, the host and docking station are manufactured by the same vendor).

Figure 31. Data Paths of a Type-C Hub Design



6 Summary

This application note showed that all forms of Type-C design can be implemented with the CCG1 controller. The reference designs, including full schematics and application software, will get you off to a running start. The examples are based on SuperSpeed, but remember that Type-C Power Delivery is also applicable to USB 2.0 high-speed systems.

7 Related Documents

User Guides

- CY4501 CCG1 Development Kit Guide
- CY8CKIT-040 – PSoC 4000 Pioneer Development Kit Guide

Application Note

- AN94150 – Designing a SuperSpeed Hub Using HX3 with Optimized BOM

About the Authors

Name: John Hyde

Title: Principal Consultant at USB Design By Example

Background: John has been involved in USB since its inception and wrote his first *USB Design By Example* book in 1999. He works with a wide variety of USB folk in architecture, design, and debug of their embedded products. He has written five "how to" books and is now working on an update to his SuperSpeed book and a new *USB Type-C Design By Example* book. He earned a B.Sc in Electronics from Southampton University, UK, and now lives in Portland, OR.

Name: Vihang Trivedi

Title: Applications Engineer Sr

Background: Vihang Trivedi supports USB Type-C controllers at Cypress Semiconductors. He holds a Masters degree in Electrical Engineering from California State University, Chico.

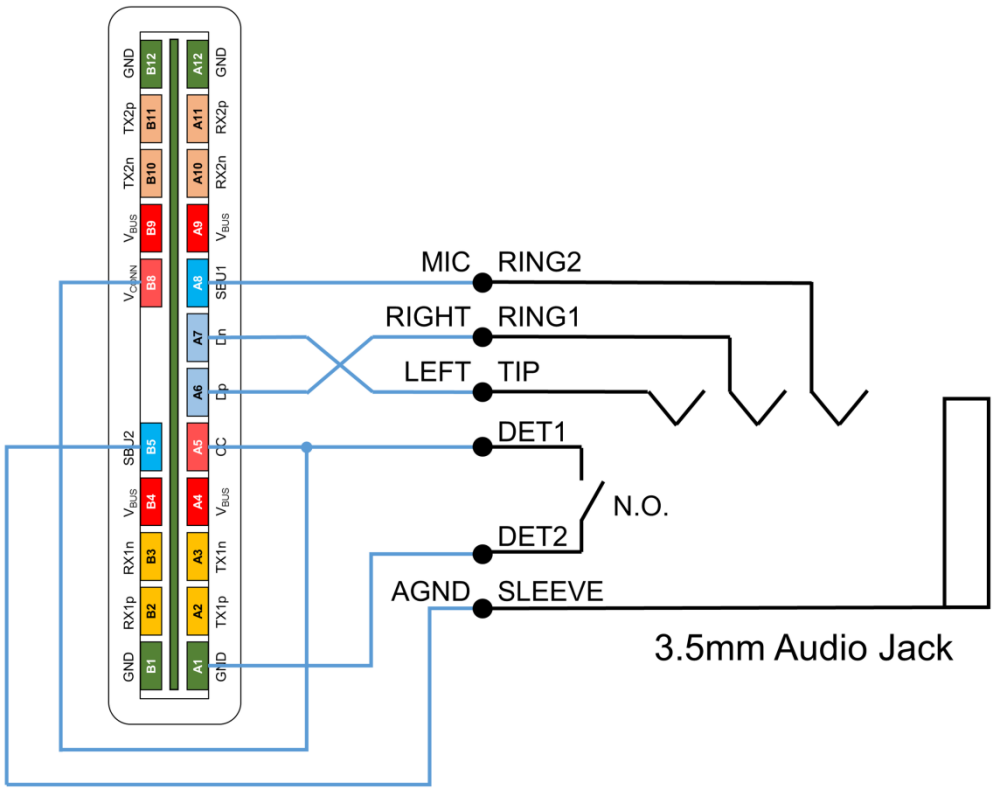
8 Appendix A: USB and Type-C Acronyms

BMC	biphase mark code
CC	configuration channel
CCGx	configuration channel controller generation 1 or 2
CDP	charging downstream port
CRC	cyclic redundancy check
DCP	dedicated charging port
DFP	downstream facing port
DRP	dual-role port
EMCA	electronically marked cable assembly
I^2C	inter-integrated circuit
OCP	overcurrent protection
OVP	overvoltage protection
PD	power Delivery
PD Specification	USB Power Delivery Specification Revision 2.0, Version V1.0, a free download from www.usb.org/developers
PDO	power delivery object
PHY	physical Layer
SCB	serial communication block
SDP	standard downstream port
SOP	Start of Packet symbol directed at USB-PD Port Partner.
SOP'	Start of Packet symbol directed at CCGx at the near end of the EMCA
SOP''	Start of Packet symbol directed at CCGx at the far end of the EMCA
SOP*	Collectively denoting the Start of Packet symbol (SOP, SOP', and SOP'')
SWD	serial wire debug
Type-C Specification	USB Type-C Cable and Connector Specification Revision 1.0, a free download from www.usb.org/developers
UFP	upstream facing port
USB	Universal Serial Bus
USB-IF	USB Implementers' Forum, a non-profit organization chartered with promoting USB
USB-PD	USB Power Delivery

9 Appendix B: Accessories

A UFP is defined as having an Rd on its CC lines. If Rd is less than Ra (Ra = 1 kΩ ± 20 percent), then the DFP would recognize this device as an audio accessory, as shown in Figure 32. Note that Vconn and CC are connected together and then connected to DET1 on the audio jack. When an audio plug is inserted, the switch is closed, and DET1 is shorted to ground (a short is less than 1 kΩ). The DFP will see this as a connection event. It is essential that the DFP does not apply +5 V to Vconn on this audio accessory, since it is connected directly to ground.

Figure 32. Connection of an Audio Jack to Type-C Plug



The simplicity of this design is that it contains only wires. There are no active or passive components. In this way, headphones could be made with a Type-C plug instead of an audio jack with little cost added.

The Type-C Specification also describes a debug accessory, but no details have been disclosed.

Document History

Document Title: AN96527 – Designing USB Type-C Products Using Cypress's CCG1 Controllers

Document Number: 001-96527

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	4784481	VGT	06/03/2015	New Application Note
*A	5742443	AESATMP9	05/19/2017	Updated logo and copyright.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709