USB Type-C ENGINEERING CHANGE NOTICE

Title: ErrorRecovery Exit for VCONN Source
Applied to: USB Type-C Specification Release 1.3, July 14, 2017

Brief description of the functional changes proposed:

Increase the minimum time that a VCONN source must wait before exiting the ErrorRecovery state. The minimum time is increased to be tVconndischarge (max).

Benefits as a result of the proposed changes:

This allows time for the cable to discharge VCONN before an Rp is applied to it. This will prevent interoperability issues that may arise because the VCONN pin is not fully discharged before the port moves to the Unattached.SRC state and applies an Rp to the VCONN pin.

Case 1: VCONN source was the Source, and the source goes from Attached.SRC to ErrorRecovery.
   ➔ The VCONN bulk capacitance will be disconnected, but the source does not discharge VCONN nor give the cable time to discharge. This is problem is addressed.

Case 2: VCONN source was the Source, and the sink goes from Attached.SNK to ErrorRecovery.
   ➔ The source goes from Attached.SRC to UnattachedWait.SRC and discharges VCONN. There is no problem here.

Case 3: VCONN source was the Sink, and the source goes from Attached.SRC to ErrorRecovery
   ➔ Sink goes to Unattached.SNK which discharges VCONN. There is no problem here.

Case 4: VCONN source was the Sink, and the sink goes from Attached.SNK to ErrorRecovery
   ➔ The sink disconnects the VCONN bulk capacitance, but the source goes to Unattached.SRC immediately and does not discharge VCONN. This means the source's Rp may prevent the cable from discharging VCONN when it is connected through the cable (Fig. 4-40). This is problem is addressed.

An assessment of the impact to the existing revision and systems that currently conform to the USB specification:

Sources that support VCONN_Swap need to transition from Attached.SRC to UnattachedWait.SRC instead of Unattached.SRC upon disconnect even when they are not supplying VCONN.

Sources that supply VCONN need to remain in the ErrorRecovery state longer than previously required. Existing implementations may or may not already remain in the ErrorRecovery state long enough.

An analysis of the hardware implications:

An analysis of the software implications:

Systems may just increase their timeout value for tErrorRecovery or change to always transition through UnattachedWait.SRC.

An analysis of the compliance testing implications:

The compliance would need to change how it tests tErrorRecovery and exiting Attached.SRC.

USB Type-C ENGINEERING CHANGE NOTICE

Actual Change Requested

(a). Section 4.11.2, Page 201
From Text:

tErrorRecovery	25ms		Time a self-powered port shall remain in the ErrorRecovery state.

To Text:

tErrorRecovery	25ms		Time a self-powered port shall remain in the ErrorRecovery state.
	240ms		Time a source shall remain in the ErrorRecovery state if it was sourcing Vconn in the previous state.

(b). Section 4.5.2.2.9.2, Page 157
From Text:

A Source that is supplying V$_{CONN}$ shall transition to UnattachedWait.SRC when the SRC.Open state is detected on the monitored CC pin. The Source shall detect the SRC.Open state within tSRCDisconnect, but should detect it as quickly as possible.

A Source that is not supplying V$_{CONN}$ shall transition to Unattached.SRC when the SRC.Open state is detected on the monitored CC pin. The Source shall detect the SRC.Open state within tSRCDisconnect, but should detect it as quickly as possible.

To Text:

A Source that is supplying V$_{CONN}$ or has yielded V$_{CONN}$ source responsibility to the Sink through USB PD V$_{CONN}$ Swap messaging shall transition to UnattachedWait.SRC when the SRC.Open state is detected on the monitored CC pin. The Source shall detect the SRC.Open state withintSRCDisconnect, but should detect it as quickly as possible.

A Source that is not supplying V$_{CONN}$ and has not yielded V$_{CONN}$ responsibility to the Sink through USB PD V$_{CONN}$ Swap messaging shall transition to Unattached.SRC when the SRC.Open state is detected on the monitored CC pin. The Source shall detect the SRC.Open state within tSRCDisconnect, but should detect it as quickly as possible.