

USB Power Delivery ENGINEERING CHANGE NOTICE

Title: Delay of VCONN_SWAP

**Applied to: USB Power Delivery Specification Revision 3.2
Version 1.0**

Brief description of the functional changes proposed:
This ECN allows the VCONN provider to send WAIT response if the far-end port supports same or newer specification revision. The ECR also requires that a port delays its future VCONN_Swap request by at least tVconnSwapDelay* after losing VCONN ownership.

Benefits as a result of the proposed changes:
Avoid possibility of ports getting stuck in constant VCONN_Swap battle

An assessment of the impact to the existing revision and systems that currently conform to the USB specification:
No impact to existing PD controllers

An analysis of the hardware implications:

An analysis of the software implications:
The PD controllers will have to be updated to implement new behavior

An analysis of the compliance testing implications:
Compliance tester should ensure that both the tester and DUT support PD spec revision same or newer than 3.2.1.1 if the DUT in VCONN Source role responds with WAIT to VCONN_Swap message from the tester. Also if the tester supports an older version of the spec then the DUT should Accept the request and should not send another VCONN_Swap request within tVconnSwapDelay* of PS_RDY to previous VCONN_Swap sequence The tester as VCONN-Source can send WAIT response to DUT that supports specification version earlier than 3.2.1.1 and expect a SoftReset as WAIT was an unexpected response.

USB Power Delivery ENGINEERING CHANGE NOTICE

Rationale for Change

This ECN tries to prevent the DFP and UFP getting stuck in constant VCONN_Swap (VCS) sequence to get VCONN ownership. The following sequence of events could lead to such scenario:

1. A DRP host that is preferred to be a DFP connects as a SOURCE to a DRP dock that prefers to be a SOURCE
2. The ports make PD contract
3. The dock sends PR_Swap to become a SOURCE
4. After PD contract that follows PR_Swap
 - The host starts the alternate mode discovery and mode entry sequence
 - The dock wants to update its Source Cap to advertise new Source Capability PDOs with higher power
5. The dock sends VCONN_Swap to read cable discovery ID response prior to issuing new Source Cap with higher power
 - The host Accept VCONN_Swap and completes VCONN_Swap sequence
6. The dock waits for tVconnStable (min 50ms) prior to communication with the cable
8. Meanwhile the host wants to initiate discovery on cable plug and issues VCONN_Swap
9. The port completes VCONN_Swap sequence there by UFP losing VCONN ownership
10. While the DFP is waiting for tVconnStable timer, the UFP sends VCONN_Swap since it still has a pending task to send updated Source Capabilities

This results in steps 5 to 10 keep repeating.

At least one of the ports should take an action to prevent ports getting stuck in this cycle.

We have discussed various proposals and decided to address this issue with two changes proposed in this ECR.

- (a) Allow VCONN provider to respond with WAIT if it intends to talk to cable plug controller. This will allow new hosts to prevent this lock-up scenario with devices that support this or newer specification revision
- (b) Force a port that loses VCONN-ownership to delay next VCONN_swap request. This allows new host/devices to avoid this problem with existing devices/hosts

USB Power Delivery ENGINEERING CHANGE NOTICE

Actual Change Requested

(a). Section 6.3.11 VCONN_Swap Message, Page 124

From Text:

- If a **Wait** Message is sent, the requester is informed that a VCONN Swap might be possible in the future but that no immediate action **Shall** be taken. A **Wait** Message **Shall** only be sent by the Port that is not presently the Vconn Source in response to a **VCONN_Swap** Message. The Port that is presently the Vconn Source **Shall Not** send a **Wait** Message in response to **VCONN_Swap** Message.

To Text:

- If a **Wait** Message is sent, the requester is informed that a VCONN Swap might be possible in the future but that no immediate action **Shall** be taken. ~~A **Wait** Message **Shall** only be sent by the Port that is not presently the Vconn Source in response to a **VCONN_Swap** Message. The Port that is presently the Vconn Source **Shall Not** send a **Wait** Message in response to **VCONN_Swap** Message.~~

A port after losing VCONN Source role due to incoming VCONN_Swap request **Shall** not initiate VCONN_Swap until at least $tVconnSwapDelayDFP$ / $tVconnSwapDelayUFP$ after completing the previous VCONN_Swap sequence.

(b) 6.3.12 Wait Message, page 137

From Text:

- It **Shall** be sent by the recipient of a **VCONN_Swap** Message that is not presently the VCONN Source to indicate it is currently unable to do a VCONN Swap.

To Text:

- It **Shall** be sent by the recipient of a **VCONN_Swap** Message ~~that is not presently the VCONN Source~~ to indicate it is currently unable to do a VCONN Swap.

(c) 6.3.12.4 Wait in response to a VCONN_Swap Message, Page 138

From Text:

The **Wait** Message is used when responding to a **VCONN_Swap** Message to indicate that a **VCONN_Swap** might be possible in the future. This can occur in any case where the device receiving the **VCONN_Swap** Message needs to evaluate the request further. A **Wait** Message **Shall** only be sent by the Port that is not presently the VCONN Source in response to a **VCONN_Swap** Message. The Port that is presently the VCONN Source **Shall Not** send a **Wait** Message in response to **VCONN_Swap** Message. Once it has completed this evaluation one of the Port Partners **Should** initiate the VCONN Swap process again by sending a **VCONN_Swap** Message.

To Text:

The **Wait** Message is used when responding to a **VCONN_Swap** Message to indicate that a **VCONN_Swap** might be possible in the future. This can occur in any case where the device receiving the **VCONN_Swap** Message needs to evaluate the request further. ~~A **Wait** Message **Shall** only be sent by the Port that is not presently the VCONN Source in response to a **VCONN_Swap** Message. The Port that is presently the VCONN Source **Shall Not** send a **Wait** Message in response to **VCONN_Swap** Message.~~ Once it has completed this evaluation one of the Port Partners **Should** initiate the VCONN Swap process again by sending a **VCONN_Swap** Message.

USB Power Delivery ENGINEERING CHANGE NOTICE

A Port that is currently the Vconn Source Shall respond with an Error! Reference source not found. Message if the Port Partner's Revision and Version, as reported in the Revision Message, is earlier than R3.2 V1.1. A Port Partner supporting an earlier Revision and Version will not expect a Error! Reference source not found. Message and will generate a Soft Reset in response.

(d). Section 6.6.22 Time Values and Timers, Page 235, Table 6-68

From Text:

Page 293

<i>tEnterUSBWait</i>	100			ms	<i>Section 6.6.4.5</i>
-----------------------------	-----	--	--	----	-------------------------------

Page 294

<i>tVCONNSwapWait</i>	100			ms	<i>Section 6.6.4.4</i>
------------------------------	-----	--	--	----	-------------------------------

To Text:

Page 293

<i>tEnterUSBWait</i>	100			ms	<i>Section 6.6.4.7</i>
-----------------------------	-----	--	--	----	-------------------------------

Page 294

<i>tVCONNSwapWait</i>	100			ms	<i>Section 6.6.4.4</i>
<i>tVCONNSwapDelayDFP</i>	100			ms	<i>Section 6.6.4.5</i>
<i>tVCONNSwapDelayUFP</i>	500			ms	<i>Section 6.6.4.6</i>

(e). 6.6.4 Wait Timers and Times, Page 274

From Text:

6.6.4.4 tVconnSwapWait

The time before the next ***VCONN_Swap*** Message, after a ***Wait*** Message has been received in response to a ***VCONN_Swap*** Message is a minimum of ***tVCONNSwapWait*** min (see ***6.3.12 “Wait Message”***). ***The Port Shall*** wait at least ***tVCONNSwapWait*** after receiving the ***EOP*** of a ***Wait*** Message sent in response to a ***VCONN_Swap*** Message, before sending a new ***VCONN_Swap*** Message.

6.6.4.5 tEnterUSBWait

The time before the next ***Enter_USB*** Message, after a ***Wait*** Message has been received in response to a ***Enter_USB*** Message is a minimum of ***tEnterUSBWait*** min (see ***6.3.12 “Wait Message”***). ***The DFP Shall*** wait at least ***tEnterUSBWait*** after receiving the ***EOP*** of a ***Wait*** Message sent in response to an ***Enter_USB*** Message, before sending a new ***Enter_USB*** Message.

To Text:

6.6.4.4 tVconnSwapWait

The time before the next ***VCONN_Swap*** Message, after a ***Wait*** Message has been received in response to a ***VCONN_Swap*** Message is a minimum of ***tVCONNSwapWait*** min (see ***6.3.12 “Wait Message”***). ***The Port Shall*** wait at least ***tVCONNSwapWait*** after receiving the ***EOP*** of a ***Wait*** Message sent in response to a ***VCONN_Swap*** Message, before sending a new ***VCONN_Swap*** Message.

USB Power Delivery ENGINEERING CHANGE NOTICE

6.6.4.5 tVconnSwapDelayDFP

The time delay for DFP after losing Vconn Source role due to an incoming Vconn Swap request from UFP and before sending the next VCONN_Swap Message. The DFP Shall wait at least tVCONNswapDelayDFP after sending the EOP of the GoodCRC Message in response to PS_RDY Message received at the end of the previous Vconn Swap sequence.

6.6.4.6 tVconnSwapDelayUFP

The time delay for UFP after losing Vconn Source role due to an incoming Vconn Swap request from DFP and before sending the next VCONN_Swap Message. The UFP Shall wait at least tVCONNswapDelayUFP after sending the EOP of the GoodCRC Message in response to PS_RDY Message received at the end of the previous Vconn Swap sequence.

6.6.4.7 tEnterUSBWait

The time before the next *Enter_USB* Message, after a *Wait* Message has been received in response to a *Enter_USB* Message is a minimum of *tEnterUSBWait* min (see 6.3.12 “*Wait Message*”). *The DFP Shall* wait at least *tEnterUSBWait* after receiving the *EOP* of a *Wait* Message sent in response to an *Enter_USB* Message, before sending a new *Enter_USB* Message.

(f). Table 8.14 “AMS: VCONN Swap”, Page 465

From Text:

Vconn Source Swap, initiated by non- Vconn Source (Wait)

To Text:

Vconn Source Swap (Wait)

(g). 8.3.2.11.2.3 VCONN Source Swap, initiated by non- VCONN Source (Wait), Page 654

From Text:

8.3.2.11.2.3 VCONN Source Swap, initiated by non- VCONN Source (Wait)

Figure 8-59 “Vconn Source Swap with Wait, initiated by non- Vconn Source” shows an example where the Port which is not initially supplying VCONN and requests a VCONN Swap which is delayed with a wait. During the process the Port Partners, keep their role as Source or Sink, maintain their operation as either a Source or a Sink (power remains constant) and don’t exchange the VCONN Source.

Figure 8-59 “VCONN Source Swap with Wait, initiated by non- VCONN Source”

Table 8.87 “Steps for Vconn Source Swap with Wait, Initiated by non- Vconn Source” below provides a detailed explanation of what happens at each labeled step in *Figure 8-59 “Vconn Source Swap with Wait, initiated by non- Vconn Source”* above.

Table 8.87 “Steps for VCONN Source Swap with Wait, Initiated by non- VCONN Source”

To Text:

USB Power Delivery ENGINEERING CHANGE NOTICE

8.3.2.11.2.3 VCONN Source Swap, ~~initiated by non-VCONN Source~~ (Wait)

Figure 8-59 “Vconn Source Swap with Wait, ~~initiated by non-Vconn Source~~” shows an example where the Port ~~which is not initially supplying VCONN and~~ requests a VCONN Swap which is delayed with a wait. During the process the Port Partners, keep their role as Source or Sink, maintain their operation as either a Source or a Sink (power remains constant) and don’t exchange the VCONN Source.

Figure 8-59 “VCONN Source Swap with Wait, ~~initiated by non-VCONN Source~~”

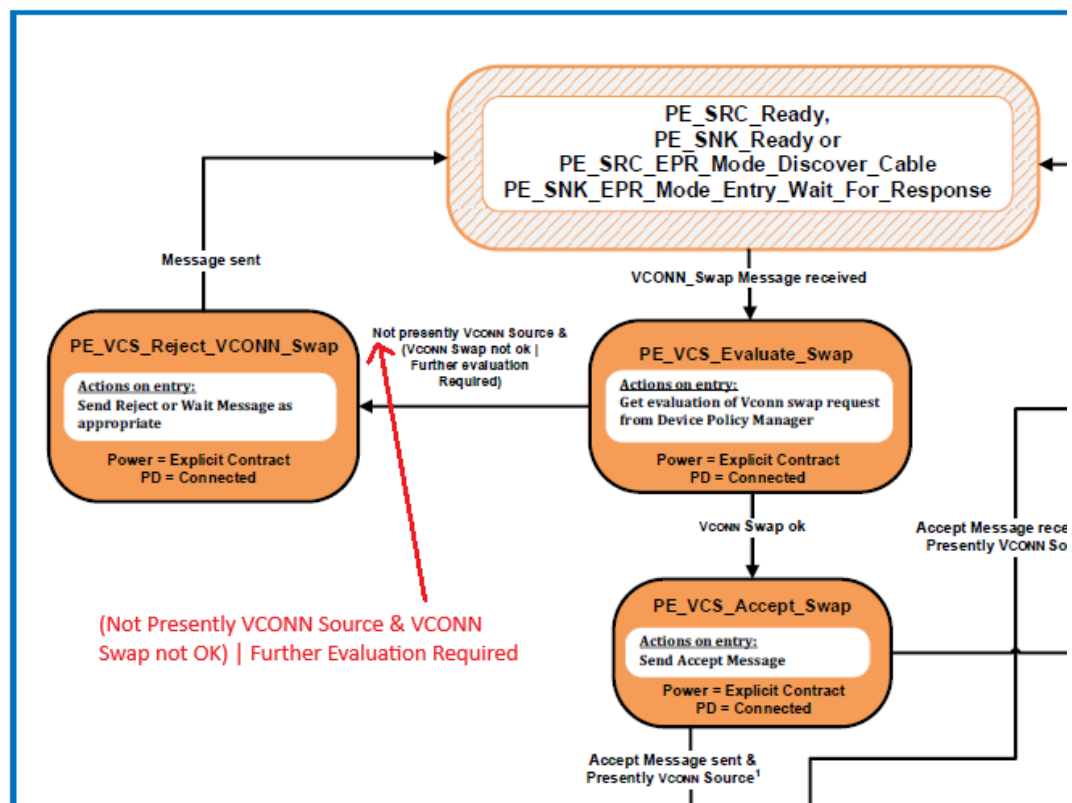
Table 8.87 “Steps for Vconn Source Swap with Wait, ~~Initiated by non-Vconn Source~~” below provides a detailed explanation of what happens at each labeled step in Figure 8-59 “Vconn Source Swap with Wait, ~~initiated by non-Vconn Source~~” above.

Table 8.87 “Steps for VCONN Source Swap with Wait, ~~Initiated by non-VCONN Source~~”

(g). 8.3.3.21 VCONN Swap State Diagram, Page 959

Figure 8-193 “VCONN Swap State Diagram”

Figure 8-193 “VCONN Swap State Diagram”



(h) 8.3.3.21.2 PE_VCS_Evaluate_Swap State, Page 960

From Text:

USB Power Delivery ENGINEERING CHANGE NOTICE

The **PE_VCS_Evaluate_Swap** state is entered from either the **PE_SRC_Ready** or **PE_SNK_Ready** state when the Policy Engine receives a **VCONN_Swap** Message.

On entry to the **PE_VCS_Evaluate_Swap** state the Policy Engine **Shall** request the Device Policy Manager for an evaluation of the VCONN Swap request. Note Ports that are presently the VCONN Source must always accept a VCONN swap request (see **Section 6.3.11 "VCONN_Swap Message"**).

The Policy Engine **Shall** transition to the **PE_VCS_Accept_Swap** state when:

- The Device Policy Manager indicates that a VCONN Swap is ok.

The Policy Engine **Shall** transition to the **PE_VCS_Reject_Swap** state when:

- The Port is not presently the VCONN Source and
- The Device Policy Manager indicates that a VCONN Swap is not ok or
- The Device Policy Manager indicates that a VCONN Swap cannot be done at this time.

To Text:

The **PE_VCS_Evaluate_Swap** state is entered from either the **PE_SRC_Ready** or **PE_SNK_Ready** state when the Policy Engine receives a **VCONN_Swap** Message.

On entry to the **PE_VCS_Evaluate_Swap** state the Policy Engine **Shall** request the Device Policy Manager for an evaluation of the VCONN Swap request. **Note Ports that are presently the VCONN Source must always accept a VCONN swap request (see Section 6.3.11 "VCONN_Swap Message").**

The Policy Engine **Shall** transition to the **PE_VCS_Accept_Swap** state when:

- The Device Policy Manager indicates that a VCONN Swap is ok.

The Policy Engine **Shall** transition to the **PE_VCS_Reject_Swap** state when:

- **The Port is not presently the VCONN Source and**
- **The Port is not presently the VCONN Source and** The Device Policy Manager indicates that a VCONN Swap is not ok or
- The Device Policy Manager indicates that a VCONN Swap cannot be done at this time.