USB Power Delivery ENGINEERING CHANGE NOTICE

Title: USB PD Wait Timing

Applied to: USB Power Delivery Specification Revision 2.0 Version 1.2

Brief description of the functional changes:

Clarify the usage of the Wait Message and ensure that there is a minimum time of 100ms after a Wait Message has been received before any Message is repeated.

Benefits as a result of the changes:

Ensures sufficient time for the Port which generated the Wait Message to generate other messaging such as getting capabilities in order to inform its decision making process.

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

Some systems are immediately resending messages if a wait is received. These systems conform to the letter but not the spirit of the specification and therefore cause interoperability issues in the eco-system. The intention would be to fail such systems in Compliance.

An analysis of the hardware implications:

None.

An analysis of the software implications:

Software would need to wait at least 100ms after having received a Wait Message.

An analysis of the compliance testing implications:

This change would result in additional testing in relation to Wait Messages.

Actual Change

(a). Section 6.3.12, Page 149

From Text:

6.3.12 Wait Message

The ***Wait*** Message is a valid response to a ***Request***, a ***PR_Swap***, ***DR_Swap*** or ***VCONN_Swap*** Message.

- It shall be sent to signal the Sink that the Source is unable to meet the request at this time.
- It shall be sent by the recipient of a ***PR_Swap*** Message to indicate it is unable to do a Power Role Swap at this time.
- It shall be sent by the recipient of a ***DR_Swap*** Message to indicate it is unable to do a Data Role Swap at this time.

The ***Wait*** Message shall be sent within ***tReceiverResponse*** of the receipt of the last bit of the Message (see Section 6.5.2).

6.3.12.1 Wait in response to a Request Message

The ***Wait*** Message is used by the Source when a Sink that has reserved power, requests it. The ***Wait*** Message allows the Source time to recover the power it requires to meet the request through the GotoMin process. A Source shall only send a ***Wait*** Message in response to a ***Request*** Message when an Explicit Contract exists between the Port Partners.

The Sink is allowed to repeat the ***Request*** Message using the ***SinkRequestTimer*** to ensure that there is ***tSinkRequest*** between requests.

6.3.12.2 Wait in response to a PR_Swap Message

The ***Wait*** Message is used when responding to a ***PR_Swap*** Message to indicate that a Power Role Swap might be possible in the future. This can occur in any case where the device receiving the ***PR_Swap*** Message needs to evaluate the request further e.g. by requesting Capabilities from the originator of the ***PR_Swap*** Message. Once it has completed this evaluation one of the Port Partners should initiate the Power Role Swap process again by sending a ***PR_Swap*** Message.

The ***Wait*** Message is also used where a Hub is operating in hybrid mode when a request cannot be satisfied (see ***[USBCBridge 1.0]*)**.

6.3.12.3 Wait in response to a DR_Swap Message

The ***Wait*** Message is used when responding to a ***DR_Swap*** Message to indicate that a Date Role Swap might be possible in the future. This can occur in any case where the device receiving the ***DR_Swap*** Message needs to evaluate the request further. Once it has completed this evaluation one of the Port Partners should initiate the Data Role Swap process again by sending a ***DR_Swap*** Message.

6.3.12.4 Wait in response to a VCONN_Swap Message

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. Sender of the ***VCONN_Swap*** Message should initiate the VCONN Swap process again at a future time by resending a ***VCONN_Swap*** Message.

To Text:

6.3.12 Wait Message

The ***Wait*** Message is a valid response to a ***Request***, a ***PR_Swap***, ***DR_Swap*** or ***VCONN_Swap*** Message.

- It shall be sent to signal the Sink that the Source is unable to meet the request at this time.
- It shall be sent by the recipient of a ***PR_Swap*** Message to indicate it is unable to do a Power Role Swap at this time.
- It shall be sent by the recipient of a ***DR_Swap*** Message to indicate it is unable to do a Data Role Swap at this time.
- It shall be sent by the recipient of a ***VCONN_Swap*** Message to indicate it is unable to do a VCONN Swap at this time.

The ***Wait*** Message shall be sent within ***tReceiverResponse*** of the receipt of the last bit of the Message (see Section 6.5.2).

6.3.12.1 Wait in response to a Request Message

The ***Wait*** Message is used by the Source when a Sink that has reserved power, requests it. The ***Wait*** Message allows the Source time to recover the power it requires to meet the request through the GotoMin process. A Source shall only send a ***Wait*** Message in response to a ***Request*** Message when an Explicit Contract exists between the Port Partners.

The Sink is allowed to repeat the ***Request*** Message using the ***SinkRequestTimer*** ~~to~~and shall ensure that there is ***tSinkRequest*** ~~between requests~~ after receiving the ***Wait*** Message before sending another ***Request*** Message.

6.3.12.2 Wait in response to a PR_Swap Message

The ***Wait*** Message is used when responding to a ***PR_Swap*** Message to indicate that a Power Role Swap might be possible in the future. This can occur in any case where the device receiving the ***PR_Swap*** Message needs to evaluate the request further e.g. by requesting Capabilities from the originator of the ***PR_Swap*** Message. Once it has completed this evaluation one of the Port Partners should initiate the Power Role Swap process again by sending a ***PR_Swap*** Message.

The ***Wait*** Message is also used where a Hub is operating in hybrid mode when a request cannot be satisfied (see ***[USBCBridge 1.0]***).

A Port that receives a ***Wait*** Message in response to a ***PR_Swap*** Message shall wait ***tPRSwapWait*** after receiving the ***Wait*** Message before sending another ***PR_Swap*** Message.

6.3.12.3 Wait in response to a DR_Swap Message

The ***Wait*** Message is used when responding to a ***DR_Swap*** Message to indicate that a Date Role Swap might be possible in the future. This can occur in any case where the device receiving the ***DR_Swap*** Message needs to evaluate the request further. Once it has completed this evaluation one of the Port Partners should initiate the Data Role Swap process again by sending a ***DR_Swap*** Message.

A Port that receives a ***Wait*** Message in response to a ***DR_Swap*** Message shall wait ***tDRSwapWait*** after receiving the ***Wait*** Message before sending another ***DR_Swap*** Message.

6.3.12.4 Wait in response to a VCONN_Swap Message

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. The sender of the ***VCONN_Swap*** Message should initiate the VCONN Swap process again at a future time by resending a ***VCONN_Swap*** Message.

A Port that receives a ***Wait*** Message in response to a ***VCONN_Swap*** Message shall wait ***tVCONNSwapWait*** after receiving the ***Wait*** Message before sending another ***VCONN_Swap*** Message.

(a). Section 6.5.5, Page 185

From Text:

6.5.5 SinkRequestTimer

The ***SinkRequestTimer*** is used to ensure that the time between Sink ***Request*** Messages, after a ***Wait*** Message has been received from the Source, is a minimum of ***tSinkRequest*** (see Section 6.3.12).

The ***SinkRequestTimer*** shall be started when a ***Wait*** Message has been received and shall be stopped if any other Message is received or during a Hard Reset.

The Sink shall wait at least ***tSinkRequest,*** after receiving a ***Wait*** Message, before issuing a new ***Request*** Message. Whenever there is a ***SinkRequestTimer*** timeout the Sink may send a ***Request*** Message. It shall then re-initialize and restart the ***SinkRequestTimer***.

To Text:

6.6.5 Wait Timers and Times

6.6.5.1 SinkRequestTimer

The ***SinkRequestTimer*** is used to ensure that the time ~~between~~ before the next Sink ***Request*** Message~~s~~, after a ***Wait*** Message has been received from the Source in response to a Sink ***Request*** Message, is a minimum of ***tSinkRequest*** min (see Section 6.3.12).

The ***SinkRequestTimer*** shall be started when the ***EOP*** of a ***Wait*** Message has been received and shall be stopped if any other Message is received or during a Hard Reset.

The Sink shall wait at least ***tSinkRequest,*** after receiving the ***EOP*** of a ***Wait*** Message sent in response to a Sink ***Request*** Message, before ~~issuing~~sending a new ***Request*** Message. Whenever there is a ***SinkRequestTimer*** timeout the Sink may send a ***Request*** Message. It shall then re-initialize and restart the ***SinkRequestTimer***.

6.6.5.2 tPRSwapWait

The time before the next ***PR_Swap*** Message, after a ***Wait*** Message has been received in response to a ***PR_Swap*** Message is a minimum of ***tPRSwapWait*** min (see Section 6.3.12). The Port shall wait at least ***tPRSwapWait*** after receiving the ***EOP*** of a ***Wait*** Message sent in response to a ***PR_Swap*** Message, before sending a new ***PR_Swap*** Message.

6.6.5.3 tDRSwapWait

The time before the next ***DR_Swap*** Message, after a ***Wait*** Message has been received in response to a ***DR_Swap*** Message is a minimum of ***tDRSwapWait*** min (see Section 6.3.12). The Port shall wait at least ***tDRSwapWait*** after receiving the ***EOP*** of a ***Wait*** Message sent in response to a ***DR_Swap*** Message, before sending a new ***DR_Swap*** Message.

6.6.5.4 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 Section 6.3.12). 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.

(a). Section 6.5.16, Page 190, Table 6-30

From Text:

Table 6-32 Time Values

Parameter	Value (min)	Value (max)	Units	Reference
tAttentionAverage		10	s	Section 6.5.16
tAttentionBurstSpacing	100		ms	Section 6.5.16
tAttentionSpacing	250		ms	Section 6.5.16
tBISTMode		300	ms	Section 6.5.8.1
tBISTContMode	30	60	ms	Section 6.5.8.4
tBISTReceive	1.0	1.2	ms	Section 6.5.8.5
tBISTResponse		15	ms	Section 6.5.8.2
tCableMessage	750		µs	Section 6.5.14
tDiscoverIdentity	40	50	ms	
tDRSwapHardReset		15	ms	Section 6.5.11.3
tFirstSourceCap		250	ms	
tHardReset		5	ms	Section 6.3.13
tHardResetComplete	4	5	ms	Section 6.5.10
tNoResponse	4.5	5.5	s	Section 6.5.7
tProtErrHardReset		15	ms	Section 6.5.11.4
tProtErrSoftReset		15	ms	Section 6.5.10.2
tPSHardReset	25	35	ms	Section 6.5.11.2
tPSSourceOff	750	920	ms	Section 6.5.6.2
tPSSourceOn	390	480	ms	Section 6.5.6.3
tPSTransition	450	550	ms	Section 6.5.6.1
tReceive	0.9	1.1	ms	Section 6.5.1
tReceiverResponse		15	ms	Section 6.5.2
tRetry		75	µs	Section 6.5.1
tSenderResponse	24	30	ms	Section 6.5.2
tSendSourceCap	1	2	s	Section 6.5.4.1
tSinkActivity	120	150	ms	Section 6.5.3.2
tSinkRequest	100		ms	Section 6.5.5.1
tSinkWaitCap	2.1	2.5	s	Section 6.5.4.2
tSoftReset		15	ms	Section 6.5.3.1, 6.7.1
tSourceActivity	40	50	ms	Section 6.5.3.1
tSwapSinkReady		15	ms	Section 6.5.9.2
tSwapSourceStart	20		ms	Section 6.5.9.2

Parameter	Value (min)	Value (max)	Units	Reference
tTransmit		195	µs	Section 6.5.1
tTypeCSendSourceCap	100	200	ms	Section 6.5.4.1
tTypeCSinkWaitCap	310	620	ms	Section 6.5.4.2
tVCONNSourceOff		25	ms	Section 6.5.13
tVCONNSourceOn		100	ms	Section 6.5.13
tVDMBusy	50		ms	Section 6.5.12.4
tVDMEnterMode		25	ms	Section 6.5.12.2
tVDMExitMode		25	ms	Section 6.5.12.3
tVDMReceiverResponse		15	ms	Section 6.5.12.1
tVDMSenderResponse	24	30	ms	Section 6.5.12.1
tVDMWaitModeEntry	40	50	ms	Section 6.5.12.2
tVDMWaitModeExit	40	50	ms	Section 6.5.12.3

To Text:

Table 6-32 Time Values

Parameter	Value (min)	Value (max)	Units	Reference
tAttentionAverage		10	s	Section 6.5.16
tAttentionBurstSpacing	100		ms	Section 6.5.16
tAttentionSpacing	250		ms	Section 6.5.16
tBISTMode		300	ms	Section 6.5.8.1
tBISTContMode	30	60	ms	Section 6.5.8.4
tBISTReceive	1.0	1.2	ms	Section 6.5.8.5
tBISTResponse		15	ms	Section 6.5.8.2
tCableMessage	750		µs	Section 6.5.14
tDiscoverIdentity	40	50	ms	Section 6.5.14
tDRSwapHardReset		15	ms	Section 6.5.11.3
tDRSwapWait	100		ms	Section 6.6.5.3
tFirstSourceCap		250	ms	Section 6.5.4.3
tHardReset		5	ms	Section 6.3.13
tHardResetComplete	4	5	ms	Section 6.5.10
tNoResponse	4.5	5.5	s	Section 6.5.7
tProtErrHardReset		15	ms	Section 6.5.11.4
tProtErrSoftReset		15	ms	Section 6.5.10.2
tPRSwapWait	100		ms	Section 6.6.5.2
tPSHardReset	25	35	ms	Section 6.5.11.2
tPSSourceOff	750	920	ms	Section 6.5.6.2

Parameter	Value (min)	Value (max)	Units	Reference
tPSSourceOn	390	480	ms	Section 6.5.6.3
tPSTransition	450	550	ms	Section 6.5.6.1
tReceive	0.9	1.1	ms	Section 6.5.1
tReceiverResponse		15	ms	Section 6.5.2
tRetry		75	µs	Section 6.5.1
tSenderResponse	24	30	ms	Section 6.5.2
tSendSourceCap	1	2	s	Section 6.5.4.1
tSinkActivity	120	150	ms	Section 6.5.3.2
tSinkRequest	100		ms	Section 6.5.5
tSinkWaitCap	2.1	2.5	s	Section 6.5.4.2
tSoftReset		15	ms	Section 6.5.3.1, 6.7.1
tSourceActivity	40	50	ms	Section 6.5.3.1
tSwapSinkReady		15	ms	Section 6.5.9.2
tSwapSourceStart	20		ms	Section 6.5.9.2
tTransmit		195	µs	Section 6.5.1
tTypeCSendSourceCap	100	200	ms	Section 6.5.4.1
tTypeCSinkWaitCap	310	620	ms	Section 6.5.4.2
tVCONNSourceOff		25	ms	Section 6.5.13
tVCONNSourceOn		100	ms	Section 6.5.13
tVCONNSwapWait	100		ms	Section 6.6.5.4
tVDMBusy	50		ms	Section 6.5.12.4
tVDMEnterMode		25	ms	Section 6.5.12.2
tVDMExitMode		25	ms	Section 6.5.12.3
tVDMReceiverResponse		15	ms	Section 6.5.12.1
tVDMSenderResponse	24	30	ms	Section 6.5.12.1
tVDMWaitModeEntry	40	50	ms	Section 6.5.12.2
tVDMWaitModeExit	40	50	ms	Section 6.5.12.3