

USB Power Delivery ENGINEERING CHANGE NOTIFICATION

Title: Timer Descriptions Corrections

Applied to: USB Power Delivery Specification Revision 3.1

Version 1.4

Brief description of the functional changes proposed:
Correct specification of timers based on the Policy Engine receiving a message. (Message is only seen by Policy Engine at end of following GoodCRC)

Benefits as a result of the proposed changes:
Timer text descriptions will match the state diagrams.

An assessment of the impact to the existing revision and systems that currently conform to the USB specification:
Should make no change as most likely actual implementations will already match the state diagrams.

An analysis of the hardware implications:
Should be none.

An analysis of the software implications:
Should be none.

An analysis of the compliance testing implications:
Minor. May require examination of timing analysis algorithms. Could be causing a maximum error of around 0.5ms.

USB Power Delivery ENGINEERING CHANGE NOTIFICATION

Description of the Problem

It was realised that the detailed text description of timers relying on the reception of a message were in some cases being incorrectly timed from the end of the message rather than the end of the following GoodCRC. This realisation is based on the fact that the Protocol Layer State diagrams make it clear that a received message is only passed up to the Policy Engine after the GoodCRC has been sent. So far (in previous ECRs) only the ReceiverResponseTimer has been corrected. (However, the correction somehow got missed from the last release, so is repeated here for completeness.)

USB Power Delivery ENGINEERING CHANGE NOTIFICATION

Actual Change Requested

(a) Section 6.6.2 SenderResponseTimer, p231

From Text:

The *SenderResponseTimer* *Shall* be started from the time the last bit of the *GoodCRC* Message *EOP* (i.e. the *GoodCRC* Message corresponding to the Message requesting a response) has been received by the Physical Layer.

The *SenderResponseTimer* *Shall* be stopped when the last bit of the expected response Message *EOP* has been received by the Physical Layer.

To Text:

The *SenderResponseTimer* *Shall* be started from the time the last bit of the *GoodCRC* Message *EOP*, (i.e. the *GoodCRC* Message corresponding to the Message requesting a response), has been received by the Physical Layer.

The *SenderResponseTimer* *Shall* be stopped when the last bit of the *GoodCRC* Message *EOP*, corresponding to the expected response Message, *EOP* has been received-transmitted by the Physical Layer.

(b) Section 6.6.5.1 PSTransitionTimer, p234

From Text:

The *PSTransitionTimer* *Shall* be started when the last bit of an *Accept* or *GotoMin* Message *EOP* has been received by the Physical Layer. The *PSTransitionTimer* *Shall* be stopped when the last bit of the *PS_RDY* Message *EOP* has been received by the Physical Layer.

To Text:

The *PSTransitionTimer* *Shall* be started when the last bit of the *GoodCRC* Message *EOP* corresponding to an *Accept* or *GotoMin* Message *EOP* has been received-transmitted by the Physical Layer. The *PSTransitionTimer* *Shall* be stopped when the last bit of the *GoodCRC* Message *EOP*, corresponding to the *PS_RDY* Message, *EOP* has been received-transmitted by the Physical Layer.

(c) Section 6.6.5.2.1, (PSSourceOffTimer) Use during Power Role Swap, p233

From Text:

USB Power Delivery ENGINEERING CHANGE NOTIFICATION

If a **PR_Swap** Message request has been sent by the Dual-Role Power Device currently acting as a Source the Sink can respond with an **Accept** Message. When the last bit of the **EOP** of the **GoodCRC** Message corresponding to this **Accept** Message is received by the Sink, then the **PSSourceOffTimer** *Shall* be started.

If a **PR_Swap** Message request has been sent by the Dual-Role Power Device currently acting as a Sink the Source can respond with an **Accept** Message. When the last bit of the **EOP** of this **Accept** Message is received by the Sink then the **PSSourceOffTimer** *Shall* be started.

The **PSSourceOffTimer** *Shall* be stopped when:

- The last bit of the **EOP** of the **PS_RDY** Message is received.

To Text:

If a **PR_Swap** Message request has been sent by the Dual-Role Power Device currently acting as a Source the Sink can respond with an **Accept** Message. When the last bit of the **GoodCRC** Message **EOP**, corresponding to this **transmitted Accept** Message, is received by the Sink's **Physical Layer**, then the **PSSourceOffTimer** *Shall* be started.

If a **PR_Swap** Message request has been sent by the Dual-Role Power Device currently acting as a Sink the Source can respond with an **Accept** Message. When the last bit of the **GoodCRC** Message **EOP**, corresponding to **EOP** of this **received Accept** Message, is **transmitted** by the Sink's **Physical Layer** then the **PSSourceOffTimer** *Shall* be started.

The **PSSourceOffTimer** *Shall* be stopped when:

- The last bit of the **GoodCRC** Message **EOP**, corresponding to the received **PS_RDY** Message, is **transmitted** by the **Physical Layer**.

(d) Section 6.6.5.2.2, (PSSourceOffTimer) Use during Fast Role Swap, p233

From Text:

When the **FR_Swap** Message request has been sent by the initial Sink, the initial Source *Shall* respond with an **Accept** Message. When the last bit of the **EOP** of the **GoodCRC** Message corresponding to this **Accept** Message is received by the initial Sink, then the **PSSourceOffTimer** *Shall* be started.

The **PSSourceOffTimer** *Shall* be stopped when:

- The last bit of the **EOP** of the **PS_RDY** Message is received.

To Text:

When the **FR_Swap** Message request has been sent by the initial Sink, the initial Source *Shall* respond with an **Accept** Message. When the last bit of the **GoodCRC** Message **EOP**, corresponding to this **Accept** Message, is received by the initial Sink's **Physical Layer**, then the **PSSourceOffTimer** *Shall* be started.

The **PSSourceOffTimer** *Shall* be stopped when:

- The last bit of the **GoodCRC** Message **EOP**, corresponding to the received **PS_RDY** Message, is **transmitted** by the **Physical Layer**.

USB Power Delivery ENGINEERING CHANGE NOTIFICATION

(e) Section 6.6.5.3.1 (PSSourceOnTimer) Use during Power Role Swap, p234

From Text:

The *PSSourceOnTimer* *Shall* be started when:

- The last bit of the *EOP* of the *GoodCRC* Message, corresponding to the transmitted *PS_RDY* Message, is received by the Physical Layer.

The *PSSourceOnTimer* *Shall* be stopped when:

- The last bit of the *EOP* of the *PS_RDY* Message is received by the Physical Layer.

To Text:

The *PSSourceOnTimer* *Shall* be started when:

- The last bit of the *GoodCRC* Message *EOP*, corresponding to the transmitted *PS_RDY* Message, is received by the Physical Layer.

The *PSSourceOnTimer* *Shall* be stopped when:

- The last bit of the *GoodCRC* Message *EOP*, corresponding to the received *PS_RDY* Message, is transmitted by the Physical Layer.

(f) Section 6.6.5.3.2 (PSSourceOnTimer) Use during Fast Role Swap, p234

From Text:

The *PSSourceOnTimer* *Shall* be started when:

- The last bit of the *EOP* of the *GoodCRC* Message, corresponding to the transmitted *PS_RDY* Message, is received by the Physical Layer.

The *PSSourceOnTimer* *Shall* be stopped when:

- The last bit of the *EOP* of the *PS_RDY* Message is received by the Physical Layer.

To Text:

The *PSSourceOnTimer* *Shall* be started when:

- The last bit of the *GoodCRC* Message *EOP*, corresponding to the transmitted *PS_RDY* Message, is received by the Physical Layer.

The *PSSourceOnTimer* *Shall* be stopped when:

- The last bit of the *GoodCRC* Message *EOP*, corresponding to the received *PS_RDY* Message, is transmitted by the Physical Layer.

USB Power Delivery ENGINEERING CHANGE NOTIFICATION

(g) Section 6.6.7.1 tBISTCarrierMode, p235

From Text:

A UUT *Shall* enter BIST Carrier Mode within *tBISTCarrierMode* of the last bit of the *EOP* of the *BIST* Message used to initiate the test is received by the Physical Layer. In *BIST Carrier Mode* when transmitting a continuous carrier signal transmission *Shall* start as soon as the UUT enters BIST mode.

To Text:

A UUT *Shall* enter BIST Carrier Mode within *tBISTCarrierMode* of the last bit of the *GoodCRC* Message *EOP*, corresponding to the received *BIST* Message used to initiate the test, being transmitted by the Physical Layer. In *BIST Carrier Mode* when transmitting a continuous carrier signal transmission *Shall* start as soon as the UUT enters BIST mode.

(h) Section 6.6.7.3 tBISTSharedTestMode, p235

From Text:

A UUT *Shall* enter BIST Shared Capacity Test Mode and send a new *Source_Capabilities* Message from all Ports within the shared capacity group within *tBISTSharedTestMode* of the last bit of the *EOP* of the *BIST* Message used to initiate the test is received by the Physical Layer.

To Text:

A UUT *Shall* enter BIST Shared Capacity Test Mode and send a new *Source_Capabilities* Message from all Ports within the shared capacity group within *tBISTSharedTestMode* of the last bit of the *GoodCRC* Message *EOP*, corresponding to the *BIST* Message used to initiate the test, being transmitted by the Physical Layer.

(i) Section 6.6.10.2 tDataReset, p236

From Text:

The DFP *Shall* complete the Data_Reset process (as defined in Section 6.3.14) within *tDataReset* of either:

- The last bit of the *GoodCRC* acknowledging the *Accept* Message when the DFP sent the *Data_Reset* Message.
- The last bit of the *Accept* Message when the UFP sent the *Data_Reset* Message.

USB Power Delivery ENGINEERING CHANGE NOTIFICATION

To Text:

The DFP *shall* complete the Data_Reset process (as defined in Section 6.3.14) within *tDataReset* of the last bit of the *GoodCRC* Message *EOP*, corresponding to the *Accept* Message, being transmitted by the Physical Layer.

(j) Section 6.6.12.1 VDMResponseTimer, p237

From Text:

The *VDMResponseTimer* *shall* be started from the time the last bit of the *GoodCRC* Message *EOP* (i.e. the *GoodCRC* Message corresponding to the VDM Command requesting a response) has been received by the Physical Layer. The *VDMResponseTimer* *shall* be stopped when the last bit of the *GoodCRC* Message *EOP*, corresponding to the expected VDM Command response, has been transmitted by the Physical Layer.

The *VDMResponseTimer* *shall* be started from the time the last bit of the *GoodCRC* Message *EOP* (i.e. the *GoodCRC* Message corresponding to the VDM Command requesting a response) has been received by the Physical Layer. The *VDMResponseTimer* *shall* be stopped when the last bit of the expected VDM Command response *EOP* has been received by the Physical Layer.

To Text:

The *VDMResponseTimer* *shall* be started from the time the last bit of the *GoodCRC* Message *EOP* (i.e. the *GoodCRC* Message, corresponding to the VDM Command requesting a response) has been received by the Physical Layer. The *VDMResponseTimer* *shall* be stopped when the last bit of the *GoodCRC* Message *EOP*, corresponding to the expected VDM Command response, has been transmitted by the Physical Layer.

The *VDMResponseTimer* *shall* be started from the time the last bit of the *GoodCRC* Message *EOP* (i.e. the *GoodCRC* Message corresponding to the VDM Command requesting a response) has been received by the Physical Layer. The *VDMResponseTimer* *shall* be stopped when the last bit of the *GoodCRC* Message *EOP*, corresponding to the expected VDM Command response, *EOP* has been transmitted by the Physical Layer.

(k) Section 6.6.12.2 VDMModeEntryTimer, p237

From Text:

The *VDMModeEntryTimer* *shall* be started from the time the last bit of the *GoodCRC* Message *EOP* (i.e. the *GoodCRC* Message corresponding to the VDM Command request) has been received by the Physical Layer. The *VDMModeEntryTimer* *shall* be stopped when the last bit of the expected Structured VDM Command response (ACK, NAK or BUSY) *EOP* has been received by the Physical Layer.

USB Power Delivery ENGINEERING CHANGE NOTIFICATION

To Text:

The *VDMModeEntryTimer* *Shall* be started from the time the last bit of the *GoodCRC* Message *EOP* (i.e. the *GoodCRC* Message corresponding to the VDM Command request), has been received by the Physical Layer. The *VDMModeEntryTimer* *Shall* be stopped when the last bit of the *GoodCRC* Message *EOP*, corresponding to the expected Structured VDM Command response (ACK, NAK or BUSY), *EOP* has been transmitted by the Physical Layer.

(l) Section 6.6.12.3 VDMModeExitTimer, p237

From Text:

The *VDMModeExitTimer* *Shall* be started from the time the last bit of the *GoodCRC* Message *EOP* (i.e. the *GoodCRC* Message corresponding to the VDM Command requesting a response) has been received by the Physical Layer. The *VDMModeExitTimer* *Shall* be stopped when the last bit of the expected Structured VDM Command response ACK *EOP* has been received by the Physical Layer.

To Text:

The *VDMModeExitTimer* *Shall* be started from the time the last bit of the *GoodCRC* Message *EOP* (i.e. the *GoodCRC* Message corresponding to the VDM Command requesting a response) has been received by the Physical Layer. The *VDMModeExitTimer* *Shall* be stopped when the last bit of the *GoodCRC* Message *EOP*, corresponding to the expected Structured VDM Command response ACK, *EOP* has been transmitted by the Physical Layer.

(m) Section 6.6.13.1 VCONNOnTimer, p238

From Text:

The *VCONNOnTimer* is used during a VCONN Swap.

The *VCONNOnTimer* *Shall* be started when:

- The last bit of the *EOP* of the *Accept* Message is received.
- The last bit of the *EOP* of *GoodCRC* Message corresponding to the *Accept* Message is received.

The *VCONNOnTimer* *Shall* be stopped when:

- The last bit of the *EOP* of the *PS_RDY* Message is received.

Prior to sending the *PS_RDY* Message, the Port *Shall* have turned VCONN On.

To Text:

The *VCONNOnTimer* is used during a VCONN Swap.

The *VCONNOnTimer* *Shall* be started when:

• ~~The last bit of the *EOP* of the *Accept* Message is received.~~

- The last bit of the *GoodCRC* Message *EOP*, corresponding to the *Accept* Message, is transmitted or received by the Physical Layer.

The *VCONNOnTimer* *Shall* be stopped when:

USB Power Delivery ENGINEERING CHANGE NOTIFICATION

- The last bit of the **GoodCRC Message EOP, corresponding to the received PS_RDY Message** is **transmitted by the Physical Layer**.

Prior to sending the **PS_RDY** Message, the Port *Shall* have turned VCONN On.

(n) Section 6.6.13.2 tVCONNSourceOff, p238

From Text:

The **tVCONNSourceOff** time applies during a VCONN Swap. The initial VCONN Source *Shall* cease sourcing VCONN within **tVCONNSourceOff** of receipt of the last bit of the **EOP** of the **PS_RDY** Message.

To Text:

The **tVCONNSourceOff** time applies during a VCONN Swap. The initial VCONN Source *Shall* cease sourcing VCONN within **tVCONNSourceOff** of the last bit of the **GoodCRC Message EOP, corresponding to the PS_RDY Message**, **being transmitted by the Physical Layer**.

(o) Section 6.6.17.2 tFRSwapComplete, p239

From Text:

During a fast-role swap, the initial Sink *Shall* respond with a the **PS_RDY** Message within **tFRSwapComplete** after it has received the **PS_RDY** Message from the Initial Source. The **tFRSwapComplete** time Shall be measured from the time the last bit of the **PS_RDY** Message **EOP** has been received by the Physical Layer until the first bit of the response **PS_RD** Message Preamble has been transmitted by the Physical Layer.

To Text:

During a fast-role swap, the initial Sink *Shall* respond with a the **PS_RDY** Message within **tFRSwapComplete** after it has received the **PS_RDY** Message from the Initial Source. The **tFRSwapComplete** time Shall be measured from the time the last bit of the **GoodCRC Message EOP, corresponding to the PS_RDY Message** has been **transmitted** by the Physical Layer until the first bit of the response **PS_RD PS_RDY** Message Preamble has been transmitted by the Physical Layer.

(p) Section 6.6.18.1 ChunkingNotSupportedTimer, p239

From Text:

The **ChunkingNotSupportedTimer** *Shall* be started when:

- The last bit of the **EOP** of a Message Chunk of a multi-chunk Message is received. The Policy Engine *Shall Not* send its **Not_Supported** Message before the **ChunkingNotSupportedTimer** expires.

To Text:

The **ChunkingNotSupportedTimer** *Shall* be started when:

USB Power Delivery ENGINEERING CHANGE NOTIFICATION

- The last bit of the **GoodCRC Message EOP, corresponding to the** Message Chunk of a multi-chunk Message, is **transmitted by the Physical Layer**. The Policy Engine **Shall Not** send its **Not_Supported** Message before the **ChunkingNotSupportedTimer** expires.

(q) Section 6.6.18.2 ChunkSenderRequestTimer, p239

From Text:

The **ChunkSenderRequestTimer** **Shall** be stopped when:

- The last bit of the **EOP** of the Chunk Request Message is received.
- A Message other than a Chunk Request is received from the Protocol Layer Rx.

To Text:

The **ChunkSenderRequestTimer** **Shall** be stopped when:

- The last bit of the **EOP** of the **EOP of the GoodCRC Message corresponding to the** Chunk Request Message is **transmitted by the Physical Layer**.
- A Message other than a Chunk Request is received from the Protocol Layer Rx.

(r) Section 6.6.18.3 ChunkSenderResponseTimer, p240

From Text:

6.6.18.2 ChunkSenderResponseTimer

The **ChunkSenderResponseTimer** **Shall** be stopped when:

- The last bit of the EOP of the Chunk Response Message is received.
- A Message other than a Chunk is received from the Protocol Layer.

To Text:

6.6.18.3 ChunkSenderResponseTimer

The **ChunkSenderResponseTimer** **Shall** be stopped when:

- The last bit of the **EOP of the GoodCRC Message corresponding to the** Chunk Response Message is **transmitted by the Physical Layer**.
- A Message other than a Chunk is received from the Protocol Layer.

(s) Section 6.6.19.1 SinkPPSPeriodicTimer, p240

From Text:

SinkPPSPeriodicTimer **Shall** be re-initialized and restarted when the last bit of the **EOP** of any Message is received that causes the Sink to enter the **PE_SNK_Ready** state.

The Sink **Shall** stop the **SinkPPSPeriodicTimer** when the last bit of the **EOP** of any Message or the last bit of any Signaling is received from the Source and by the Sink that causes the Sink to leave the **PE_SNK_Ready** state.

USB Power Delivery ENGINEERING CHANGE NOTIFICATION

To Text:

SinkPPSPeriodicTimer **Shall** be re-initialized and restarted **on transmission, by the Physical Layer, of** the last bit of the **GoodCRC** Message **EOP, corresponding to** any **received** Message, that causes the Sink to enter the **PE_SNK_Ready** state. The Sink **Shall** stop the *SinkPPSPeriodicTimer* **on the transmission of the EOP of the GoodCRC Message corresponding to** any Message or the last bit of any Signaling is received from the Source and by the Sink that causes the Sink to leave the **PE_SNK_Ready** state.

(t) Section 6.6.19.2 SourcePPSCCommTimer, p240

From Text:

SourcePPSCCommTimer **Shall** be re-initialized and restarted when the last bit of the **EOP** of any Message is received that causes the Source to enter the **PE_SRC_Ready** state.
The Source **Shall** stop the *SourcePPSCCommTimer* when the last bit of the **EOP** of any Message or the last bit of any Signaling is received from the Sink by the Source that causes the Source to leave the **PE_SRC_Ready** state.

To Text:

SourcePPSCCommTimer **Shall** be re-initialized and restarted when, **after receiving any message that causes the Source to enter the PE_SRC_Ready state, the last bit of the corresponding GoodCRC Message EOP is transmitted by the Physical Layer.**

The Source **Shall** stop the *SourcePPSCCommTimer* when:

- **after receiving any message that causes the Source to leave the PE_SRC_Ready state, the last bit the corresponding GoodCRC Message EOP is transmitted by the Physical Layer or**
- **the last bit of any Signaling is received by the Physical Layer from the Sink by the Source that causes the Source to leave the PE_SRC_Ready state.**

(u) Section 6.6.21.1 SinkEPREnterTimer, p241

From Text:

The *SinkEPREnterTimer* is used to ensure the EPR Mode entry process completes within **tEnterEPR**. The Sink **Shall** start the timer when it sees the last bit of the **GoodCRC** Message in response to the **EPR_Mode** Message with the Action field set to 1 (“Enter”). The Sink **Shall** stop the timer when it receives the last bit of the **EPR_Mode** Message with the Action field set to 3 (“Enter Succeeded”). If the timer expires the Sink **Shall** send a **Soft_Reset** Message.

To Text:

The *SinkEPREnterTimer* is used to ensure the EPR Mode entry process completes within **tEnterEPR**. The Sink **Shall** start the timer when it sees the last bit of the **GoodCRC** Message in response to the **EPR_Mode** Message with the Action field set to 1 (“Enter”). The Sink **Shall** stop the timer when **the last bit of the GoodCRC Message EOP, corresponding to the received EPR_Mode Message with the Action field set to 3 (“Enter Succeeded”), has been transmitted by the Physical Layer.** If the timer expires the Sink **Shall** send a **Soft_Reset** Message.

USB Power Delivery ENGINEERING CHANGE NOTIFICATION

(v) Section 6.6.21.2 SinkEPRKeepAliveTimer, p241

From Text:

While operating in EPR mode, the Sink *Shall* stop the *SinkEPRKeepAliveTimer* timer whenever:

- Sends the last bit of the *GoodCRC* Message in response any Message from the Source.
- Sends the last bit of any Message sent to the Source.

To Text:

While operating in EPR mode, the Sink *Shall* stop the *SinkEPRKeepAliveTimer* timer whenever:

- the last bit of the *GoodCRC* Message *EOP*, in response any Message from the Source, is transmitted by the Physical Layer.
- The Physical Layer Receives the last bit of the *GoodCRC* Message *EOP* in response to any Message sent to the Source.
- ~~Sends the last bit of any Message sent to the Source.~~

(w) Section 6.6.21.3 SourceEPRKeepAliveTimer, p241

From Text:

While operating in EPR mode, the Source *Shall* stop the *SourceEPRKeepAliveTimer* timer whenever:

- Sends the last bit of the *GoodCRC* Message in response any Message from the Sink.
- Sends the last bit of any Message sent to the Sink.

To Text:

6.6.21.3 SourceEPRKeepAlive Timer

While operating in EPR mode, the Source *Shall* stop the *SourceEPRKeepAliveTimer* timer whenever:

- The last bit of the *GoodCRC* Message *EOP*, in response any Message from the Sink, is transmitted by the Physical Layer.
- The Physical Layer receives the last bit of the *GoodCRC* Message *EOP* in response to any Message sent to the Source.
- ~~Sends the last bit of any Message sent to the Source.~~