USB Power Delivery ENGINEERING CHANGE NOTICE

Title: Deprecate Interruptibility of AMSes
Applied to: USB Power Delivery Specification Revision 3.1 Version 1.7

Brief description of the functional changes proposed:
Make all AMSes non-interuptible

Benefits as a result of the proposed changes:
Over time this will simplify design of devices by removing a complication. The specification will also become simpler to read. Testing has seen very few devices, if any, actually interrupting any AMS. The CTS has been written incorrectly without any of the group members realizing this until last workshop. This shows that it is not a well understood part of the spec. Looking at the interruptible AMSes, none of them have more than two messages, one in each direction. The added delay from not being able to interrupt an AMS is less that 30ms and normally more like 10ms.

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

An analysis of the hardware implications:
None expected

An analysis of the software implications:
Will lead to simpler SW

An analysis of the compliance testing implications:
A few changes

USB Power Delivery ENGINEERING CHANGE NOTICE

Actual Change Requested

(a). Table 1-1 Terms and Abbreviations, p48

From Text:

Atomic Message Sequence (AMS)	A fixed sequence of Messages as defined in Section 8.3.2 typically starting and ending in one of the following states: ***PE_SRC_Ready***, ***PE_SNK_Ready*** or ***PE_CBL_Ready***. An AMS can be Interruptible or Non-interruptible.

To Text:

Atomic Message Sequence (AMS)	A fixed sequence of Messages as defined in Section 8.3.2 typically starting and ending in one of the following states: ***PE_SRC_Ready***, ***PE_SNK_Ready*** or ***PE_CBL_Ready***. An AMS ~~can be Interruptible or~~is Non-interruptible.

(b). Table 1-1 Terms and Abbreviations, p51

From Text:

Interruptible	An AMS that, on receiving a Protocol Error, returns to the appropriate ready state to process the incoming Message is said to be Interruptible. Every AMS is Interruptible until the first Message in the AMS has been sent (a ***GoodCRC*** Message has been received). An AMS of Vendor Messages is Interruptible during the entire sequence.

To Text:

~~Interruptible~~	~~An AMS that, on receiving a Protocol Error, returns to the appropriate ready state to process the incoming Message is said to be Interruptible. Every AMS is Interruptible until the first Message in the AMS has been sent (a ***GoodCRC*** Message has been received). An AMS of Vendor Messages is Interruptible during the entire sequence.~~

(c). Table 1-1 Terms and Abbreviations, p51

From Text:

Non-interruptible	An AMS that, on receiving a Protocol Error, generates either a Soft Reset or Hard Reset. Any power related AMS is non-interruptible once the first Message in the AMS has been sent (a ***GoodCRC*** Message has been received).

To Text:

Non-interruptible	~~An AMS that, on receiving a Protocol Error, generates either a Soft Reset or Hard Reset.~~ There cannot be any unexpected Messages during an AMS; it is therefore Non-interruptible. An ~~Any power related~~ AMS starts ~~is Non-interruptible~~ once the first Message in the AMS has been sent (a ***GoodCRC*** Message has been received). There cannot be any unexpected Messages during an AMS; it is therefore Non-interruptible. An AMS starts when the first Message in the AMS has been sent (i.e., a GoodCRC Message has been received acknowledging the Message). See Section 8.3.2.1.3.

Commented [RPM1]: Deprecate Interruptibility.

(d). Table 1-1 Terms and Abbreviations, p53

From Text:

Protocol Error	An unexpected Message during an Atomic Message Sequence. A Protocol Error during an Non-interruptible AMS will result in either a Soft Reset or a Hard Reset. A Protocol Error during an Interruptible AMS will result in a return to the appropriate ready state where the Message will be handled.

To Text:

Protocol Error	**An unexpected Message during an Atomic Message Sequence. A Protocol Error during an ~~Non-interruptible~~ AMS will result in either a Soft Reset or a Hard Reset. ~~A Protocol Error during an Interruptible AMS will result in a return to the appropriate ready state where the Message will be handled.~~**

(e). Table 1-1 Terms and Abbreviations, p53

From Text:

Re-run	**Start an Interruptible AMS again from the beginning after a Protocol Error.**

To Text:

~~**Re-run**~~	~~**Start an Interruptible AMS again from the beginning after a Protocol Error.**~~

(f). Section 2.5.3 SOP Communication, p61

From Text:

SOP Communication is used for Port-to-Port communication between the Source and the Sink. SOP Communication is recognized by both Port Partners but not by any intervening Cable Plugs. SOP Communication takes priority over other SOP* Communications since it is critical to complete power related operations as soon as possible. Message sequences relating to power are also allowed to interrupt other sequences to ensure that negotiation and control of power is given priority on the bus.

To Text:

SOP Communication is used for Port-to-Port communication between the Source and the Sink. SOP Communication is recognized by both Port Partners but not by any intervening Cable Plugs. SOP Communication takes priority over other SOP* Communications since it is critical to complete power related operations as soon as possible. ~~Message sequences relating to power are also allowed to interrupt other sequences to ensure that negotiation and control of power is given priority on the bus.~~

(g). Section 6.3.19 FR_Swap Message, p128

From Text:

Note: A Fast Role Swap is a "best effort" solution to a situation where a PDUSB Device has lost its external power. This process can occur at any time, even during a Non-interruptible AMS in which case error handling such as Hard Reset or ***[USB Type-C 2.2][USB Type-C 2.2]*** Error Recovery will be triggered.

To Text:

Note: A Fast Role Swap is a "best effort" solution to a situation where a PDUSB Device has lost its external power. This process can occur at any time, even during ~~an Non-interruptible~~ AMS in which case error handling such as Hard Reset or ***[USB Type-C 2.2][USB Type-C 2.2]*** Error Recovery will be triggered.

(h). Section 6.4.4 Vendor Defined Message, p151

From Text:

Both Unstructured and Structured VDMs ***Shall*** only be sent and received after an Explicit Contract has been established. The only exception to this is the ***Discover Identity*** Command which ***May*** be sent by Source when no Contract or an Implicit Contract (in place after a Power Role Swap or Fast Role Swap) is in place in order to discover Cable capabilities (see Section 8.3.3.24.3). A VDM Message sequence ***Shall Not*** interrupt any other PD Message Sequence. A VDM Message sequence ***Shall*** be interruptible by any other PD Message Sequence.

To Text:

Both Unstructured and Structured VDMs ***Shall*** only be sent and received after an Explicit Contract has been established. The only exception to this is the ***Discover Identity*** Command which ***May*** be sent by Source when no Contract or an Implicit Contract (in place after a Power Role Swap or Fast Role Swap) is in place in order to discover Cable capabilities (see Section 8.3.3.24.3). ~~A VDM Message sequence ***Shall Not*** interrupt any other PD Message Sequence.~~ ~~A VDM Message sequence ***Shall*** be interruptible by any other PD Message Sequence.~~

(i). Section 6.4.4.1 Unstructured VDM, p151

From Text:

- Unstructured VDMs ***Shall Not*** be initiated or responded to under any other circumstances.
- A Vendor Defined Message sequence ***Shall*** be interruptible e.g., due to the need for a power related AMS.

- Unstructured VDMs ***Shall*** only be used during Modal Operation in the context of an Active Mode i.e., only after the UFP has Ack'ed the ***Enter Mode*** Command can Unstructured VDMs be sent or received. The Active Mode and the associated Unstructured VDMs ***Shall*** use the same SVID.

To Text:

- Unstructured VDMs ***Shall Not*** be initiated or responded to under any other circumstances.
- ~~A Vendor Defined Message sequence ***Shall*** be interruptible e.g., due to the need for a power related AMS.~~
- Unstructured VDMs ***Shall*** only be used during Modal Operation in the context of an Active Mode i.e., only after the UFP has Ack'ed the ***Enter Mode*** Command can Unstructured VDMs be sent or received. The Active Mode and the associated Unstructured VDMs ***Shall*** use the same SVID.

(j). Section 6.4.4.2 Structured VDM, p152

From Text:

- A Structured VDM Command sequence ***Shall*** be interruptible e.g., due to the need for a power related AMS.

To Text:

- ~~A Structured VDM Command sequence ***Shall*** be interruptible e.g., due to the need for a power related AMS.~~

(k). Section 6.4.4.3.4 Enter Mode Command, p173

From Text:

Figure 6-21 shows the sequence of events during the transition between USB operation and entering a Mode. It illustrates when the Responder's Mode changes and when the Initiator's Mode changes. Figure 6-22 shows a sequence that is Interrupted by a ***Source_Capabilities*** Message, that completes a Contract Negotiation, and then the sequence is Re-run. Figure 6-23 illustrates that when the Responder returns a NAK the transition to a Mode do not take place and the Responder and Initiator remain in their default USB roles.

To Text:

Figure 6-21 shows the sequence of events during the transition between USB operation and entering a Mode. It illustrates when the Responder's Mode changes and when the Initiator's Mode changes.~~ Figure 6-22 shows a sequence that is Interrupted by a ***Source_Capabilities*** Message, that completes a Contract Negotiation, and then the sequence is Re-run.~~ Figure 6-23 illustrates that when the Responder returns a NAK the transition to a Mode do not take place and the Responder and Initiator remain in their default USB roles.

(l). Figure 6-22 Enter Mode sequence Interrupted by Source Capabilities and then Re-run, p174

To Text:

Remove figure.

(m). Section 6.4.4.5 VDM Message Timing and Normal PD Messages, p177

From Text:

Any Command Process or other VDM sequence ***May*** be interrupted by any other USB PD Message. The Vendor or Standards defined state operation ***Shall*** comprehend this and continue to operate as expected when processing any other USB PD Messages.

To Text:

~~Any Command Process or other VDM sequence ***May*** be interrupted by any other USB PD Message. The Vendor or Standards defined state operation ***Shall*** comprehend this and continue to operate as expected when processing any other USB PD Messages.~~

(n). Section 6.4.4.5 VDM Message Timing and Normal PD Messages, p177

From Text:

VDM sequences ***Shall*** be interruptible after the return of a ***GoodCRC*** Message has been completed. In the case where there is an error in transmission of the ***Vendor_Defined*** Message, as for any other PD Message, the ***Vendor_Defined*** Message will not be retried, but instead the incoming Message will be processed by the Policy Engine. This means that the ***Vendor_Defined*** Message sequence will need to be Re-run after the USB PD Message sequence has completed.

To Text:

~~VDM sequences ***Shall*** be interruptible after the return of a ***GoodCRC*** Message has been completed. In the case where there is an error in transmission of the ***Vendor_Defined*** Message, as for any other PD Message, the ***Vendor_Defined*** Message will not be retried, but instead the incoming Message will be processed by the Policy Engine. This means that the ***Vendor_Defined*** Message sequence will need to be Re-run after the USB PD Message sequence has completed.~~

(o). Section 6.5.16 Vendor_Defined_Extended Message, p215

USB Power Delivery ENGINEERING CHANGE NOTICE

From Text:

A VDEM Message sequence ***Shall Not*** interrupt any other PD Message Sequence. A VDEM Message sequence ***Shall*** be interruptible by any other PD Message Sequence.

To Text:

A VDEM Message sequence ***Shall Not*** interrupt any other PD Message Sequence. ~~A VDEM Message sequence ***Shall*** be interruptible by any other PD Message Sequence.~~

(p). Section 6.5.16 Vendor_Defined_Extended Message, p216

From Text:

- A Vendor Defined Extended Message sequence ***Shall*** be interruptible e.g., due to the need for a power related AMS.

To Text:

- ~~A Vendor Defined Extended Message sequence ***Shall*** be interruptible e.g., due to the need for a power related AMS.~~

(q). Section 6.8.1 Soft Reset and Protocol Error, p234

From Text:

Protocol Errors are any unexpected Message during an AMS. If the first Message in an AMS has been passed to the Protocol Layer by the Policy Engine but has not yet been sent (***GoodCRC*** Message not received) when the Protocol Error occurs, the Policy Engine ***Shall Not*** issue a Soft Reset but ***Shall*** return to the ***PE_SNK_Ready*** or ***PE_SRC_Ready*** state and then process the incoming Message. If the Protocol Error occurs during an Interruptible AMS then the Policy Engine ***Shall Not*** issue a Soft Reset but ***Shall*** return to the ***PE_SNK_Ready*** or ***PE_SRC_Ready*** state and then process the incoming Message. If the incoming Message is an Unexpected Message received in the ***PE_SNK_Ready*** or ***PE_SRC_Ready*** state, the Policy Engine ***Shall*** issue a Soft Reset. If the Protocol Error occurs during an Non- interruptible AMS this ***Shall*** lead to a Soft Reset in order to re-synchronize the Policy Engine state machines (see Section 8.3.3.4) except when the Voltage is transition when a Protocol Error ***Shall*** lead to a Hard Reset (see Section 6.6.11.4 and Section 8.3.3.2). Details of Interruptible and Non-interruptible AMS's can be found in Section 8.3.2.1.3.

To Text:

Protocol Errors are any unexpected Message during an AMS. If the first Message in an AMS has been passed to the Protocol Layer by the Policy Engine but has not yet been sent (i.e., a ***GoodCRC*** Message acknowledging the Message has not been received) when the Protocol Error occurs, the Policy Engine ***Shall Not*** issue a Soft Reset

but ***Shall*** return to the *PE_SNK_Ready* or *PE_SRC_Ready* state and then process the incoming Message. ~~If the Protocol Error occurs during an Interruptible AMS then the Policy Engine ***Shall Not*** issue a Soft Reset but ***Shall*** return to the *PE_SNK_Ready* or *PE_SRC_Ready* state and then process the incoming Message.~~ If the incoming Message is an Unexpected Message received in the *PE_SNK_Ready* or *PE_SRC_Ready* state, the Policy Engine ***Shall*** issue a Soft Reset. If the Protocol Error occurs during an ~~Non- interruptible~~ AMS this ***Shall*** lead to a Soft Reset in order to re-synchronize the Policy Engine state machines (see Section 8.3.3.4) except when the Voltage is transition when a Protocol Error ***Shall*** lead to a Hard Reset (see Section 6.6.11.4 and Section 8.3.3.2). Details of ~~Interruptible and Non-interruptible~~ AMS's can be found in Section 8.3.2.1.3.

(r). Table 6-72 Response to an incoming Message (except VDM), p235

From Text:

Recipient's Power Role	Recipient's state	Incoming Message			
		Recognized			Unrecognized
		Supported		Unsupported	
		Expected	Unexpected		
Source	PE_SRC_Ready	Process Message	*Soft_Reset* Message[2]	*Not_Supported* Message[3]	*Not_Supported* Message[3] (except for VDM) See 6.4.4.1 for UVDM, 6.4.4.2 for SVDM
	During Interruptible AMS (In Explicit Contract)	Process Message	return to *PE_SRC_Ready* state and process Message		
	During Interruptible AMS (Not in Explicit Contract)	Process Message	*Soft_Reset* Message[2]		
	During Non-interruptible AMS (power not transitioning[1])	Process Message	*Soft_Reset* Message[2]		
	During Non-interruptible AMS (power transitioning[1])	Process Message	*Hard Reset* Signaling		
Sink	PE_SNK_Ready	Process Message	*Soft_Reset* Message[2]	*Not_Supported* Message[3]	*Not_Supported* Message[3] (except for VDM) See 6.4.4.1 for UVDM, 6.4.4.2 for SVDM
	During Interruptible AMS (In Explicit Contract)	Process Message	return to *PE_SNK_Ready* state and process Message		
	During Interruptible AMS (Not in Explicit Contract)	Process Message	*Soft_Reset* Message[2]		

Recipient's Power Role	Recipient's state	Incoming Message			
		Recognized			Unrecognized
		Supported		Unsupported	
		Expected	Unexpected		
	During Non-interruptible AMS (not power transitioned)	Process Message	***Soft_Reset*** Message[2]		
	During Non-interruptible AMS (power transitioned)	Process Message	***Hard Reset*** Signaling		

1. *"power transitioning" means the policy engine is in PE_SRC_Transition_Supply State or PE_SNK_Transition_Sink State or PE_FRS_SNK_SRC_Send_Swap State.*
2. *The Soft_Reset Message **Shall** be sent using the SOP* of the incoming message.*
3. *The Not_Supported Message **Shall** be sent using the SOP* of the incoming message.*

To Text:

Recipient's Power Role	Recipient's state	Incoming Message			
		Recognized			Unrecognized
		Supported		Unsupported	
		Expected	Unexpected		
Source	PE_SRC_Ready	Process Message	***Soft_Reset*** Message[2]	***Not_Supported*** Message[3]	***Not_Supported*** Message[3] (except for VDM) See 6.4.4.1 for UVDM, 6.4.4.2 for SVDM
	~~During Interruptible AMS (In Explicit Contract)~~	~~Process Message~~	~~return to ***PE_SRC_Ready*** state and process Message~~		
	~~During Interruptible AMS (Not in Explicit Contract)~~	~~Process Message~~	~~***Soft_Reset*** Message[2]~~		
	During ~~Non-interruptible~~ AMS (power not transitioning[1])	Process Message	***Soft_Reset*** Message[2]		
	During ~~Non-interruptible~~ AMS (power transitioning[1])	Process Message	***Hard Reset*** Signaling		
Sink	PE_SNK_Ready	Process Message	***Soft_Reset*** Message[2]	***Not_Supported*** Message[3]	***Not_Supported*** Message[3] (except for VDM) See 6.4.4.1 for UVDM, 6.4.4.2 for SVDM
	~~During Interruptible AMS (In Explicit Contract)~~	~~Process Message~~	~~return to ***PE_SNK_Ready*** state and process Message~~		

Recipient's Power Role	Recipient's state	Incoming Message			
		Recognized			Unrecognized
		Supported		Unsupported	
		Expected	Unexpected		
	~~During Interruptible AMS (Not in Explicit Contract)~~	~~Process Message~~	~~***Soft_Reset*** Message[2]~~		
	During ~~Non-interruptible~~ AMS (not power transitioned)	Process Message	***Soft_Reset*** Message[2]		
	During ~~Non-interruptible~~ AMS (power transitioned)	Process Message	***Hard Reset*** Signaling		

1. *"power transitioning" means the policy engine is in PE_SRC_Transition_Supply State or PE_SNK_Transition_Sink State or PE_FRS_SNK_SRC_Send_Swap State.*
2. *The Soft_Reset Message **Shall** be sent using the SOP* of the incoming message.*
3. *The Not_Supported Message **Shall** be sent using the SOP* of the incoming message.*

(s). Section 8.3.2.1 Introduction, p366

From Text:

An AMS ***Shall*** be considered to have ended:

- When the Protocol Engine signals the Policy Engine that transmission of the final Message in the AMS is a success and for the opposite port when the final Message has been received.
- In the case of an Interrupted AMS, the ending of the AMS ***Shall*** be considered to be the start of the interrupting AMS by the port that initiates the interrupting AMS. For the opposing port the reception of the starting message marks the end of the interrupted AMS.
- A ***Soft_Reset*** Message, ***Hard Reset*** Signaling for SOP' or SOP'' or ***Cable Reset*** Signaling has been sent or received.

Section 8.3.2.1.3 gives details of which of these AMS's are interruptible or non-interruptible.

To Text:

An AMS ***Shall*** be considered to have ended:

- When the Protocol Engine signals the Policy Engine that transmission of the final Message in the AMS is a success and for the opposite port when the final Message has been received.
- ~~In the case of an Interrupted AMS, the ending of the AMS ***Shall*** be considered to be the start of the interrupting AMS by the port that initiates the interrupting AMS. For the opposing port the reception of the starting message marks the end of the interrupted AMS.~~
- A ***Soft_Reset*** Message, ***Hard Reset*** Signaling for SOP' or SOP'' or ***Cable Reset*** Signaling has been sent or received.

USB Power Delivery ENGINEERING CHANGE NOTICE

Section 8.3.2.1.3 gives details of ~~which of these~~ AMS's ~~are interruptible or non-interruptible~~.

(t). 8.3.2.1.3 Interruptible and Non-Interruptible Atomic Message Sequences, p370

From Text:

8.3.2.1.3 Interruptible and Non-Interruptible Atomic Message Sequences

Table 8-4 details which AMS (as defined in Section 8.3.2) ***Shall*** be treated as Interruptible or Non-interruptible during the sequence. Every AMS which starts with the same Message ***Shall*** obey the Interruptible/Non-interruptible requirement. Note that every AMS is Interruptible until the first Message in the sequence has been successfully sent (***GoodCRC*** Message received). Any Sequence of VDMs ***Shall*** be Interruptible. After the AMS that caused the interruption has completed, if the original AMS is still needed the interrupted AMS ***Shall*** be Re-run.

The Message Sequences in Table 8-4 list sequences of either Messages or combinations of Messages and one or more embedded AMSes. Where there is an embedded AMS the entire Message Sequence is treated as an AMS and the Rp value used for collision avoidance (see Section 5.7) ***Shall*** only be changed on leaving or entering the ready state at the beginning or end of the entire Message Sequence, and not at the start or end of the embedded AMS.

Table 8-5Table 8-5 details a Hard Reset (which is Signaling not an AMS) followed by an SPR Contract Negotiation AMS which ***Shall*** be treated as Non-Interruptible.

To Text:

8.3.2.1.3 ~~Interruptible and Non-Interruptible~~ Atomic Message Sequences

~~Table 8-4 details which AMS (as defined in Section 8.3.2) ***Shall*** be treated as Interruptible or Non-interruptible during the sequence. Every AMS which starts with the same Message ***Shall*** obey the Interruptible/Non-interruptible requirement.~~ Note that every AMS is interruptible until the first Message in the sequence has been successfully sent (***GoodCRC*** Message received)~~. Any Sequence of VDMs ***Shall*** be Interruptible. After the AMS that caused the interruption has completed, if the original AMS is still needed the interrupted AMS ***Shall*** be Re-run.~~

The Atomic Message Sequences (AMS) in Table 8-4 list sequences of either Messages or combinations of Messages and one or more embedded AMSes which are Non-Interruptible. Where there is an embedded AMS the entire Message Sequence is treated as an AMS and the Rp value used for collision avoidance (see Section 5.7) ***Shall*** only be changed on leaving or entering the ready state at the beginning or end of the entire Message Sequence, and not at the start or end of the embedded AMS. Note that an AMS has not started until the first Message in the sequence has been successfully sent (i.e., a ***GoodCRC*** Message has been received acknowledging the Message).

Table 8-5 details a Hard Reset (which is Signaling not an AMS) followed by an SPR Contract Negotiation AMS ~~which ***Shall*** be treated as Non-Interruptible~~.

USB Power Delivery ENGINEERING CHANGE NOTICE

(u). Table 8-4 Interruptible and Non-interruptible AMS, p371

From Text:

Table 8-4 Interruptible and Non-interruptible AMS

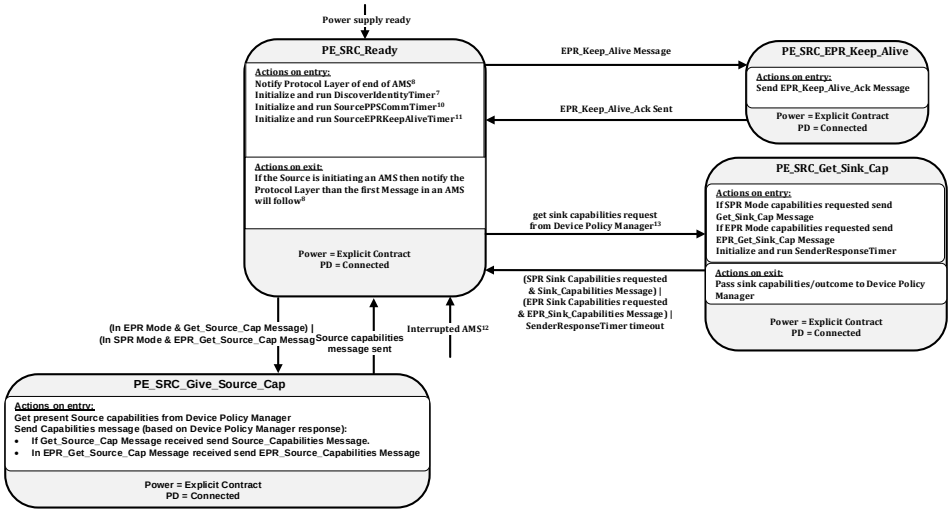
To Text:

Table 8-4 ~~Interruptible and Non-interruptible~~ Atomic Message Sequences

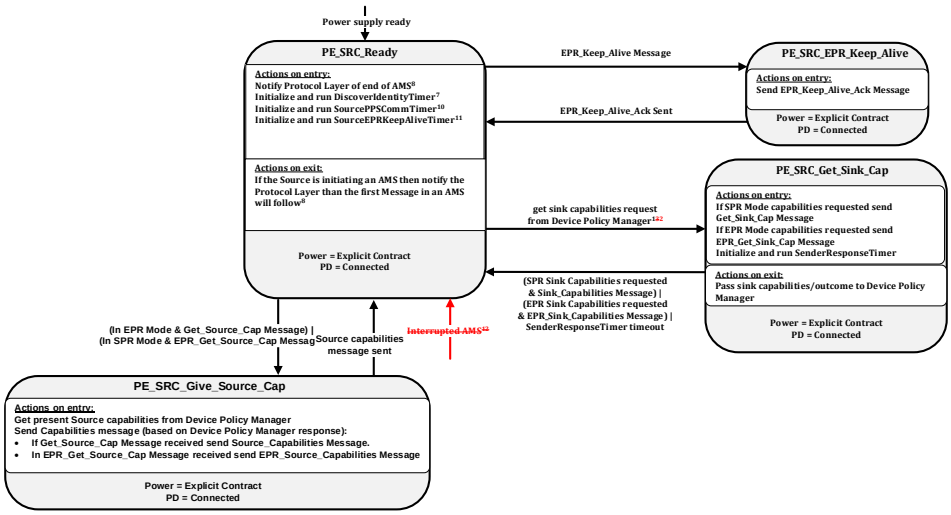
Remove column titled interruptibility.

(v) Figure 8-134 Source Port Policy Engine State Diagram, p680

From Text:



To Text:



(w) Figure 8-134 Source Port Policy Engine State Diagram, p681

From Text:

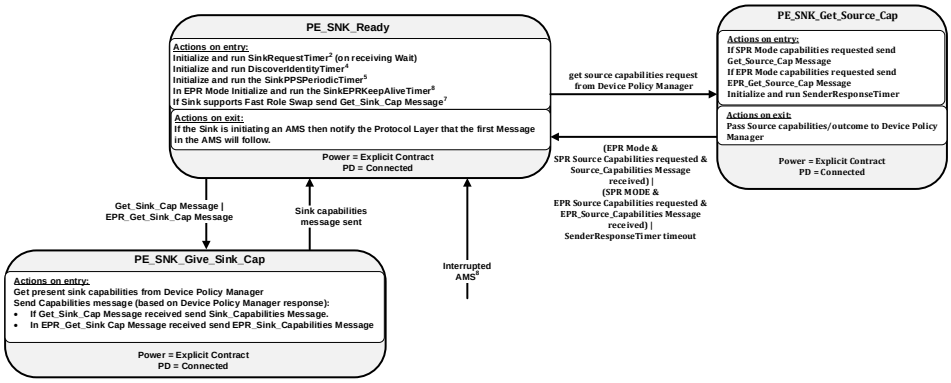
[12] Entry from any State where the start of an AMS was interrupted by an incoming Message.
[13] Either SPR or EPR Sink Capabilities ***May*** be requested, regardless of whether or not the Source is currently operating in SPR or EPR Mode.

To Text:

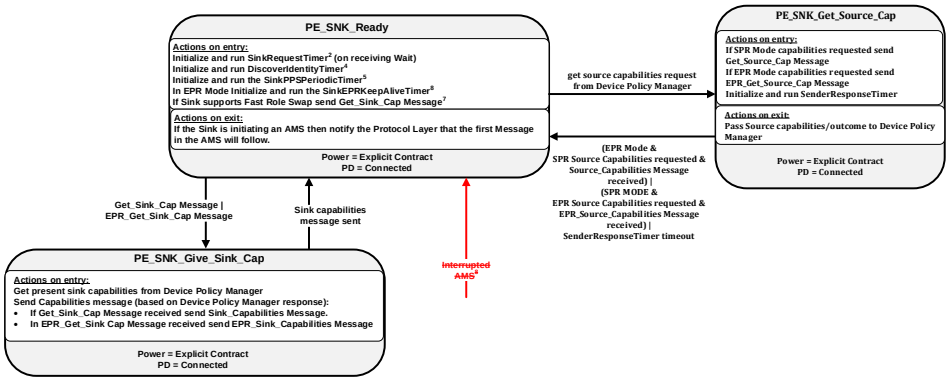
~~[12] Entry from any State where the start of an AMS was interrupted by an incoming Message.~~
[13] Either SPR or EPR Sink Capabilities ***May*** be requested, regardless of whether or not the Source is currently operating in SPR or EPR Mode.

(x) Figure 8-135 Sink Port Policy Engine State Diagram, p681

From Text:



To Text:



(y) Figure 8-135 Sink Port Policy Engine State Diagram, p681

From Text:

[8] Entry from any State where the start of an AMS was interrupted by an incoming Message.

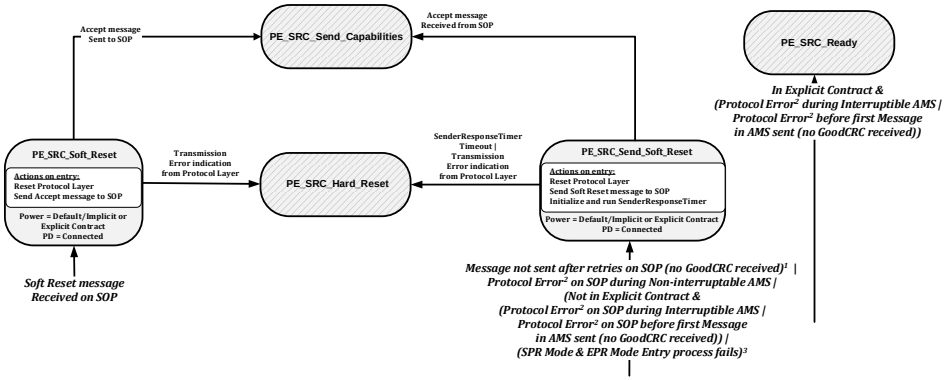
To Text:

~~[8] Entry from any State where the start of an AMS was interrupted by an incoming Message.~~

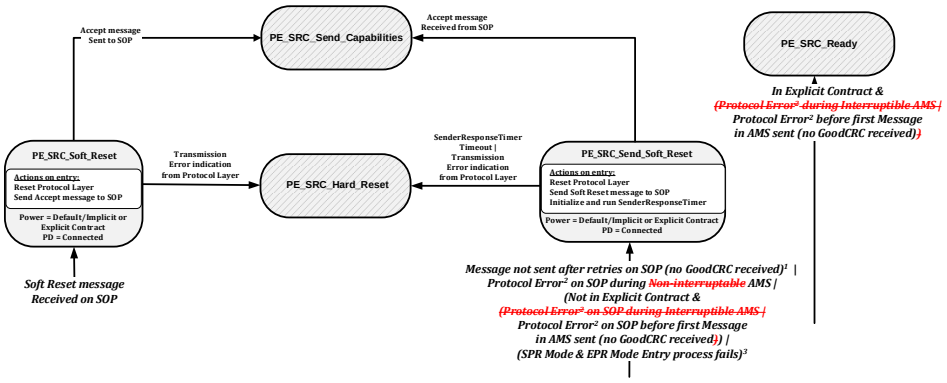
USB Power Delivery ENGINEERING CHANGE NOTICE

(z). Figure 8-136 SOP Source Port Soft Reset and Protocol Error State Diagram, p694

From Text:



To Text:



(aa). Section 8.3.3.4.1.1 PE_SRC_Send_Soft_Reset State, p694

From Text:

The *PE_SRC_Send_Soft_Reset* state ***Shall*** be entered from any state when:

- A Protocol Error on ***SOP*** is detected by the Protocol Layer during a Non-interruptible AMS (see Section 6.8.1) or
- A Message has not been sent after retries to the Sink or
- When not in an Explicit Contract and
  - Protocol Errors occurred on ***SOP*** during an Interruptible AMS or
  - Protocol Errors occurred on ***SOP*** during any AMS where the first Message in the sequence has not yet been sent i.e., an unexpected Message is received instead of the expected ***GoodCRC*** Message response.
- When in SPR Mode and the EPR Mode entry process fails.

To Text:

The *PE_SRC_Send_Soft_Reset* state ***Shall*** be entered from any state when:

- A Protocol Error on ***SOP*** is detected by the Protocol Layer during a ~~Non-interruptiblen~~ AMS (see Section 6.8.1) or
- A Message has not been sent after retries to the Sink or
- When not in an Explicit Contract and
  - ~~Protocol Errors occurred on ***SOP*** during an Interruptible AMS or~~
  - Protocol Errors occurred on ***SOP*** during any AMS where the first Message in the sequence has not yet been sent i.e., an unexpected Message is received instead of the expected ***GoodCRC*** Message response.
- When in SPR Mode and the EPR Mode entry process fails.

(ab). 8.3.3.4.1.1 PE_SRC_Send_Soft_Reset State, p695

From Text:

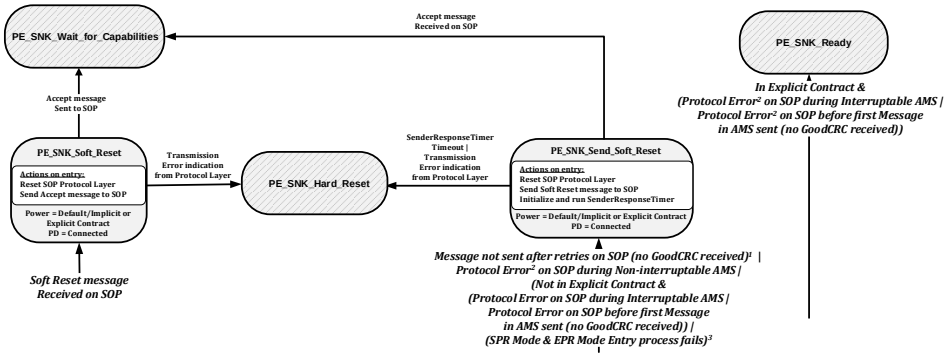
• When in an Explicit Contract
○ Protocol Errors occurred on ***SOP*** during an Interruptible AMS.
○ Protocol Errors occurred on ***SOP*** during any AMS where the first Message in the sequence has not yet been sent i.e., an unexpected Message is received instead of the expected ***GoodCRC*** Message response.

To Text:

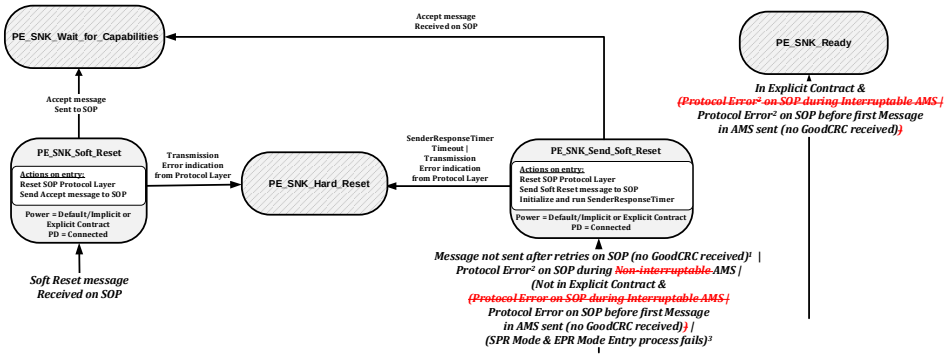
• When in an Explicit Contract and
~~○ Protocol Errors occurred on ***SOP*** during an Interruptible AMS.~~
○ Protocol Errors occurred on ***SOP*** during any AMS where the first Message in the sequence has not yet been sent i.e., an unexpected Message is received instead of the expected ***GoodCRC*** Message response.

(ac). Figure 8-137 Sink Port Soft Reset and Protocol Error Diagram, p696

From Text:



To Text:



(ad). 8.3.3.4.2.1 PE_SNK_Send_Soft_Reset State, p696

From Text:

The *PE_SNK_Send_Soft_Reset* state ***Shall*** be entered from any state when:

- A Protocol Error on ***SOP*** is detected by the Protocol Layer during a Non-interruptiblen AMS (see Section 6.8.1) or
- A Message has not been sent after retries to the Sink or
- When not in an Explicit Contract and
  - Protocol Errors occurred on ***SOP*** during an Interruptible AMS or
  - Protocol Errors occurred on ***SOP*** during any AMS where the first Message in the sequence has not yet been sent i.e., an unexpected Message is received instead of the expected ***GoodCRC*** Message response.
- When in SPR Mode and the EPR Mode entry process fails.

To Text:

The *PE_SNK_Send_Soft_Reset* state ***Shall*** be entered from any state when:

- A Protocol Error on ***SOP*** is detected by the Protocol Layer during a ~~Non-interruptible~~n AMS (see Section 6.8.1) or
- A Message has not been sent after retries to the Sink or
- When not in an Explicit Contract and
  - ~~Protocol Errors occurred on ***SOP*** during an Interruptible AMS or~~
    - Protocol Errors occurred on ***SOP*** during any AMS where the first Message in the sequence has not yet been sent i.e., an unexpected Message is received instead of the expected ***GoodCRC*** Message response.
- When in SPR Mode and the EPR Mode entry process fails.

(ae). 8.3.3.4.2.1 PE_SNK_Send_Soft_Reset State, p696

From Text:

• When in an Explicit Contract
○ Protocol Errors occurred on ***SOP*** during an Interruptible AMS.

To Text:

• When in an Explicit Contract and
~~○ Protocol Errors occurred on ***SOP*** during an Interruptible AMS.~~

(af). Section 8.3.3.6.1 Source Port Not Supported Message State Diagram, p703

From Text:

[1] Transition can either be the result of a Protocol Error during an interruptible AMS or as a result of an unsupported Message being received in the *PE_SRC_Ready* state directly (see also Section 8.3.3.4.1).

To Text:

[1] Transition ~~can either be the result of a Protocol Error during an interruptible AMS or~~ as a result of an unsupported Message being received in the *PE_SRC_Ready* state directly (see also Section 8.3.3.4.1).

(ag). Section 8.3.3.6.1.3 PE_SRC_Chunk_Received State, p703

From Text:

The *PE_SRC_Chunk_Received* state ***Shall*** be entered from the *PE_SRC_Ready* state either as the result of a Protocol Error received during an interruptible AMS or as a result of an unsupported Message being received in the *PE_SRC_Ready* state directly where the Message is a Chunk in a multi-Chunk Message (see also Section 6.6.18.1 and Section 8.3.3.4.1).

To Text:

The *PE_SRC_Chunk_Received* state ***Shall*** be entered from the *PE_SRC_Ready* state ~~either as the result of a Protocol Error received during an interruptible AMS or~~ as a result of an unsupported Message being received in the *PE_SRC_Ready* state directly where the Message is a Chunk in a multi-Chunk Message (see also Section 6.6.18.1 and Section 8.3.3.4.1).

(ah). Section 8.3.3.6.2 Sink Port Not Supported Message State Diagram, p704

From Text:

[1] Transition can either be the result of a Protocol Error during an interruptible AMS or as a result of an unsupported Message being received in the *PE_SNK_Ready* state directly (see also Section 8.3.3.4.2).

To Text:

[1] Transition ~~can either be the result of a Protocol Error during an interruptible AMS or~~ as a result of an unsupported Message being received in the *PE_SNK_Ready* state directly (see also Section 8.3.3.4.2).

(ag). Section 8.3.3.6.2.3 PE_SNK_Chunk_Received State, p704

From Text:

The *PE_SNK_Chunk_Received* state ***Shall*** be entered from the *PE_SNK_Ready* state either as the result of a Protocol Error received during an interruptible AMS or as a result of an unsupported Message being received in the *PE_SNK_Ready* state directly where the Message is a Chunk in a multi-Chunk Message (see also Section 6.6.18.1 and Section 8.3.3.4.1).

To Text:

The *PE_SNK_Chunk_Received* state ***Shall*** be entered from the *PE_SNK_Ready* state ~~either as the result of a Protocol Error received during an interruptible AMS or~~ as a result of an unsupported Message being received in the *PE_SNK_Ready* state directly where the Message is a Chunk in a multi-Chunk Message (see also Section 6.6.18.1 and Section 8.3.3.4.1).