USB Power Delivery ENGINEERING CHANGE NOTICE

Title: Source State Diagram

Applied to: USB Power Delivery Specification Revision 2.0 Version 1.2

Brief description of the functional changes:

Revision 3.0 Source state diagram which had been incorrectly integrated into Revision 2.0 and has been replaced with the correct version.

Updates to note numbering.

Additions and amendments to textual descriptions to match diagrams.

Benefits as a result of the changes:

Diagrams match operation as defined elsewhere in the specification.

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

Aligns specification with the requirements expected on publication of the specification.

An analysis of the hardware implications:

None.

An analysis of the software implications:

None.

An analysis of the compliance testing implications:

USB Power Delivery ENGINEERING CHANGE NOTICE

Ensures that there are no unexpected timers or transitions to be tested in the state diagrams.
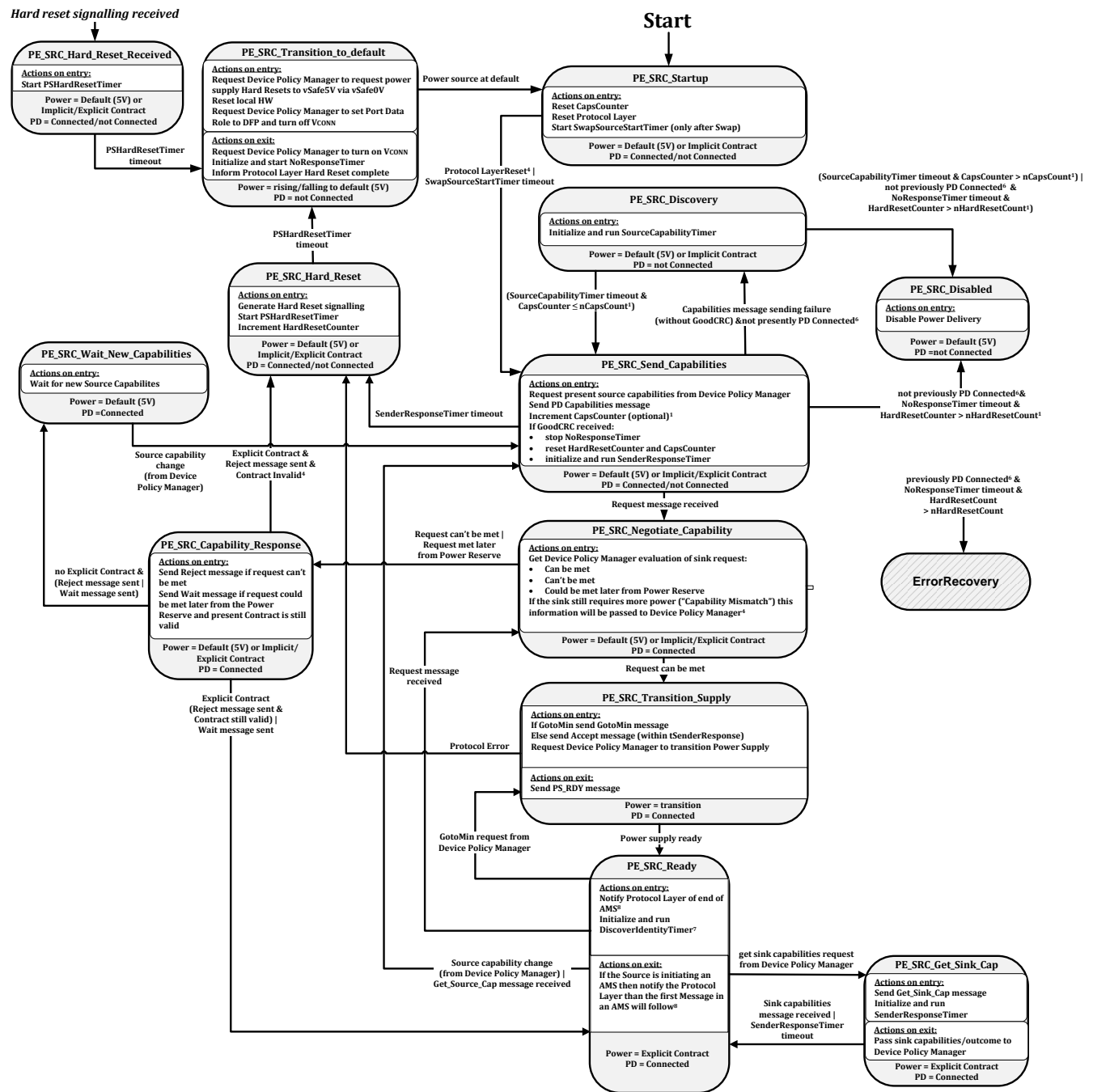
USB Power Delivery ENGINEERING CHANGE NOTICE

Actual Change

(a). Section 8.3.3.2, Page 398, Figure 8-42

From Text:

USB Power Delivery ENGINEERING CHANGE NOTICE

Figure 8-42 Source Port Policy Engine state diagram

Hard reset signalling received

Start

PE_SRC_Hard_Reset_Received
Actions on entry:
Start PSHardResetTimer

Power = Default (5V) or
Implicit/Explicit Contract
PD = Connected/not Connected

PE_SRC_Transition_to_default
Actions on entry:
Request Device Policy Manager to request power
supply Hard Resets to vSafe5V via vSafe0V
Reset local HW
Request Device Policy Manager to set Port Data
Role to DFP and turn off V$_{CONN}$

Actions on exit:
Request Device Policy Manager to turn on V$_{CONN}$
Initialize and start NoResponseTimer
Inform Protocol Layer Hard Reset complete

Power = rising/falling to default (5V)
PD = not Connected

PE_SRC_Startup
Actions on entry:
Reset CapsCounter
Reset Protocol Layer
Start SwapSourceStartTimer (only after Swap)

Power = Default (5V) or Implicit Contract
PD = Connected/not Connected

Power source at default

PSHardResetTimer timeout

Protocol LayerReset[4] |
SwapSourceStartTimer timeout

PE_SRC_Discovery
Actions on entry:
Initialize and run SourceCapabilityTimer

Power = Default (5V) or Implicit Contract
PD = not Connected

(SourceCapabilityTimer timeout & CapsCounter > nCapsCount[1]) |
not previously PD Connected[6] &
NoResponseTimer timeout &
HardResetCounter > nHardResetCount[1]

PE_SRC_Hard_Reset
Actions on entry:
Generate Hard Reset signalling
Start PSHardResetTimer
Increment HardResetCounter

Power = Default (5V) or
Implicit/Explicit Contract
PD = Connected/not Connected

PSHardResetTimer timeout

(SourceCapabilityTimer timeout &
CapsCounter ≤ nCapsCount[1])

Capabilities message sending failure
(without GoodCRC) &not presently PD Connected[6]

PE_SRC_Disabled
Actions on entry:
Disable Power Delivery

Power = Default (5V)
PD =not Connected

PE_SRC_Wait_New_Capabilities
Actions on entry:
Wait for new Source Capabilites

Power = Default (5V)
PD =Connected

PE_SRC_Send_Capabilities
Actions on entry:
Request present source capabilities from Device Policy Manager
Send PD Capabilities message
Increment CapsCounter (optional)[1]
If GoodCRC received:
• stop NoResponseTimer
• reset HardResetCounter and CapsCounter
• initialize and run SenderResponseTimer

Power = Default (5V) or Implicit/Explicit Contract
PD = Connected/not Connected

SenderResponseTimer timeout

not previously PD Connected[6]&
NoResponseTimer timeout &
HardResetCounter > nHardResetCount[1]

Source capability
change
(from Device
Policy Manager)

Explicit Contract &
Reject message sent &
Contract Invalid[4]

previously PD Connected[6] &
NoResponseTimer timeout &
HardResetCount
> nHardResetCount

ErrorRecovery

Request message received

PE_SRC_Capability_Response
Actions on entry:
Send Reject message if request can't be
met
Send Wait message if request could
be met later from the Power
Reserve and present Contract is still
valid

Power = Default (5V) or Implicit/
Explicit Contract
PD = Connected

no Explicit Contract &
(Reject message sent |
Wait message sent)

Request can't be met |
Request met later
from Power Reserve

PE_SRC_Negotiate_Capability
Actions on entry:
Get Device Policy Manager evaluation of sink request:
• Can be met
• Can't be met
• Could be met later from Power Reserve
If the sink still requires more power ("Capability Mismatch") this
information will be passed to Device Policy Manager[4]

Power = Default (5V) or Implicit/Explicit Contract
PD = Connected

Explicit Contract
(Reject message sent &
Contract still valid) |
Wait message sent

Request message
received

Request can be met

Protocol Error

PE_SRC_Transition_Supply
Actions on entry:
If GotoMin send GotoMin message
Else send Accept message (within tSenderResponse)
Request Device Policy Manager to transition Power Supply

Actions on exit:
Send PS_RDY message

Power = transition
PD = Connected

GotoMin request from
Device Policy Manager

Power supply ready

PE_SRC_Ready
Actions on entry:
Notify Protocol Layer of end of
AMS[8]
Initialize and run
DiscoverIdentityTimer[7]

Actions on exit:
If the Source is initiating an
AMS then notify the Protocol
Layer than the first Message in
an AMS will follow[8]

Power = Explicit Contract
PD = Connected

Source capability change
(from Device Policy Manager) |
Get_Source_Cap message received

get sink capabilities request
from Device Policy Manager

Sink capabilities
message received |
SenderResponseTimer
timeout

PE_SRC_Get_Sink_Cap
Actions on entry:
Send Get_Sink_Cap message
Initialize and run
SenderResponseTimer

Actions on exit:
Pass sink capabilities/outcome to
Device Policy Manager

Power = Explicit Contract
PD = Connected

---

[1] For a USB non-Type-C Port If the NoResponseTimer times Out and the HardResetCounter > nHardResetCount the Policy Engine can either continue sending capabilities or go to the **PE_SRC_Disabled** state.

USB Power Delivery ENGINEERING CHANGE NOTICE

[2] Implementation of the *CapsCounter* is **Optional**. In the case where this is not implemented the Source shall continue to send *Source_Capabilities* Messages each time the *SourceCapabilityTimer* times out.

[3] When operating at *vSafe5V* and not swapped, or when two systems both using the Type-C connector are communicating, *Ping* messages are optional so the *SourceActivityTimer* is not required to run in these circumstances.

[4] Since the Sink is required to make a valid request from the offered capabilities the expected transition is via "Request can be met" unless the Source capabilities have changed since the last offer.

[5] "Contract **Invalid**" means that the previously negotiated Voltage and Current values are no longer included in the Source's new Capabilities. If the Sink fails to make a valid Request in this case then Power Delivery operation is no longer possible and Power Delivery mode is exited with a Hard Reset.

[6] After a Power Swap the new Source is required to wait an additional *tSwapSourceStart* before sending a *Source_Capabilities* Message. This delay is not required when first starting up a system.

[7] PD Connected is defined as a situation when the Port Partners are actively communicating. The Port Partners remain PD Connected after a Swap until there is a transition to Disabled or the connector is able to identify a disconnect (USB Type-C, USB Type-A with insert detect, USB Micro-AB).
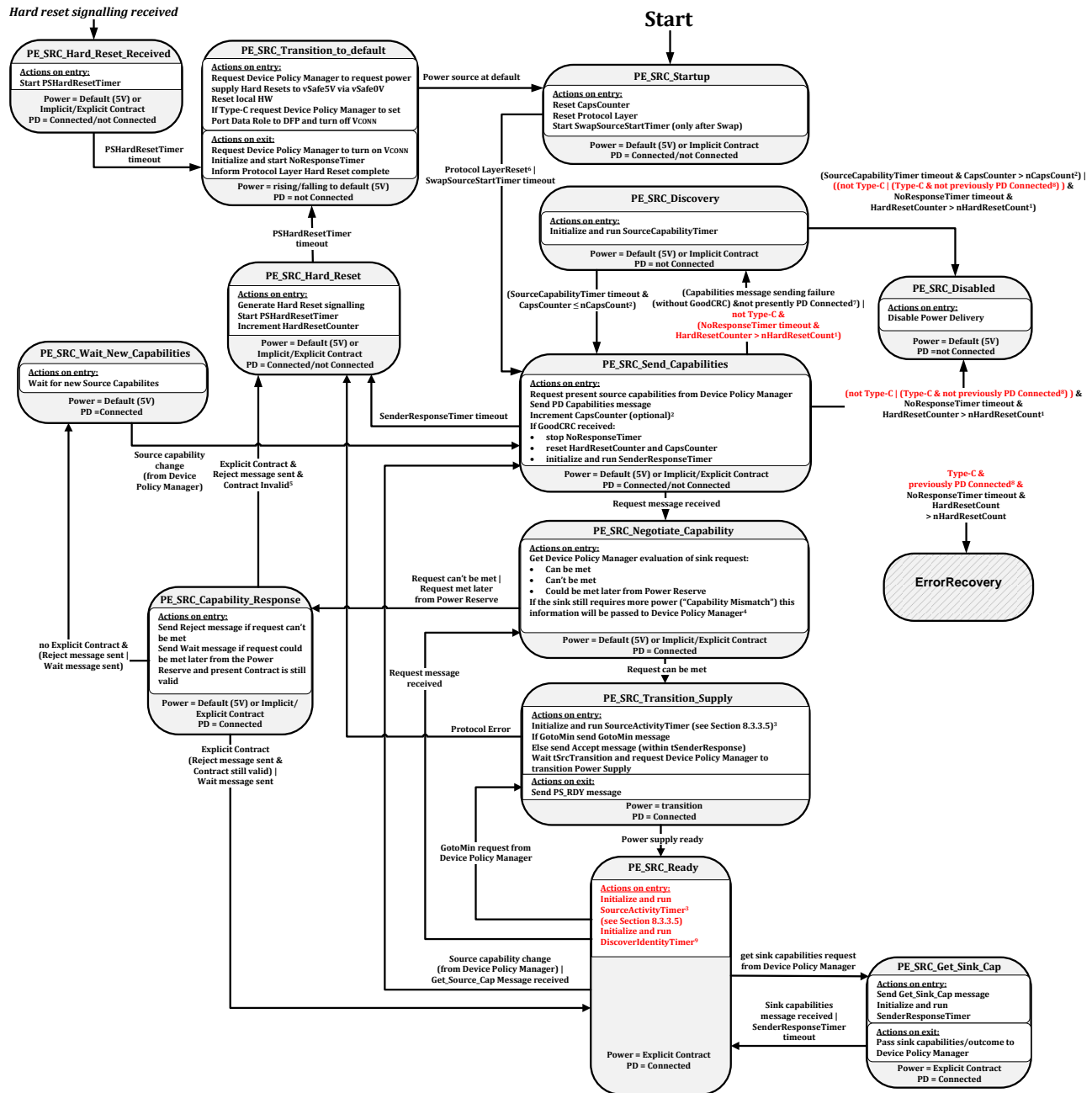
[8] Port Partners are no longer PD Connected after a Hard Reset but for USB Type-C connections consideration needs to be given as to whether there has been a PD Connection while the Ports have been Attached to prevent unnecessary USB Type-C Error Recovery.

[9] The *DiscoverIdentityTimer* is run when this is a DFP and a PD Connection with a Cable Plug needs to be established i.e. no GoodCRC has yet been received in response to a Discover Identity message. PE_SRC_Startup state

To Text:

USB Power Delivery ENGINEERING CHANGE NOTICE

Figure 8-42 Source Port Policy Engine state diagram

Hard reset signalling received

Start

PE_SRC_Hard_Reset_Received
Actions on entry:
Start PSHardResetTimer

Power = Default (5V) or
Implicit/Explicit Contract
PD = Connected/not Connected

PE_SRC_Transition_to_default
Actions on entry:
Request Device Policy Manager to request power
supply Hard Resets to vSafe5V via vSafe0V
Reset local HW
If Type-C request Device Policy Manager to set
Port Data Role to DFP and turn off Vconn

Actions on exit:
Request Device Policy Manager to turn on Vconn
Initialize and start NoResponseTimer
Inform Protocol Layer Hard Reset complete

Power = rising/falling to default (5V)
PD = not Connected

PE_SRC_Startup
Actions on entry:
Reset CapsCounter
Reset Protocol Layer
Start SwapSourceStartTimer (only after Swap)

Power = Default (5V) or Implicit Contract
PD = Connected/not Connected

Power source at default

PSHardResetTimer timeout

Protocol LayerReset[6] |
SwapSourceStartTimer timeout

PSHardResetTimer timeout

PE_SRC_Hard_Reset
Actions on entry:
Generate Hard Reset signalling
Start PSHardResetTimer
Increment HardResetCounter

Power = Default (5V) or
Implicit/Explicit Contract
PD = Connected/not Connected

PE_SRC_Discovery
Actions on entry:
Initialize and run SourceCapabilityTimer

Power = Default (5V) or Implicit Contract
PD = not Connected

(SourceCapabilityTimer timeout & CapsCounter > nCapsCount[2]) |
((not Type-C | (Type-C & not previously PD Connected[8])) &
NoResponseTimer timeout &
HardResetCounter > nHardResetCount[1])

PE_SRC_Disabled
Actions on entry:
Disable Power Delivery

Power = Default (5V)
PD = not Connected

(SourceCapabilityTimer timeout &
CapsCounter ≤ nCapsCount[2])

(Capabilities message sending failure
(without GoodCRC) &not presently PD Connected[7]) |
not Type-C &
(NoResponseTimer timeout &
HardResetCounter > nHardResetCount[1])

SenderResponseTimer timeout

PE_SRC_Wait_New_Capabilities
Actions on entry:
Wait for new Source Capabilities

Power = Default (5V)
PD = Connected

PE_SRC_Send_Capabilities
Actions on entry:
Request present source capabilities from Device Policy Manager
Send PD Capabilities message
Increment CapsCounter (optional)[2]
If GoodCRC received:
• stop NoResponseTimer
• reset HardResetCounter and CapsCounter
• initialize and run SenderResponseTimer

Power = Default (5V) or Implicit/Explicit Contract
PD = Connected/not Connected

(not Type-C | (Type-C & not previously PD Connected[8])) &
NoResponseTimer timeout &
HardResetCounter > nHardResetCount[1]

Source capability
change
(from Device
Policy Manager)

Explicit Contract &
Reject message sent &
Contract Invalid[5]

Request message received

Type-C &
previously PD Connected[8] &
NoResponseTimer timeout &
HardResetCount
> nHardResetCount

ErrorRecovery

PE_SRC_Negotiate_Capability
Actions on entry:
Get Device Policy Manager evaluation of sink request:
• Can be met
• Can't be met
• Could be met later from Power Reserve
If the sink still requires more power ("Capability Mismatch") this
information will be passed to Device Policy Manager[4]

Power = Default (5V) or Implicit/Explicit Contract
PD = Connected

Request can't be met |
Request met later
from Power Reserve

PE_SRC_Capability_Response
Actions on entry:
Send Reject message if request can't
be met
Send Wait message if request could
be met later from the Power
Reserve and present Contract is still
valid

Power = Default (5V) or Implicit/
Explicit Contract
PD = Connected

no Explicit Contract &
(Reject message sent |
Wait message sent)

Request message
received

Protocol Error

Request can be met

PE_SRC_Transition_Supply
Actions on entry:
Initialize and run SourceActivityTimer (see Section 8.3.3.5)[3]
If GotoMin send GotoMin message
Else send Accept message (within tSenderResponse)
Wait tSrcTransition and request Device Policy Manager to
transition Power Supply

Actions on exit:
Send PS_RDY message

Power = transition
PD = Connected

Explicit Contract
(Reject message sent &
Contract still valid) |
Wait message sent

GotoMin request from
Device Policy Manager

Power supply ready

PE_SRC_Ready
Actions on entry:
Initialize and run
SourceActivityTimer[3]
(see Section 8.3.3.5)
Initialize and run
DiscoverIdentityTimer[9]

Power = Explicit Contract
PD = Connected

Source capability change
(from Device Policy Manager) |
Get_Source_Cap Message received

get sink capabilities request
from Device Policy Manager

Sink capabilities
message received |
SenderResponseTimer
timeout

PE_SRC_Get_Sink_Cap
Actions on entry:
Send Get_Sink_Cap message
Initialize and run
SenderResponseTimer

Actions on exit:
Pass sink capabilities/outcome to
Device Policy Manager

Power = Explicit Contract
PD = Connected

---

[1] For a USB non-Type-C Port If the NoResponseTimer times Out and the HardResetCounter > nHardResetCount the Policy Engine can either continue sending capabilities or go to the **PE_SRC_Disabled** state.

[2] Implementation of the *CapsCounter* is **Optional**. In the case where this is not implemented the Source shall continue to send *Source_Capabilities* Messages each time the *SourceCapabilityTimer* times out.

[3] When operating at *vSafe5V* and not swapped, or when two systems both using the Type-C connector are communicating, *Ping* messages are optional so the *SourceActivityTimer* is not required to run in these circumstances.

[4] Since the Sink is required to make a valid request from the offered capabilities the expected transition is via "Request can be met" unless the Source capabilities have changed since the last offer.

[5] "Contract *Invalid*" means that the previously negotiated Voltage and Current values are no longer included in the Source's new Capabilities. If the Sink fails to make a valid Request in this case then Power Delivery operation is no longer possible and Power Delivery mode is exited with a Hard Reset.

[6] After a Power Swap the new Source is required to wait an additional *tSwapSourceStart* before sending a *Source_Capabilities* Message. This delay is not required when first starting up a system.

[7] PD Connected is defined as a situation when the Port Partners are actively communicating. The Port Partners remain PD Connected after a Swap until there is a transition to Disabled or the connector is able to identify a disconnect (USB Type-C, USB Type-A with insert detect, USB Micro-AB).

[8] Port Partners are no longer PD Connected after a Hard Reset but for USB Type-C connections consideration needs to be given as to whether there has been a PD Connection while the Ports have been Attached to prevent unnecessary USB Type-C Error Recovery.

[9] The *DiscoverIdentityTimer* is run when this is a DFP and a PD Connection with a Cable Plug needs to be established i.e. no GoodCRC has yet been received in response to a Discover Identity message. *PE_SRC_Startup* state

(b). Section 8.3.3.2.5, Page 401

From Text:

8.3.3.2.5   PE_SRC_Transition_Supply state

The Policy Engine shall be in the *PE_SRC_Transition_Supply* state while the power supply is transitioning from one power to another.

On entry to the *PE_SRC_Transition_Supply* state, the Policy Engine shall initialize and run the *SourceActivityTimer* (see Section 8.3.3.6 for details of *Ping* messaging for Source Ports), request the Protocol Layer to either send a *GotoMin* Message (if this was requested by the Device Policy Manager) or otherwise an *Accept* Message and inform the Device Policy Manager that it shall transition the power supply to the Requested power level. Note: that if the power supply is currently operating at the requested power no change will be necessary.

On exit from the *PE_SRC_Transition_Supply* state the Policy Engine shall request the Protocol Layer to send a *PS_RDY* Message.

The Policy Engine shall transition to the *PE_SRC_Ready* state when:

- The Device Policy Manager informs the Policy Engine that the power supply is ready.

USB Power Delivery ENGINEERING CHANGE NOTICE

To Text:

8.3.3.2.6   PE_SRC_Transition_Supply state

The Policy Engine shall be in the *PE_SRC_Transition_Supply* state while the power supply is transitioning from one power to another.

On entry to the *PE_SRC_Transition_Supply* state, the Policy Engine shall initialize and run the *SourceActivityTimer* (see Section 8.3.3.6 for details of *Ping* messaging for Source Ports), request the Protocol Layer to either send a *GotoMin* Message (if this was requested by the Device Policy Manager) or otherwise an *Accept* Message and inform the Device Policy Manager that it shall transition the power supply to the Requested power level. Note: that if the power supply is currently operating at the requested power no change will be necessary.

On exit from the *PE_SRC_Transition_Supply* state the Policy Engine shall request the Protocol Layer to send a *PS_RDY* Message.

The Policy Engine shall transition to the *PE_SRC_Ready* state when:

- The Device Policy Manager informs the Policy Engine that the power supply is ready.

The Policy Engine shall transition to the *PE_SRC_Hard_Reset* state when:

- A Protocol Error occurs.