Changing assigned Class through DLL v101

Info (not part of baseline)

In the legacy state diagram, the PD would move from the MDI_POWER1 to the MDI_POWER2 state if it started out in Class 3 and successfully increased its power budget through LLDP. The legacy SD enforces that only PDs that request Class 4 are able to move to a higher power state.
This currently does not work: changes through DLL are not reflected in the pd_max_power variable. This baseline implements that same functionality for the Type 3/4 state diagrams. For now focus is on single-signature, once this is stable, a sync with the dual-signature state diagrams needs to happen.

for

Which is correct....

PSE section

33.2.5.9 Type 3 and Type 4 variables

Add variable as follows:

pse_power_update

A variable that is set when the PSEAllocatedPowerValue in the DLL state diagram in Figure 33–46 has been updated.
Values:

FALSE: The value of PSEAllocatedPowerValue has not changed.
TRUE: The value of PSEAllocatedPowerValue has changed.

33.2.5.11 Type 3 and Type 4 functions

Add function as follows:

do_update_pd_allocated_pwr

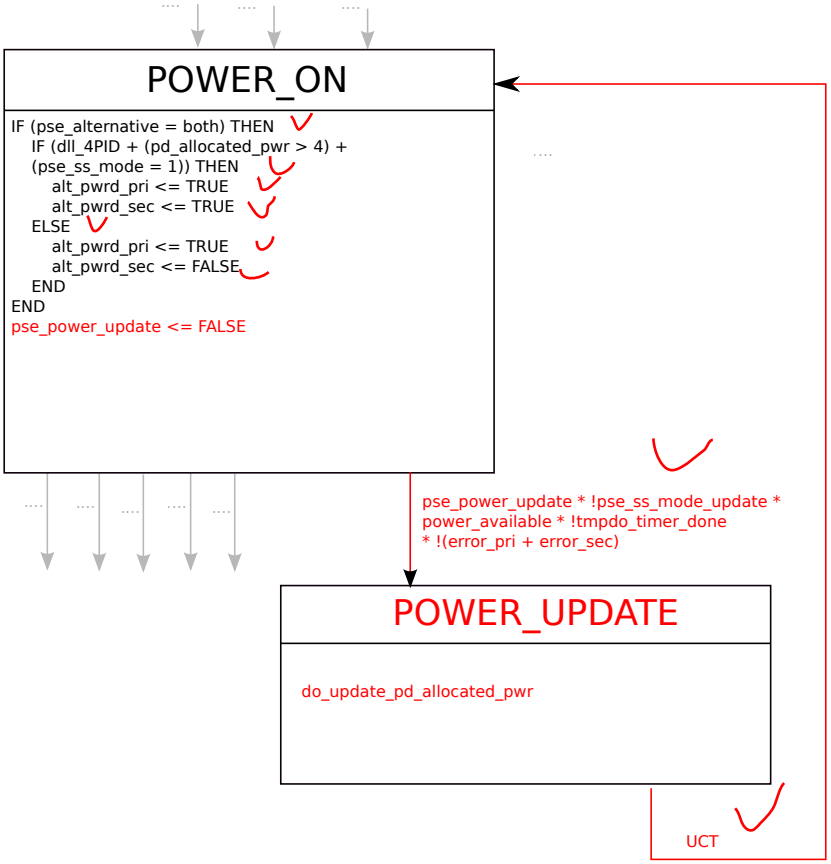
A function that updates the pd_allocated_value based on the value of PSEAllocatedPowerValue as defined in Table 33–14. This function returns the following variable:

pd_allocated_pwr: this variable indicates the Class assigned to the PD.
Values:

1: Class 1
2: Class 2
3: Class 3
4: Class 4
5: Class 5
6: Class 6
7: Class 7
8: Class 8

33.2.5.12 Type 3 and Type 4 state diagrams

Change Figure 33–15 on page 95 as follows:



I guess it doesn't changes that in our project the PSE has to issue all necessary fingers regardless of LLDP capability. Please confirm.

Correct, this is unchanged.

PD section

33.3.3.8 Type 3 and Type 4 single-signature variables

Info (not part of baseline)

Variable summary:

pd_req_pwr	A constant that denotes the requested Class of the PD. This variable determines what signatures present_class_sig_A and present_class_sig_B are. It is also the maximum power the PD can ever draw.
pse_power_level	A variable the counts the number of classification events. One event sets it to 3. Two or three events sets it to 4. Four events sets it to 6. Five events sets it to 8. These values are chosen such that it becomes easy to derive pd_max_power.
pse_assigned_class	Is the Class assigned by the PSE. This is initially set based on the min(pse_power_level, pd_req_class) but can change after a DLL negotiation. Note that the assigned Class can be higher than the requested Class. See pd_max_power.
pd_max_power	This variable is the maximum power limit of the PD. In normal operation this will be equal to the pse_assigned_class variable. In the exceptional case that the PSE has assigned a higher Class than the requested Class, this variable will be equal to the requested Class and thus prevent the PD from going above its Physical Layer requested power.

Add the following variables:

In the exceptional case: Do you mean to extended power?

pse_assigned_class

A variable that indicates the assigned Class. This variable is initially set by Physical Layer classification and may be updated through DLL classification.

append to teh end of this text: "to lower values than the initially set by Physical Layer classification"

Values:

1: Class 1
2: Class 2
3: Class 3
4: Class 4

This is true, but is already handled by a SHALL, why do we keep repeating this ?

Power demotion, not ext. pwr.

5: Class 5
6: Class 6
7: Class 7
8: Class 8

pd_power_update

A variable that is set when the PDMaxPowerValue in the DLL state diagram in Figure 33–49 has been updated.
Values:

FALSE: The value of PDMaxPowerValue has not changed.
TRUE: The value of PDMaxPowerValue has changed.

33.3.3.10 Type 3 and Type 4 single-signature functions

Add the following function:

do_update_pse_assigned_class

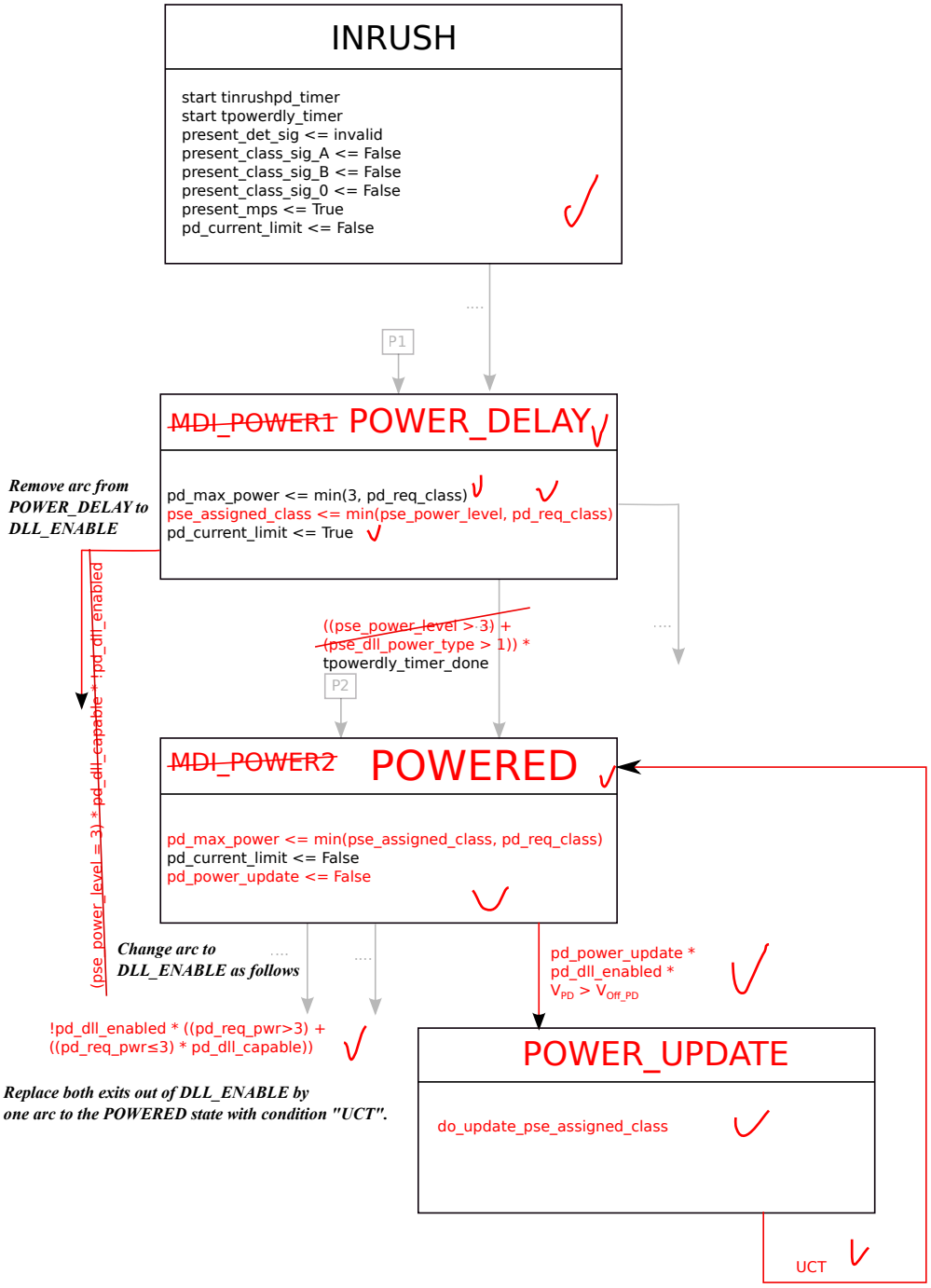
A function that updates the pse_assigned_class based on the value of PDMaxPowerValue as defined in Table 33–24.
This function returns the following variable:

pse_assigned_class: this variable indicates the Class assigned to the PD.
Values:

1: Class 1
2: Class 2
3: Class 3
4: Class 4
5: Class 5
6: Class 6
7: Class 7
8: Class 8

33.3.3.11 Type 3 and Type 4 single-signature PD state diagrams

Change Figure 33–32 as follows:



Info (not part of baseline)

The following non-normative text is added to explain that it is not permitted to draw more power than the requested Class. The actual requirement is in the PD state diagram, as well as on D2.2, page 153, line 47.

33.3.6 PD classifications

Make changes to the text as follows:

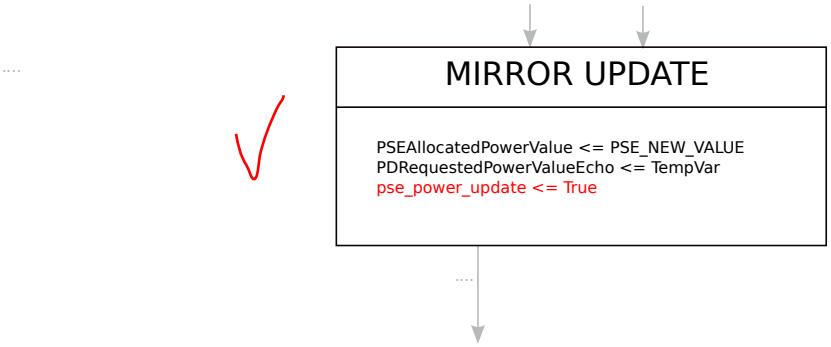
After a successful DLL classification, the assigned Class changes depending on the value of PDMaxPowerValue variable, as defined in Table 33–24. A PD may be able to get a Class assigned that is higher than its requested Class, however it is not permitted to draw more power than what corresponds to the PDs requested Class.

Why to allow it in the text???

Fair point, I will change/delete it.

33.5.3.6 Power control S state diagrams

Change Figure 33–46 as follows:



Change Figure 33–49 as follows: