DLL Constants v101

Info (not part of baseline)

Three constants are listed in 145.5.3.2:

- PD_DLLMAX_VALUE
- PD_INITIAL_VALUE
- PSE_INITIAL_VALUE

They are however not constants: their value can only be determined after the PSE or PD has completed classification, and, moreover, the value changes when a new classification cycle occurs. I was initially unsure how best to fix this, but a lightweight solution is possible.

A second problem is regarding the use of PD_DLLMAX_VALUE. This value is used to limit what the maximum value of PDRequestedPowerValue can be. Currently PD_DLLMAX_VALUE is tied to pd_max_power, which means a PD can never ask for more power than what it initially got via Physical Layer classification. The desired behavior is that it is restricted to its requested Class. This in turn makes PD_DLLMAX_VALUE a true constant.

145.5.3.2 Single-signature constants

Change the description of PD_DLLMAX_VALUE as follows:

This value is derived from ~~pd_max_power variable (145.3.3.4)~~ pd_req_class (145.3.3.3) described as follows:

~~pd_max_power~~ pd_req_class	PD_DLLMAX_VALUE
~~0~~	~~130~~
1	39
2	65
3	130
4	255
5	400
6	600
7	620
8	999

Change the description of PD_INITIAL_VALUE as follows:

~~This~~ The value of this variable is valid after classification and is derived ~~as follows~~ from the pd_max_power (145.3.3.4) variable used in the PD state diagrams; defined in Figure 145–26 as follows:

Change the description of PSE_INITIAL_VALUE as follows:

~~This~~ The value of this variable is valid after classification and is derived from the pd_allocated_pwr variable, as defined in 145.2.5.4, which is used in the PSE state diagrams in 145.2.5.7 as follows:

Move the text in 145.5.3.2 (Constants) and the variables (except PD_DLLMAX_VALUE) to 145.5.3.3 (Variables).

Also apply fixes in the document to dual-signature as appropriate.