Simplified classification in the State Diagram II v100

Info (not part of baseline)

This baseline is the follow up to ➠yseboodt_01_1116_simpleclass.pdf. Aims are as follows:

1. The current D2.2 state diagram uses a complex combination of class_num_events and pse_avail_pwr to control the classification flow. A further problem is that in the D2.1 text several permutations that should be possible are not (eg. A Type 4 PSE can only have pse_avail_pwr = 7 or 8) This baseline simplifies this by removing class_num_events and only work with pse_avail_pwr.
2. The variables pd_req_power and pd_allocated_power are currently returned by a function. The assignment of those variables is however complex (due to power demotion) and should be done explicitly by the SD, rather than 'magically' get filled out by a function.
3. An open ticket on the TDL is to implement improved discovery for the PD's requested Class, even for PSEs that have pse_power_available ≤ 3. This is implemented through a function do_class_probe (which allows flexible implementation) in the new states CLASS_PROBE and CLASS_RESET.
4. There are several 'shalls' in the Type 3/4 variable list. These are moved to the appropriate sections and struck from the variable list.

Simplified classification

33.2.5.9 Type 3 and Type 4 variables

Add the following variables:

option_classprobe

This variable indicates if the PSE should determine the requested Class of the PD when pse_avail_pwr is less than 3. When set to TRUE, the PSE will issue 3 class events to determine the requested Class, perform a classification reset by applying V_{Reset} for at least T_{Reset} to the PI (see Table 33–17), followed by a normal classification procedure.
Values:

FALSE: The PSE will not probe for the PD requested Class
TRUE: The PSE probes for the PD requested Class

pd_allocated_pwr

A variable that indicates the Class that has been 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

Info (not part of baseline)

The pd_req_pwr variable (returned by do_classification) had a shall in it and a may. Both of these have been moved to the classification section. The original text was:

> pd_req_pwr: This variable indicates the power class requested by the PD. When a PD requests a higher class than a PSE can support, the PSE shall assign the PD Class 3, 4, or 6, whichever is the highest that it can support. For Type 3 and Type 4 PSEs, when connected to a single-signature PD, operating over 4-pairs, classification events may appear on one or both pairsets. See 33.2.7.

pd_req_pwr

The variable indicates the power class requested by the PD. When a PD requests a higher Class than a PSE can support, the PSE assigns the PD to Class 3, Class 4, or Class 6, whichever is the highest Class it can support. If pse_avail_pwr is less than 5, this variable may not contain the actual requested Class by the PSE; see pd_req_pwr_probe.
Values:

0: Class 0
1: Class 1
2: Class 2
3: Class 3
4: Class 4
5: Class 5
6: Class 6
7: Class 7
8: Class 8

Info (not part of baseline)

We can now remove Table 33–7 (which ties PSE Type to class_num_events) and Table 33–8 (which ties class_num_events to pse_avail_power). This is then replaced by a Table that links PSE Type to the allowed values of pse_avail_power.

Change text on page 85 as follows:

~~Type 3 and Type 4 PSEs shall meet at least one of the allowable variable definition permutations described in Table 33–7 and Table 33–8.~~
Type 3 and Type 4 PSEs shall set pse_avail_pwr, pse_avail_pwr_pri, and pse_avail_pwr_sec from the range in Table 33–7.

Remove Table 33–7 and 33–8.

Insert a new Table as follows:

Table 33–7 — Allowed Type 3 and Type 4 values for pse_avail_pwr, pse_avail_pwr_pri, and pse_avail_pwr_sec.

PSE Type	pse_avail_pwr	pse_avail_pwr_pri, pse_avail_pwr_sec
Type 3	1 to 6	1 to 4
Type 4	1 to 8	1 to 5

33.2.5.11 Type 3 and Type 4 functions

Remove variables pd_req_pwr and pd_allocated_pwr from the do_classification function.

Rename variables as follows:

Old name	**New name**
pd_class_detected	pd_class_sig
pd_class_detected_pri	pd_class_sig_pri
pd_class_detected_sec	pd_class_sig_sec

Add a new function do_class_probe as follows:

do_class_probe

This functions discovers the requested Class of the PD by producing a number of classification events. This function returns the following variables:

pd_req_pwr_probe: This variable contains the requested Class of the PD.
Values:

0: Class 0
1: Class 1
2: Class 2
3: Class 3
4: Class 4
5: Class 5
6: Class 6

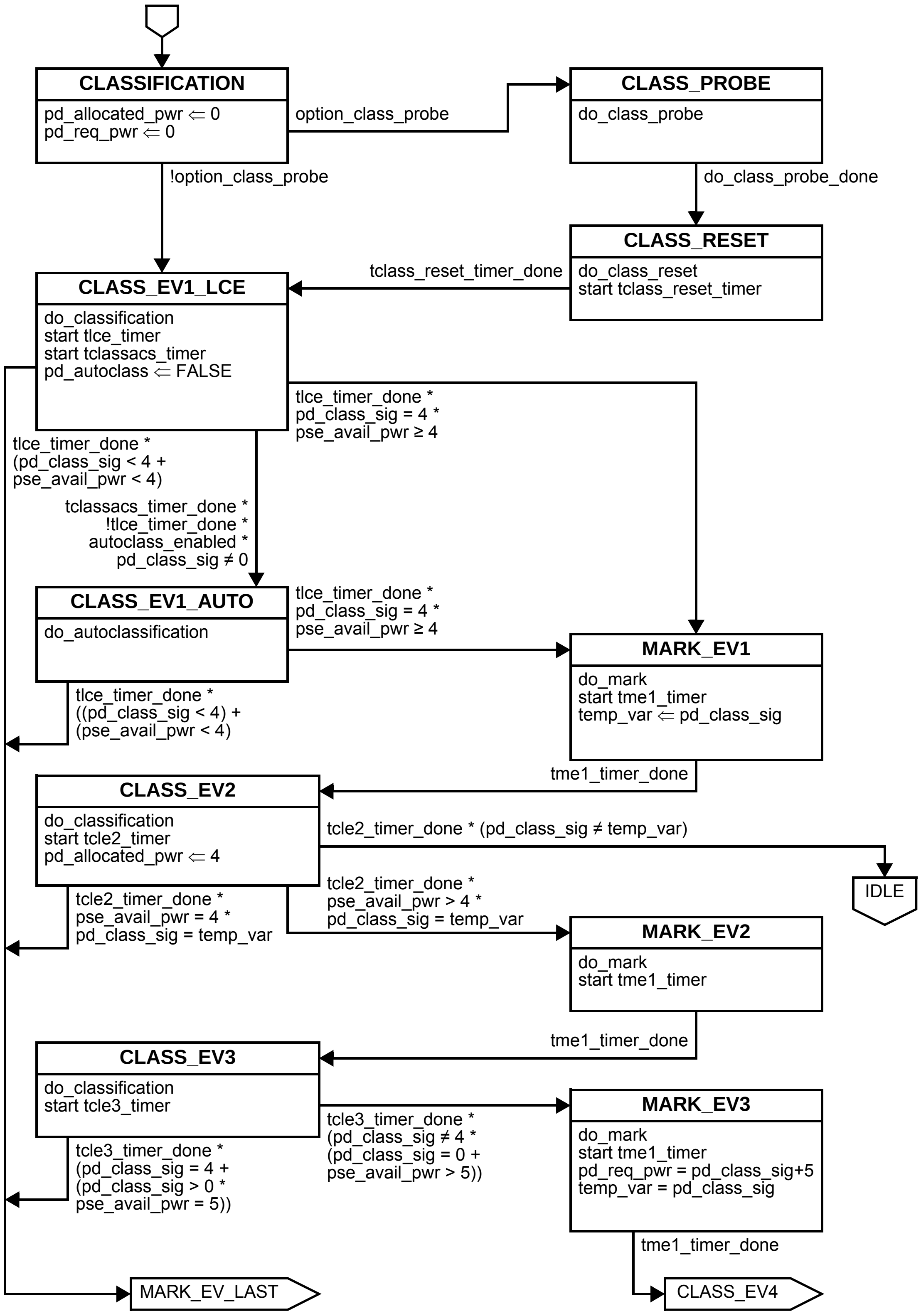
7: Class 7
8: Class 8

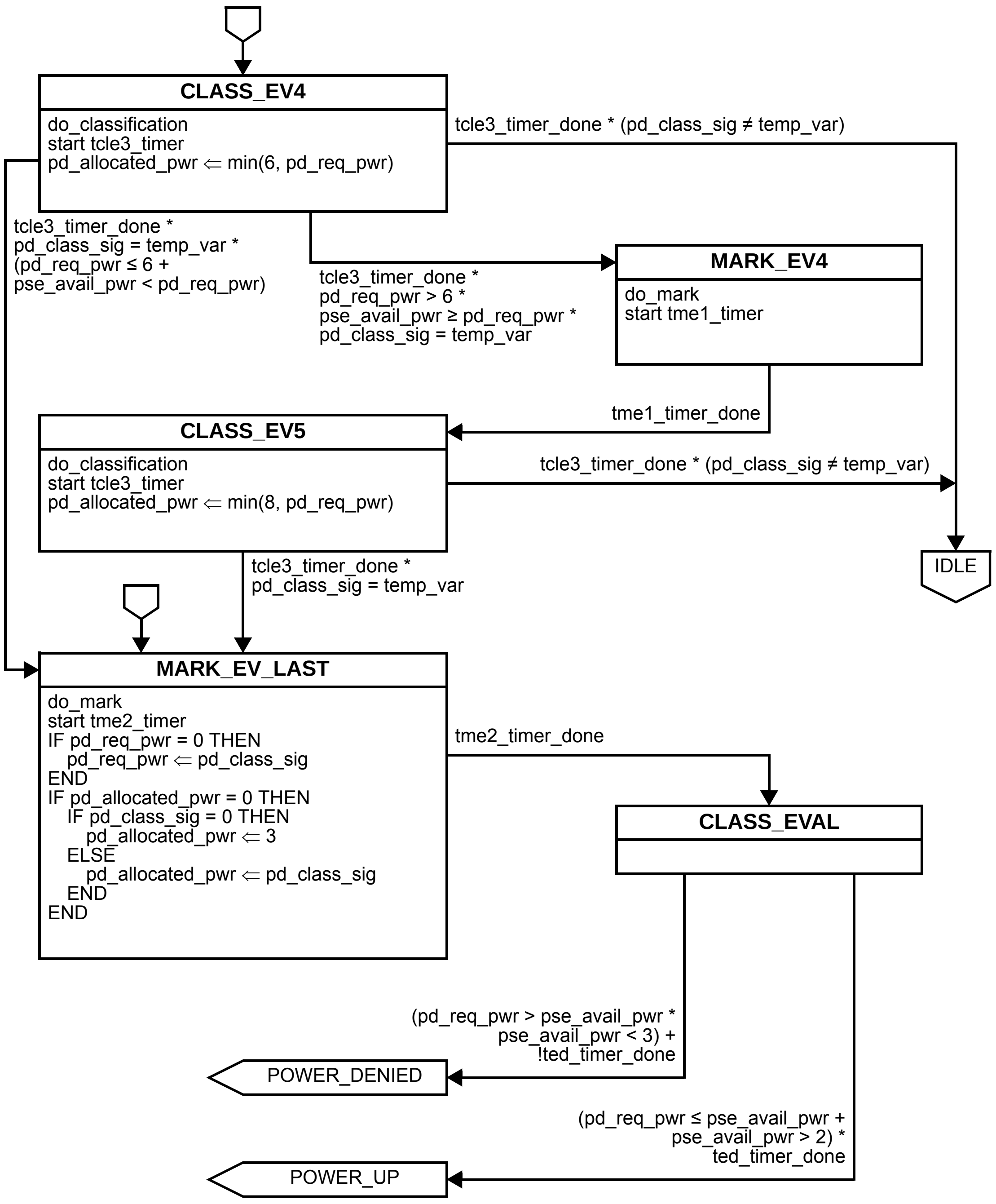
33.2.5.12 Type 3 and Type 4 state diagrams

Change the arcs going from DETECT_EVAL and CXN_CHK_DETECT_EVAL to CLASS_EV1_LCE to go to CLASSIFICATION in stead.

Remove the CLASS_EVAL state from page 95.

Replace the classification SD on page 97 as follows:


CLASSIFICATION
pd_allocated_pwr ⇐ 0
pd_req_pwr ⇐ 0
option_class_probe
!option_class_probe
CLASS_PROBE
do_class_probe
do_class_probe_done
CLASS_RESET
do_class_reset
start tclass_reset_timer
tclass_reset_timer_done
CLASS_EV1_LCE
do_classification
start tlce_timer
start tclassacs_timer
pd_autoclass ⇐ FALSE
tlce_timer_done * pd_class_sig = 4 * pse_avail_pwr ≥ 4
tlce_timer_done * (pd_class_sig < 4 + pse_avail_pwr < 4)
tclassacs_timer_done * !tlce_timer_done * autoclass_enabled * pd_class_sig ≠ 0
CLASS_EV1_AUTO
do_autoclassification
tlce_timer_done * pd_class_sig = 4 * pse_avail_pwr ≥ 4
tlce_timer_done * ((pd_class_sig < 4) + (pse_avail_pwr < 4)
MARK_EV1
do_mark
start tme1_timer
temp_var ⇐ pd_class_sig
tme1_timer_done
CLASS_EV2
do_classification
start tcle2_timer
pd_allocated_pwr ⇐ 4
tcle2_timer_done * (pd_class_sig ≠ temp_var)
IDLE
tcle2_timer_done * pse_avail_pwr > 4 * pd_class_sig = temp_var
tcle2_timer_done * pse_avail_pwr = 4 * pd_class_sig = temp_var
MARK_EV2
do_mark
start tme1_timer
tme1_timer_done
CLASS_EV3
do_classification
start tcle3_timer
tcle3_timer_done * (pd_class_sig ≠ 4 * (pd_class_sig = 0 + pse_avail_pwr > 5))
tcle3_timer_done * (pd_class_sig = 4 + (pd_class_sig > 0 * pse_avail_pwr = 5))
MARK_EV3
do_mark
start tme1_timer
pd_req_pwr = pd_class_sig+5
temp_var = pd_class_sig
tme1_timer_done
MARK_EV_LAST
CLASS_EV4



CLASS_EV4
do_classification
start tcle3_timer
pd_allocated_pwr ⇐ min(6, pd_req_pwr)
tcle3_timer_done * (pd_class_sig ≠ temp_var)
tcle3_timer_done * pd_class_sig = temp_var * (pd_req_pwr ≤ 6 + pse_avail_pwr < pd_req_pwr)
tcle3_timer_done * pd_req_pwr > 6 * pse_avail_pwr ≥ pd_req_pwr * pd_class_sig = temp_var
MARK_EV4
do_mark
start tme1_timer
tme1_timer_done
CLASS_EV5
do_classification
start tcle3_timer
pd_allocated_pwr ⇐ min(8, pd_req_pwr)
tcle3_timer_done * (pd_class_sig ≠ temp_var)
IDLE
tcle3_timer_done * pd_class_sig = temp_var
MARK_EV_LAST
do_mark
start tme2_timer
IF pd_req_pwr = 0 THEN
pd_req_pwr ⇐ pd_class_sig
END
IF pd_allocated_pwr = 0 THEN
IF pd_class_sig = 0 THEN
pd_allocated_pwr ⇐ 3
ELSE
pd_allocated_pwr ⇐ pd_class_sig
END
END
tme2_timer_done
CLASS_EVAL
(pd_req_pwr > pse_avail_pwr * pse_avail_pwr < 3) + !ted_timer_done
POWER_DENIED
(pd_req_pwr ≤ pse_avail_pwr + pse_avail_pwr > 2) * ted_timer_done
POWER_UP

Info (not part of baseline)

Below are the 'shall' statements currently still in the variable list implemented in the text below.

33.2.7 PSE classification of PDs and mutual identification

Change the text on page 106, line 5 as follows:

Physical Layer classification occurs before a PSE supplies power to a PD, when the PSE asserts a voltage in the range of V Class as defined in Table 33–17 onto one or both pairsets. This is called a class event. The PD responds to each class event with a current representing one of a limited number of classification signatures.

The assigned Class is the result of the PDs requested Class and the number of class events produced by the PSE as shown in Table 33–13 and Table 33–14. See 33.3.6 for PD classification behavior. When a single- signature PD requests a higher Class than a Type 3 or Type 4 PSE can support, the PSE shall assign~~s~~ the PD to Class 3, 4, or 6, whichever is the highest that it can support. When a dual-signature PD requests a higher Class than a Type 3 or Type 4 PSE can support, the PSE assigns the PD Class 3 or 4, whichever is the highest that it can support.

33.2.7.2 PSE Multiple-Event Physical Layer classification

Make changes to 33.2.7.2 as follows:

Type 2 PSEs shall provide a maximum of two class events and two mark events. ~~Type 3 and Type 4 PSEs that require more class events for mutual identification than the available power allows may issue a class reset event after performing mutual identification.~~

Type 3 PSEs

— shall . . .

Type 4 PSEs

— shall . . .

Type 3 and Type 4 PSEs that require more class events for mutual identification, or to discover the requested Class of the PD, than the available power allows may issue a class reset event after performing mutual identification or classification.

. . .

All class event voltages and mark event voltages shall have the same polarity as defined for $V_{\text{Port\_PSE-2P}}$ in 33.2.4. Type 3 and Type 4 PSEs may issue classification events on one or both pairsets, when connected to a single-signature PD and operating over 4-pairs.

The PSE shall complete Multiple-Event Physical Layer classification and transition to the POWER_ON, POWER_ON_PRI, or POWER_ON_SEC state without allowing the voltage at the PI or pairset to go below V_{Mark} min, unless in the CLASS_-RESET, CLASS_RESET_PRI, or CLASS_RESET_SEC states. If the PSE returns to the IDLE state, it shall maintain the PI voltage at V_{Reset} for a period of at least T_{Reset} min before starting a new detection cycle. If the PSE is in the CLASS_RESET, CLASS_RESET_PRI, or CLASS_RESET_SEC state it shall maintain the PI or pairset voltage at V_{Reset} for a period of at least T_{Reset} min.

Fixes to the PD state diagram

Info (not part of baseline)

No default value is set for the variable pse_dll_power_type. This causes an invalid comparison on the exit from MDI_POWER1 to MDI_POWER2. Best solution is to set pse_dll_power_type to ‘1’ in the IDLE state.

Append the IDLE state in Figure 33–32 with the statement as follows:

pse_dll_power_type $\Leftarrow$ 1