Comment (Clause **33.2.5.9** Page 66 line 39) Jean address it

The variable class_4PID_mult_events_sec is used in figure 33-21 (Figure 33-21 - Type 3 and Type 4 PSE dual-signature classification state diagram on the ~~Primary~~ Secondary Alternative) but is not defined.

[notes:

1. The typo Secondary instead of Primary is addressed in another comment
2. The variable class_4PID_mult_events_sec is missing also in the updated SM in Picard_03_0116.pdf]

Suggested Remedy

Add at line 39 the following Text:

class_4PID_mult_events_sec

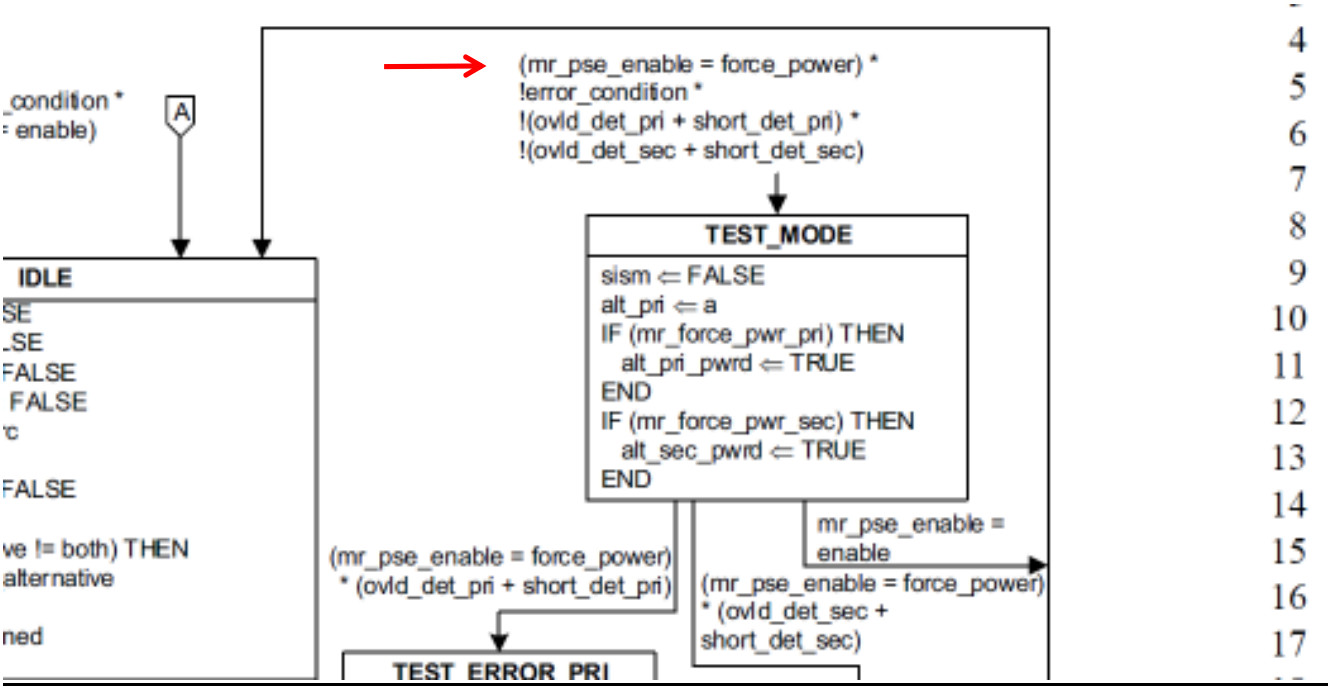
A variable indicating if the PSE uses the method consisting in generating 3 class events to determine if the dual signature PD is a candidate for 4-pair power.

Values:

FALSE: the PSE does not need to generate 3 class events to determine if the PD is a candidate for 4-pair power.

TRUE: the PSE generates at least 3 class events to determine if the PD is a candidate for 4-pair power.

Comment (**33.2.5.11 page 78 line 7, Figure 33-15)**



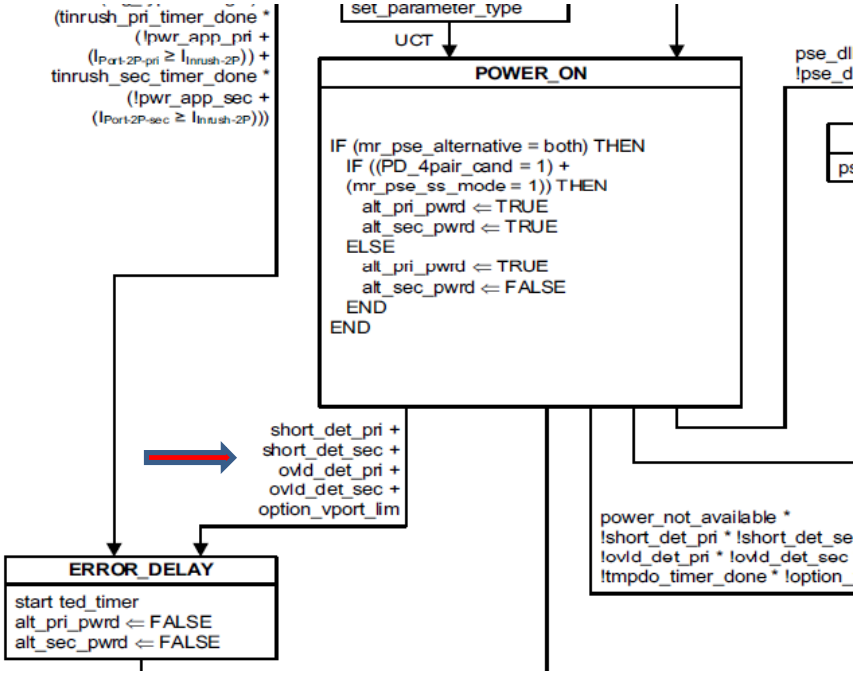
The input to TEST_MODE is incorrect since it is not allow to test each pairset individually.
In addition, overload is optional.

(mr_pse_enable = force_power) *!error_condition *!(ovld_det_pri + short_det_pri) *!(ovld_det_sec + short_det_sec)

Suggested Remedy.

(mr_pse_enable = force_power) *!error_condition *[!(ovld_det_pri + short_det_pri) + !(ovld_det_sec + short_det_sec)]
[The issue of overload is optional is addressed in separate comment.]

Comment (**clause 33.2.5.12,** page 80 line 34**, Figure 33-15**)
[Updates to comment #202]



In the exit from **POWER_ON to ERROR_DELAY** Turning off the power due to overload is optional and not mandatory. According to the state machine it is mandatory.
The current text is: short_det_pri + short_det_sec + ovld_det_pri + ovld_det_sec + option_vport_lim

Suggested Remedy:
Task force to discuss the following 3 options:
Option 1:
[If we remove: + ovld_det_pri + ovld_det_sec it will fix the problem. The text outside the state machine (in 33.2.8.6 Overload current) allows shutting of the power in case of overload"
So if state machine have the priority to set the requirements, the text will clarify the optional features.
The same is correct for legacy state machine in Figure 33-13 page 63 line 51]

1. Change the text to:

short_det_pri + short_det_sec + option_vport_lim

[Consider to apply the concept of this option for Type 1 and 2 SM.]

2. Apply the same solution to legacy state machine in Figure 33-13 page 63 line 51:

Change to: short_detected+ option_vport_lim

[it doesn't considered changing legacy since the feature is optional]

Option 2:

1. To add to the list of Type 3, 4 variables the variables option_ovld_pri and option_ovld_sec with the following values:

option_ovld_pri

A variable that controls the circuitry that the PSE Primary Alternative uses to go to ERROR_DELAY state in case of overload conditions.

Values: FALSE: The PSE is not turning off the Primary Alternative under overload conditions.

TRUE: The PSE is turning off the Primary Alternative under overload conditions.

option_ovld_sec

A variable that controls the circuitry that the PSE Secondary Alternative uses to go to ERROR_DELAY in case of overload conditions.

Values: FALSE: The PSE is not turning off the Secondary Alternative under overload conditions.

TRUE: The PSE is turning off the Secondary Alternative under overload conditions.

1.1 Change the list of conditions to:

short_det_pri + short_det_sec + option_ovld_pri*ovld_det_pri + option_ ovld _sec *ovld_det_sec + option_vport_lim

[Consider to apply the concept of this option for Type 1 and 2 SM.]

2. Update the legacy state machine in Figure 33-13 page 63 line 51:

2.1 To add to the list of Type 1, 2 variables the variable option_ovld with the following values:

option_ovld A variable that controls the circuitry that the PSE uses to go to ERROR_DELAY state in case of overload conditions.

Values: FALSE: The PSE is not turning off the power under overload conditions.

TRUE: The PSE is turning off the power under overload conditions.

2.2 Change the list of conditions to:

short_detected+ option_ovld *ovld_detected + option_vport_lim

Option 3:

To keep the state machine as it is and to add the following text at the beginning of the state machine clause:

”State machine has priority over text unless a waver is explicitly specified.” or equivalent wording.

[This will allow simplifications of state machine description in many cases including PD section.]

Comment (33.2.5.12, page 85 line 22**, Figure 33-19**):

[Updates to comment #200]
[This comment addresses comment #181 as well.]

The objective of this comment is to allow PSE to do class reset any time within Tpon to generate 1st classification sequence to do some testing such:

a) Checking unbalance with classification voltage
b) In addition to other ways to know if single-signature PD is 4-pairs capable i.e. the 4PID check, we can positively know it by finding the PD class code by generating 3 classifications events and if there just power for Type 1 the PSE can redo classification by issuing one class event.

Allowing doing class reset during Tpon doesn't supported in Figure 33-19 as it does in dual-signature classification state diagram in figures 33-20 and 33-21.

In addition, there is a need to allow generate 1 class event if PSE knows that the power available in Type 1 without the need to know what is the PD requested power.
The above was meant to increase PSE design flexibility.

Suggested Remedy:

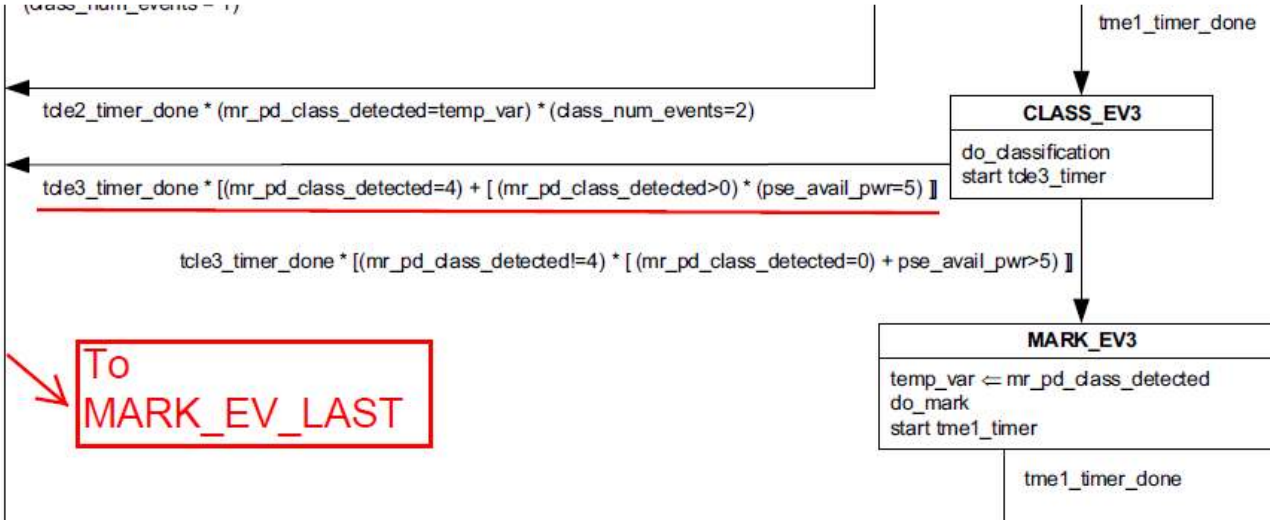
To add the following text to classification section page 97 line 30:
"PSE is allowed to reset the PD classification during class event sequence and redo its classification sequence at any time between the end of detection and POWER_UP time duration (Tpon) without redoing connection check and detection."

or equivalent wording.

To add the following Editor Notes:
"Editor Note: To add in Figure 33-19 the ability to reset classification after at least 1 classification events with long first class event or with short first class event without doing connection check and detection again when Tpon is still not done."

"Editor Note: To add in Figure 33-19 the ability generate 1 class event if PSE knows that the power available is Type 1 without the need to know what is the PD requested power."

Comment (33.2.5.12, page 85 line 22, **Figure 33-15**)



In the exit from CLASS_EV3 to MARK_EV_LAST:

tcle3_timer_done * [(mr_pd_class_detected=4) + [(mr_pd_class_detected>0) * (pse_avail_pwr=5)]]

The statement is true if:

a) tcle3_timer_done *(mr_pd_class_detected=4) +
b) tcle3_timer_done *(mr_pd_class_detected>0) * (pse_avail_pwr=5)

According to Table 33-11, for pse_avail_pwr=5, the number of classification events is 4 and this exist is coming from CLASS_EV3.

The suggested fix is:

The statement is true if:

tcle3_timer_done *(mr_pd_class_detected=4) +
tcle3_timer_done *(mr_pd_class_detected>0) * (pse_avail_pwr=4)

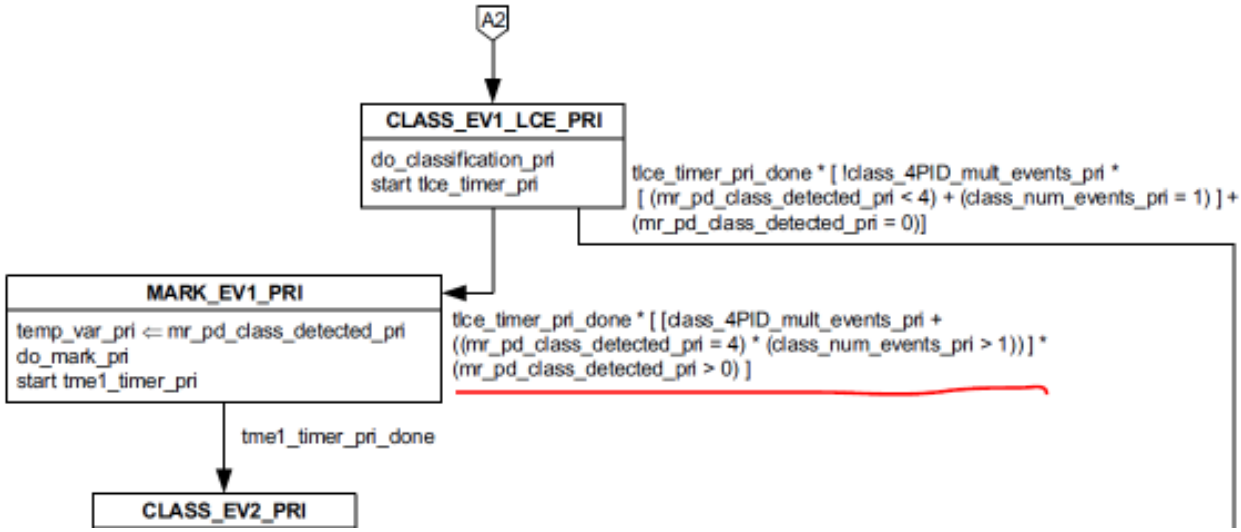
Suggested Remedy

tcle3_timer_done *(mr_pd_class_detected=4) +
tcle3_timer_done *(mr_pd_class_detected>0) * (pse_avail_pwr= 4)=

=tcle3_timer_done * [(mr_pd_class_detected=4) + [(mr_pd_class_detected>0) * (pse_avail_pwr=4)]]

Comment (clause 33.2.5.12, page 86 line 10, , Figure 33-20)

[Updates to comment #231]



In the following text of the exit from CLASS_EV1_LCE_PRI to MARK_EV1_PRI:

tlce_timer_pri_done * **[** [class_4PID_mult_events_pri +((mr_pd_class_detected_pri = 4) * (class_num_events_pri > 1))] *(mr_pd_class_detected_pri > 0)]

The conditions are true if:

a) tlce_timer_pri_done * class_4PID_mult_events_pri *(mr_pd_class_detected_pri > 0) which means tlce timer is done and PSE uses 3 class event for 4PID and it has to be true when the first class signature>0 since it is dual-signature PD that starts with class signature 1 to 5.

b) tlce_timer_pri_done *((mr_pd_class_detected_pri = 4) * (class_num_events_pri > 1)) *(mr_pd_class_detected_pri > 0)

There are 2 issues:

1. Redundant round parenthesis in the part: +((mr_pd_class_detected_pri = 4) * (class_num_events_pri > 1))
2. Why in part (b) we need (mr_pd_class_detected_pri > 0) if we have (mr_pd_class_detected_pri = 4).

Suggested Remedy:

Change from:

tlce_timer_pri_done * **[** [class_4PID_mult_events_pri +((mr_pd_class_detected_pri = 4) * (class_num_events_pri > 1))] *(mr_pd_class_detected_pri > 0)]

To:

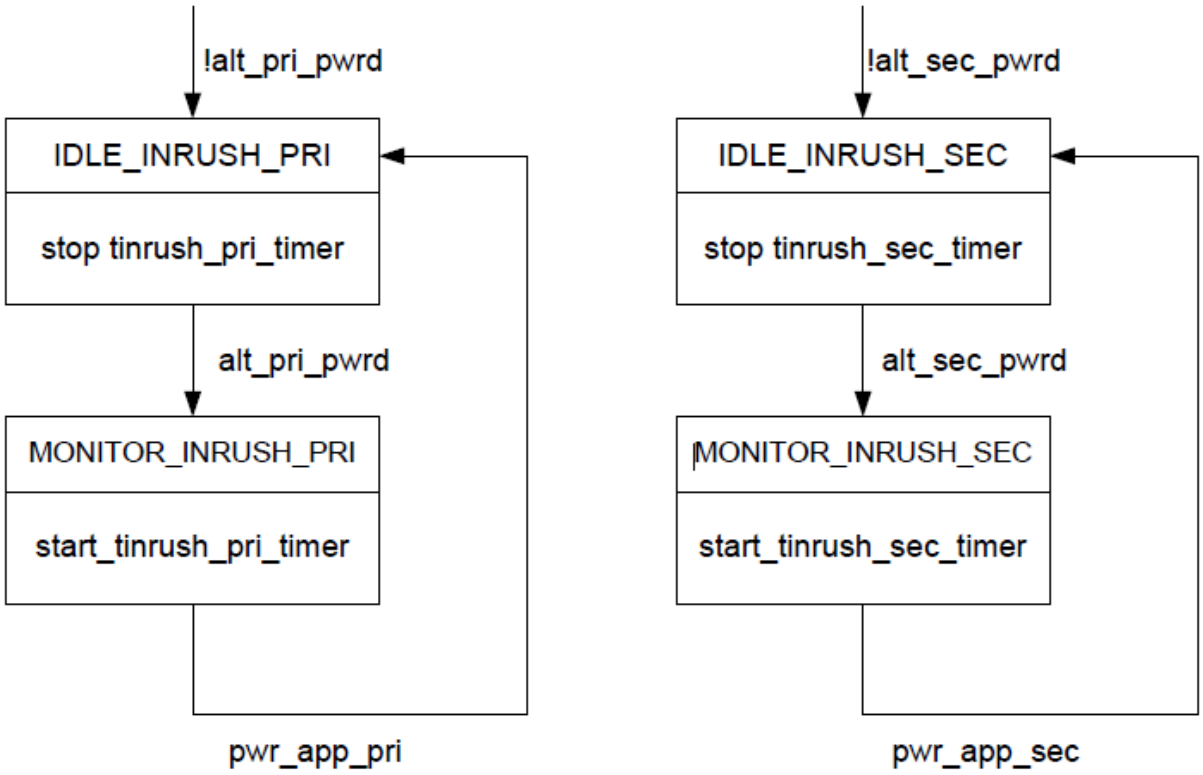
tlce_timer_pri_done * class_4PID_mult_events_pri *(mr_pd_class_detected_pri > 0)+
tlce_timer_pri_done *(mr_pd_class_detected_pri = 4) * (class_num_events_pri > 1) =

= tlce_timer_pri_done*
[class_4PID_mult_events_pri *(mr_pd_class_detected_pri > 0) +(mr_pd_class_detected_pri = 4)*(class_num_events_pri > 1)]

Comment (clause 33.2.5.12, page 88 line 25, Figure 33-23)

[Comment #229]

See darshan_08_0316.pdf for new Figure 33-23.
Figure 33–23—Type 3 and Type 4 inrush monitor state diagram does not reflect the case where POWER_UP for ALT A and ALT B may be done in different time and not simultaneously.

Suggested Remedy:

Replace Figure 33–23 as proposed in darshan_08_0316.pdf for new Figure 33-23.