Contents

2 ..... Comments
3 ..... Proposed Remedy
6 ..... Annex A (145.2.5.7, Page 142 Line 3)
6 ..... Annex C: (145.2.5.7, P142, L9 and Page 143, L23)
7 ..... Annex D – Derivation or power demotion logics for the primary
7 ..... Annex E – Derivation or power demotion logics for the secondary
8 ..... Annex F – Issues related to the use of temp_var_pri per stover_02_1117.pdf.

Comments

(r01-313, **145.2.5.7, P142, L3)**
(r01-312, **145.2.5.7, P142, L7)**
(r01-317, **145.2.5.7, P143, L7, L10)**
(r01-391, **145.2.5.7, P143, L23, L26)**
(r01-484, **P144, L23, L10)**

In D3.1 we add the following functionality, and after simulating some parts and analyzing the changes we did, we found some errors in state machine and variable definitions that need to be corrected.

The intent of adding CLASSIFICATION_PRI, CLASS_PROBE_PRI and the exits to IDLE_PRI and CLASS_RESET_PRI was to allow the following functionality:

a) Allowing detection + class event cycles until host decides to powerup. This seems working.

b) To use the class code detected during do_class_prob_pri for 4PID as well. This seems working too.

c) The definition of option_class_probe is incorrect to be used for dual-signature PD. This variable is defined as to be used when pse_avail_pwr_pri < 4 which make it impossible to have an exit from CLASS_PROBE_PRI with a condition pse_avail_pwr_pri≥4. As a result, we need to use different variables for single and dual signature and differentiates between primary and secondary. *There are additional reasons to differentiate between option_class_probe for primary and secondary*. See Annex A for details.

d) This item was deleted.

e) The intent when doing CLASS_PROBE_PRI with 3 short class events when pse_avail_pwr_pri<4 was to go to class reset and then to single long class event and then power up. Now in D3.1, If option_class_probe is set, state machine will either go to IDLE_PRI or to CLASS_RESET_PRI and from CLASS_RESET_PRI eventually to IDLE_PRI instead of continuing towards MARK_EV_LAST_PRI and powering the port while allowing powering with only single long class event and not twice. This is not working. See Annex C for details.

f) Missing logic for power demotion in the exits from CLASS_EVAL_PRI (and CLASS_EVAL_PRI). See Annex D.

g) All the above issues are the same for the secondary as well.

h) Clause 145.2.5.6, page 131 line 5: Definition of do_class_reset function is missing for the high-level state machine.

i) In the functions do classification, do_classification_pri and declassification's, there are missing description of the need to perform class event etc.

Proposed Remedy

Baseline starts here

1. Add the following variables to the variable list in 145.2.5.4:

option_class_probe_pri
This variable indicates if the PSE should determine the PD requested Class on the Primary Alternative by issuing 3 class events. When set to TRUE, the PSE will issue 3 class events to determine the PD requested Class, perform a classification reset by applying Verse for at least Tre set to the PI (see Table 145– 14), 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.

option_class_probe_sec
This variable indicates if the PSE should determine the PD requested Class on the Secondary Alternative by issuing 3 class events. When set to TRUE, the PSE will issue 3 class events to determine the PD requested Class, perform a classification reset by applying VReset for at least TReset to the PI (see Table 145– 14), 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.

do_class_reset
This function produces the classification reset voltage; See VReset in Table 145–14. This function does not return any variables.

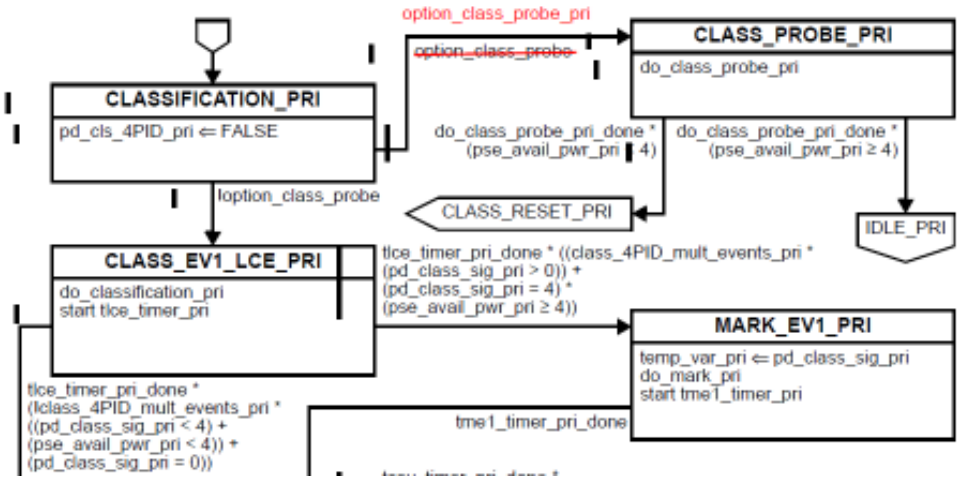
2. Make the following modifications in the following functions:

do_classification This function produces the classification event voltage and determines the PD's class signature. This function returns the following variable……."

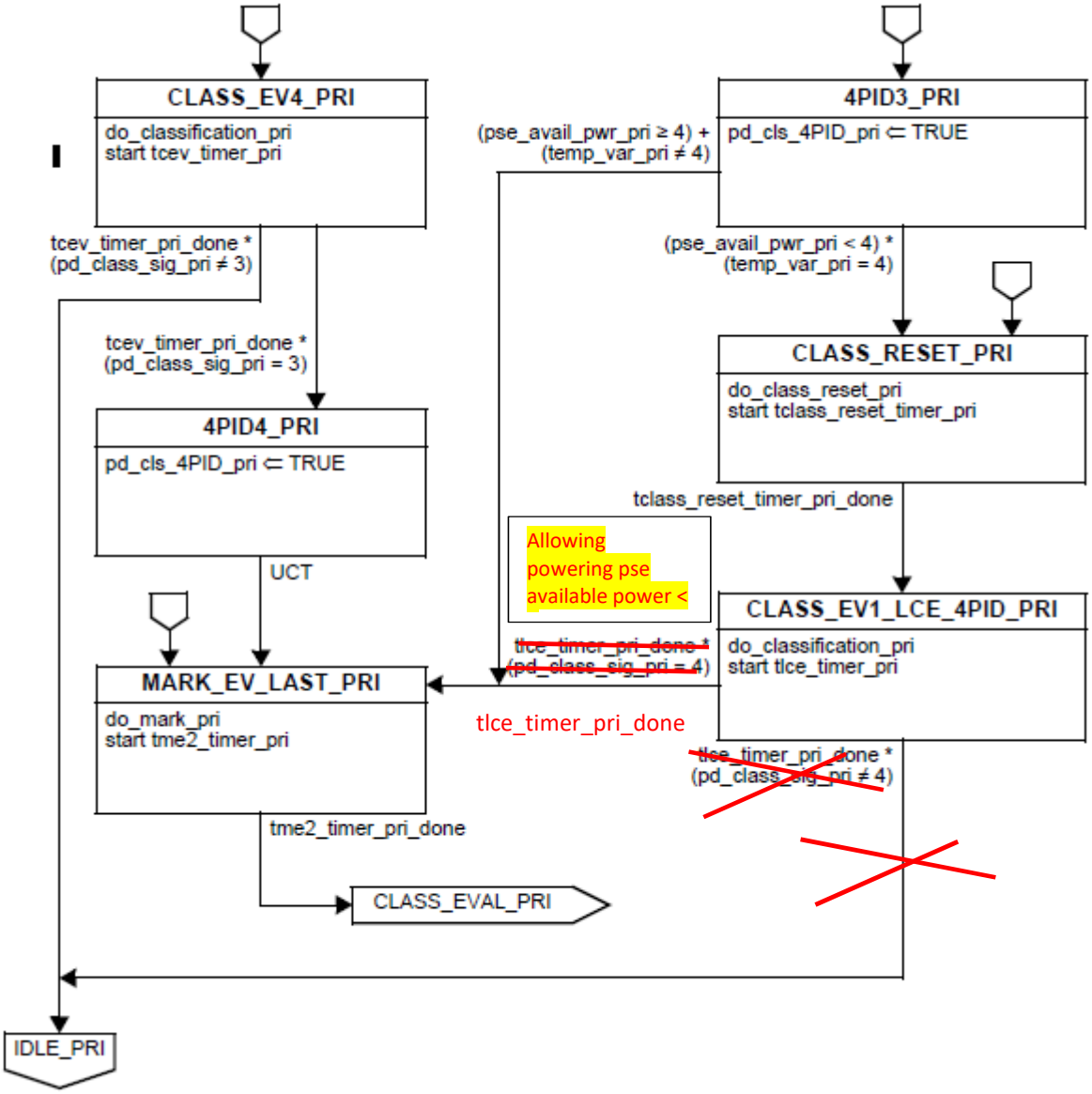
do_classification_pri This function produces the classification event voltage and determines the PD's class signature for the Primary Alternative. This function returns the following variables for the Primary Alternative:

do_classification_sec This function produces the classification event voltage and determines the PD's class signature for the Secondary Alternative. This function returns the following variables for the Primary Alternative:

3. change the following state machine for the primary and secondary as shown for the primary.

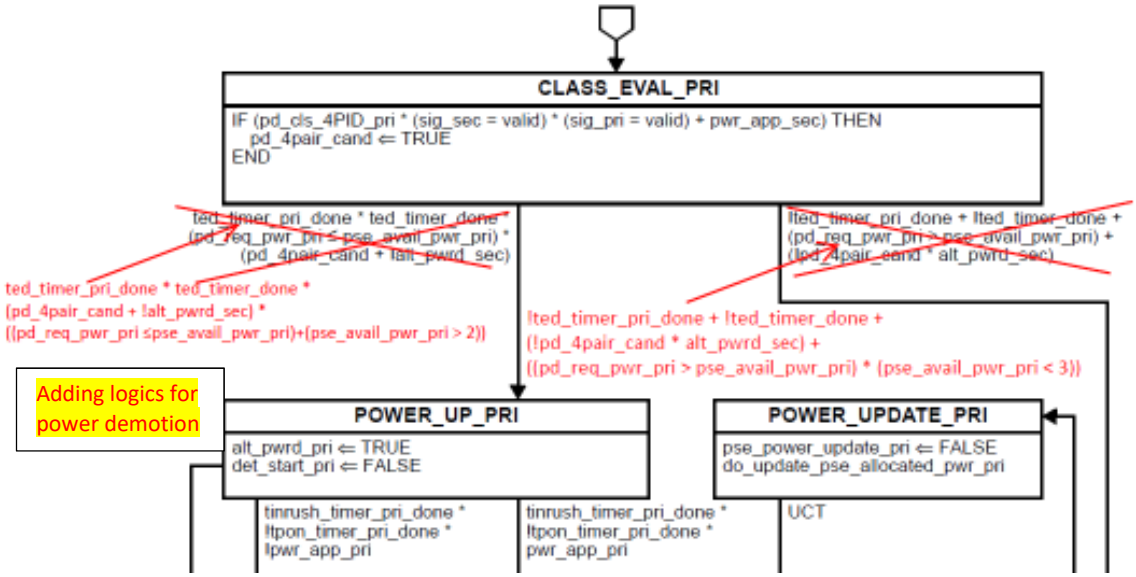


4. change the following state machine for the primary and secondary as shown for the primary.

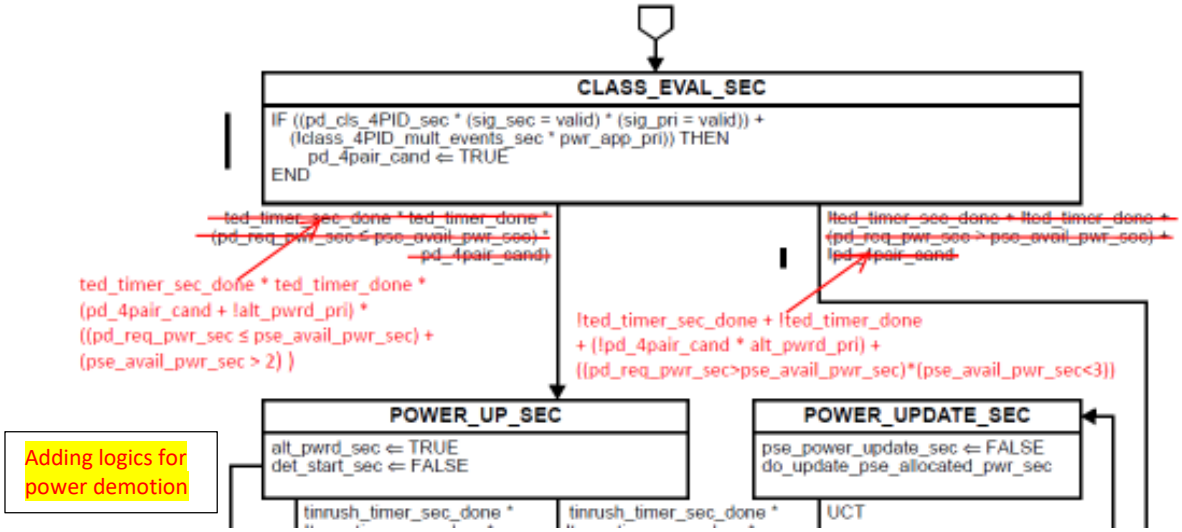


5. change the following state machine for the primary.

3.



6. change the following state machine for the secondary.



END of Baseline

Annex A (145.2.5.7, Page 142 Line 3)

1. In the definition of option_class_probe it says that this variable is used only when pse_avail_pwr_pri<4. As a result, it is not possible that we will have exit from CLASS_PROBE_PRI to IDLE_PRI with a condition pse_avail_pwr_pri ≥ 4 since it will never happen in this route! As a result, we need to use different option_class_probe for dual-signature since the current option_class_probe variable definition is only good for single-signature.

1-1) As a result of (1) we need new variables for option_class_probe_pri and option_class_probe _sec with a definition that **should not** be depend in pse_avail_pwr_pri<4.

2) It is possible that for the primary alternative the available power will be <4 and for the secondary the available power >4. Therefore, the option_class_probe need to be separate for primary and secondary as recommended above.

Proposed Remedy:

From all the above arguments, it is proposed to differentiate between option_class_probe_variable used in the single-signature PSE SM part and the dual-signature PSE SM part by using option_class_probe_pri/sec with a bit different definition compared to option_class_probe. See variable list in the base line for details.

Annex B was deleted from previous revision.

Annex C: (145.2.5.7, P142, L9 and Page 143, L23)

If option_class_probe is set and pse_avail_pwr_pri<4, the state machine will either go to IDLE_PRI or to CLASS_RESET_PRI and from CLASS_RESET_PRI through CLASS_EV1_LCE_4PID_PRI eventually to IDLE_PRI instead of continuing to MARK_EV_LAST_PRI towards powering the port, which is an error (The intent was to allow powering while doing only single long class event and not twice as we had in D3.0. Now it will not power at all regardless the route we take from CLASS_EV1_LCE_4PID_PRI.
Details (Staring from page 142):
From CLASSIFICATION_PRI to CLASS_PROBE_PRI option_class_probe_is TRUE.
From CLASS_PROBE_PRI to CLASS_RESET_PRI pse_avail_pwr_pri<4.
Moving to CLASS_RESET_PRI on page 143 and then continue to CLASS_EV1_LCE_4PID_PRI.
Now we have two exits:
Exit 1: if pd_class_sig_pri=4 we go to MARK_EV_LAST_PRI and then to CLASS_EVAL_PRI on page 144. Now, if we want to POWER_UP_PRI we need to meet the condition pd_req_pwr_pri<=pse_avail_pwr_pri. Since pse_avail_pwr_pri<4 which means pse_avail_pwr_pri=3 which means PSE supports only class 3 or lower and pd_class_sig_pri=4 means PD class is 4 or 5 (4,4,0 is class 4. 4,4,3 is class 5), we will fail the exit from CLASS_EVAL_PRI to POWER_UP_PRI due to (pd_req_pwr_pri<pse_avail_pwr_pri) which will end with POWER_DENIDED_PRI *(in addition to the fact that the logic for power demotion is missing and is addressed in separate item in this document)*.
Exit 2: If pd_class_sig_pri ≠ 4, which means PD class is 3 or lower, we go directly to IDLE_PRI which will also fail to power which is an error since surely, we can power all classes below class 4 starting with pd_class_sig_pri=3 or pd_class_sig_pri=2 etc.
The result is no matter what we do when available power<4 (and PD class is 1-3) we can't go to MARK_EV_LAST and continue towards power up!
The solution: To flip the exits from CLASS_EV1_LCE_4PID_PRI to allow powering PDs with class 1 to 3 that matches pse_avail_pwr_pri<4 after doing CLASS_RESET_PRI.

Proposed Remedy:

1. **Change the exit from CLASS_EV1_LCE_4PID_PRI to MARK_EV_LAST_PRI from:**

tlce_timer_pri_done * (pd_class_sig_pri = 4) to: tlce_timer_pri_done * (pd_class_sig_pri ≠ 4)

2. **Change the exit from CLASS_EV1_LCE_4PID_PRI to IDLE_PRI from:**

tlce_timer_pri_done * (pd_class_sig_pri = 4) to: tlce_timer_pri_done * (pd_class_sig_pri = 4).

Annex D – Derivation or power demotion logics for the primary

Looking at the high-level single-signature state machine and use the logics for power demotions with the necessary changes for dual-signature.

1. Change the exit from CLASS_EVAL_PRI to POWER_DENIGED_PRI from:

!ted_timer_pri_done + !ted_timer_done + (pd_req_pwr_pri > pse_avail_pwr_pri) + (!pd_4pair_cand * alt_pwrd_sec)

To:

!ted_timer_pri_done + !ted_timer_done +
(!pd_4pair_cand * alt_pwrd_sec) +
((pd_req_pwr_pri > pse_avail_pwr_pri) * (pse_avail_pwr_pri < 3))

2. Change the exit from CLASS_EVAL_PRI to POWER_UP_PRI from:

ted_timer_pri_done * ted_timer_done * (pd_req_pwr_pri ≤ pse_avail_pwr_pri) * (pd_4pair_cand + !alt_pwrd_sec)

To:

ted_timer_pri_done * ted_timer_done *
(pd_4pair_cand + !alt_pwrd_sec) *((pd_req_pwr_pri ≤ pse_avail_pwr_pri) + (pse_avail_pwr_pri > 2))

Annex E – Derivation or power demotion logics for the secondary

1. Change the exit from CLASS_EVAL_SEC to POWER_DENIGED_SEC from:

!ted_timer_sec_done + !ted_timer_done + (pd_req_pwr_sec > pse_avail_pwr_sec) + !pd_4pair_cand

To:

!ted_timer_sec_done + !ted_timer_done + (!pd_4pair_cand * alt_pwrd_pri) +
((pd_req_pwr_sec > pse_avail_pwr_sec) * (pse_avail_pwr_sec < 3))

2. Change the exit from CLASS_EVAL_SEC to POWER_UP_SEC from:

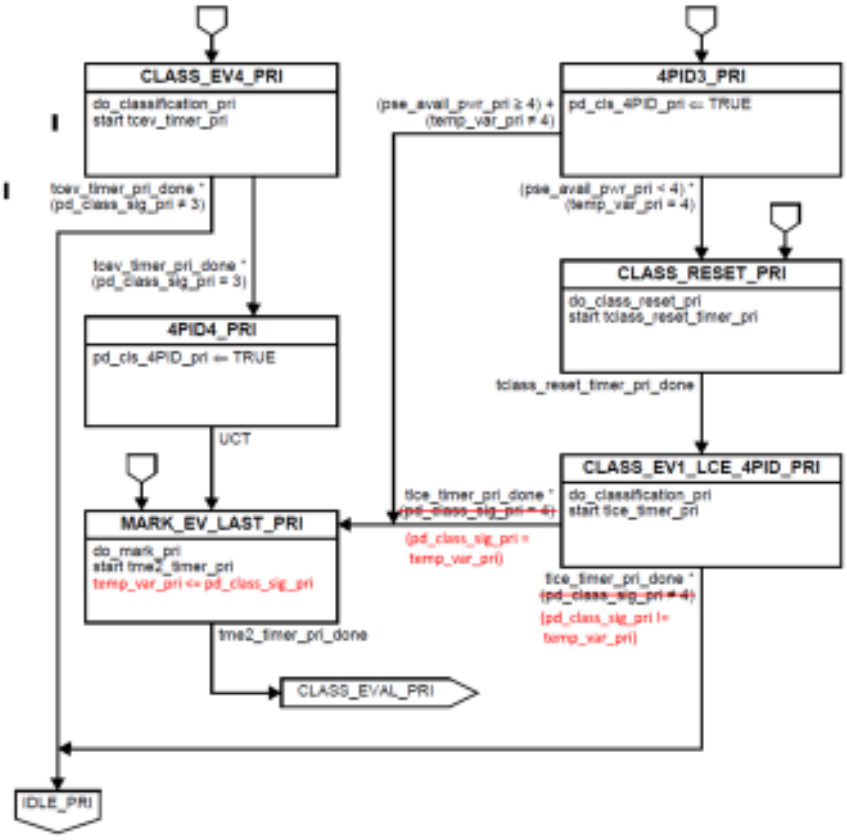
ted_timer_sec_done * ted_timer_done * (pd_req_pwr_sec ≤ pse_avail_pwr_sec) * pd_4pair_cand)

To:

ted_timer_sec_done * ted_timer_done *
(pd_4pair_cand + !alt_pwrd_pri) *((pd_req_pwr_sec ≤ pse_avail_pwr_sec) + (pse_avail_pwr_sec > 2))

Annex F – Issues related to the use of temp_var_pri per stover_02_1117.pdf.

See: http://www.ieee802.org/3/bt/public/nov17/stover_02_1117.pdf



3 REASONS WHY WE HAVE PROBLEM TO USE THE ABOVE PROPOSED SOLUTION:

1. In MARK_EV_LAST_PRI, adding the assignment, temp_var_pri <== pd_class_sig_pri will not solve the problem. This condition will not be used in the state machine after MARK_EV_LAST_PRI (when going to CLASS_EVAL_PRI and afterwards), therefore it is redundant.

2. The temp_var_pri current value in CLASS_EV1_LCE_4PID_PRI has the value of pd_class_sig_pri=4 if we got to CLASS_EV1_LCE_4PID_PRI from 4PID3_PRI, resulting with no possibility to continue MARK_EV_LAST_PRI which is an error. Please note that temp_var_pri is updated only once as a result of CLASS_EV1_LCE_PRI and will be assigned once in MARK_EV1_PRI.

3. In the exit from CLASS_EV1_LCE_4PID_PRI to MARK_EV_LAST_PRI the use of temp_var_pri is not possible since it contains undefined value due to the following:
   We got to CLASS_EV1_LCE_4PID_PRI from CLASS_RESET_PRI and before it we were in CLASS_PROBE_PRI.
   In CLASS_PROBE_PRI we execute the function do_class_probe_pri which doesn't return pd_class_sig_pri. It returns only pd_req_pwr_pri and pd_cls_4PID_pri. It means that temp_var_pri will not get a value.
   (When running simulation, we got "use of a variable before initialization" warning. Now, even if CLASS_PROBE_PRI was returning pd_class_sig_pri, due to reason (2)).