

# Contents

2..... Comments

3..... Proposed Remedy

5.....PD state machine

7..... Annex A (145.2.5.7, Page 142 Line 3)

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

8..... Annex D – Derivation or power demotion logics for the primary

8..... Annex E – Derivation or power demotion logics for the secondary

9.....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  $pse\_avail\_pwr\_pri \geq 4$ . 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 do\_classification\_sec, there are missing description of the need to perform class event etc.
- j) In the PD state machine, in the state DO\_MARK\_EVENT2 the assignment  $pse\_power\_level\_mode(X)=4$  is only TRUE if PD requested class is  $\geq 4$



## 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 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.

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:***

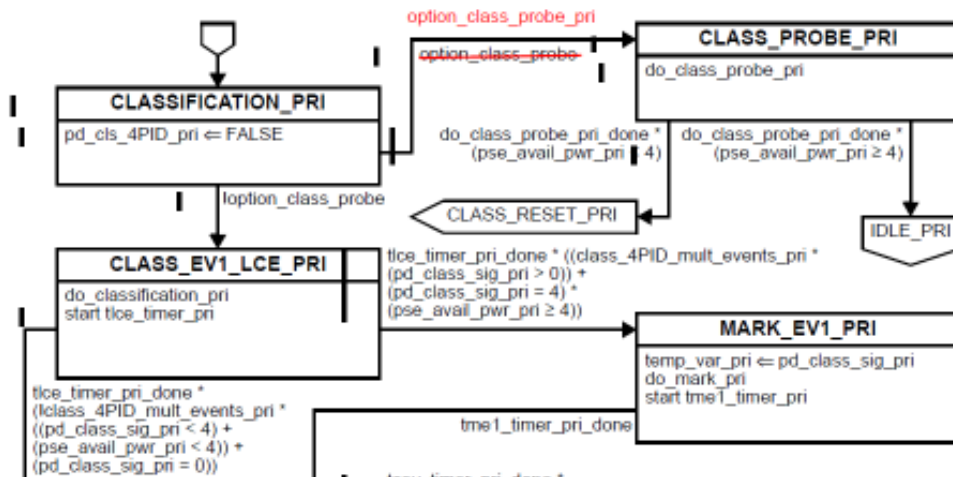
do\_classification [This function produces the classification event voltage and determines the PD's class signature.](#) This function returns the following variable:....."

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:

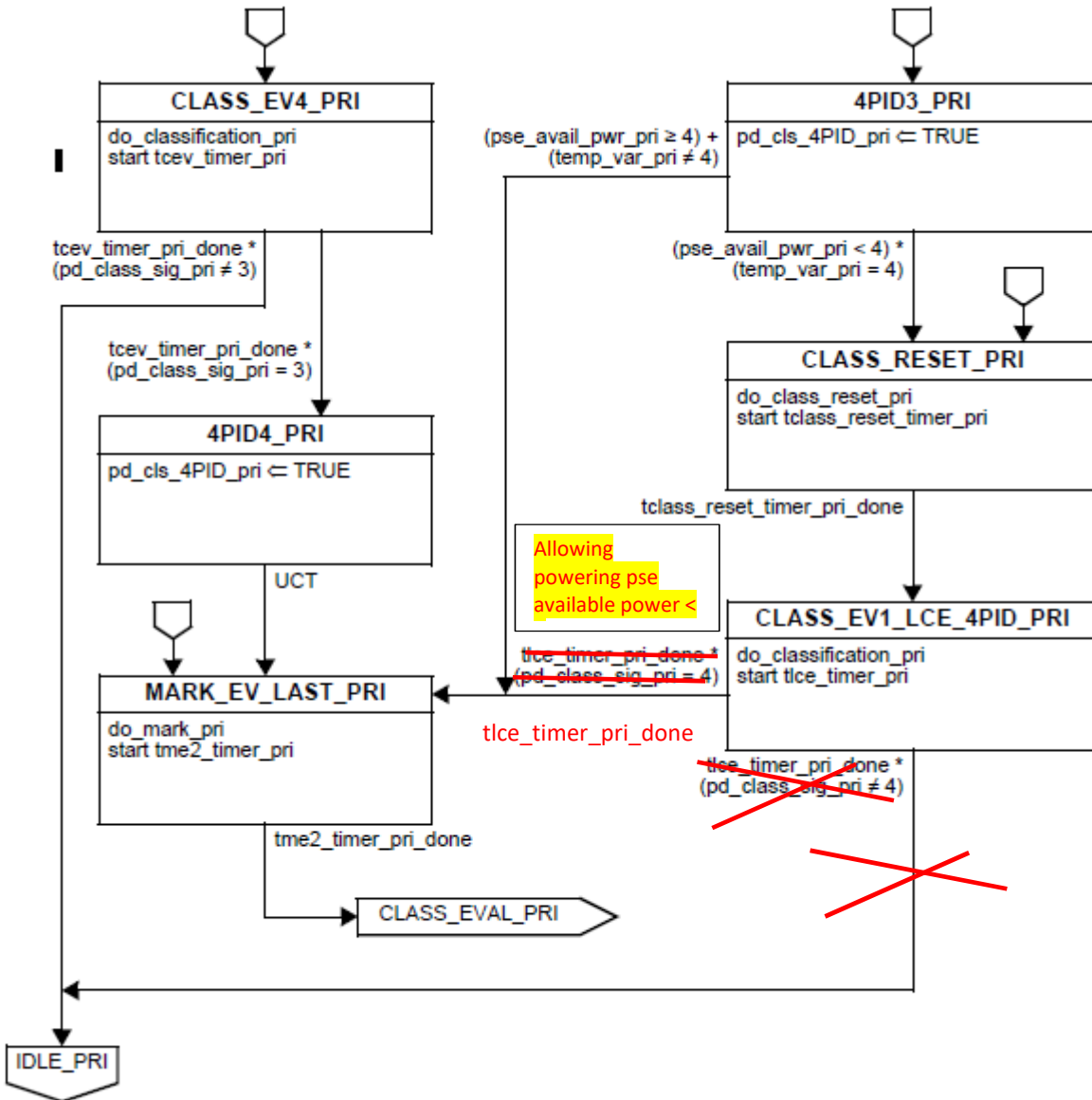
do\_classification\_sec [This function produces the classification event voltage and determines the PD's class signature for the Secondary Alternative.](#) This function returns the following variables for the Primary Alternative:



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

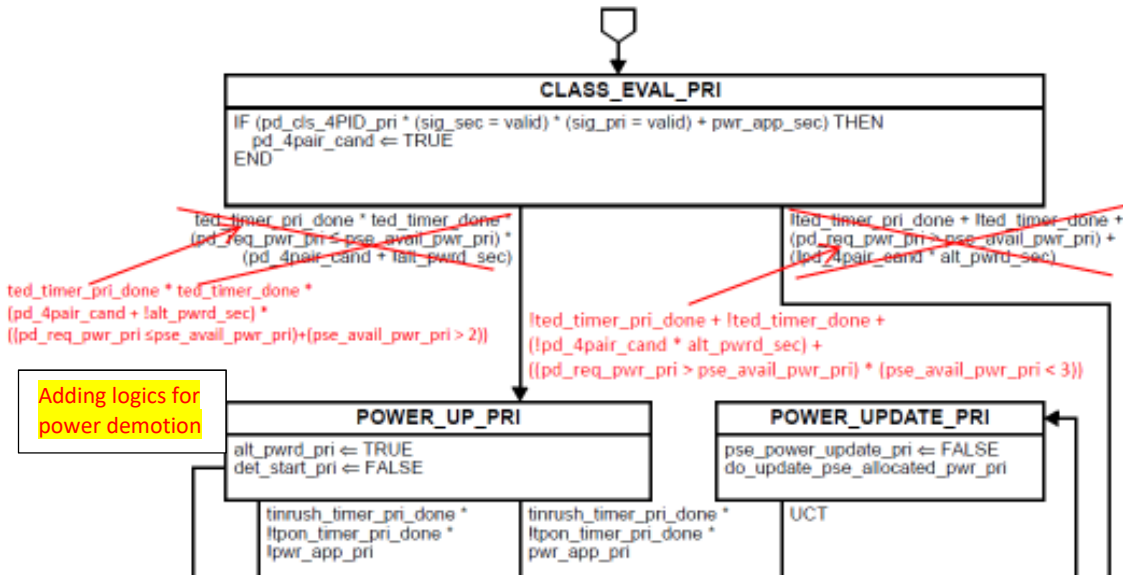


4. change the following state machine for the primary and secondary as shown for the primary.

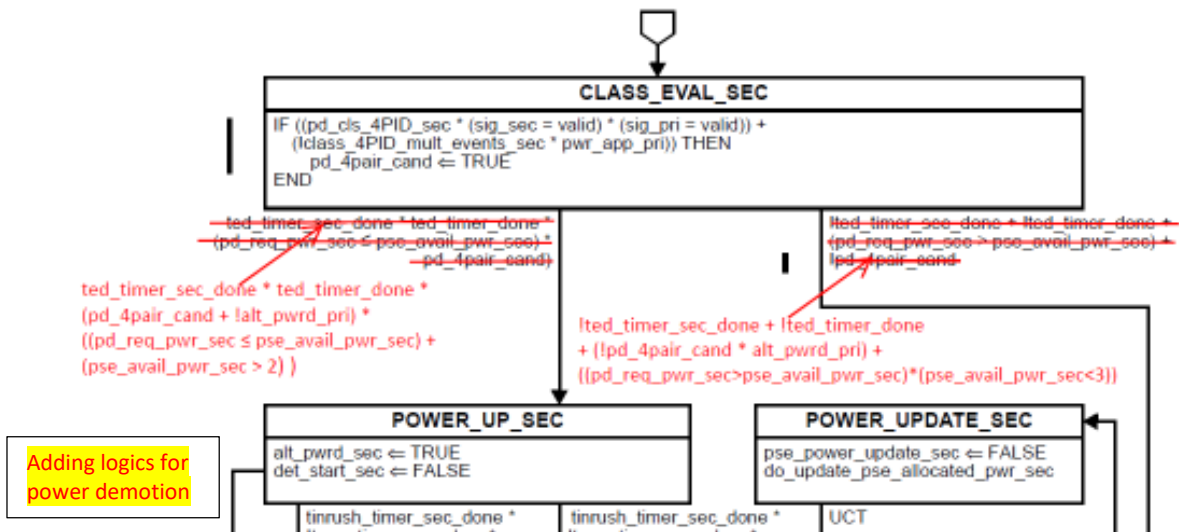


5. change the following state machine for the primary.

3.



6. change the following state machine for the secondary.



PD state machine

**Comment (Clause 145.3.3.12 P189 L32)**

In the state DO\_MARK\_EVENT2 the assignment `pse_power_level_mode(X)=4` is only TRURE if PD requested class is  $\geq 4$ .

**Proposed Remedy**

Change "`pse_power_level_mode(X)=4`"

To:

IF `pd_req_class_mode(X) ≥ 4` THEN

`pse_power_level_mode(X)= 4`

END



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 (Starting 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\_DENIED\_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:

$!ted\_timer\_pri\_done + !ted\_timer\_done + (pd\_req\_pwr\_pri > pse\_avail\_pwr\_pri) + (!pd\_4pair\_cand * alt\_pwr\_sec)$

To:

$!ted\_timer\_pri\_done + !ted\_timer\_done +$   
 $(!pd\_4pair\_cand * alt\_pwr\_sec) +$   
 $((pd\_req\_pwr\_pri > pse\_avail\_pwr\_pri) * (pse\_avail\_pwr\_pri < 3))$

### 2. Change the exit from CLASS\_EVAL\_PRI to POWER\_UP\_PRI from:

$ted\_timer\_pri\_done * ted\_timer\_done * (pd\_req\_pwr\_pri \leq pse\_avail\_pwr\_pri) * (pd\_4pair\_cand + !alt\_pwr\_sec)$

To:

$ted\_timer\_pri\_done * ted\_timer\_done *$   
 $(pd\_4pair\_cand + !alt\_pwr\_sec) * ((pd\_req\_pwr\_pri \leq pse\_avail\_pwr\_pri) + (pse\_avail\_pwr\_pri > 2))$

## 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:

$!ted\_timer\_sec\_done + !ted\_timer\_done + (!pd\_4pair\_cand * alt\_pwr\_pri) +$   
 $((pd\_req\_pwr\_sec > pse\_avail\_pwr\_sec) * (pse\_avail\_pwr\_sec < 3))$

### 2. Change the exit from CLASS\_EVAL\_SEC to POWER\_UP\_SEC from:

$ted\_timer\_sec\_done * ted\_timer\_done * (pd\_req\_pwr\_sec \leq pse\_avail\_pwr\_sec) * pd\_4pair\_cand$

To:

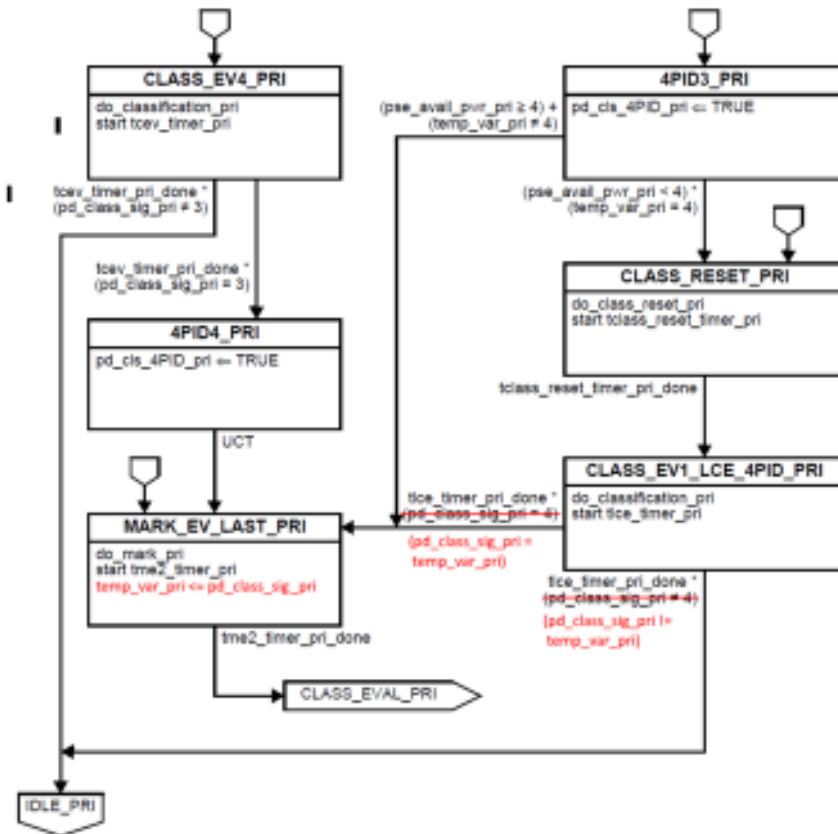
$ted\_timer\_sec\_done * ted\_timer\_done *$   
 $(pd\_4pair\_cand + !alt\_pwr\_pri) * ((pd\_req\_pwr\_sec \leq pse\_avail\_pwr\_sec) + (pse\_avail\_pwr\_sec > 2))$





# 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](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) ).

