## Comment (#85, #100, #101, #125, #127, #132, #133, #134, #136, #137 )

(TDL from D2.2 Comments:#185, #358, #143 )

Single signature state machines

1. PSE state machine when supporting dual-signature needs some updated based on the changes made for D2.2 per the changes made by yseboodt_02_0117_lldpupdate made for single-signature with the necessary updates for dual-signature.
2. Some corrections need to be made for the single-signature variable list.
3. DLL_ENABLE state in the PSE was deleted and replaced by compact IF statement.
4. Due to the decision to keep clause 33 for Type 1 and 2 as it is and add new clause for Type 3 and 4, some deletions where made.
5. A TODO list action item was added to set some TLV fields to 0 before PD is going to IDLE.

Dual-single signature state machines

6. DLL_ENABLE state in dual-signature state machine was deleted and replaced by compact IF statement that support new and legacy devices. This approach make DLL mandatory for power level >3 as it was and <3 if DLL is capable.
7. Some corrections in variable definitions where made.
8. Due to the decision to keep clause 33 for Type 1 and 2 as it is and add new clause for Type 3 and 4, some deletions where made.
9. The suffix “M” was replaced with the suffix “X” to prevent confusion with the word Mode.
10. To change in Table 145-12 from “_mode(M)” to “_Alt(M)”. Two locations.
11. A TODO list action item was added to set some TLV fields to 0 before mode(X) is going to IDLE.

## Suggested Remedy (See next page):

Suggested Remedy:

**Baseline starts here:**

[PSE section]

***1. Update according to the following proposed baseline.***

***2. Whenever there is suffix "M", change it to"X" in mode(M) and Alt(M).***

***3. Change in*** Clause 145.2.7 Table 145-12, page 140 line 4: ***from "_mode(M)" to "_Alt(M)". Two locations.***

145.2.5.4 Variables

***Add the following variables:***

pse_power_update_pri

A variable that is set when the PSEAllocatedPowerValue_Alt(X) in the DLL state diagram in Figure 145–47 has been updated.

Values:

FALSE: The value of PSEAllocatedPowerValue_Alt(X) has not changed.

TRUE: The value of PSEAllocatedPowerValue_Alt(X) has changed.

pse_power_update_sec

A variable that is set when the PSEAllocatedPowerValue_Alt(X) in the DLL state diagram in Figure 145–47 has been updated.

Values:

FALSE: The value of PSEAllocatedPowerValue_Alt(X) has not changed.

TRUE: The value of PSEAllocatedPowerValue_Alt(X) has changed.

***Update the following variables:***

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 4, this variable may not contain the actual requested Class by the ~~PSE~~PD; see pq_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

145.2.5.6 Functions

***Change the following functions:***

do_update_~~pd~~pse_allocated_pwr

A function that updates the ~~pd~~pse allocated value based on the value of PSEAllocatedPowerValue as defined in Table 145–12. This function returns the following variable:

~~pd~~pse _allocated_pwr:

this variable indicates the Class 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

***Add the following functions:***

do_update_pse_allocated_pwr_pri

A function that updates the pse_allocated_pwr_Alt(X) based on the value of PSEAllocatedPowerValue_mode(X) as defined in Table 145–12. This function returns the following variable:

pse_allocated_pwr_pri: this variable indicates the Class assigned to the PD.

Values:

1: Class 1
2: Class 2
3: Class 3
4: Class 4
5: Class 5

do_update_pse_allocated_pwr_sec

A function that updates the pse_allocated_pwr_Alt(X) based on the value of PSEAllocatedPowerValue_mode(X) as defined in Table 145–12. This function returns the following variable:

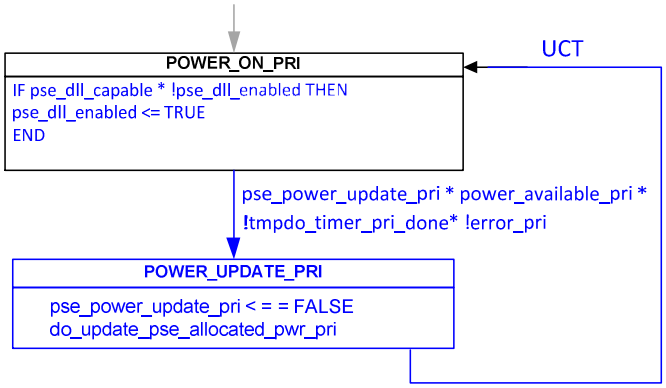
pse_allocated_pwr_sec: this variable indicates the Class assigned to the PD.

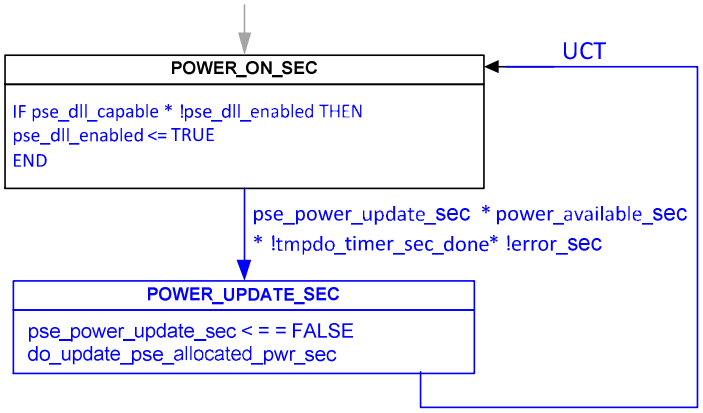
Values:

1: Class 1
2: Class 2
3: Class 3
4: Class 4
5: Class 5

145.2.5.7 State diagram

***1. Delete DLL_ENABLE states and the in/out arrows to I and Change Figure 145-15 and Figure 145-16 as follows:***



# [PD SECTION : SINGLE-SIGNATURE]

## 145.3.3 PD state diagram

***-Editor to fix Figure 145-26, 145-27 and 145-28 numbering in order to make the single-signature PD two parts state machine with single number and update the references to it accordingly.***

***-Editor to fix Figure 145-29, 145-30 numbering in order to make the dual-signature PD two parts state machine with single number and update the references to it accordingly.***

***- Make the changes following changes:***

The PD state diagram specifies the externally observable behavior of a PD. Single-signature Type 3 and Type 4 PDs shall provide the behavior of the state diagram show in Figure 145-26.

Dual-signature Type 3 and Type 4 PDs shall provide the behavior of the state diagram shown in Figure 145-29 over each pairset independently unless otherwise specified. All the parameters that apply to Mode A and Mode B are denoted with the suffix "_~~mode(M)~~mode(X)" where "~~M~~X" can be "A" or "B". A parameter that ends with the suffix "_~~mode(M)~~mode(X)" may have different values for Mode A and Mode B.

**145.3.3.4 Single-signature variables**

pd_reset

An implementation-specific control variable that unconditionally resets the PD state diagram to the OFFLINE state.

Values:

FALSE: The device has not been reset ~~(default)~~.

TRUE: The device has been reset.

pd_undefined

A control variable that indicates that the PD is in an undefined condition. The PD may or may not show a valid or invalid detection signature, may or may not draw mark current, may or may not draw any class current, may or may not show MPS and may change the pse_power_level variable.

Values:

FALSE: The PD is in a defined condition ~~(default)~~.

TRUE: The PD is an undefined condition.

~~pse_dll_power_type~~

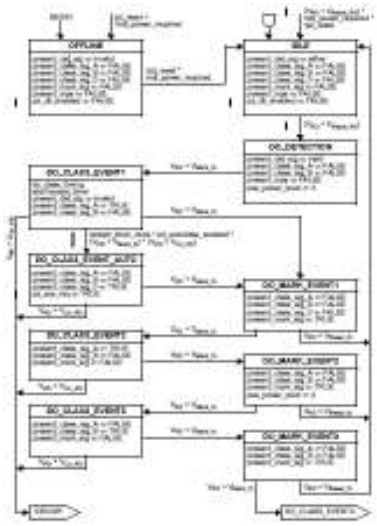
~~A control variable output by the PD power control state diagram, defined in Figure 145-44, that indicates the PSE Type as 1 or 2, see 79.3.2.4.1.~~

~~Values:~~

~~1: The PSE is a Type 1 PSE, for a Type 1 PSE~~

~~2: The PSE is a Type 2 PSE, for Type 2, Type 3, or Type 4 PSEs~~

**145.3.3.7 Single-signature PD state diagrams**



*-Editor to replace pd_req_pwr variable (it is PSE and not PD variable) with pd_req_class in in Figure 145-27 in all locations.*

*-Editor to remove "BEGIN" from the relevant states per yseboodt_07_0317.pdf proposal*

**Figure 145–26—Type 3 and Type 4 single-signature PD state diagram**

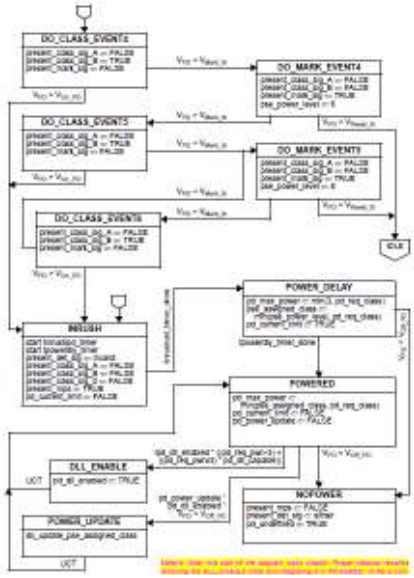


**Figure 145–27—Type 3 and Type 4 single-signature PD state diagram (Continued)**

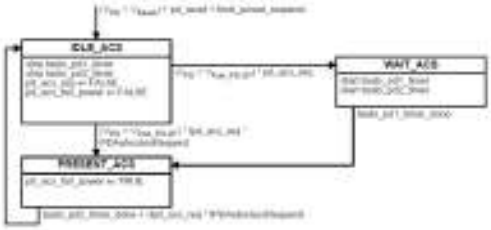


**Figure 145–28—Type 3 and Type 4 single-signature PD Autoclass state diagram**

NOTE 1—DO_CLASS_EVENT6 creates a defined behavior for a Type 3 or Type 4 PD that is brought into the classification range more than 5 times.

NOTE 2—In general, there is no requirement for a PD to respond with a valid classification signature for any DO_CLASS_EVENT duration less than TClass_PD as defined in Table 145–28.

# [PD SECTION: DUAL-SIGNATURE]

### 145.3.3.8 Dual-signature constants

pd_req_class_~~mode(M)~~mode(X)
A constant indicating the requested Class of the PD over Mode M
Values:
1: The PD requests Class 1.
2: The PD requests Class 2.
3: The PD requests Class 3.
4: The PD requests Class 4.
5: The PD requests Class 5.

VMark_th
Mark event voltage threshold per pairset (see Table 145–26)

VOff_PD
PD power supply turn off voltage (see Table 145–28)

VOn_PD
PD power supply turn on voltage (see Table 145–28)

VReset_PD
Reset voltage per pairset (see Table 145–26)

VReset_th
Reset voltage threshold per pairset (see Table 145–26)

### 145.3.3.9 Dual-signature variables

pse assigned class_mode(X)
A variable **(generated by the PD)** that indicates the **PSE assigned** Class over mode M to the PD. This variable is initially set by Physical Layer classification and may be updated through DLL classification.
Values:
1: Class 1
2: Class 2
3: Class 3
4: Class 4
5: Class 5

pd power update_mode(X)
A variable that is set when the PDMaxPowerValue_mode(X) in the DLL state diagram in Figure 145–48 has been updated.
Values:
FALSE: The value of PDMaxPowerValue_mode(X) has not changed.
TRUE: The value of PDMaxPowerValue_mode(X) has changed.

mdi_power_required_ ~~mode(M)~~mode(X)
A control variable indicating that over mode M, the PD is enabled and should request power from the PSE by applying a PD detection signature to the link, and when the PSE sources power to apply the MPS to keep the PSE sourcing power. A variable that is set in an implementation-dependent manner.
Values:
FALSE: PD functionality is disabled.
TRUE: PD functionality is enabled.

pd_current_limit_mode(X)
Control on Mode M limiting the input current to a value conforming to IInrush_PD-2P, as defined in Table 145–28. Values:
FALSE: The PD is not required to control the input current.
TRUE: The PD is required to control the input current.

pd_dll_capable_mode(X)
This variable indicates whether the PD implements Data Link Layer classification over mode (X).
Values:
FALSE: The PD does not implement Data Link Layer classification.
TRUE: The PD does implement Data Link Layer classification.

pd_dll_enabled_mode(X)
A variable indicating whether the Data Link Layer classification mechanism is enabled over mode (X).
Values:
FALSE: Data Link Layer classification is not enabled.
TRUE: Data Link Layer classification is enabled.

pd_max_power_ ~~mode(M)~~mode(X)
A control variable indicating the max power that the PD may draw from the PSE over mode M. See power classifications in Table 145–28.
Values:
1: PD may draw Class 1 power
2: PD may draw Class 2 power
3: PD may draw Class 3 power
4: PD may draw Class 4 power
5: PD may draw Class 5 power

pd_reset_ ~~mode(M)~~mode(X)
An implementation-specific control variable that unconditionally resets the PD state diagram over mode M to the OFFLINE_ ~~mode(M)~~mode(X) state.
Values:
FALSE: The device has not been reset ~~(default)~~.
TRUE: The device has been reset.

pd_undefined_ ~~mode(M)~~mode(X)
A control variable that indicates that the PD is in an undefined condition over mode M. The PD may or may not show a valid or invalid detection signature, may or may not draw mark current, may or may not draw any class current, may or may not show MPS and may change the pse_power_level_modeA variable.
Values:
FALSE: The PD is in a defined condition ~~(default)~~.
TRUE: The PD is an undefined condition.

present_class_sig_A_ ~~mode(M)~~mode(X)
Controls presenting the classification signature that is used during first two class events (see 145.3.6) by the PD over mode M.
Values:
FALSE: The PD classification signature is not to be applied to the link.
TRUE: The PD classification signature is to be applied to the link.

present_class_sig_B_ ~~mode(M)~~mode(X)
Controls presenting the classification signature that is used during the third class event and all subsequent class events (see 145.3.6) by the PD over mode M.
Values:
FALSE: The PD classification signature is not to be applied to the link.
TRUE: The PD classification signature is to be applied to the link.

present_det_sig_ ~~mode(M)~~mode(X)
Controls presenting the detection signature (see 145.3.4) by the PD over mode M.
Values:
invalid: A non-valid PD detection signature is to be applied to the pairset over Mode M.
valid: A valid PD detection signature is to be applied to the pairset over Mode M.
either: Either a valid or non-valid PD detection signature may be applied to the pairset.

present_mark_sig_ ~~mode(M)~~mode(X)
Controls presenting the mark event current and impedance (see 145.3.6.1.1) by the PD over mode M.
Values:
FALSE: The PD does not present mark event behavior.
TRUE: The PD does present mark event behavior.

present_mps_~~mode(M)~~mode(X)

Controls applying MPS over mode M (see 145.3.9) to the PD's PI.

Values:

FALSE: The Maintain Power Signature (MPS) is not to be applied to the PD's PI.

TRUE: The MPS is to be applied to the PD's PI.

~~pse_dll_power_type~~

~~A control variable output by the PD power control state diagram (Figure 145-44) that indicates the PSE type connected to the PD as 1 or 2, see 79.3.2.4.1.~~

~~Values:~~

~~1: The PSE is a Type 1 PSE, for a Type 1 PSE.~~

~~2: The PSE is a Type 2 PSE, for a Type 2, 3 and, 4 PSE.~~

pse_power_level_~~mode(M)~~mode(X)

A control variable that indicates to the PD the level of power the PSE is supplying over Mode M.

Values:

3: The PSE has allocated the PD's requested power or Class 3 power, whichever is less.

4: The PSE has allocated the PD's requested power or Class 4 power.

5: The PSE has allocated the PD's requested power or Class 5 power.

VPD_~~mode(M)~~mode(X)

The voltage at the PD PI measured between any positive conductor and any negative conductor of the Mode M pairs; see 145.1.3.

### 145.3.3.10 Dual-signature timers

All timers operate in the manner described in 14.2.3.2 with the following addition. A timer is reset and stops counting upon entering a state where "stop x_timer" is asserted.

tinrushpd_timer_mode(X)

A timer used to determine when the PD exits INRUSH over mode (X) and meets the requirements of POWER_DELAY; see $T_{Inrush\_PD}$ in Table 145–28.

tpowerdly_timer_~~mode(M)~~mode(X)

A timer used to prevent PD from drawing more than $I_{Inrush\_PD}$ and $I_{Inrush\_PD\text{-}2P}$ from $T_{Inrush\_PD}$ to $T_{delay\text{-}2P}$. See Table 145–28.

### 145.3.3.11 Dual-signature functions

do_class_timing_~~mode(M)~~mode(X)

This function is used by ~~a~~ the PD ~~to~~ to evaluate the Type of PSE connected to the pairset by measuring the length of the class event over Mode M. The class event timing requirements are defined in Table 145–26. This function returns the following variable:

long_class_event_~~mode(M)~~mode(X): A control variable that indicates to the PD the Type of PSE to which it is connected. This variable is used to indicate which MPS timing requirements (see 145.3.9) the PD should use. See 145.3.7.

do_update_pse_assigned_class_mode(X)

A function that updates the pse_assigned_class_mode(X) based on the value of PDMaxPowerValue_mode(X) as defined in Table 33–24. This function returns the following variable:

pse_assigned_class_mode(X): this variable indicates the Class assigned to the PD.

Values:

1: Class 1

2: Class 2

3: Class 3

4: Class 4

5: Class 5

## 145.3.3.12 Dual-signature PD state diagrams

*-Change the suffix _mode(M) to _mode(X).*

*-Change from VPD_mode to $V_{PD}$_mode in all occurrences.*

***-Remove "BEGIN" from the relevant states per yseboodt_07_0317.pdf proposal***

***-Make the following changes in Figure 145-29.***

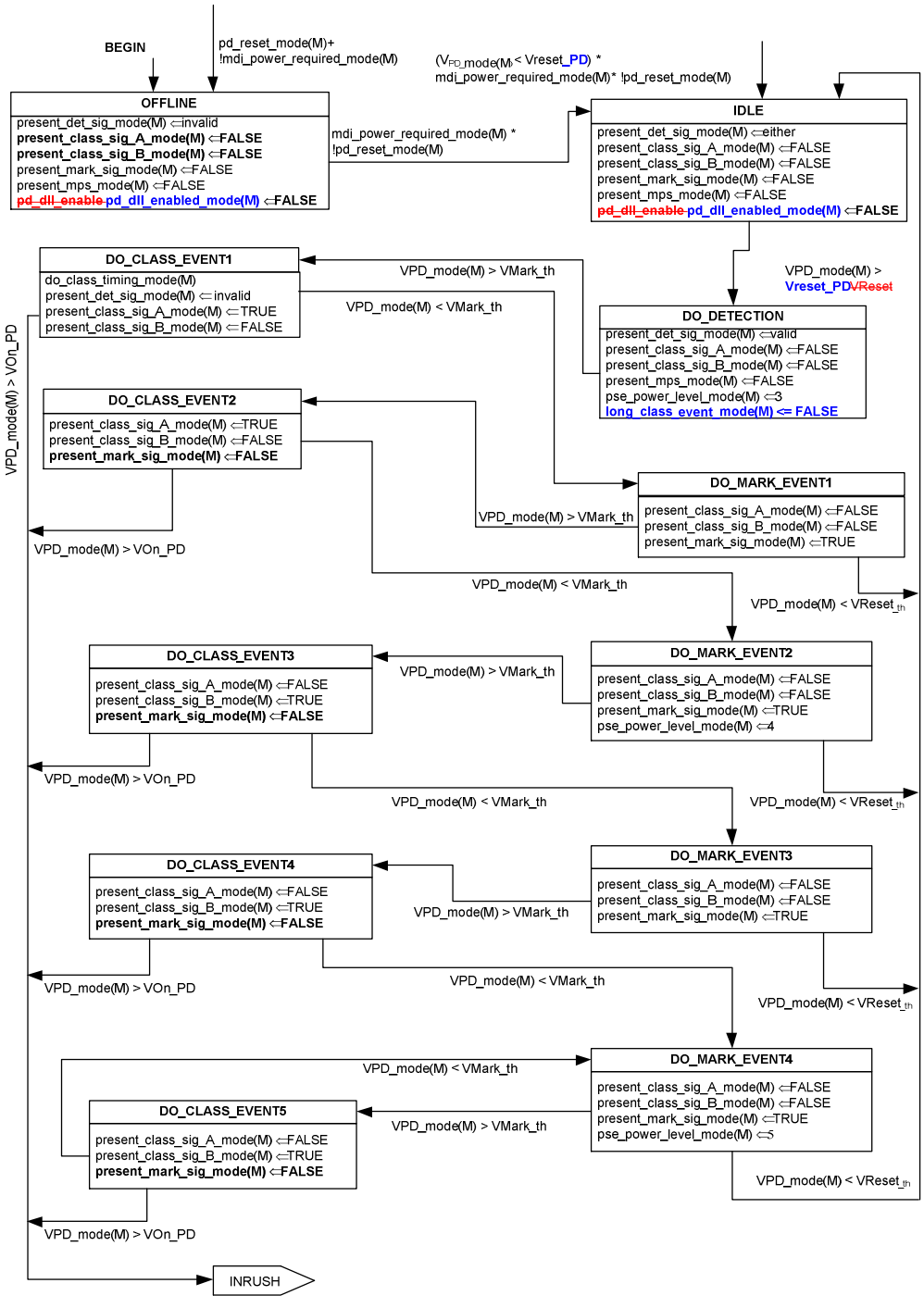
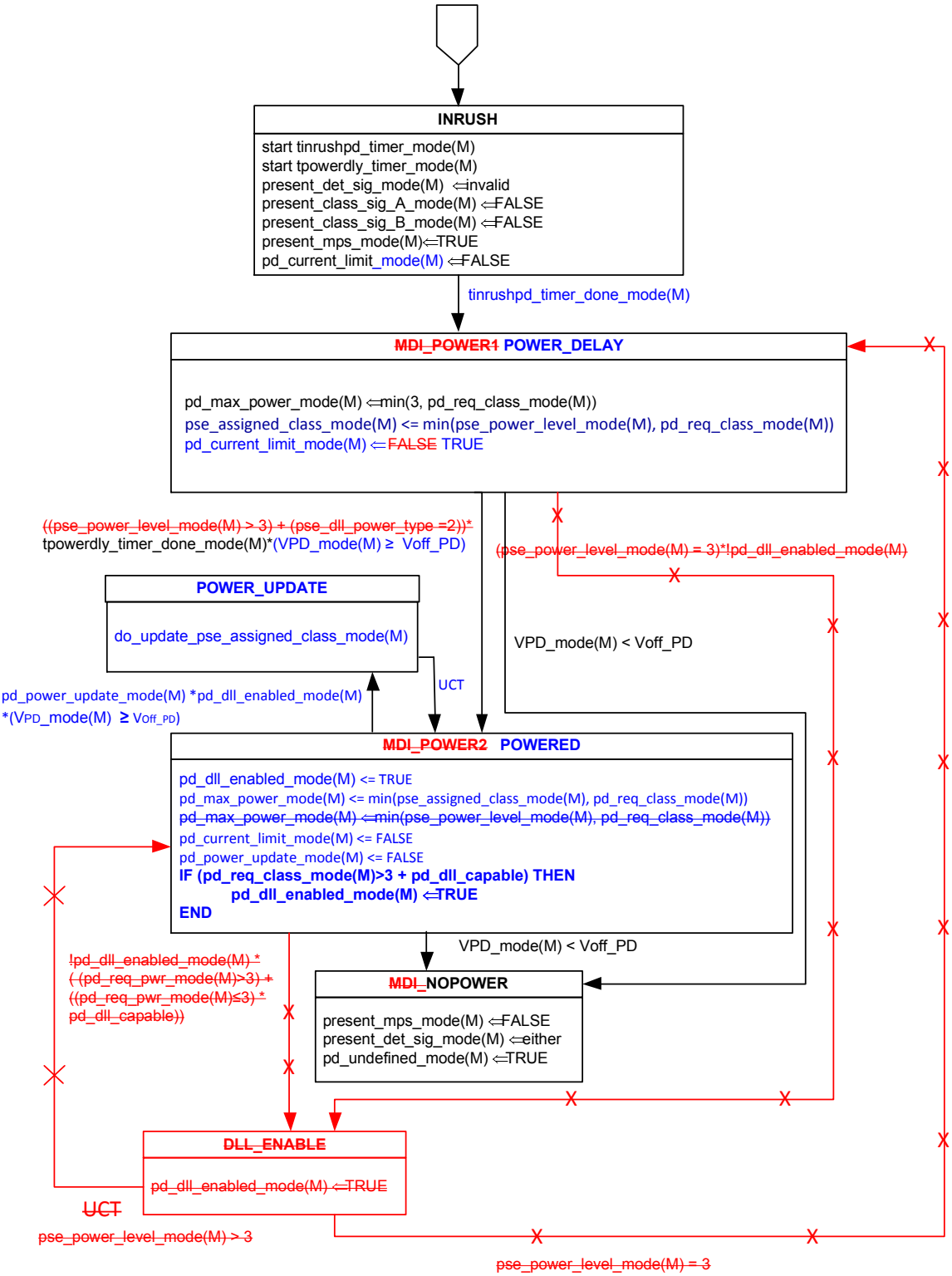


**Figure 145–29—Type 3 and Type 4 dual-signature PD state diagram**

*-Change the suffix _mode(M) to _mode(X).*

*-Make the following changes in Figure 145-30.*




Base Line ends here