IEEE P802.3bt
PSE State Diagram Comments

Dan Dove, DNS for LTC

November 6, 2015

Objectives of this Presentation:

A number of issues have been observed within the D1.4 version of the PSE State Diagram but ***this presentation only targets a few key areas*.**

• D1.4 fails to initialize some key variables, and I suggest a few more variables that would be required to address other issues with the state diagram. Its recommended these be initialized in the IDLE state.
• D1.4 shows multiple states being entered from a single state based upon the same conditions. I offer a way to address that.
• In Catania, it was stated that a means for alternating which Alt would perform detection first should be addressed. Text was added to infer this. I'm suggesting a more explicit means to communicate how this occurs.
• I have some recommendations for removing intra-page connectors to clean up the drawings and make them more readable.
• For "sig_type=single' POWER_UP and POWER_ON states, there are some issues with the logic for which I have suggested changes.

I believe that the classification state diagram only holds valid for the case of sig_type=single. For dual-signature cases, the current detection->powering paths need to include a classification path for each Alt and language to address variable duplication.

Insert new state diagrams Figure 33-9a to Figure 33-9d for Type 3 and Type 4 PSEs as follows:

Dual Path State Machine Discussion:

The E and M arcs go to multiple locations. To fix this, we create a state that sets flags telling those sub-state-machines to start up.

- •PI_SM: Flag that tells PI_State Machines to start
- •StagDet: Tells PI_State Machines whether to run detection sequentically (staggered) or simultaneously.

When both sub-state-machines are done, they set flags to cause the PI_SM_START exit arc to be TRUE.
Alt_Done_A: Says alt_a detection is stopped
Alt_Done_B: Says alt_b detection is stopped

Additional variable initialization in IDLE state.

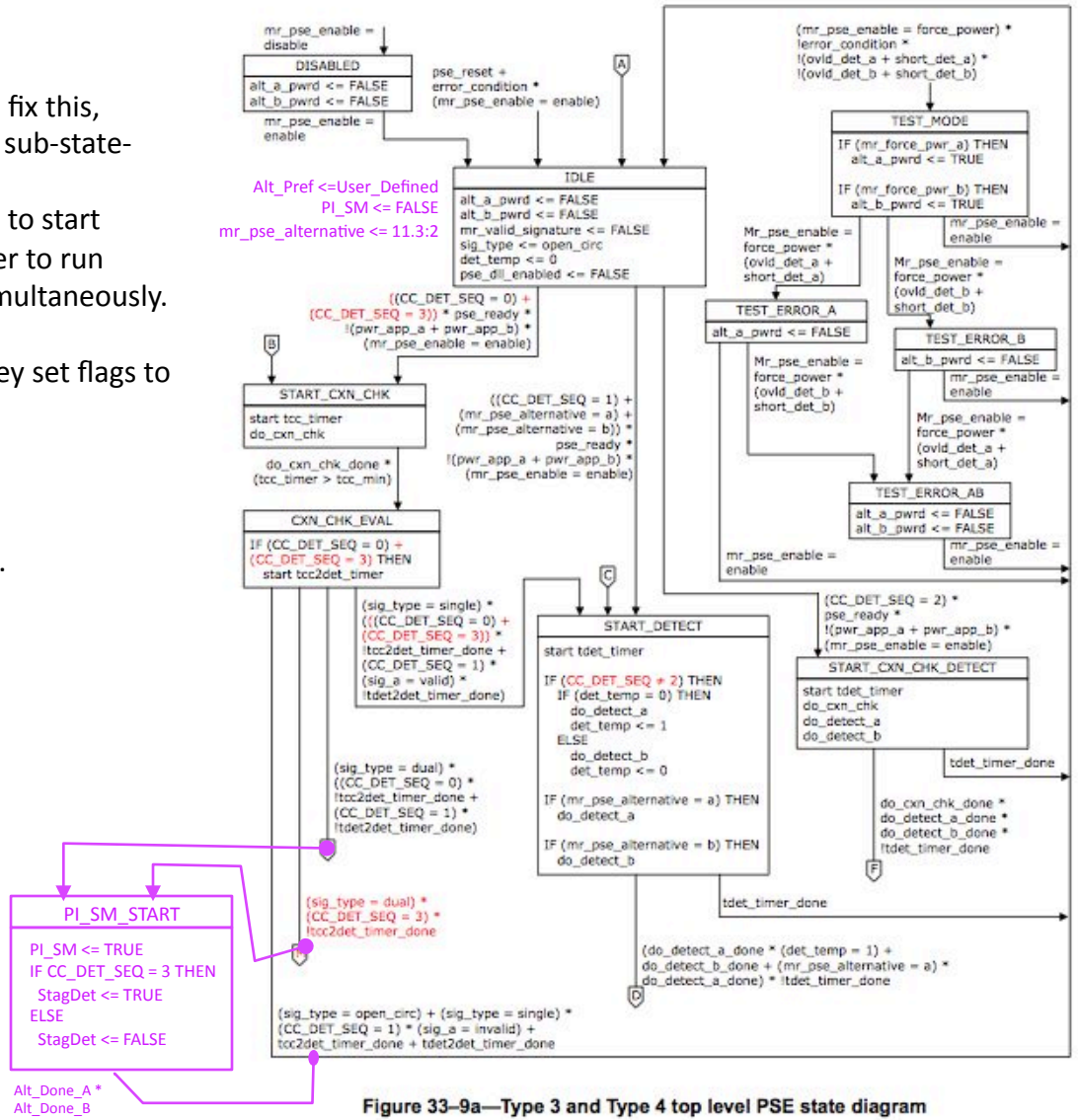


Figure 33–9a—Type 3 and Type 4 top level PSE state diagram

Note: This state is only drawn here for ease of comment. Final drawing would integrate the additional state cleanly.

Dual Path State Machine Discussion: (continued)

Create an Entry_A state that is entered whenever PI_SM is FALSE. (Initialized by IDLE in upper state machine)

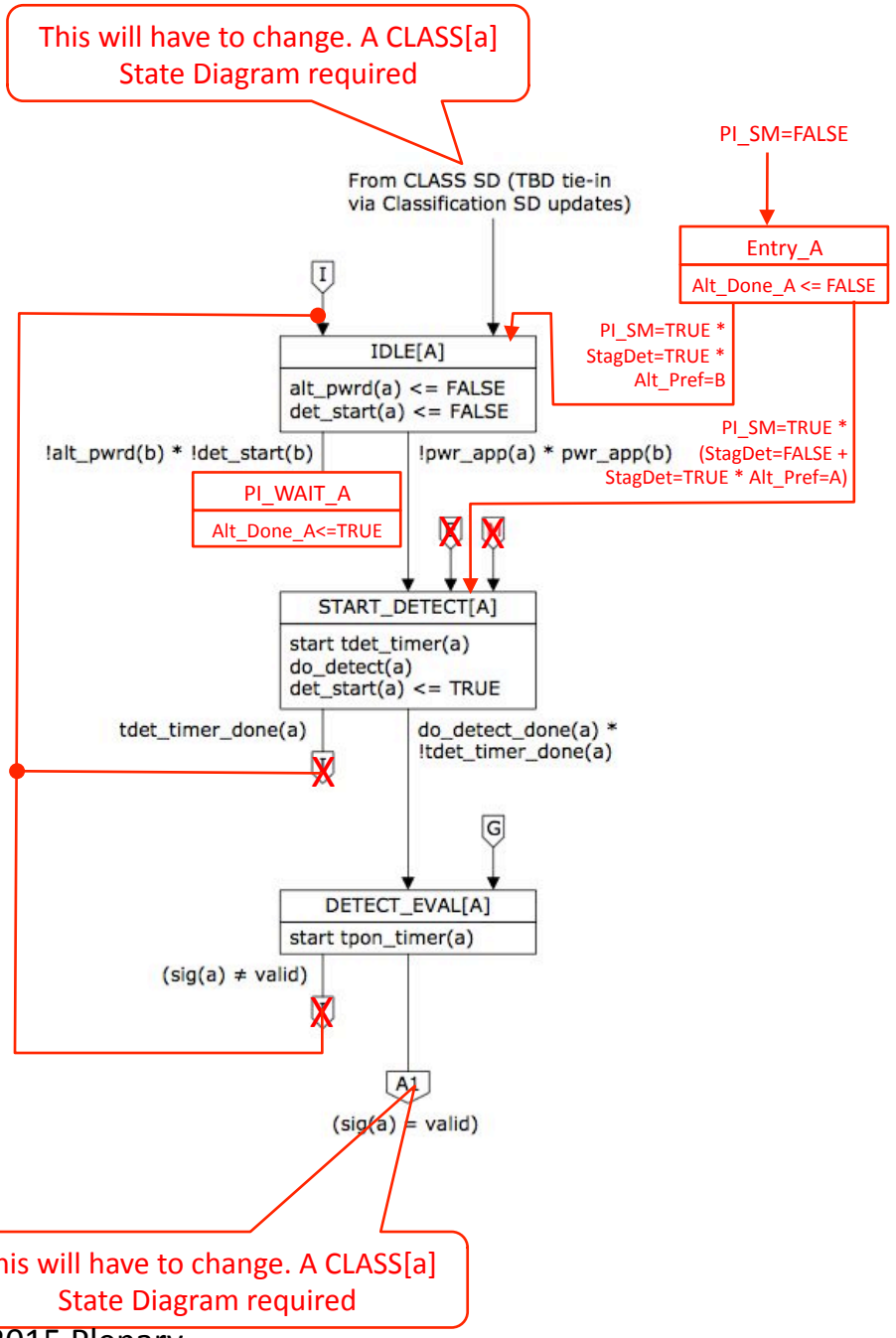
Exit from this state is controlled by PI_SM=TRUE which occurs when the upper state machine enters the PI_SM_ START state.

Direction to either IDLE_A or START_DETECT_A is based on whether StagDet is TRUE, and if so, whether the Alt_Pref is A or B.
For the case where StagDet is FALSE, both alt_a and alt_b detections will occur immediately.
For the case where StagDet is TRUE, Alt_Pref will decide which alt goes to IDLE and which one goes to START_DETECT.

Eliminated the intrapage connectors for cleanliness.

When in IDLE[A] and the other alt is in IDLE[B], both alts drop into the PI_Wait_A/B states, set the Alt_Done_A/B flags. This triggers the upper level state machine to return to IDLE.

Arc M and Arc E entries are no longer required.



Dual Path State Machine Discussion: (continued)

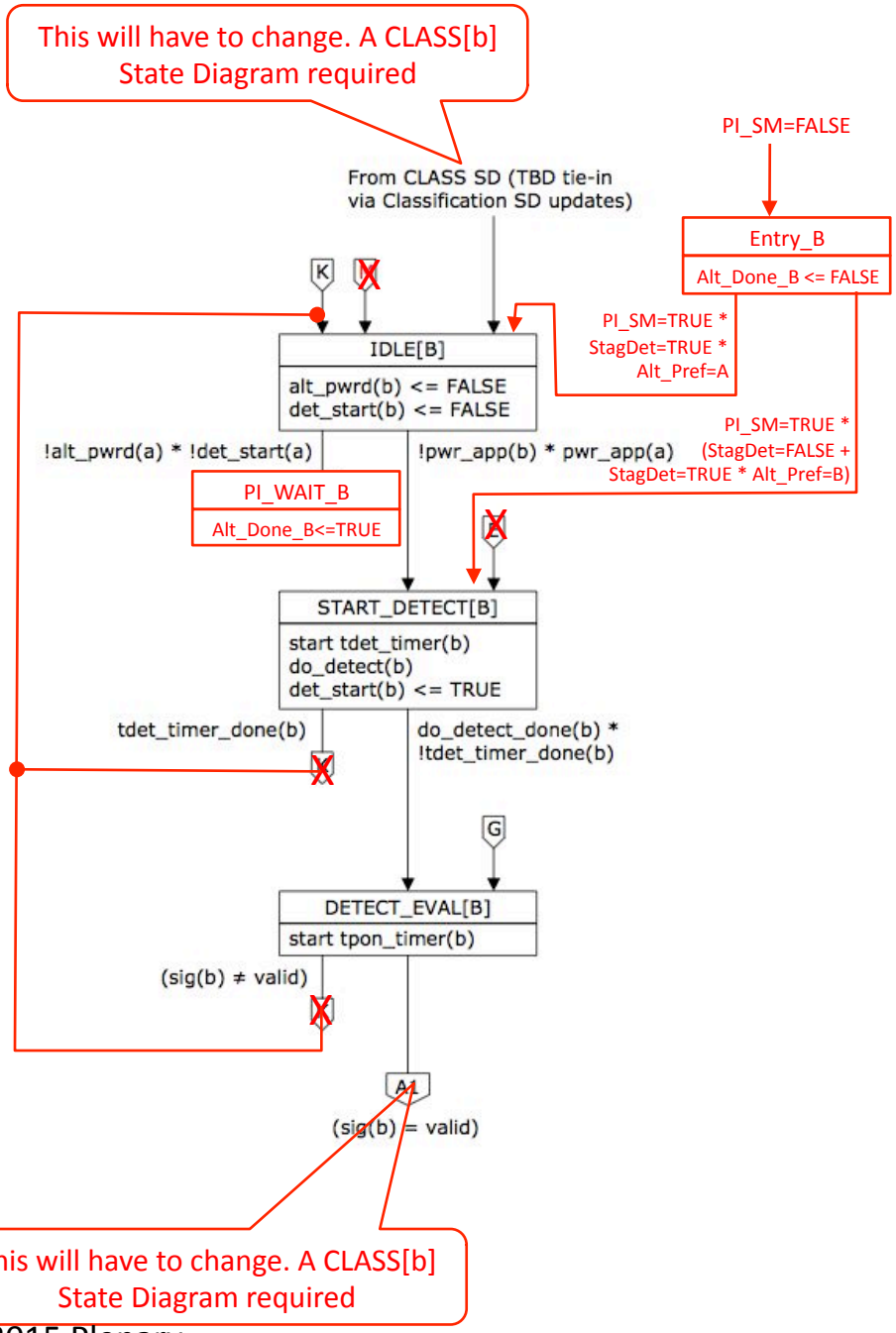
Create an Entry_B state that is entered whenever PI_SM is FALSE. (Initialized by IDLE in upper state machine)

Exit from this state is controlled by PI_SM=TRUE
Direction to either IDLE or START_DETECT is based on whether StagDet is TRUE, and if so, whether the Alt_Pref is A or B.
For the case where StagDet is FALSE, both alt_a and alt_b detections will occur immediately.
For the case where StagDet is TRUE, Alt_Pref will decide which alt goes to IDLE and which one goes to START_DETECT.

Eliminated the intrapage connectors for cleanliness.

When in IDLE[B] and the other alt is in IDLE[A], both alts drop into the PI_Wait_A/B states, set the Alt_Done_A/B flags. This triggers the upper level state machine to return to IDLE.

Arc M and Arc E entries are no longer required.



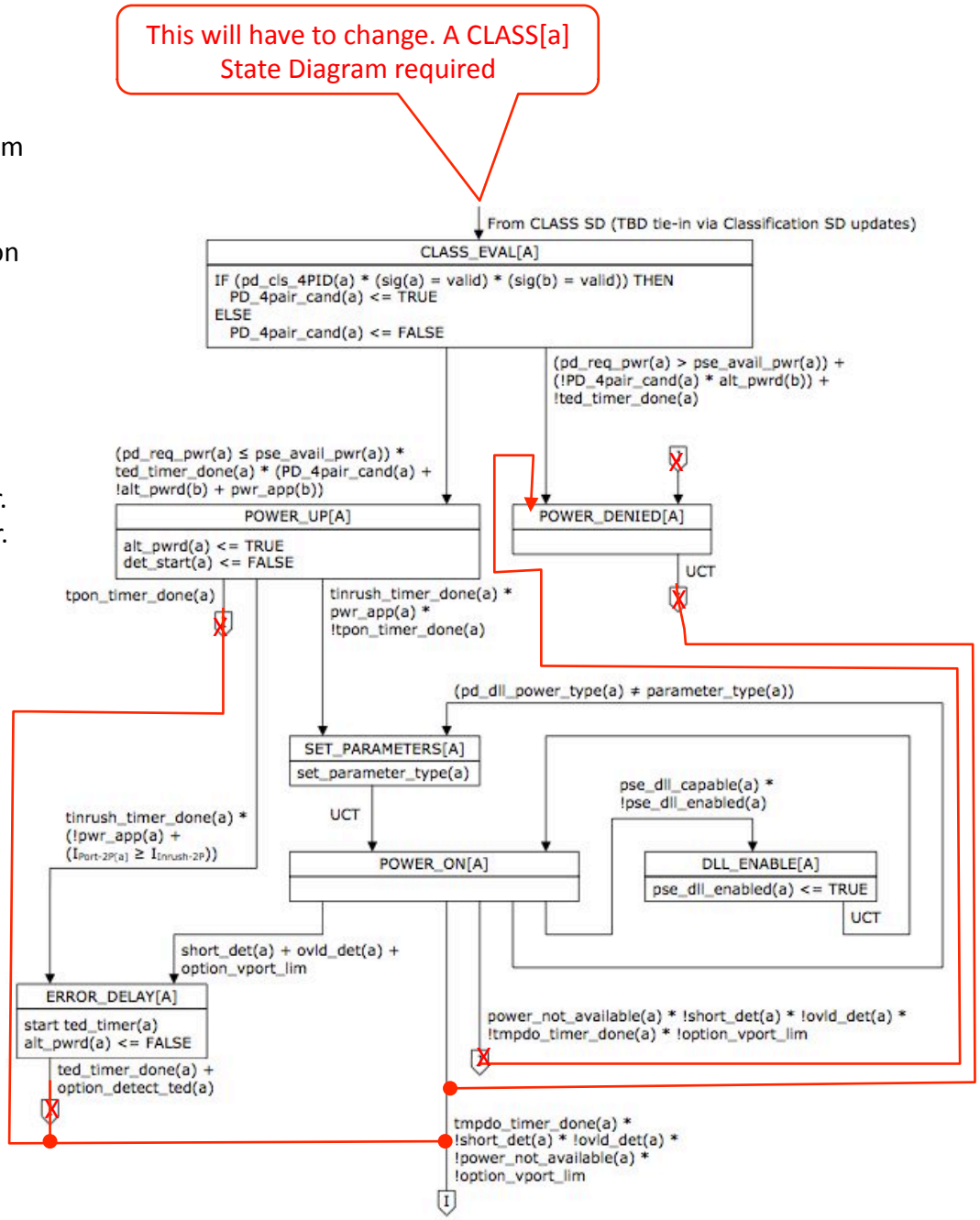
CLASS EVAL[A] Page Discussion:

Its expected that the entry point into this page will be from a new CLASS[a] page that addresses how to perform classification on the alt_a pairset while alt_b is either awaiting detection, in detection, or in its own classification and/or power_on state processes.

There is not much to this page to recommend other than getting rid of the intrapage connectors.

Note: For the J path, I would move POWER_DENIED[A] over to the right a bit to clean & clear the pathway better. Also, move the arc from CLASS_EVAL[A] to make it neater.

Note: CLASS_EVAL[B] page has same changes.



Resolving the POWER_UP state

sig_type=single POWER_UP Discussion:

Logic term “sig_type=single should not exist in this diagram. For cases where sig_type=dual, would be in the other alt_a / alt_b state diagram paths.

In POWER_UP, dll_4PID=1 term is not possible since pse_dll_enable has not yet been set to TRUE.

If ANY logical condition was added, perhaps

IF mr_pse_alternative = both * mr_pse_ss_mode = 1
THEN
alt_a_powered <= TRUE
alt_b_powered <= TRUE
ELSE
alt_a_powered <= TRUE
alt_b_powered <= FALSE

This is nice because it allows the PSE “ubiquitous management” to decide that it does NOT want to power up all 4 pairs even though the PSE is configured to power 4 pairs via the mr_pse_alternative variable.

*Note: We may need to initialize mr_ss_pse_mode in IDLE or another state to ensure it does not change values between states.

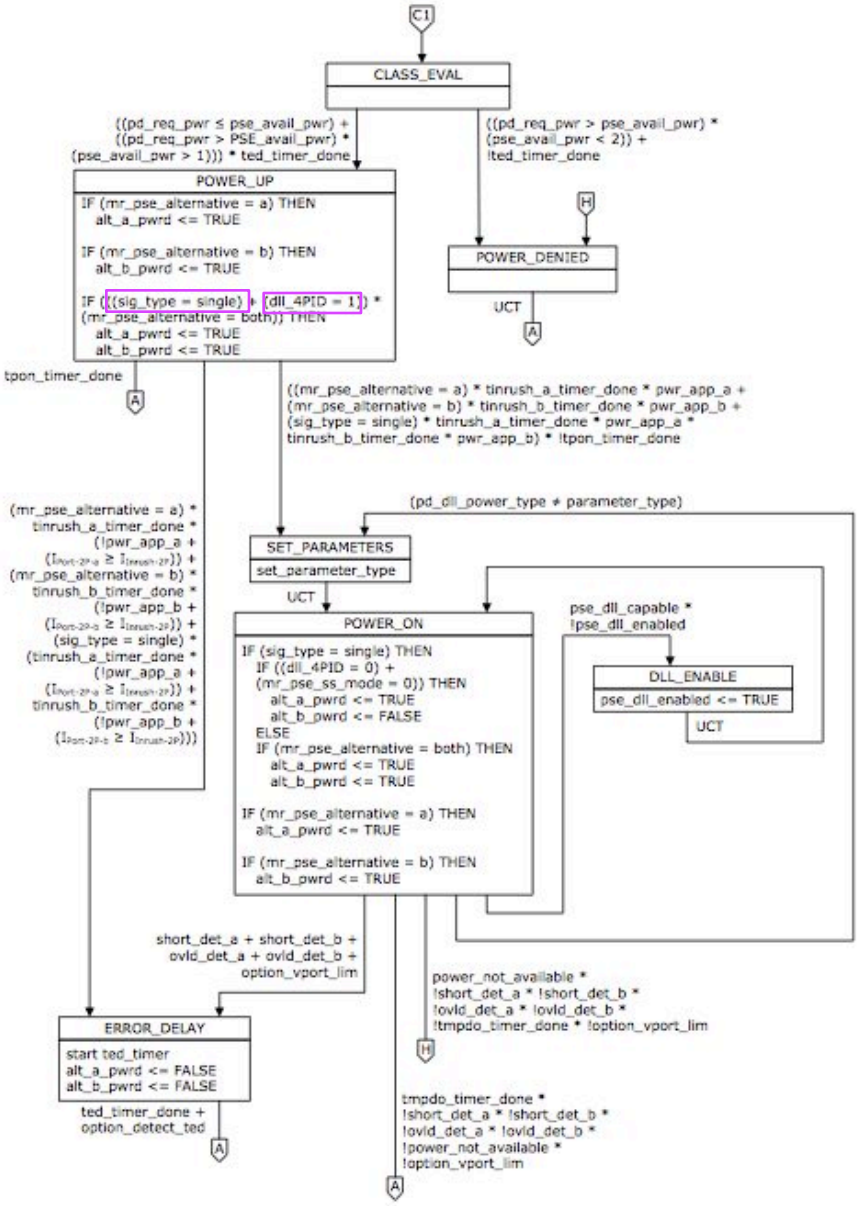


Figure 33–9a—Type 3 and Type 4 top level PSE state diagram (continued)

Resolving the POWER_ON state

sig_type=single POWER_ON Discussion:

Logic term “sig_type=single should not exist in this diagram either. For cases where sig_type=dual, would be in the other alt_a / alt_b state diagram paths.

In POWER_ON, dll_4PID=0 term, and the logic with mr_pse_ss_mode = 0 would ALWAYS be TRUE on entry into this state. So a PSE that powered up both pair-sets, would be required to immediately un-power a pairset [B] upon entry. This makes no sense and is an odd behavior.

I believe the goal should be smooth operation with the ability to turn OFF a pairset due to DLL intervention, not to power it up, turn it off, then turn it back on with DLL intervention only.

Replace first logic block with:
If (mr_pse_alternative = both) THEN
 IF ((dll_4PID= 1) + mr_pse_ss_mode = 1)) THEN
  alt_a_pwrd <= TRUE
  alt_b_pwrd <= TRUE
 ELSE
  alt_a_pwrd <= TRUE
  alt_b_pwrd <= FALSE

This logic would allow all 4-pair PSEs with mr_pse_ss_mode=1 maintain power on all 4 pairs. They would only drop power to alt_b if dll_4PID remains 0 and mr_pse_ss_mode is set to 0.
Ubiquitous mgmt, upon receiving dll_4PID=0, could clear mr_pse_ss_mode to 0 if they wish to power down alt_b.

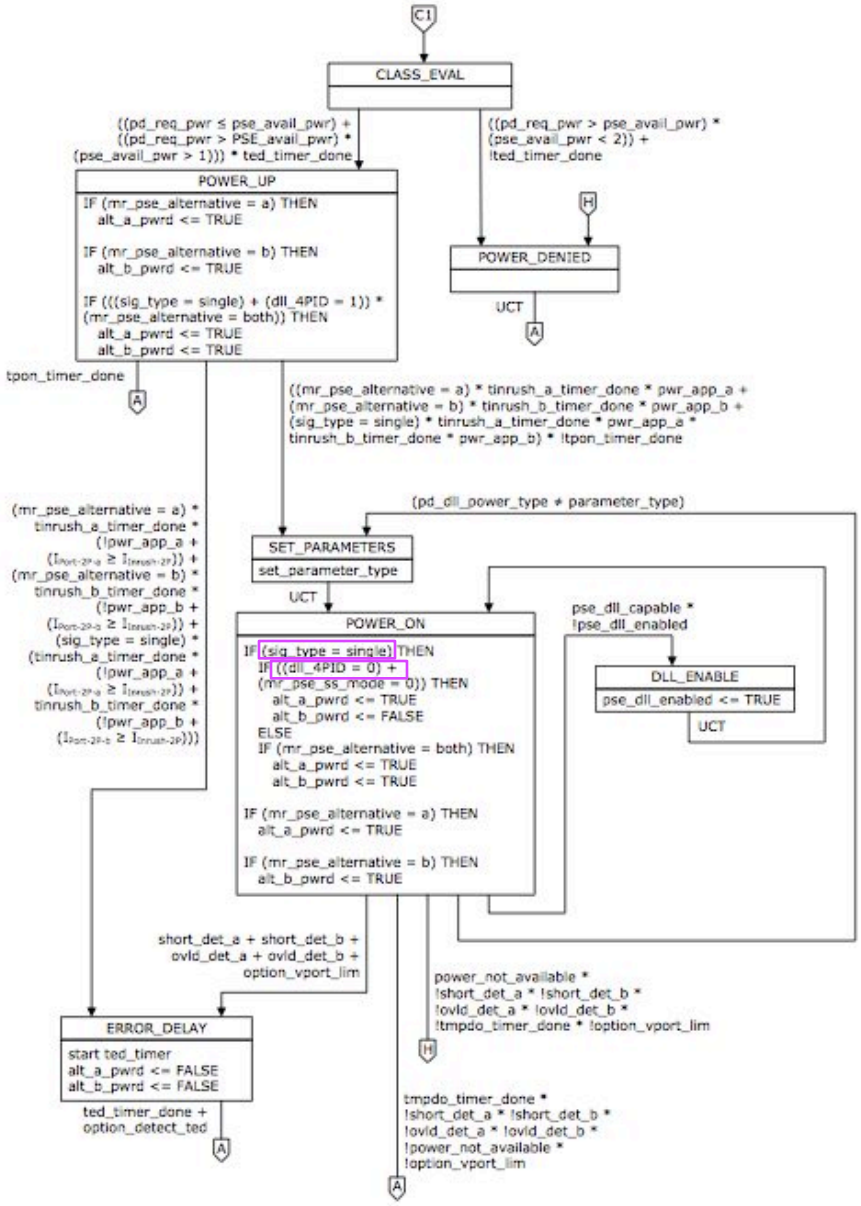


Figure 33–9a—Type 3 and Type 4 top level PSE state diagram (continued)

Thank You!