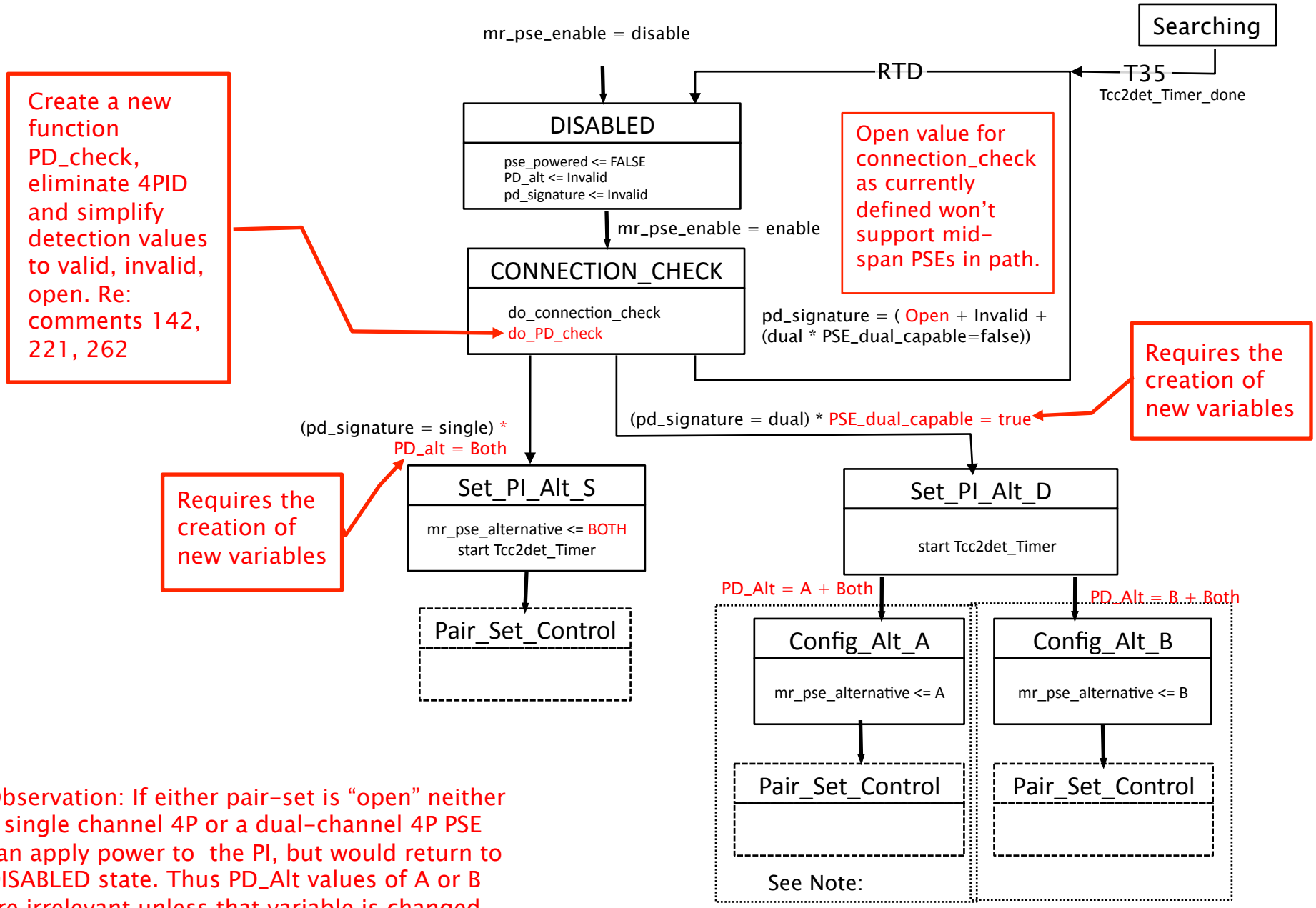
IEEE P802.3bt
PSE State Diagram Update

Dan Dove, DNS for LTC

July 10, 2015

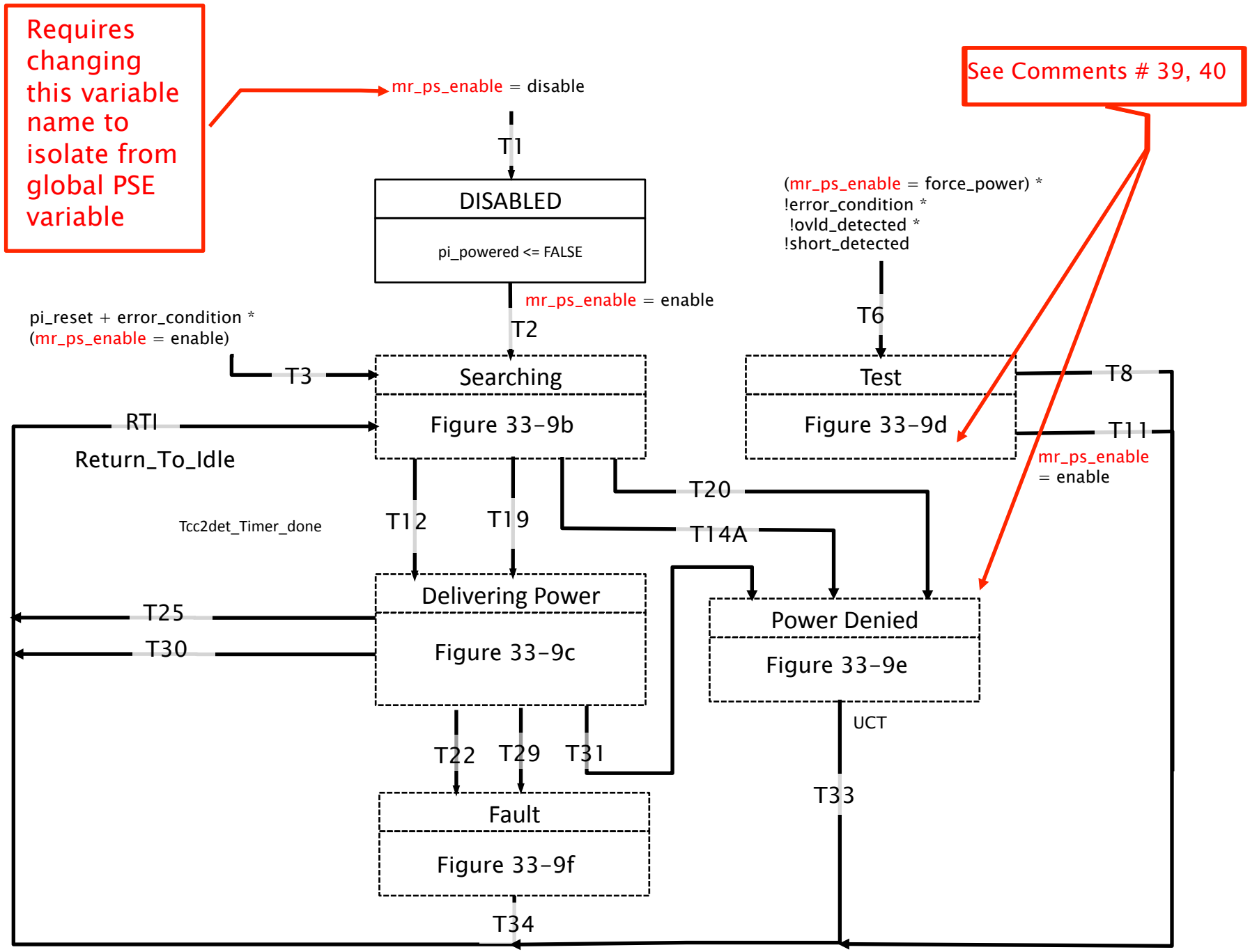
The Challenge

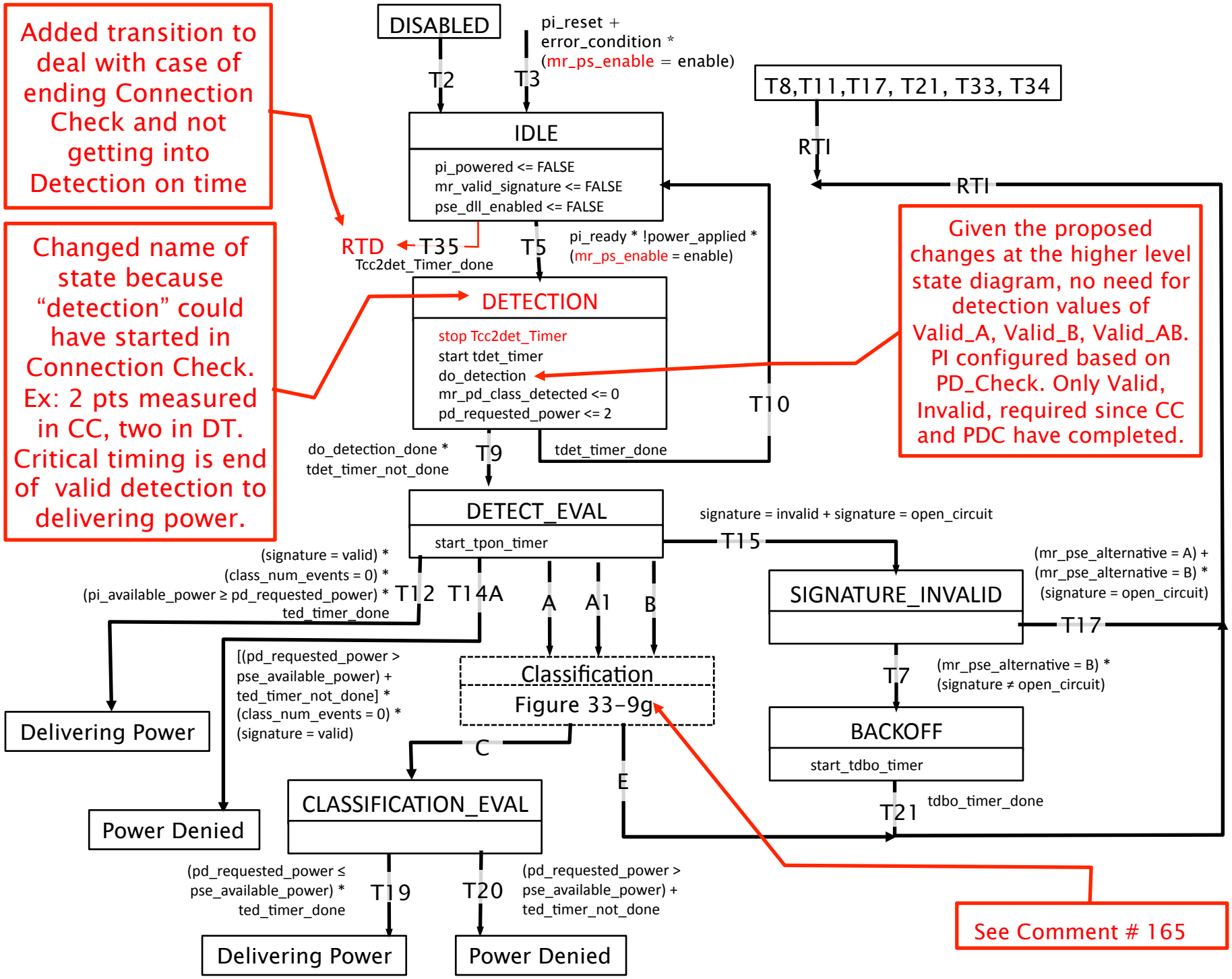
- The case of "dual PSE with dual signature PD" has a potential to exponentially increase the number of states and transitions in the state diagram.
  - Previous examination of the impact of adding connection check into the DO DETECTION state showed substantial expansion of transitions. Did not even include impact of dual-PSE w/dual-sig-PD.
- To simplify the diagram, and allow the widest range of implementation, I am taking the approach of including two independent state machines defined as capable of operating simultaneously.
- This dramatically simplifies the problem, allowing the existing PSE State Diagram to become "Pair Set Control State Diagram" and then adding a layer above it to decide which pse_alt to use.
- For single-channel PSEs and PDs, the state diagram follows existing design with minor modifications (variable name changes)
- For dual-channel PSEs and dual-sig PDs, it enables two state machines to run pretty much independently of each other. This enables dual-dual configurations, and also dual-single configurations.
- For Type 1 and Type 2 PSE fallback, the PSE host can simply disable Type 3 and Type 4 PSE State Machine and enable the Type 1 and Type 2 PSE State Machine using the "Ubiquitous Management Entity" or it may choose to power Type 1 and Type 2 PSEs using the Type 3 and Type 4 PSE State Machine.

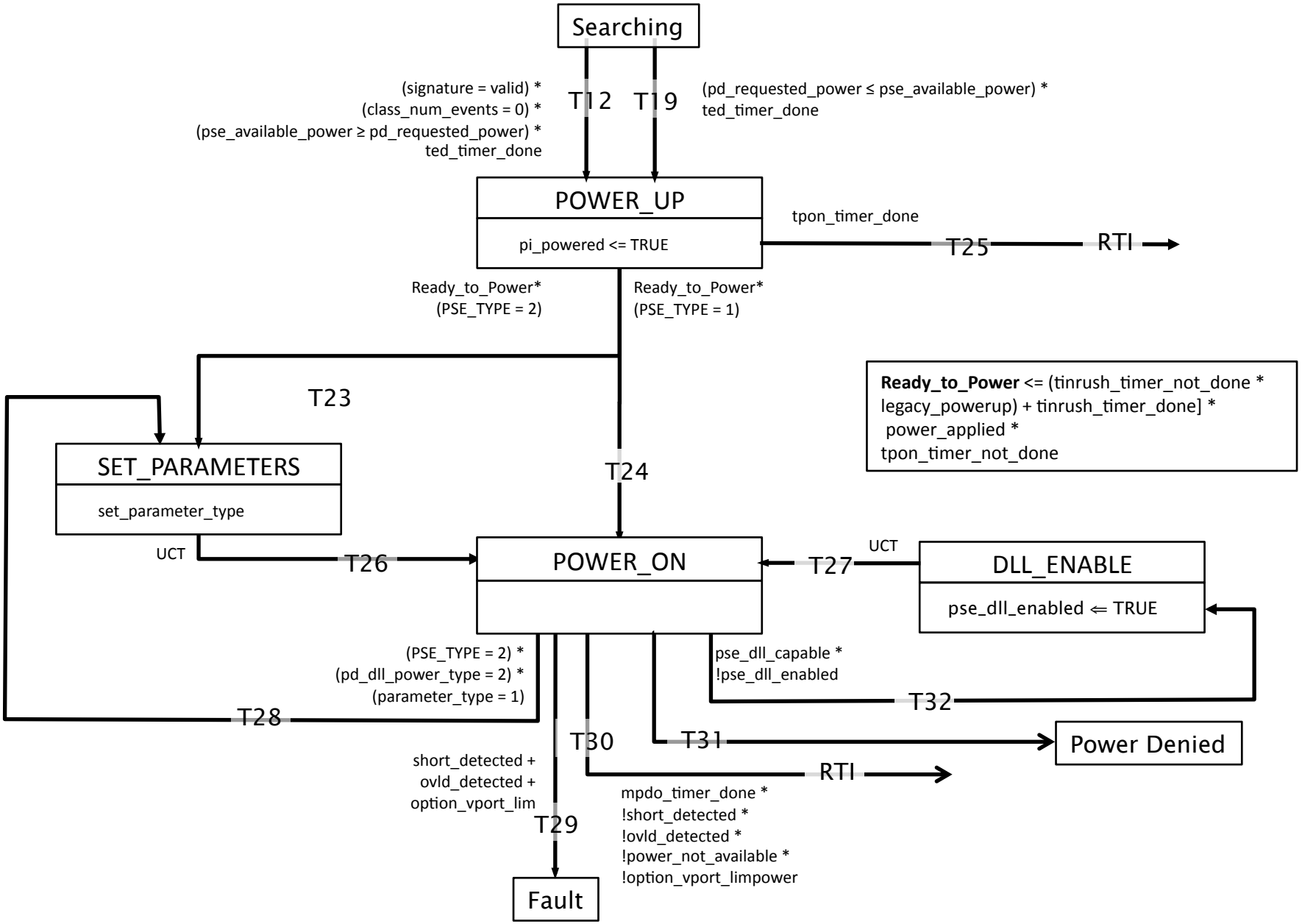


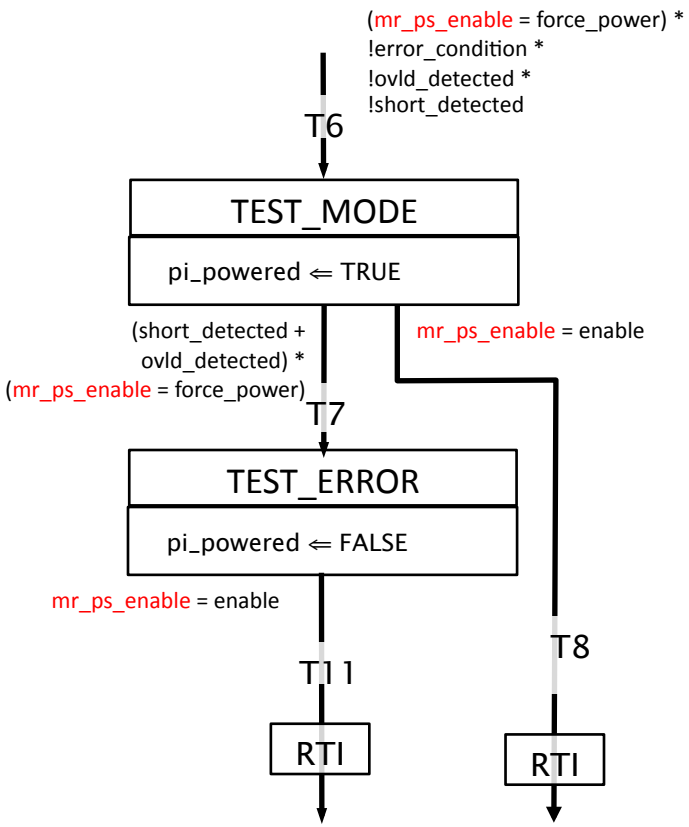
Observation: If either pair-set is "open" neither a single channel 4P or a dual-channel 4P PSE can apply power to the PI, but would return to DISABLED state. Thus PD_Alt values of A or B are irrelevant unless that variable is changed.

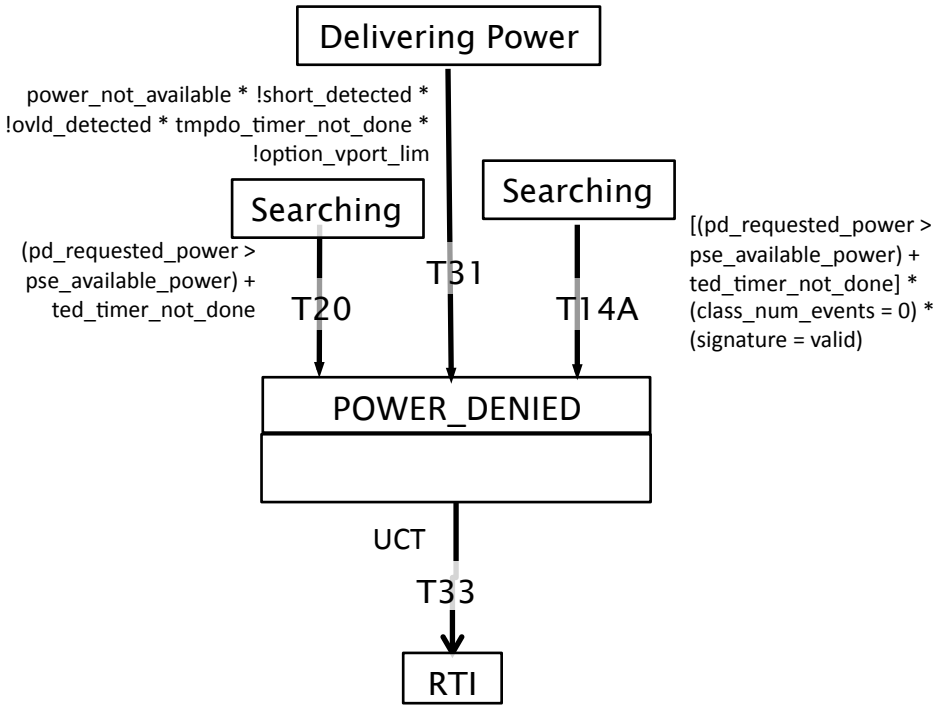
Note: For dual capable PSEs with dual-signature PDs, this diagram indicates the PSE will operate as two distinct Pair_Set_Control state machines with distinct variables for each.

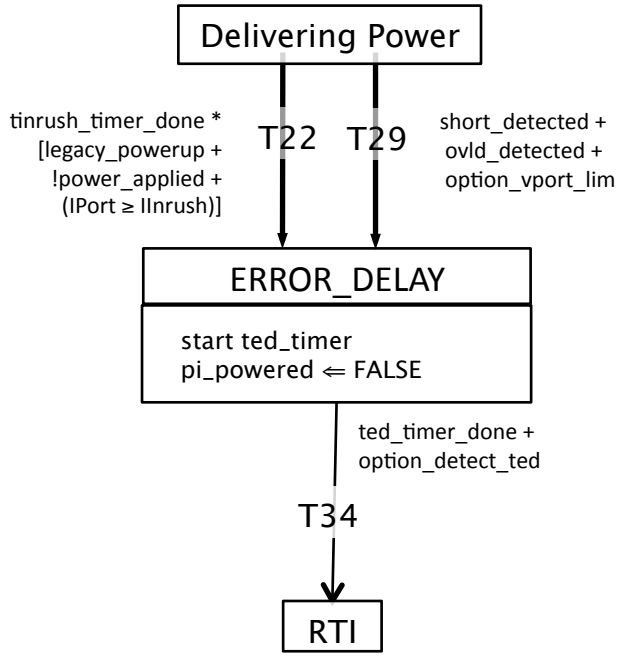

Requires changing this variable name to isolate from global PSE variable
See Comments # 39, 40
mr_ps_enable = disable
T1
DISABLED
pi_powered <= FALSE
mr_ps_enable = enable
T2
pi_reset + error_condition *
(mr_ps_enable = enable)
T3
Searching
Figure 33-9b
(mr_ps_enable = force_power) *
!error_condition *
!ovld_detected *
!short_detected
T6
Test
Figure 33-9d
T8
T11
mr_ps_enable
= enable
RTI
Return_To_Idle
T20
T14A
Tcc2det_Timer_done
T12
T19
Delivering Power
Figure 33-9c
T25
T30
Power Denied
Figure 33-9e
UCT
T22
T29
T31
T33
Fault
Figure 33-9f
T34


Added transition to deal with case of ending Connection Check and not getting into Detection on time
Changed name of state because "detection" could have started in Connection Check. Ex: 2 pts measured in CC, two in DT. Critical timing is end of valid detection to delivering power.
DISABLED
pi_reset + error_condition * (mr_ps_enable = enable)
T2
T3
IDLE
pi_powered <= FALSE
mr_valid_signature <= FALSE
pse_dll_enabled <= FALSE
RTD
T35
Tcc2det_Timer_done
T5
pi_ready * !power_applied * (mr_ps_enable = enable)
DETECTION
stop Tcc2det_Timer
start tdet_timer
do_detection
mr_pd_class_detected <= 0
pd_requested_power <= 2
T10
tdet_timer_done
do_detection_done * tdet_timer_not_done
T9
DETECT_EVAL
start_tpon_timer
T8,T11,T17, T21, T33, T34
RTI
RTI
Given the proposed changes at the higher level state diagram, no need for detection values of Valid_A, Valid_B, Valid_AB. PI configured based on PD_Check. Only Valid, Invalid, required since CC and PDC have completed.
signature = invalid + signature = open_circuit
T15
SIGNATURE_INVALID
(mr_pse_alternative = A) + (mr_pse_alternative = B) * (signature = open_circuit)
T17
T7
(mr_pse_alternative = B) * (signature ≠ open_circuit)
BACKOFF
start_tdbo_timer
T21
tdbo_timer_done
(signature = valid) * (class_num_events = 0) * (pi_available_power ≥ pd_requested_power) * ted_timer_done
T12
T14A
A
A1
B
Delivering Power
[(pd_requested_power > pse_available_power) + ted_timer_not_done] * (class_num_events = 0) * (signature = valid)
Classification
Figure 33-9g
C
E
Power Denied
CLASSIFICATION_EVAL
(pd_requested_power ≤ pse_available_power) * ted_timer_done
T19
T20
(pd_requested_power > pse_available_power) + ted_timer_not_done
Delivering Power
Power Denied
See Comment # 165


Searching
(signature = valid) *
(class_num_events = 0) *
(pse_available_power ≥ pd_requested_power) *
ted_timer_done
T12
T19
(pd_requested_power ≤ pse_available_power) *
ted_timer_done
POWER_UP
pi_powered <= TRUE
tpon_timer_done
T25
RTI
Ready_to_Power*
(PSE_TYPE = 2)
Ready_to_Power*
(PSE_TYPE = 1)
T23
T24
Ready_to_Power <= (tinrush_timer_not_done *
legacy_powerup) + tinrush_timer_done] *
power_applied *
tpon_timer_not_done
SET_PARAMETERS
set_parameter_type
UCT
T26
POWER_ON
T27
UCT
DLL_ENABLE
pse_dll_enabled ⇐ TRUE
(PSE_TYPE = 2) *
(pd_dll_power_type = 2) *
(parameter_type = 1)
pse_dll_capable *
!pse_dll_enabled
T32
T28
T30
T31
Power Denied
RTI
short_detected +
ovld_detected +
option_vport_lim
T29
mpdo_timer_done *
!short_detected *
!ovld_detected *
!power_not_available *
!option_vport_limpower
Fault


(mr_ps_enable = force_power) *
!error_condition *
!ovld_detected *
!short_detected
T6
TEST_MODE
pi_powered ⇐ TRUE
(short_detected +
ovld_detected) *
(mr_ps_enable = force_power)
T7
mr_ps_enable = enable
TEST_ERROR
pi_powered ⇐ FALSE
mr_ps_enable = enable
T11
T8
RTI
RTI


Delivering Power
power_not_available * !short_detected *
!ovld_detected * tmpdo_timer_not_done *
!option_vport_lim
Searching
Searching
(pd_requested_power >
pse_available_power) +
ted_timer_not_done
T20
T31
T14A
[(pd_requested_power >
pse_available_power) +
ted_timer_not_done] *
(class_num_events = 0) *
(signature = valid)
POWER_DENIED
UCT
T33
RTI


Delivering Power
tinrush_timer_done *
[legacy_powerup +
!power_applied +
(IPort ≥ IInrush)]
T22
T29
short_detected +
ovld_detected +
option_vport_lim
ERROR_DELAY
start ted_timer
pi_powered ⇐ FALSE
ted_timer_done +
option_detect_ted
T34
RTI

Connection Check & Detection

The goal is to allow the widest range of implementation possible.

One can capture portions of data used for the "Detection" function during Connection Check and store them, then later when operating within the PI Control state machine, use that data in conjunction with other data captured during do_detection to complete the detection process. This would potentially speed up the process and ensures that a robust detection is completed no longer before power is applied than the current Type 1 and Type 2 state diagram.

OR

One can perform Connection Check and disregard the data captured after the decision is made to exit that state, and then when entering the DETECTION state, perform an existing do_detection function as currently implemented.

The former would be faster and reduce the number of steps required to complete detection.

Example Text for PD_Check

- ***Add new function do_PD_check as follows:***

do_PD_check

This function is to be used only for Type 3 and Type 4 PSEs and works in conjunction with connection check defined in Section 33.2.5.0 and determines whether a PD can accept power over a single alternative PI configuration or both at the same time. This function returns the following variables:

PD_alt: This variable indicates the type of PD signature is connected to the PI, with respect to 4-pair operation.

Values:

A: The PSE has determined PD appears capable of accepting power only on Alt-A.
B: The PSE has determined PD appears capable of accepting power only on Alt-B.
Both: The PSE has determined PD appears capable of accepting power on Alt-A and Alt-B
Invalid: The PD_check function has not determined a valid value.

Remaining Work

- Define the new variables and function required to make the decision for operating Pair_Set_Control properly
  - **PSE_dual_capable:**
    - True – PSE is capable of operating on two PI instances simultaneously
    - False – PSE is capable of operating on one PI instance
  - **mr_ps_enable:**
    - enable – Pair_Set_Control state machine is enabled
    - disable – Pair_Set_Control state machine is disabled
  - **PD_alt:**
    - A – PD appears capable of accepting power on Alt-A
    - B – PD appears capable of accepting power on Alt-B
    - Both – PD appears capable of accepting power on Alt-A and Alt-B
- Review the overall operation and confirm that it behaves as intended
- Identify if there are any conditions on a dual-PSE to dual-sig-PD configuration that would require both PI instances to return to the top-level diagram
  - Equivalent to global "mr_pse_enable = disable"
- Incorporate appropriate text changes
  - Could deprecate need for 4PID text/variables.
- Incorporate FrameMaker version of state diagrams.

Q&A