



IEEE P802.3bt PSE State Diagram Update

Dan Dove, DNS for LTC

March 15



March 4, 2015

Outline

- Introduction
- PSE States
- Diagram Hierarchical Concept
- Diagram Transition Simplification



Introduction

- First, we create a high-level state diagram with blocks defined by the management status register.
- Next, we identify all arcs in the existing PSE state diagram with numbers.
- Next, we build a top-level block diagram showing all of the relevant arcs between those blocks
- Then we redraw the individual blocks with arcs coming in, exiting.



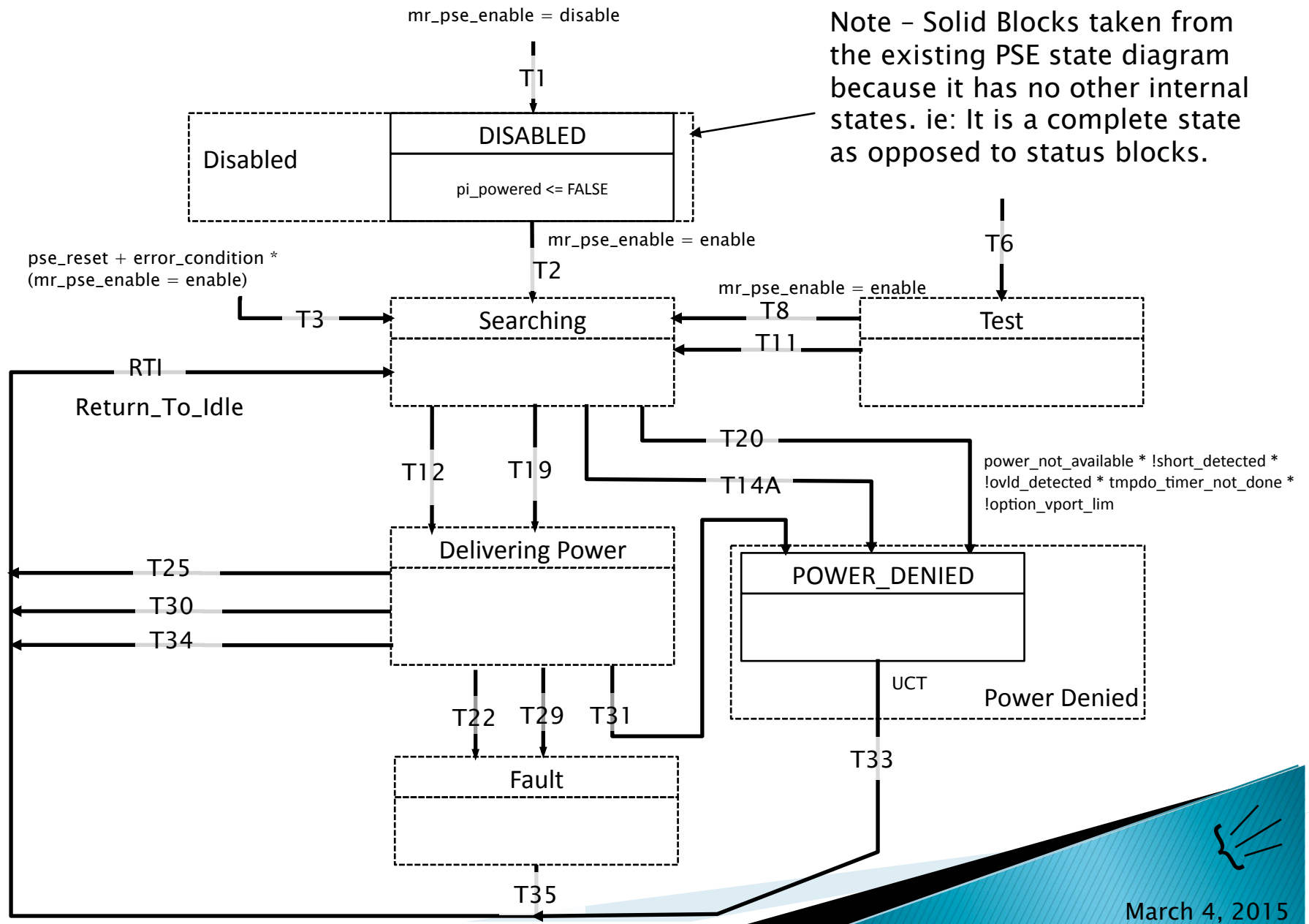
33.2.4.7 State diagrams

The diagram illustrates the PSE state machine with the following states and transitions:

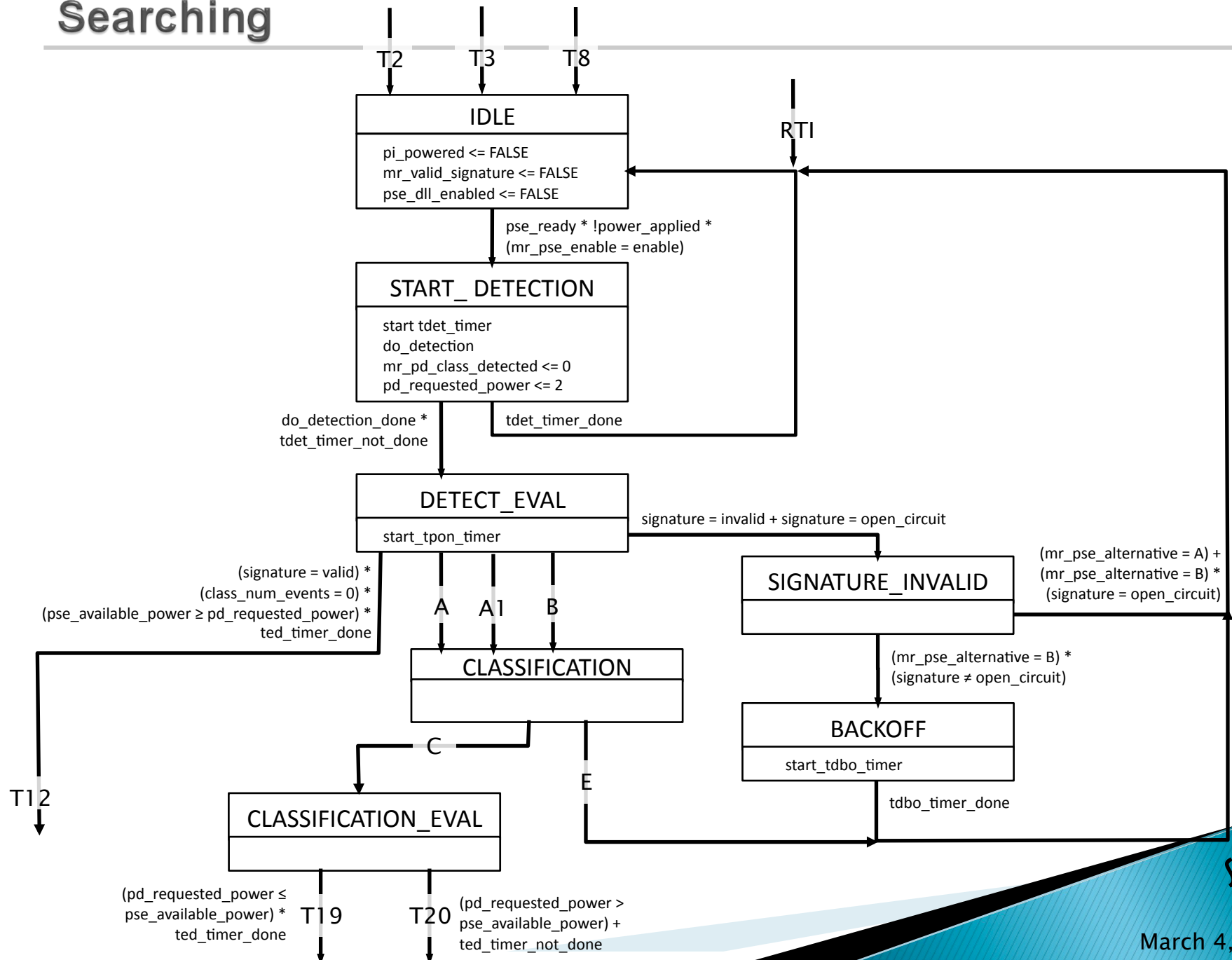
- Disabled State:**
 - Initial state: `DISABLED` (`pi_powered <= FALSE`).
 - Transition **T1** (Event: `mr_pse_enable = disable`) leads to `pi_powered <= FALSE`.
 - Transition **T2** (Event: `mr_pse_enable = enable`) leads to the `IDLE` state.
- IDLE State:**
 - Initial state: `IDLE` (`pi_powered <= FALSE`, `mr_valid_signature <= FALSE`, `pse_dll_enabled <= FALSE`).
 - Transition **T3** (Event: `+ error_condition * (mr_pse_enable = enable)`) leads to the `TEST_MODE` state.
 - Transition **T4** (Event: `+ error_condition * (mr_pse_enable = enable)`) leads to the `TEST_ERROR` state.
 - Transition **T5** (Event: `pse_ready * !power_applied * (mr_pse_enable = enable)`) leads to the `START_DETECTION` state.
- TEST_MODE State:**
 - Initial state: `TEST_MODE` (`pi_powered <= TRUE`).
 - Transition **T6** (Event: `(mr_pse_enable = force_power) * error_condition * !ovld_detected * !short_detected`) leads to the `TEST_ERROR` state.
 - Transition **T7** (Event: `(short_detected + ovld_detected) * (mr_pse_enable = force_power)`) leads to the `TEST_ERROR` state.
 - Transition **T8** (Event: `mr_pse_enable = enable`) leads to the `TEST_ERROR` state.
- TEST_ERROR State:**
 - Initial state: `TEST_ERROR` (`pi_powered <= FALSE`).
 - Transition **T11** (Event: `mr_pse_enable = enable`) leads to the `START_DETECTION` state.
- START_DETECTION State:**
 - Initial state: `START_DETECTION` (`start_tdet_timer`, `do_detection`, `mr_pd_class_detected <= 0`, `pd_requested_power <= 2`).
 - Transition **T9** (Event: `do_detection_done * tdet_timer_not_done`) leads to the `DETECT_EVAL` state.
 - Transition **T10** (Event: `tdet_timer_done`) leads to the `DETECT_EVAL` state.
- DETECT_EVAL State:**
 - Initial state: `DETECT_EVAL` (`start_tpon_timer`).
 - Transition **T12** (Event: `(signature = valid) * (class_num_events = 0) * (pse_available_power > pd_requested_power) * ted_timer_done`) leads to the `CLASSIFICATION_EVAL` state.
 - Transition **T13** (Event: `(signature = invalid) + (signature = open_circuit)`) leads to the `SIGNATURE_INVALID` state.
 - Transition **T14** (Event: `(signature = open_circuit)`) leads to the `SIGNATURE_INVALID` state.
- CLASSIFICATION_EVAL State:**
 - Initial state: `CLASSIFICATION_EVAL`.
 - Transition **T16** (Event: `(pd_requested_power <= pse_available_power) * ted_timer_done`) leads to the `POWER_UP` state.
 - Transition **T20** (Event: `(pd_requested_power > pse_available_power) + ted_timer_not_done`) leads to the `POWER_DENIED` state.
- POWER_UP State:**
 - Initial state: `POWER_UP` (`pi_powered <= TRUE`).
 - Transition **T23** (Event: `(ttrush_timer_not_done * legacy_powerup) * ttrush_timer_done * power_applied * tpon_timer_not_done * (PSE_TYPE = 2)`) leads to the `SET_PARAMETERS` state.
 - Transition **T24** (Event: `(ttrush_timer_not_done * legacy_powerup) + ttrush_timer_done`) leads to the `POWER_ON` state.
- SET_PARAMETERS State:**
 - Initial state: `SET_PARAMETERS` (`set_parameter_type`).
 - Transition **T26** (Event: `(PSE_TYPE = 2) * (pd_dll_power_type = 2) * (parameter_type = 1)`) leads to the `POWER_ON` state.
- POWER_ON State:**
 - Initial state: `POWER_ON`.
 - Transition **T27** (Event: `pse_dll_capable * !pse_dll_enabled`) leads to the `DLL_ENABLE` state.
 - Transition **T28** (Event: `mpdo_timer_done * !short_detected * !ovld_detected * !power_not_available * !option_vport_lm`) leads to the `ERROR_DELAY` state.
 - Transition **T29** (Event: `short_detected + ovld_detected + option_vport_lm`) leads to the `ERROR_DELAY` state.
- DLL_ENABLE State:**
 - Initial state: `DLL_ENABLE` (`pse_dll_enabled <= TRUE`).
 - Transition **T34** (Event: `UCT`) leads to the `ERROR_DELAY` state.
- ERROR_DELAY State:**
 - Initial state: `ERROR_DELAY` (`start_ted_timer`, `pi_powered <= FALSE`).
 - Transition **T35** (Event: `ted_timer_done + option_detect_ted`) leads to the `FAULT` state.
- FAULT State:**
 - Initial state: `FAULT`.

March 4, 2015

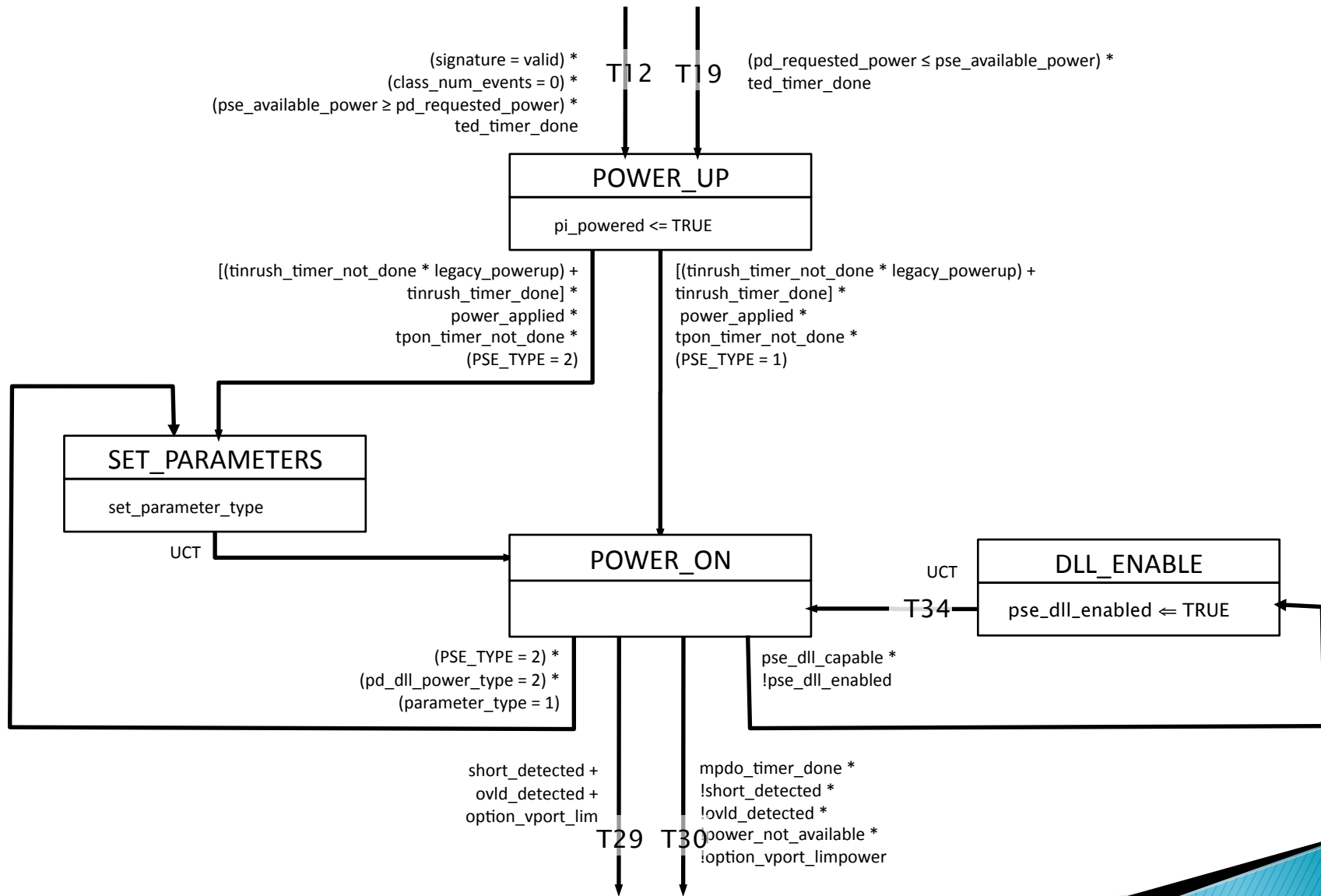
High Level State Diagram



Searching



Delivering Power



Test, Fault & Power Denied

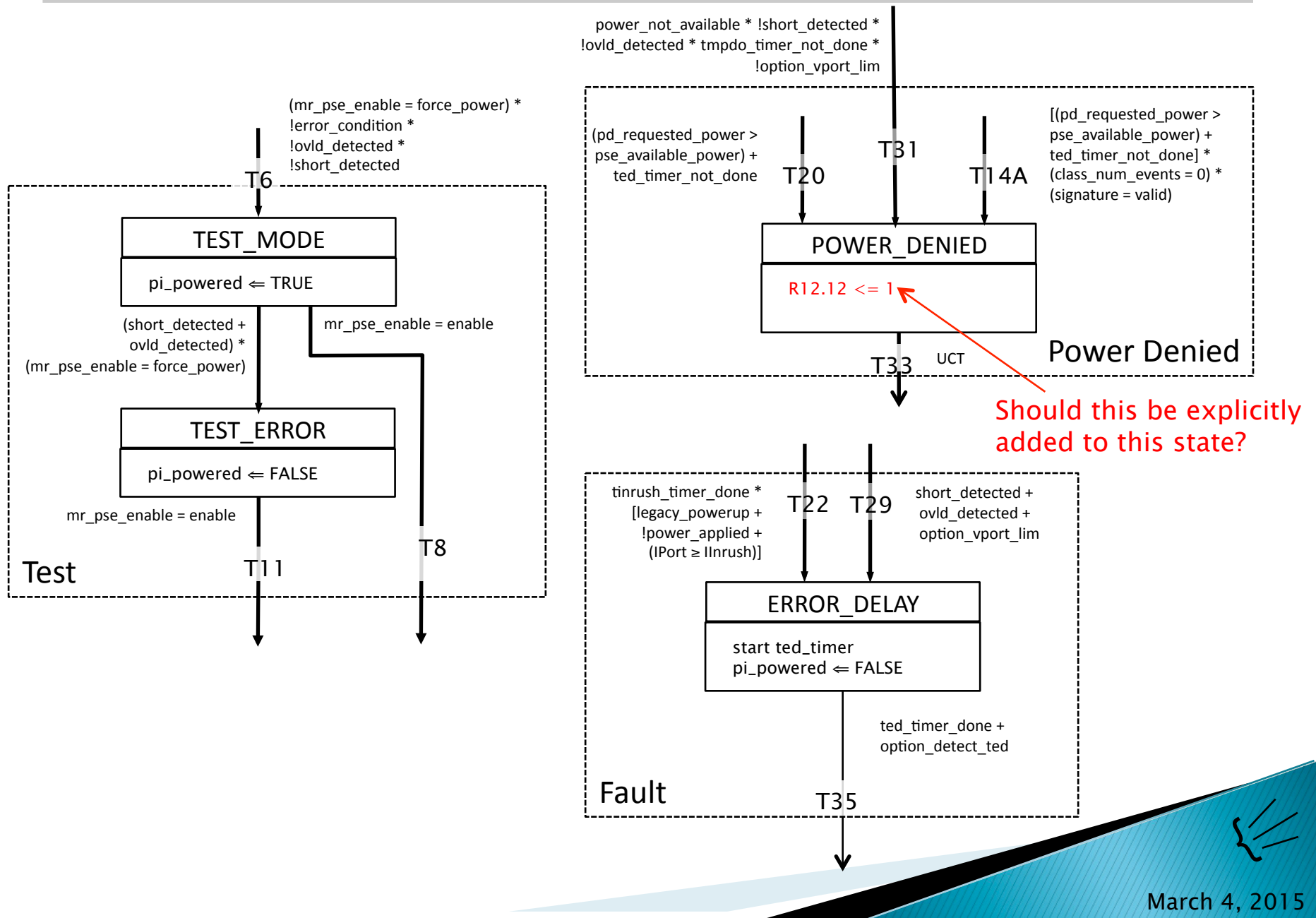


Diagram Transition Simplification

- Identify transition terms that are either common or very complex
- Define new logic terms and then describe that logic in the State Diagram Variables
- Define a common node (RTI) to simplify diagram

Examples:

Complex:

Ready_2_Power = $[(\text{tinrush_timer_not_done} * \text{legacy_powerup}) + \text{tinrush_timer_done}] * \text{power_applied} * \text{tpon_timer_not_done}$

- Makes T23 = $\text{Ready_2_Power} * (\text{PSE_TYPE} = 2)$
- Makes T25 = $\text{Ready_2_Power} * (\text{PSE_TYPE} = 1)$

Common:

Enable_Pwr = $(\text{mr_pse_enable} = \text{enable})$

Disable_Pwr = $(\text{mr_pse_enable} = \text{disable})$

- Simplifies and reduces many terms in readability/size

RTI = Node called “Return To Idle” which is extremely common node in that many states lead to it. Much simpler than showing 8+ arcs all going into IDLE state.



Comments

- This is a “Work in Progress” and expected to draw some constructive criticism, recommendations, etc.
- The outcome if adopted, would be to have multiple diagrams in the specification rather than a single two-page diagram, but these diagrams will be smaller, easier to understand, and easier to modify to accept future changes.
- The classification portion of the existing PSE State Diagram is assumed to be the current page pretty much as is.
- A few minor modifications have been made that are assumed to be errors.
 - A typo taken out of T23 (removed hyphen)
 - A1 arc added to “Searching” block since it appears as entry into the Classification State
- It’s the author’s opinion, that a good specification is done from the top down, rather than the bottom up. The existing PSE state diagram, while it accurately represents behavior of a PSE, appears to have been designed from the bottom up to explain PSE behavior, rather than to direct design.
- I look forward to further discussion on this approach.

