***Comment i-251 and i-252***

In darshan_13_0917 there are two real issues that are identified. These issues are well described and those descriptions are repeated here, without modification.

***Comment i-251 (Page 136 Line 20).***

In the exit from ENTRY_SEC to START_DETECT_SEC, when selecting CC_DET_SEQ 0 or 1, and class_4PID_multi_event_sec = FALSE, the secondary state machine allows to move from ENTRY_SEC state to START_DETECT_SEC only if pwr_app_pri = TRUE per the existing condition:

sism * ((!class_4PID_mult_events_sec * **pwr_app_pri**) + class_4PID_mult_events_sec) * (CC_DET_SEQ=0 + CC_DET_SEQ=1)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | * | 1 | * | 1 | + | 0 | * | 1 | + | 1 |

Result: 1 ➔ Moving to START_DETECT_SEC ➔ OK

If Primary fails to powerup, the Primary state machine returns back to IDLE_PRI. As a result, pwr_app_pri variable will remain in FALSE, and the secondary state machine won't be able to exit from ENTRY_SEC i.e. will be stuck there.

sism * ((!class_4PID_mult_events_sec * **pwr_app_pri**) + class_4PID_mult_events_sec) * (CC_DET_SEQ=0 + CC_DET_SEQ=1)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | * | 1 | * | 0 | + | 0 | * | 1 | + | 1 |

Result: 0 ➔ stuck in ENTRY_SEC.

***Comment i-252 (Page 136 Line 21).***

***The exit from ENTRY_SEC to START_DETECT_SEC:***

In the transition between ENTRY_SEC to START_DET_SEC we have the following condition:
sism * ((!class_4PID_mult_events_sec * pwr_app_pri) + class_4PID_mult_events_sec) * (CC_DET_SEQ=0 + CC_DET_SEQ=1).

When class_4PID_mult_events_sec=FALSE, and CC_DET_SEQ=0 is TRUE or CC_DET_SEQ=1 is TRUE, If START_DET_PRI exit to IDLE_PRI due to tdet_timer_pri_done, the pwr_app_pri will remain in FALSE which won't allow exiting from ENTRY_SEC to START_DETECT_SEC and the secondary state machine remain stuck in ENTRY_SEC.

*Note: Even if we did detection before tdet_timer_pri is expired, we will get tdet_timer_pri_done anyway at some time. There is missing stop timer assignment.*

| | | | | |
|---|---|---|---|---|
| tdet_timer_pri_done | ➔ | FALSE | ➔ | pwr_app_pri remains in FALSE |
| | | | | |

START_DET_PRI exit to IDLE_PRI ➔

sism * ((!class_4PID_mult_events_sec * **pwr_app_pri**) + class_4PID_mult_events_sec) * (CC_DET_SEQ=0 + CC_DET_SEQ=1)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | * | 1 | * | 0 | + | 0 | * | 1 | + | 1 |

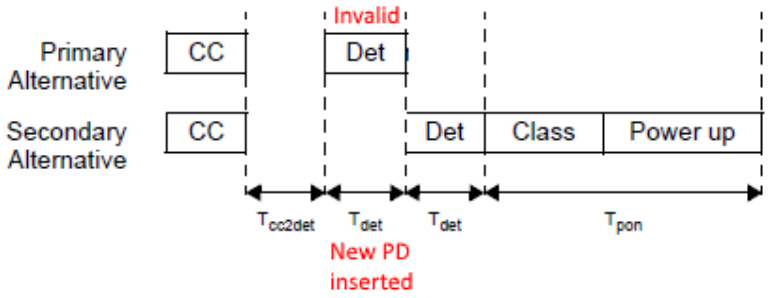
Result: 0 ➔ stuck in ENTRY_SEC.

### *Reflections on darshan_13_0917 proposed fixes for i-251 and i-252*

darshan_13_0917 creates the opportunity for a failed detection on the primary pairset to be followed by a successful detection and power-on on the secondary pairset. This will allow Tpon violations. Tpon violations may occur as do_detect_pri, even if invalid, can take up to 500ms, during which time the object connected to PSE PI may change.

The entire logic behind moving to an explicit description of the CC/DETECT sequences was to make impossible a PD removal-reinsertion during the entire CC-DETECT-CLASS-POWER sequences 0, 1, 2 and 3 for both single and dual signature PDs. Instead of allowing the secondary to power on after the primary has failed (and is in WAIT_PRI) we must instead require a new Connection Check.

Figure 145B–11 illustrates a PSE implementing CC_DET_SEQ=3 when the connection check result is dual.




### *Corrected fixes for i-251 and i-252*

The following proposal ensures that detect/class cycles on the secondary pairset (when primary pairset is not powered) are always for inspection purposes only and will not allow power on from the secondary pairset – thus preserving Tpon. In addition comments i-251 and i-252 are addressed. Further, the implementation of option_probe_alt_sec is (incidentally) corrected so an "inspection-only" detect/class on the secondary side does not apply power per the agreed behavior in stover_01_1116_rev01.pdf.

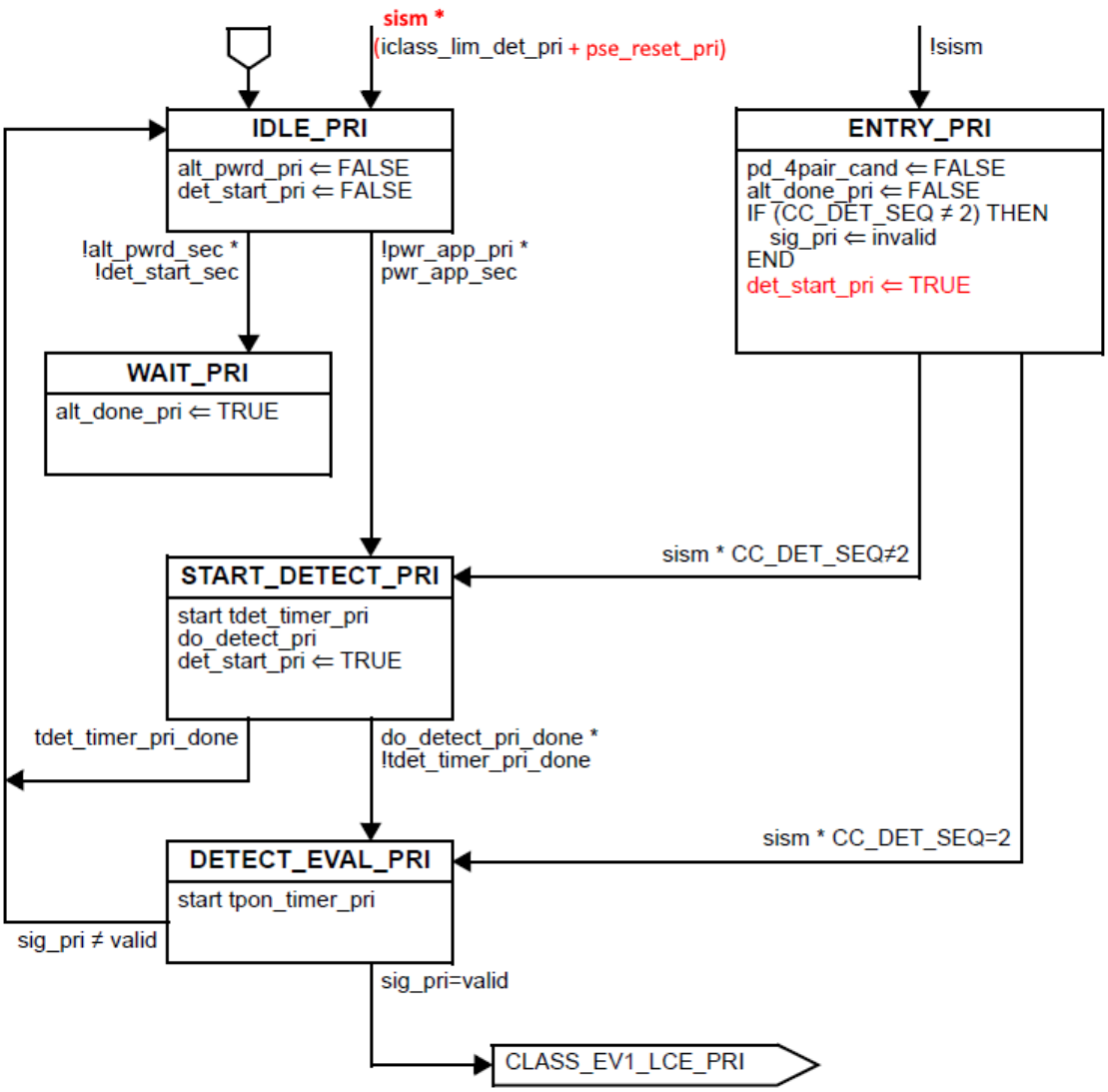
Many other race conditions are addressed per feedback from multiple people. Also invalid-valid sequences becoming powered is correctly addressed.

***1. Add the following variables to 145.2.5.4***

pse_reset_pri
Controls the resetting of the PSE state diagram on Alternative A. Condition that is TRUE until such time as the power supply for the device that contains the PSE overall state diagrams has reached the operating region. It is also TRUE when implementation-specific reasons require reset of PSE Alternative A functionality.
Values:
FALSE: Do not reset the PSE state diagram.
TRUE: Reset the PSE state diagram.

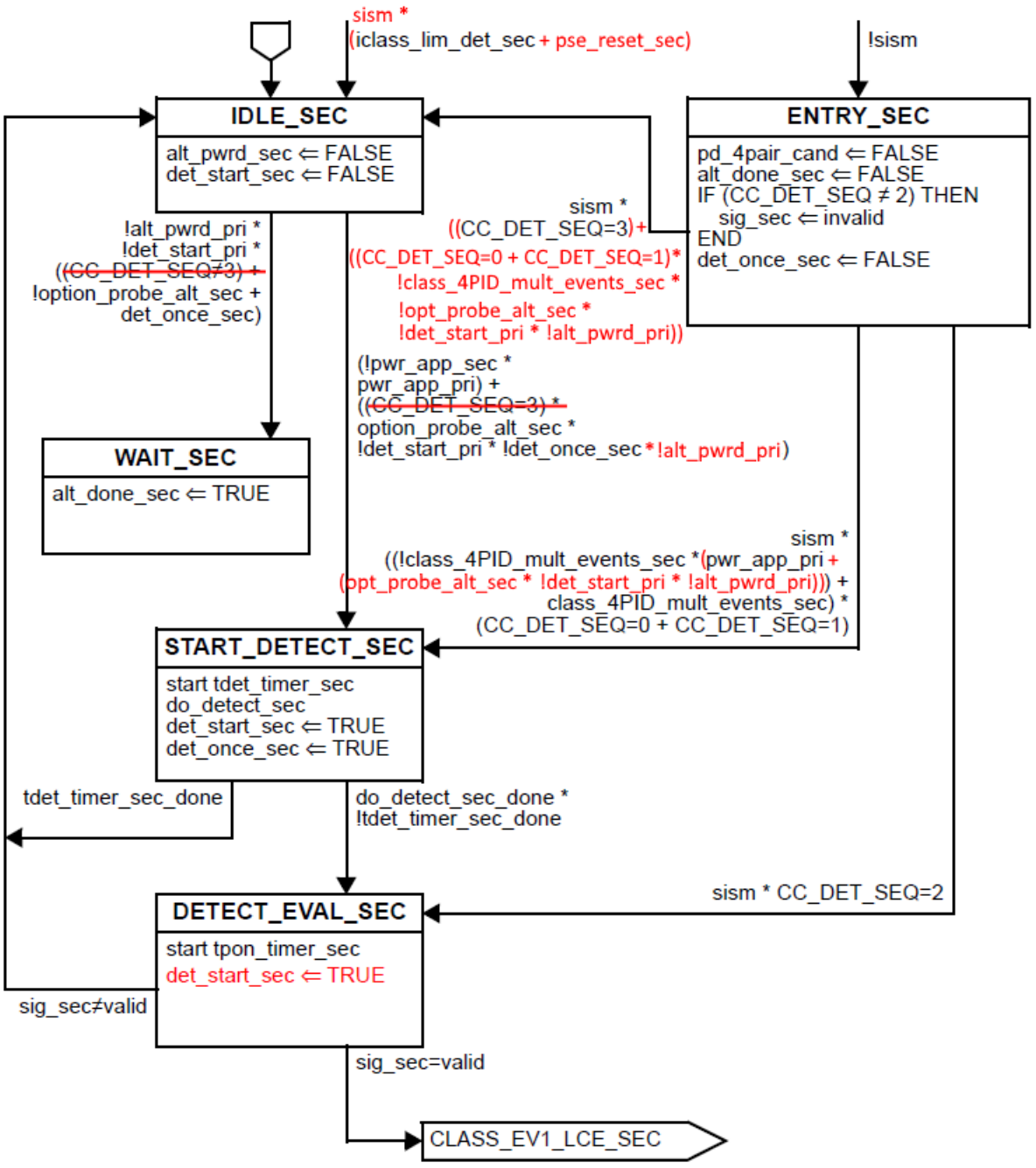
pse_reset_sec
Controls the resetting of the PSE state diagram on Alternative B. Condition that is TRUE until such time as the power supply for the device that contains the PSE overall state diagrams has reached the operating region. It is also TRUE when implementation-specific reasons require reset of PSE Alternative B functionality.
Values:
FALSE: Do not reset the PSE state diagram.
TRUE: Reset the PSE state diagram.

***Modify Figure 145-15 to add sism term IDLE_PRI entry and allow pse_reset_pri***



***Modify Figure 145-16 to allow return to***

- ***add sism term IDLE_PRI entry and allow pse_reset_sec***
- ***Allow SEQ0/1 to exit ENTRY_SEC to IDLE_SEC if staggered detect fails on primary for any reason (!valid or tdet_timer_done)***
- ***Fix SEQ3 IDLE_SEC to START_DETECT_SEC to move uniformly to START_DETECT_SEC***
    - ***Was moving after class in one case and after powerup in the other***
- ***Grant option_probe_alt_sec behavior to SEQ0/1 in ENTRY_SEC to START_DETECT_SEC arc***



***Modify Figure 145-16 to***

- ***Properly terminate an "inspection only" option_probe_alt_sec.***
  - ***This implies staggered detects cannot power secondary if primary detect result is invalid***
  - ***Also preserves tpon***