

*This presentation addresses some comments that were submitted as individual comment, per the rule one comment per issue, however they are related since they touch the same STATES and exit conditions and their solution need to be integrated for completeness and easy and clear instructions for the editor if these comments will be accepted.*

## ***PART A: Comments i-250, i-251(Page 136 Line 20)***

- 1) In the current state machine, we will be stuck in ENTRY\_SEC in the transition from ENTRY\_SEC to START\_DETECT\_SEC due to:
  - 1.1) failing detection and/or classification in the primary that will cause pwr\_app\_pri=FALSE when class\_4PID\_mult\_event\_sec=FALSE or
  - 1.2) tdet\_timer\_pri signal stay done even if we did detection in the primary, continue to DETECT\_EVAL\_PRI and then to IDLE\_PRI due to invalid signature which will keep pwr\_app\_pri=FALSE
- 2) The current state machine in the exit from IDLE\_SEC to START\_DETECT\_SEC doesn't allow doing staggered detection when CC-DET-SEQ=3 when primary is not turn on yet and also prevents doing multiple cycles of detection+classification until host decides to power on the port.

As a result, we need to:

- (1) Eliminate the possibility of being stuck in ENTRY\_SEC in the transition from ENTRY\_SEC to START\_DETECT\_SEC due to failing detection and/or classification in the primary or tdet\_timer\_pri signal stay done event if we did detection in the primary.
- (2) Per the CC\_DET\_SEQ parameter options (0 to 3) to verify that detection and detection+classification cycles can be done in staggered or parallel manner per the CC\_DET\_SEQ definition options and the state machine. This work is focused in CC\_DET\_SEQ=3 (there other options look OK).



## Comments Details:

### Comment i-250 (Page 136 Line 20).

In the exit from ENTRY\_SEC to START\_DETECT\_SEC:

There is redundant parenthesis in the exit from ENTRY\_SEC to START\_DETECT\_SEC:

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

in the part: (!class\_4PID\_mult\_events\_sec \* pwr\_app\_pri).

#### Proposed Remedy:

Change from:

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

To:

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

See darshan\_04\_0917.pdf for additional changes proposed to this condition due to other comments.

### 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.

The straightforward way to handle this problem is to enable moving to START\_DETECT\_SEC from ENTRY\_SEC, also if primary performed detection at least once and is now in IDLE\_PRI state which prevents stuck at ENTRY\_SEC. This solution requires the addition of new variable det\_once\_pri (the current draft has only det\_once\_sec) which is required also by other comments that all related to each other (e.g. to allow the possibility to do cycles of detection + class probe events on primary and secondary with the option to go to IDLE\_PRI/SEC and WAIT\_PRI/SEC and others in this presentation).

#### Proposed Remedy for comment i-251 only (See proposed baseline that addresses all related comments):

1) Add the following variable:

det\_once\_pri

This variable indicates if the PSE has probed the Primary Alternative at least once, when entering to DETECT\_EVAL\_PRI. Values:

FALSE: The PSE has not probed on the Primary Alternative since entering the Primary Alternative state diagram.

TRUE: The PSE has probed the Primary Alternative at least once since entering the Primary Alternative state diagram.

2) Change from:

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

To:

```
sism *((!class_4PID_mult_events_sec * ( pwr_app_pri + det_once_pri * !det_start_pri ) ) + class_4PID_mult_events_sec) * (CC_DET_SEQ=0 + CC_DET_SEQ=1).
```



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

**The exit from ENTRY\_SEC to START\_DETECT\_SEC:**

In the transition between ENTRY\_SEC to START\_DETECT\_SEC we have the following condition:

$\text{sism} * (!\text{class\_4PID\_mult\_events\_sec} * \text{pwr\_app\_pri}) + \text{class\_4PID\_mult\_events\_sec} * (\text{CC\_DET\_SEQ}=0 + \text{CC\_DET\_SEQ}=1)$ .

When  $\text{class\_4PID\_mult\_events\_sec}=\text{FALSE}$ , and  $\text{CC\_DET\_SEQ}=0$  is TRUE or  $\text{CC\_DET\_SEQ}=1$  is TRUE, If **START\_DET\_PRI** exit to **IDLE\_PRI due to tdet\_timer\_pri\_done**, the  $\text{pwr\_app\_pri}$  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.*

Time	t1	t2	t3	t4	t5
Pri	START_DETECT_PRI	do_detect_Pri_done = FALSE tdet_timer_pri_done = TRUE	Exit to IDLE_PRI	alt_pwr_sec=FALSE det_start_sec=FALSE	Exit to WAIT_PRI
Sec	ENTRY_SEC	<i>pwr_app_pri = FALSE</i>	<i>Stay at ENTRY_SEC</i>	<i>Stay at ENTRY_SEC</i>	<i>stay at ENTRY_SEC</i>

$\text{sism} * (!\text{class\_4PID\_mult\_events\_sec} * \text{pwr\_app\_pri}) + \text{class\_4PID\_mult\_events\_sec} * (\text{CC\_DET\_SEQ}=0 + \text{CC\_DET\_SEQ}=1)$

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

Result: 0 → **stuck in ENTRY\_SEC.**

**Proposed Remedy for comment i-252 only (See proposed baseline that addresses all related comments):**

Make the following changes to allow to move to START\_DETECT\_SEC in case that we move from START\_DETECT\_PRI to IDLE\_PRI due to tdet\_timer\_pri expiration:

1. "sism \* (!class\_4PID\_mult\_events\_sec \* ( pwr\_app\_pri + tdet\_timer\_pri\_done ) ) + class\_4PID\_mult\_events\_sec) \* (CC\_DET\_SEQ=0 + CC\_DET\_SEQ=1)"
2. Add stop\_tdet\_timer\_pri in the DETECT\_EVAL\_PRI state in order to ensure that tdet\_timer\_pri\_done signal stay FALSE when the primary state machine moves from START\_DETECT\_PRI to DETECT\_EVAL\_PRI and forward down the road.

**Due to the fact that item 2 need additional changes due to other comments, and in order to meet the requirement that we need single independent comment for each issue which I did here but may cause editor confusion of how to apply the remedies of other comments, see darshan\_13\_0917.pdf.**



## Summary for comments (i-250, i-251, i-252):

Step by step changing the condition from ENTRY\_SEC to START\_DETECT\_SEC

### 1. The original condition

$\text{sism} * (!\text{class\_4PID\_mult\_events\_sec} * \text{pwr\_app\_pri}) + \text{class\_4PID\_mult\_events\_sec} * (\text{CC\_DET\_SEQ}=0 + \text{CC\_DET\_SEQ}=1)$

### 2. (i-251): To resolve detection or classification failure that cause us to return to IDLE\_PRI and stuck in ENTRY\_SEC due to pwr\_app\_pri=FALSE

$\text{sism} * (!\text{class\_4PID\_mult\_events\_sec} * (\text{pwr\_app\_pri} + \text{det\_once\_pri} * !\text{det\_start\_pri})) + \text{class\_4PID\_mult\_events\_sec} * (\text{CC\_DET\_SEQ}=0 + \text{CC\_DET\_SEQ}=1)$ .

### 3. (i-252): To resolve tdet\_timer\_pri expiration that cause us to return to IDLE\_PRI from START\_DETECT\_PRI and remain stuck in ENTRY\_SEC due to pwr\_app\_pri=FALSE

" $\text{sism} * (!\text{class\_4PID\_mult\_events\_sec} * (\text{pwr\_app\_pri} + \text{tdet\_timer\_pri\_done})) + \text{class\_4PID\_mult\_events\_sec} * (\text{CC\_DET\_SEQ}=0 + \text{CC\_DET\_SEQ}=1)$ "

### 4. The combined solution:

$\text{sism} * (!\text{class\_4PID\_mult\_events\_sec} * (\text{pwr\_app\_pri} + \text{det\_once\_pri} * !\text{det\_start\_pri} + \text{tdet\_timer\_pri\_done})) + \text{class\_4PID\_mult\_events\_sec} * (\text{CC\_DET\_SEQ}=0 + \text{CC\_DET\_SEQ}=1)$ .

### 5. Final condition:

$\text{sism} * ( (!\text{class\_4PID\_mult\_events\_sec} * (\text{pwr\_app\_pri} + \text{det\_once\_pri} * !\text{det\_start\_pri} + \text{tdet\_timer\_pri\_done})) + \text{class\_4PID\_mult\_events\_sec} * (\text{CC\_DET\_SEQ}=0 + \text{CC\_DET\_SEQ}=1) )$ .

*Note: The red marked parenthesis can be removed for clarity since logically they are redundant and doesn't change the logic. See comment i-250 that addresses this topic.*



## PART B: Comment i-254 (Page 136 Line 11).

### The exit from IDLE\_SEC to START\_DETECT\_SEC.

In the exit from IDLE\_SEC to START\_DETECT\_SEC we have the following condition:

$(!pwr\_app\_sec * pwr\_app\_pri) + ((CC\_DET\_SEQ=3) * option\_probe\_alt\_sec * !det\_start\_pri * !det\_once\_sec.$

Based on the description in page 109 lines 37-38 for CC\_DET\_SEQ and specifically, CC\_DET\_SEQ=3 for dual-signature means:

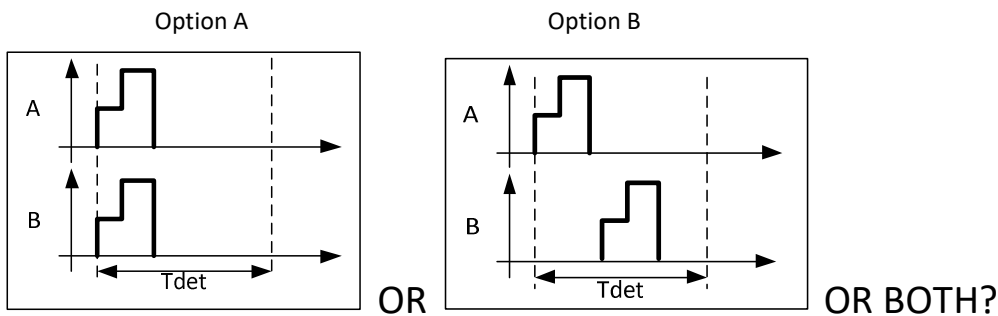
Connection check is followed by staggered detection

(The analysis and simulations results for other sequences 0, 1 and 2 are covered by other comments and most of them are OK).

The first issue is that "parallel detection" and "staggered detection" are not clearly defined in D3.0, however we know what we wanted to do per the following presentations and discussions:

- a) page 15 of [http://www.ieee802.org/3/bt/public/nov14/darshan\\_11\\_1114\\_rev\\_07.pdf](http://www.ieee802.org/3/bt/public/nov14/darshan_11_1114_rev_07.pdf), we introduce the practical definitions of it as part of the way how to find PD configuration (one implementation of connection check).
- b) page 7 in [http://www.ieee802.org/3/bt/public/jun15/abramson\\_01bt\\_0615.pdf](http://www.ieee802.org/3/bt/public/jun15/abramson_01bt_0615.pdf)

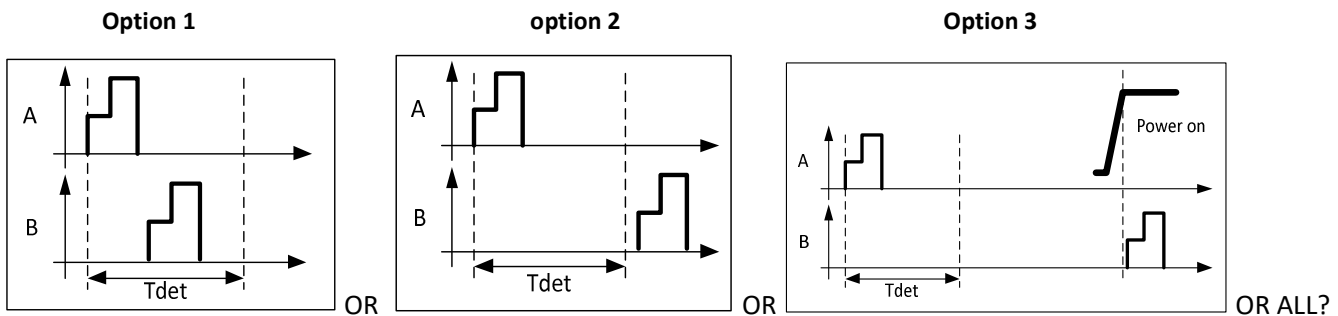
In "parallel detection" the intent was to do simultaneous detection on both pairsets in order to check if detection signature is corrupted when single-signature PD is connected. Now, in D3.0, it is not clear if it is " simultaneous detection on both pairsets" or detecting both alternative in one detection cycle (Tdet) or both.



### We need to decide which definition to adopt?

In staggered detection, it should be any non-overlapping detections, with no limits on tdet2ted.

As a result, staggered detection could be any of the following:



The last two options must be covered (2 and 3). The first option (1) depends on what is the definition for "parallel" (See above).

Staggered detection range may occur with starting the secondary detection after doing the primary detection (option 1 and 2) up to doing the secondary detection when the primary is on (option 3). This covers the full range of possibilities for staggered detection.

Option 1 and 2 in CC\_DET\_SEQ=3 is normally used when class\_4PID\_mult\_events\_sec=TRUE. This currently is not covered by the state machine for CC\_DET\_SEQ=3.

Option 3 in CC\_DET\_SEQ=3 is normally used when class\_4PID\_mult\_events\_sec=FALSE and it is covered in the 1st part of the condition:  $(!pwr\_app\_sec * pwr\_app\_pri).$



Option 3 covers ONLY the case that the primary return to IDLE\_PRI due to various reasons and the secondary didn't detect even once: ((CC\_DET\_SEQ=3) \* option\_probe\_alt\_sec \* !det\_start\_pri \* !det\_once\_sec).

The state diagram should allow staggered detection:

- before Primary power up,
- after primary power up,
- and during power up in case that class\_4PID\_mult\_events\_sec is set to FALSE.

The proposed changes in the state diagram will allow staggered detection after Primary finished its 1st detection without affecting the previous functionality and flow, by OR-ing the additional missing possibility (option 1 and 2).

The proposed changes do not affect:

- a) The behavior of other "CC\_DET\_SEQ NE 3" flows.
- b) Previous state diagram possibilities.

In addition, the proposed changes also required to cover multiple cycles of detection+classification until host decides to power on the port (which is covered by darshan\_04\_0917.pdf).

The additional missing possibility is covered by adding the following part:

+ (class\_4PID\_mult\_events\_sec\*(CC\_DET\_SEQ=3) \* !det\_once\_sec \* det\_once\_pri )

In order to implement the addition, we need to add the following variable for the primary side (similar variable is already existing for the secondary):

"det\_once\_pri

This variable indicates if the PSE has probed the Primary Alternative at least once, when entering to DETECT\_EVAL\_PRI.

Values:

FALSE: The PSE has not probed on the Primary Alternative since entering the Primary Alternative state diagram.

TRUE: The PSE has probed the Primary Alternative at least once since entering the Primary Alternative state diagram."

In the above proposed change, det\_once\_pri is used as a condition for starting detection in the secondary any time, after primary was detected at least once.

det\_once\_pri is set to FALSE when sism = FALSE at ENTRY\_PRI.

det\_once\_pri is set to TRUE when Primary state diagram reaches to "DETECT\_EVAL\_PRI", to clearly indicate that detection on primary has ended before tdet\_timer\_pri expired.

### Proposed Remedy for **Comment i-254 only:**

1. Change from:

"(!pwr\_app\_sec \*pwr\_app\_pri) + ((CC\_DET\_SEQ=3) \* option\_probe\_alt\_sec \* !det\_start\_pri \* !det\_once\_sec)"

To:

"(!pwr\_app\_sec \*pwr\_app\_pri) + ((CC\_DET\_SEQ=3) \* option\_probe\_alt\_sec \* !det\_start\_pri \* !det\_once\_sec) + (class\_4PID\_mult\_events\_sec\*(CC\_DET\_SEQ=3) \* !det\_once\_sec \* det\_once\_pri )

2. Add the following variable to the variable list:

det\_once\_pri

This variable indicates if the PSE has probed the Primary Alternative at least once, when entering to DETECT\_EVAL\_PRI. Values:

FALSE: The PSE has not probed on the Primary Alternative since entering the Primary Alternative state diagram.

TRUE: The PSE has probed the Primary Alternative at least once since entering the Primary Alternative state diagram.

3. Properly defined what is "parallel" detection and what is "staggered" detection in CC\_DET\_SEQ definition on page 109 after line by addition the following text:

"For single-signature PD, parallel detection means that detection on both pairsets are done within Tdet time period.

" For dual-signature PD, parallel detection means that detection on both pairsets are done within the same Tdet time period.

"For single-signature PD, staggered detection means [Not clearly defined. Group to discuss].

" For dual-signature PD, staggered detection means [Not clearly defined. Group to discuss].

**Baseline starts here**



## Proposed Remedy

### 1. Add the following variables to 145.2.5.4

#### det\_once\_pri

This variable indicates if the PSE has probed the Primary Alternative at least once, when entering to DETECT\_EVAL\_PRI. Values:  
 FALSE: The PSE has not probed on the Primary Alternative since entering the Primary Alternative state diagram.  
 TRUE: The PSE has probed the Primary Alternative at least once since entering the Primary Alternative state diagram.

TRUE: The PSE probes for the PD requested Class.

### 2. Properly defined what is "parallel" detection and what is "staggered" detection in CC\_DET\_SEQ definition on page 109 after line by addition the following text:

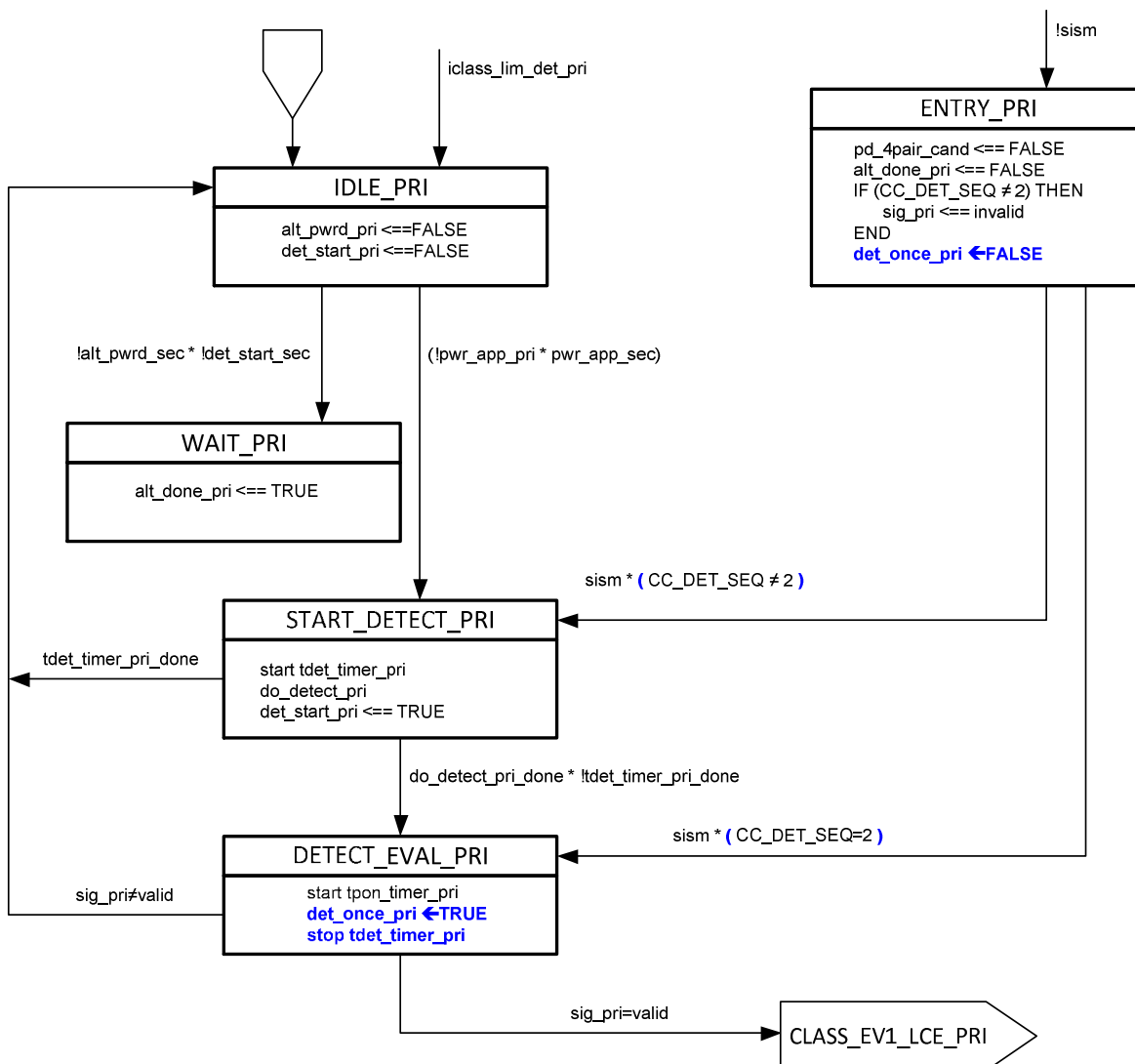
"For single-signature PD, parallel detection means that detection on both pairsets are done within Tdet time period.

" For dual-signature PD, parallel detection means that detection on both pairsets are done within the same Tdet time period.

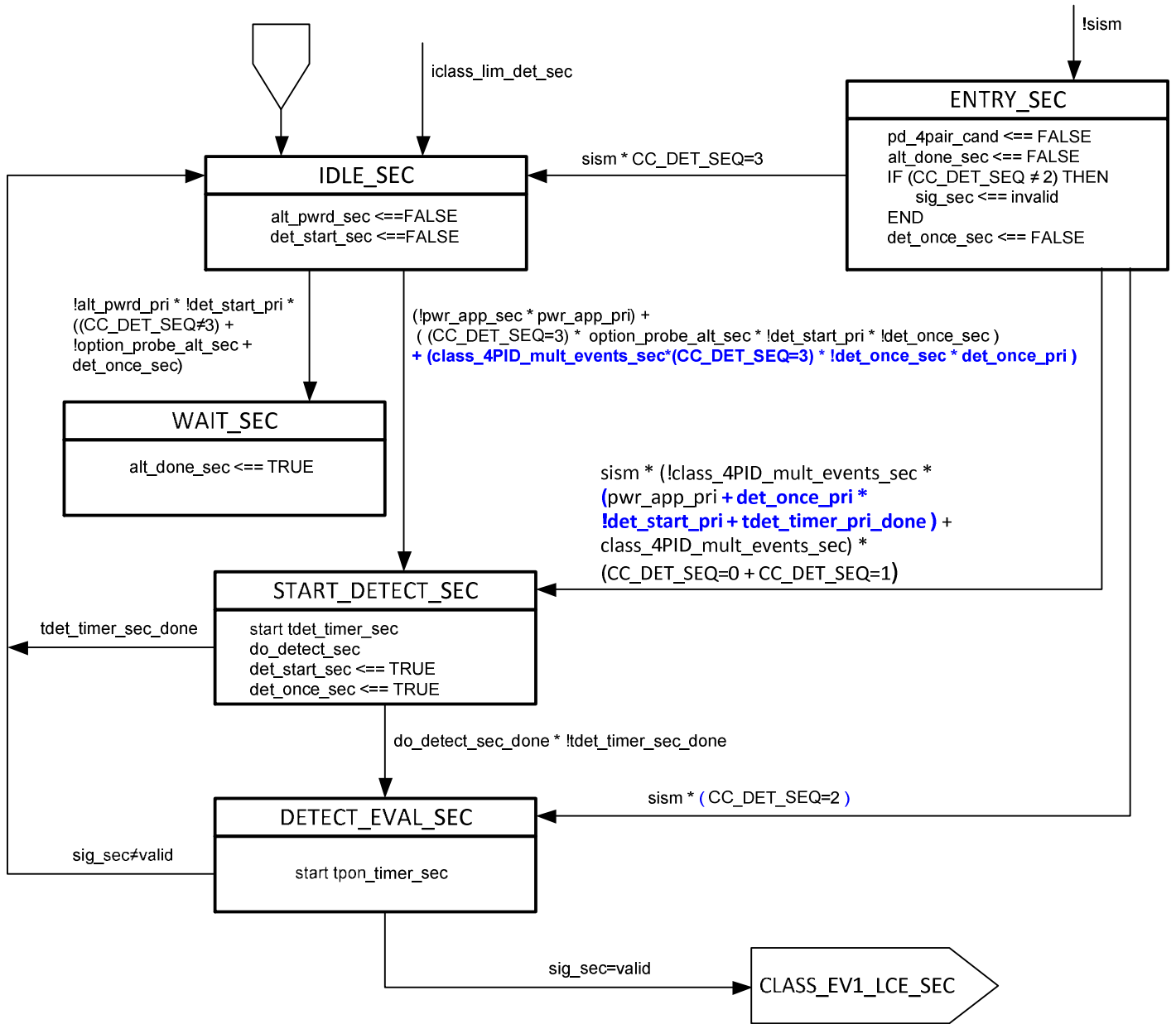
"For single-signature PD, staggered detection means [Not clearly defined. Group to discuss to be sync with SM].

" For dual-signature PD, staggered detection means [Not clearly defined. Group to discuss to be sync with SM].

### 3. Make the following changes to the state machine:



4. Make the following changes to the state machine:



End of Baseline

