D1.8 / D2.0 state diagram fix v101

Lennart Yseboodt, Matthias Wendt
Philips Lighting – Research
July 21, 2016

innovation + you

PHILIPS

Issue 1

From CLASS_EVAL to POWER_UP the condition is “pd_req_pwr < pse_avail_pwr” which has the effect that if the PSE has Class 1 available and the PD requests Class 1 the PSE will hang in CLASS_EVAL. The same applies to Class 2.
Changing it to “pd_req_pwr $\leq$ pse_avail_pwr” fixes the issue.

Issue 2

From CLASS_EV1_LCE the exits to MARK_EV1 and MARK_EV_LAST forget to check the variable pse_avail_pwr.
Currently the SD would allocate more power than is available.

Issue 3

Same as issue 2, in the state CLASS_EV2, the variable pse_avail_pwr is not taken into account. This leads to power over-allocation.

Issue 4

Again, same as issue 2 and 3, in the state CLASS_EV4, the variable pse_avail_pwr is not taken into account. This leads to power over-allocation.

Issue 5

From the IDLE state, the branch into START_CXN_CHK and the branch into START_DETECT can be True simultaneously when CC_DET_SEQ $\neq$ 1 and mr_pse_alternative $\neq$ 'both'.
Going through connection check only makes sense when mr_pse_alternative = 'both'.

PHILIPS

Issue 6

From DETECT_EVAL to IDLE (label A), parenthesis are missing around “(CC_DET_SEQ = 0) + (CC_DET_SEQ = 3)”.
Without these, the AND takes precedence over the OR.

Issue 7

The SD still uses ‘tacs_timer’ which has been renamed to ‘tclassacs_timer’.

Issue 8

The PSE inrush monitors have an inherent race condition. In the POWER_UP state, when alt_pwrd_pri is set the tinrush timer is started. Whenever inrush is completed, pwr_app_pri gets set to True. At that point, both the exit check out of POWER_UP to POWER_ON is evaluated, and, at the same time, the inrush monitor moves back to IDLE_INRUSH_PRI where the tinrush timer is stopped. This produces undefined behavior.

Note: this issue was fixed by mr. Stovers comment #122 against D1.7 but seems to have fallen through the cracks. We need also an MR to fix the legacy SD.


PHILIPS