

Rules for state diagram variables (D3.1) v110

Info (not part of baseline)

Our state diagrams are inordinately complex, with a very large number of variables (current count 163 for the PSE). Given that our state diagrams mutated out of the Clause 33 state diagrams, we have low consistency in our variable descriptions. Specifically, it is unclear what the rules are pertaining to each variable:

- may it be set externally ?
- only in IDLE, or at any time ?
- is it a state diagram internal variable ?
- is it a variable that must be set according to certain rules (eg. mps_valid) ?

The current descriptions don't help. Some examples:

Variable	Description	Implicit rules
alt_done_pri	A variable used to coordinate...	reserved for the state diagram
alt_pri	A variable used to select...	this is a config variable
alt_pwrd_pri	A variable that controls...	also reserved for the state diagram
autoclass_enable	A control variable indicating...	configuration
class_4PID_mult_events_pri	A variable indicating...	configuration
det_once_sec	This variable indicates...	reserved for state diagram
MirroredPDAutoclassRequest	A control variable output...	reserved for state diagram
mps_valid	This variable indicates the presence or absence of a valid MPS...	mandatory set per requirements

If we don't specify the 'usage rules' of variables, the state diagram can be made to do basically anything.

145. Power over Ethernet

Info (not part of baseline)

Clause 33 uses the terminology "control variable" which may have been used to indicate that the variable could be changed by the device or through the management registers. Because our state diagrams are based on the Clause 33 text, we still list several "control variables", but this has no meaning.

Replace throughout all variable lists in Clause 145 "control variable" by "variable".

145.2.5.4 Variables

Info (not part of baseline)

This baseline now assigns a variable type to each variable in Clause 145. This in many cases requires a text change of the variable description. Aim is to get everthing in the following form:

example_variable

A variable used to show an example.

Values:

FALSE: The example is false.

TRUE: The example is true.

In order not to have to duplicate the entire variable list, the baseline will list in tabular form the variable name and the designated variable type. For example:

Variable	Type	Additional instructions
alt_done_pri		—
alt_pri		—
ovld_det_pri		Add: "This variable is set per this description at any time."

Which would result in the following changes (alt_done_pri and alt_pri results in no changes):

alt_done_pri

A variable used to coordinate the main single-signature state diagram with the semi-independent dual-signature state diagram for the Primary Alternative.

Values:

FALSE: The semi-independent state diagram is not ready to return to IDLE within the single-signature state diagram.

TRUE: The semi-independent state diagram is ready to return to IDLE within the single-signature state diagram.

alt_pri

A variable used to select which Alternative assumes the role of Primary Alternative in the state diagram.

Values:

a: Alternative A assumes the role of Primary Alternative. When operating over 4 pairs, Alternative B assumes the role of Secondary Alternative.

b: Alternative B assumes the role of Primary Alternative. When operating over 4 pairs, Alternative A assumes the role of Secondary Alternative.

ovld_det_pri

A variable indicating if the PSE output current has been in an overload condition on the Primary Alternative; see 145.2.8.7. **This variable may be set at any time by the PSE.**

Values:

FALSE: The PSE has not detected an overload condition on the Primary Alternative.

TRUE: The PSE has detected an overload condition on the Primary Alternative.

For variables that control a certain behavior (eg. alt_pri), a function will be called in IDLE (initialize) that will return the complete set of variables that may be externally set in IDLE. Per the standing state diagram conventions, **all** variables may not be changed 'externally' unless this is explicitly noted in the variable description.

Update the descriptions of the variables in 145.2.5.4 per the format in the information box as follows:

Variable	Type	Additional instructions
CC_DET_SEQ		—
alt_done_pri		—
alt_done_sec		—
alt_pri		—
alt_pwrd_pri		—
alt_pwrd_sec		—
autoclass_enable		—
class_4PID_mult_events_pri		—

Variable	Type	Additional instructions
class_4PID_mult_events_sec		—
det_once_sec		—
det_start_pri		—
det_start_sec		—
det_temp		—
dll_4PID		Add: “This variable is assigned through Table 145–40.”
error_condition		Add: “This variable may be set by the PSE/PD at any time.”
error_condition_pri		Add: “This variable may be set by the PSE/PD at any time.”
error_condition_sec		Add: “This variable may be set by the PSE/PD at any time.”
iclass_lim_det		Add: “This variable is set per this description at any time.”
iclass_lim_det_pri		Add: “This variable is set per this description at any time.”
iclass_lim_det_sec		Add: “This variable is set per this description at any time.”
MirroredPDAutoclassRequest		Add: “This variable is assigned through Table 145-40.”
mps_valid		Add: “This variable is set per this description at any time.”
mps_valid_pri		Add: “This variable is set per this description at any time.”
mps_valid_sec		Add: “This variable is set per this description at any time.”
option_2ev		—
option_class_probe		—
option_detect_ted		—
option_detect_ted_pri		—
option_detect_ted_sec		—
option_probe_alt_sec		—
option_vport_lim		—
option_vport_lim_pri		—
option_vport_lim_sec		—
ovld_det_pri		Add: “This variable is set per this description at any time.”
ovld_det_sec		Add: “This variable is set per this description at any time.”
pd_4pair_cand		—
pd_class_4PID_pri		—
pd_class_4PID_sec		—
pd_req_pwr		—
power_available		Add: “This variable may be set by the PSE/PD at any time.”
power_available_pri		Add: “This variable may be set by the PSE/PD at any time.”
power_available_sec		Add: “This variable may be set by the PSE/PD at any time.”
pse_allocated_pwr		—
pse_alternative		—
pse_avail_pwr		—
pse_avail_pwr_pri		—
pse_avail_pwr_sec		—
pse_dll_capable		—
pse_dll_enable		—
pse_enable		Add: “This variable may be set by the PSE/PD at any time.”
pse_power_update		Add: “This variable may be set by the PSE/PD at any time.”
pse_power_update_pri		Add: “This variable may be set by the PSE/PD at any time.”
pse_power_update_sec		Add: “This variable may be set by the PSE/PD at any time.”
pse_ready		Add: “This variable may be set by the PSE/PD at any time.”
pse_reset		Add: “This variable is set per this description at any time.”
pse_reset_pri		Add: “This variable is set per this description at any time.”
pse_reset_sec		Add: “This variable is set per this description at any time.”
pse_ss_mode		Add: “This variable may be set by the PSE/PD at any time.”
pse_ss_mode_update		Add: “This variable may be set by the PSE/PD at any time.”
pwr_app_pri		Add: “This variable is set per this description at any time.”
pwr_app_sec		Add: “This variable is set per this description at any time.”
semi_pwr_en		—
short_det_pri		Add: “This variable is set per this description at any time.”
short_det_sec		Add: “This variable is set per this description at any time.”
sism		—
temp_var		—
temp_var_pri		—
temp_var_sec		—

145.2.5.6 Functions

Add a new function “initialize” as follows:

initialize

This function returns the following variables (see 145.2.5.4):

alt_pri
autoclass_enable
class_4PID_mult_events_pri
class_4PID_mult_events_sec
option_2ev
option_class_probe
option_detect_ted
option_detect_ted_pri
option_detect_ted_sec
option_probe_alt_sec
option_vport_lim
option_vport_lim_pri
option_vport_lim_sec
pse_alternative
pse_avail_pwr
pse_avail_pwr_pri
pse_avail_pwr_sec
pse_dll_capable
semi_pwr_en

Update the descriptions of the variables in 145.2.5.6 per the format in the information box as follows:

Function	Variable	Type	Additional instructions
do_autoclassification	pd_autoclass		—
do_class_probe	pd_req_pwr		—
do_class_probe_pri	pd_req_pwr_pri pd_cls_4PID_pri		—
do_class_probe_sec	pd_req_pwr_sec pd_cls_4PID_sec		—
do_classification	pd_class_sig		—
do_classification_pri	pd_req_pwr_pri pse_allocated_pwr_pri pd_class_sig_pri		—
do_classification_sec	pd_req_pwr_sec pse_allocated_pwr_sec pd_class_sig_sec		—
do_cxn_chk	sig_type		—
do_detect_pri	sig_pri		—
do_detect_sec	sig_sec		—
do_update_pse_allocated_pwr	pse_allocated_pwr		—
do_update_pse_allocated_pwr_pri	pse_allocated_pwr_pri		—
do_update_pse_allocated_pwr_sec	pse_allocated_pwr_sec		—

145.2.5.7 State diagrams

Add a call to “initialize” as the first statement in the IDLE state in Figure 145–13.

145.3.3.3 Single-signature PD constants

Move *pd_req_class* to 145.3.3.4.

145.3.3.4 Single-signature variables

Update the descriptions of the variables in 145.3.3.2 per the format in the information box as follows:

Variable	Type	Additional instructions
pd_req_class		—
mdi_power_required		Add: “This variable may be set by the PSE/PD at any time.”
nopower		—
pd_acs_full_power		—
pd_acs_req		—
pd_autoclass_enable		—
pd_dll_capable		—
pd_dll_enable		—
pd_max_power		—
pd_reset		Add: “This variable may be set by the PSE/PD at any time.”
PDAutoclassRequest		—
present_class_sig_0		—
present_class_sig_A		—
present_class_sig_B		—
present_det_sig		—
present_mark_sig		—
present_mps		—
pse_assigned_class		—
pse_power_level		—
V _{Mark_th}		—
V _{PD}		—
V _{Off_PD}		—
V _{On_PD}		—
V _{Reset_th}		—

145.3.3.6 Single-signature PD functions

Add a new function “initialize” as follows:

initialize

This function returns the following variables (see 145.3.3.4):

pd_req_class
pd_autoclass_enable
pd_dll_capable

Update the descriptions of the variables in 145.3.3.6 per the format in the information box as follows:

Function	Variable	Type	Additional instructions
do_class_timing	long_class_event		—
do_update_pse_assigned_class	pse_assigned_class		—

145.3.3.7 Single-signature PD state diagrams

Add a call to “initialize” to the IDLE state in Figure 145–26.

145.3.3.9 Dual-signature PD variables

Update the descriptions of the variables in 145.3.3.9 in the same way as 145.3.3.4.

145.3.3.11 Dual-signature PD functions

Update the descriptions of the variables in 145.3.3.11 in the same way as 145.3.3.6.

145.5.3.3.1 Variables

Update the descriptions of the variables in 145.5.3.3.1 per the format in the information box as follows:

Variable	Type	Additional instructions
MirroredPDAutoclassRequest		Add: “This variable is assigned through Table 145-40.”
MirroredPDRrequestedPowerValue		Add: “This variable is assigned through Table 145-40.”
MirroredPSEAllocatedPowerValueEcho		Add: “This variable is assigned through Table 145-40.”
PDRrequestedPowerValueEcho		—
PSEAllocatedPowerValue		—
PSEAutoclassCompleted		—
PSEAutoclassSupport		—
pse_initial_value		Add: “This variable is set per this description at any time.”
local_system_change		Add: “This variable may be set by the PSE/PD at any time.”
pse_alternative		—
pse_dll_enable		—
pse_dll_ready		Add: “This variable may be set by the PSE/PD at any time.”
pse_power_update		—
sig_type		—

145.5.3.3.2 Functions

Update the descriptions of the variables in 145.5.3.3.2 per the format in the information box as follows:

Function	Variable	Type	Additional instructions
pse_power_review	pse_new_value		—

145.5.3.4.1 Constants

Move pd_dllmax_value to 145.5.3.4.2.

145.5.3.4.2 Variables

Update the descriptions of the variables in 145.5.3.4.2 per the format in the information box as follows:

Variable	Type	Additional instructions
pd_dllmax_value		Add: “This variable is set per this description at any time.”
MirroredPDRrequestedPowerValueEcho		Add: “This variable is assigned through Table 145-40.”

Update the variables in 145.5.3.6 and 145.5.3.7 in the same manner as 145.5.3.3 and 145.5.3.4 respectively.