Error reporting in SVI FF

# A summary of FF error reporting

FF specifies two main methods of error reporting:

1. Block error (<Block>.BLOCK\_ERR bitmap)
2. Field Diagnostics (RB.FD\_SIMULATE.DIAGNOSTIC\_VALUE bitmap)

From there, alert notifications are generated automatically by (Softing) FF stack, with small help from application (firmware) code.

## Constrains

Scattered throughout FF specs are requirements not to duplicate error reporting between BLOCK\_ERR of different blocks (e.g. TB and AI), or between a BLOCK\_ERR and FD DIAGNOSTIC\_VALUE.

E.g. FF-891 5.5.8 states: “Configuration errors detected by Block\_Err must not be included in Field Diagnostic Conditions”.

# Native SVI FF error detection

SVI FF reports error conditions natively in TB, and then maps any detected errors to FF methods. There are two ways of detecting errors:

1. TB.COMPLETE\_STATUS indicates special conditions in the positioner processor (known as APP).  
   Not all conditions are faults or errors but some are.
2. TB.ALERT\_STATE indicates alerts configured in FF processor (known as FFP) although the last 6 alerts are mapped from TB.COMPLETE\_STATUS.

TB.COMPLETE\_STATUS shows both current and history status; the latter means “was detected but since disappeared” and is informational only. For the description of each status bit, refer to the **fault matrix**, sheet **Faults.** The status may be cleared using TB.CLEAR\_STATUS parameter but if the condition is detected again, the corresponding bit will be set again.

TB.ALERT\_STATE shows the state of

1. Configurable alerts (deviation\_alert to ip\_drive\_current\_lo); for an example, see TB.TEMPERATURE\_LO\_ALERT; every such alert can be individually enabled or disabled
2. Alerts mapped from TB.COMPLETE\_STATUS. They are not configurable and are set per **fault matrix**, sheet **FaultsToAlarms**, col. **FD Alarm**. These alerts cannot be disabled.

# Mapping SVI FF errors to FF format

## Mapping to BLOCK\_ERR

First, general BLOCK\_ERR bitmap is constructed as follows:

1. TB.XD\_ERROR\_POS, TB\_XD\_ERROR\_PRESSURE, TB\_XD\_ERROR\_TEMPERATURE bitmaps are computed from TB.COMPLETE\_STATUS per **fault matrix** sheet **FaultsToAlarms**, col. **xd\_error\_** and **Dest**. If any bit is set in any of XD\_ERROR, “Other” bit in BLOCK\_ERR is set
2. BLOCK\_ERROR bits are set based on COMPLETE\_STATUS per **fault matrix** sheet **FaultsToAlarms**, col. **Block Error**.
3. BLOCK\_ERROR bits are set based on configuration of each alert in TB.ALERT\_ACTION. That may be NO\_ERROR, MAINTENANCE\_NOW, MAINTENANCE\_SOON, or FAULT\_STATE\_SET
4. BLOCK\_ERROR bits are set based on TB.MODE.ACTUAL

Second, the general BLOCK\_ERR is apportioned between TB.BLOCK\_ERR and RB.BLOCK\_ERR per configuration of TB.ALERT\_ACTION.BLOCK\_ERR\_MAP (formerly known as TB.ALERT\_ACTION.MAPPED\_TO\_RB).

1. TB.ALERT\_ACTION. BLOCK\_ERR\_MAP =”Copy to RB.BLOCK\_ERR”  
   All of BLOCK\_ERR are written to TB.BLOCK\_ERR and the following bits are copied to RB.BLOCK\_ERR:
   1. BLK\_ERR\_DEV\_NEEDS\_MAINT\_SOON
   2. BLK\_ERR\_DEV\_NEEDS\_MAINT\_NOW
   3. BLK\_ERR\_DEV\_FAULT\_STATE\_SET
   4. BLK\_ERR\_LOST\_NV\_DATA
   5. BLK\_ERR\_OUTPUT\_FAILURE
   6. BLK\_ERR\_OTHER
   7. BLK\_ERR\_POWER\_UP
2. TB.ALERT\_ACTION. BLOCK\_ERR\_MAP = “TB.BLOCK\_ERR Only”  
   All of BLOCK\_ERR are written to TB.BLOCK\_ERR; RB.BLOCK\_ERR remains unaffected
3. TB.ALERT\_ACTION. BLOCK\_ERR\_MAP = “Report per FF-890”  
   The following bits are transferred to RB.BLOCK\_ERR:
   1. BLK\_ERR\_DEV\_NEEDS\_MAINT\_SOON
   2. BLK\_ERR\_DEV\_NEEDS\_MAINT\_NOW
   3. BLK\_ERR\_LOST\_NV\_DATA
   4. BLK\_ERR\_DEV\_FAULT\_STATE\_SET
   5. BLK\_ERR\_LOST\_STATIC\_DATA
   6. BLK\_ERR\_MEMORY\_FAILURE
   7. BLK\_ERR\_POWER\_UP

The following bits are transferred to TB.BLOCK\_ERR:

1. BLK\_ERR\_OTHER
2. BLK\_ERR\_BLOCK\_CONFIG\_ERR
3. BLK\_ERR\_LOCAL\_OVERRIDE
4. BLK\_ERR\_INPUT\_FAILURE
5. BLK\_ERR\_OUTPUT\_FAILURE
6. BLK\_ERR\_READBACK\_CHECK\_FAILED
7. BLK\_ERR\_OUT\_OF\_SERVICE

Of these options TB.ALERT\_ACTION. BLOCK\_ERR\_MAP, only the last one complies with **FF 890 4.4.3.6**. The other options are included for interoperability with some control systems, some quite old but still in use.

## Mapping to RB.FD\_SIMULATE.DIAGNOSTIC\_VALUE

First, “Transducer block doesn’t work properly” is set if TB.MODE.TARGET is not followed by TB.MODE.ACTUAL (e.g. TB is in LO)

Second, if general BLOCK\_ERR bit BLK\_ERR\_READBACK\_CHECK\_FAILED (inter-processor link failed), “Internal communications failed” bit is set

Third, the bits are transferred to RB.FD\_SIMULATE.DIAGNOSTIC\_VALUE per COMPLETE\_STATUS and the mapping in the **fault matrix**, sheet **FaultsToAlarms**, col. **FD Alarm**.

Fourth, TB.ALERT\_STATE is mapped to RB.FD\_SIMULATE.DIAGNOSTIC\_VALUE bits “many to one” as follows:

1. FDCOND\_DEVIATION\_ALERT if TB.deviation\_alert
2. FDCOND\_POSITION\_LARGE\_EXCURSION if
   1. position\_hihi\_alert
   2. position\_lolo\_alert
3. FDCOND\_POSITION\_EXCURSION if
   1. position\_hi\_alert
   2. position\_lo\_alert
4. FDCOND\_TRAVEL\_A\_ALERT if
   1. travel\_accumulation\_a\_alert
   2. cycle\_counter\_a\_alert
5. FDCOND\_TRAVEL\_B\_ALERT if
   1. travel\_accumulation\_b\_alert
   2. cycle\_counter\_b\_alert
6. FDCOND\_SUPPLY\_PRESSURE\_EXCURSION if
   1. supply\_pressure\_hi\_alert
   2. supply\_pressure\_lo\_alert
7. FDCOND\_SUPPLY\_PRESSURE\_LOLO\_ALERT if supply\_pressure\_lolo\_alert
8. FDCOND\_TEMPERATURE\_EXCURSION if
   1. temperature\_hi\_alert
   2. temperature\_lo\_alert
9. FDCOND\_SETPOINT\_TIMEOUT\_ALERT if set\_point\_timeout\_alert
10. FDCOND\_IP\_DRIVE\_CURRENT\_EXCURSION if
    1. ip\_drive\_current\_alert\_hi
    2. ip\_drive\_current\_alert\_lo
11. FDCOND\_WORKING\_TIME\_ALERT if working\_time\_alert
12. FDCOND\_NEAR\_CLOSE\_ALERT if near\_close\_alert
13. FDCOND\_ENVIRONMENT if
    1. sensor\_failure\_alert
    2. commissioning\_alert
14. FDCOND\_HARDWARE if
    1. processor\_alert
    2. supporting\_hardware\_alert
15. FDCOND\_VALVE\_CONTROL\_ALERT if
    1. valve\_control\_alert
    2. air\_supply\_alert

# Configuration suggestions

Here are some suggestions for the user on how to configure error reporting to avoid duplication.

Note that to get gory details of abnormal condition, the user needs to inspect TB.COMPLETE\_STATUS and TB.ALERT\_STATE. (There are a few bits in RB.BLOCK\_ERR outside of the scope of this document.)

So, the question is, how to get notifications of abnormal conditions using FF facilities. There two methods available: block alarm and field diagnostics. A DCS may support one of them, or both, in which latter case it is the matter of user preference.

## Field Diagnostics

Field diagnostics defines four “channels” of reporting, x=FAIL, OFFSEPC, MAINT, CHECK.

RB.FD\_<x>\_ALM is the object responsible for notifications.

It sources notifications from RB.FD\_<x>\_ACTIVE bitmap computed as RB.FD\_<x>\_MAP & RB.SIMULATE.FD\_DIAGNOSTIC\_VALUE & ~RB.FD\_<x>\_MASK.

In other words, a bit in RB.FD\_<x>\_ACTIVE is set if the corresponding bit in RB.SIMULATE.FD\_DIAGNOSTIC\_VALUE is set, is mapped to <x>, and is not masked out

## BLOCK\_ALM

BLOCK\_ALM object of RB, TB or FB is responsible for notifications of bits set in BLOCK\_ERR of that block. We are focusing on TB.

## Suggestions

### TB Alerts notifications

If the user wants to see TB alert N reported via field diagnostics, they need to find the corresponding bit in RB.SIMULATE.FD\_DIAGNOSTIC\_VALUE, map it to a desired bit in one (or even more) of RB.FD\_<x>\_MAP and ensure that the corresponding bit in RB.FD\_<x>\_MASK is not set.

In this case, the user probably doesn’t want that alert N be reported in BLOCK\_ERR (whether RB or TB). This is actually [required](#_Constrains) by FF. To do so, configure TB.ALERT\_ACTION.<N>\_ACT = “Not reported”.

Conversely, if the user wants TB alert N reported via BLOCK\_ALM, they need to configure TB.ALERT\_ACTION.<N>\_ACT to their preference, excluding “Not reported”. This will cause the alert to be reported via RB or TB block alarm, depending on configuration of TB.ALERT\_ACTION.BLOCK\_ERR\_MAP.

In this case, the user probably doesn’t want that alert N be reported in BLOCK\_ERR (whether RB or TB). This is actually [required](#_Constrains) by FF. To do so, set the corresponding bit in all RB.FD\_<x>\_MASK and/or clear the corresponding bit in all RB.FD\_<x>\_MAP.

### COMPLETE\_STATUS notifications

Oops! Working on it.