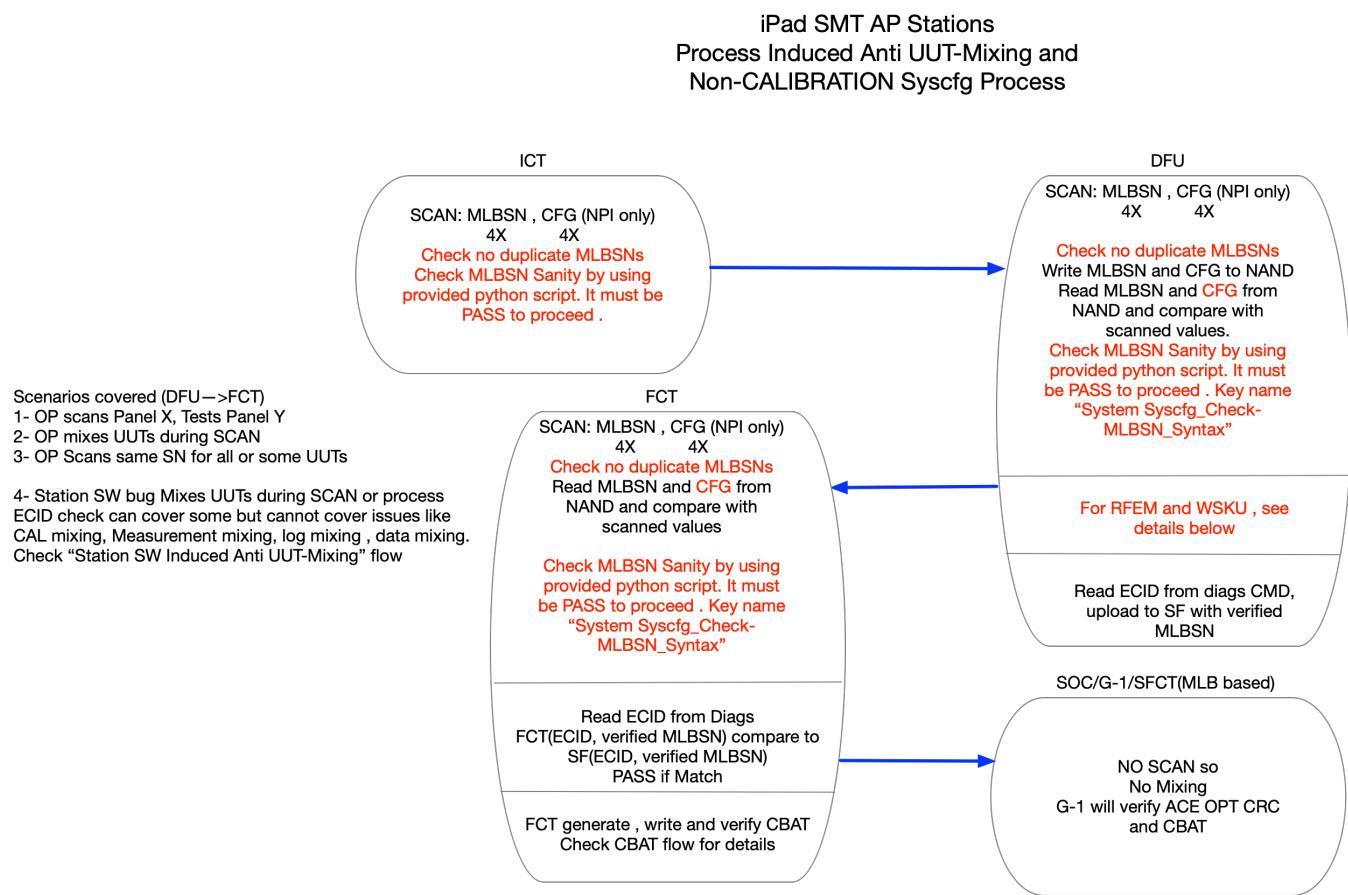


iPad SMT CALs, FWDL, OTP

DRI: Saqib Yaqub Tony Tan

Background

Document the SMT CAL, FWDL or OTP programming and Verify general flows here.



For MLBSN Syntax checker Python app , please contact your Apple DRI , if you cant download it here : [SN_Check_v1.1_overlay.zip](#)

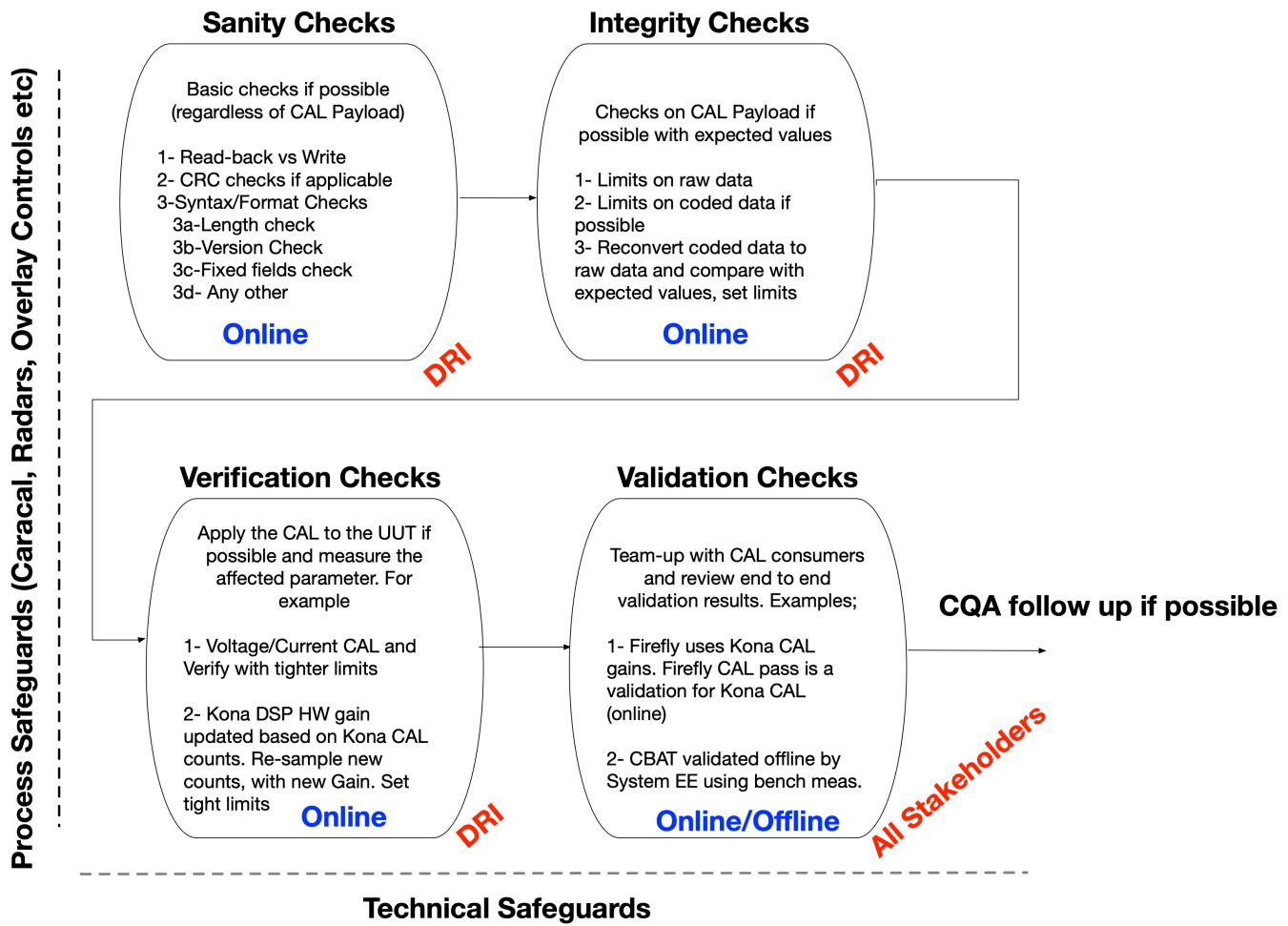
This is an optional step as most vendors might already have this MLBSN check implemented after scan. In the case, verify that the check works by injecting a fake SN. If it works, then using Python script and reporting the

System_Syscfg_Check_MLBSN_Syntax is optional

For station SW induced Anti UUT-Mixing/Anti CAL-Mixing flows, check at end of this page. Also can download here: [Station SW Induced Anti UUT-Mixing V01.graffle](#)

When loading logs or any validation Info on this radar, or any other syscfg verify radar, make sure to mention station SW version and Build stage. If more than vendor, upload for both.

CAL Syscfg / CAL Blob possible checks
Could be deployed as suitable



When loading logs or any validation Info on this radar, or any other syscfg verify radar, make sure to mention station SW version and Build stage. If more than vendor, upload for both.

After DVT , at each overlay change, please do a self -audit on RFEM and WSKU (see if they didn't change from last good version) by station DRI himself and let Apple DRI know.

Do the same self check audit on each MP overlay release/update for each vendor

Add RFEM and WSKU Self Audit to Ops handover Checklist

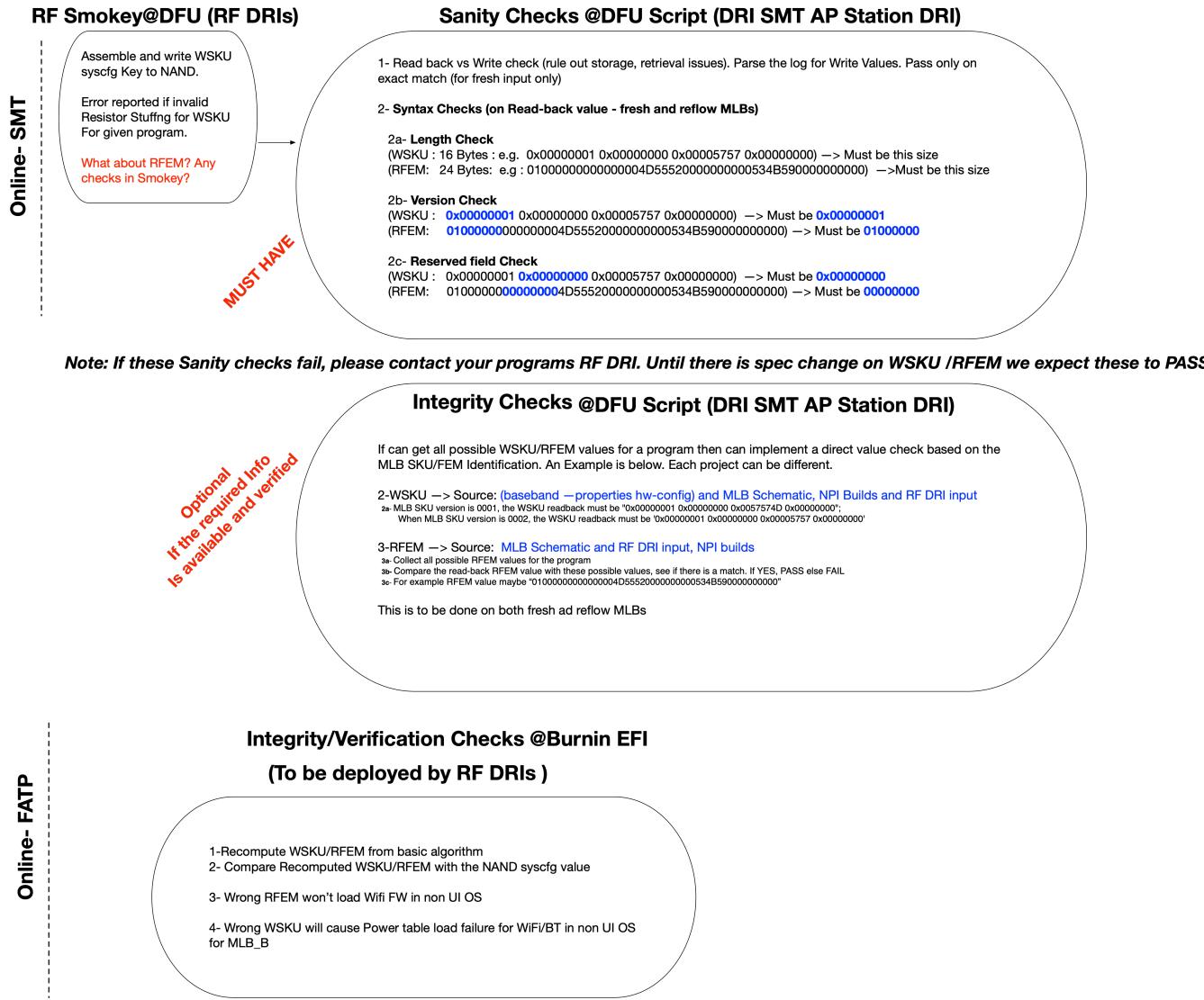
Upload WSKU and RFEM readback values to Insight as attribute (no need add to SF)

Each of the sanity checks must be shown as PASS/FAIL items on GUI . Some proposed names are like this:

System Syscfg_check-RFEM_sanity_check_Length , System Syscfg_check-RFEM_sanity_check_version, System Syscfg_check-RFEM_sanity_check_readback_compare,

System Syscfg_check-RFEM_sanity_check_expected_value

WSKU and RFEM Syscfg Keys Validation flow



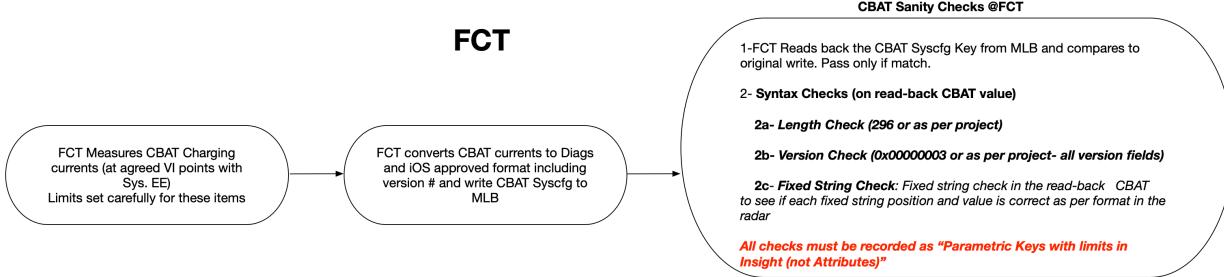
FCT

1- CBAT

Based on CBAT_CAL_Flow_V12.graffle

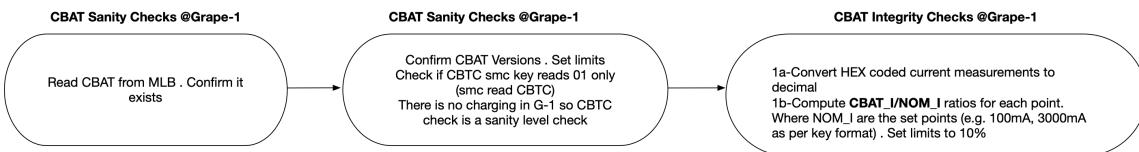
CBAT SYSCFG (USB CURRENT PULL BACK) FLOW

Online (SMT)



Grape-1 (iPad Pro, Air , Mini only)

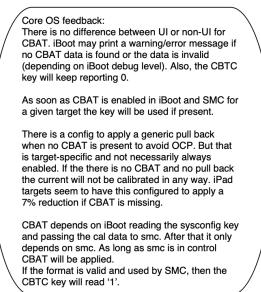
All checks must be recorded as "Parametric Keys with limits in Insight (not Attributes)"



Entry iPad doesnt have Grape-1

If there is no Grape-1 then the G-1 part of CBAT Integrity checks to be added on FCT. The CBTC related check maybe added on QT

Online (FATP)



QT/CT

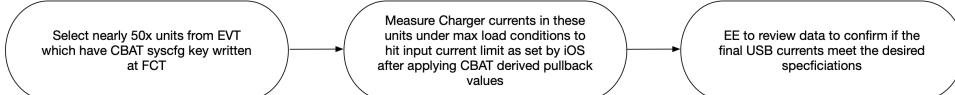
Burnin (TBD - TDL)

iCheck (TBD - TDL)

FATP checks can be added around EVT time frame when iBoot, SMC and CBAT are all finalized

Offline

Sys EE offline Validation



Create these Radars for CBAT validation per project:

1- If any change to CBAT format Or version is needed (i.e your project needs CBAT which is not used in any other existing project), Apple DRI to ask TDL to start another formal syscfg review process/radar as well. Once this new CBAT version is approved, proceed to below steps. Note that current CBAT version in iPad is 03 (rdar://51690861 (CBAT syscfg key) and is quite flexible, so changing IV points doesn't mean the version needs update. Confirm with Apple DRI if you think your new CBAT requirement can fit into V03 and a syscfg review process is needed.

2- Assume that you can use CBAT from some existing project (either exact same IV points or different, but still can fit in V03), for each new project, create a radar and assign to System EE Power DRI to confirm CBAT IV points. e.g. rdar://72031293 (BANE CBAT Format and IV Points) In this radar , you can refer/link to the radars/project whose CBAT you are using for current project e.g rdar://51690861 Power EE must sign off and approve this radar before EVT build

3- CBAT Conversion Check (Excel spreadsheet) e.g same as rdar://68253867 - assign to System Power EE

4- CBAT OS Format Check , assigned to Core OS DRI- add FCT and non UI CBAT reads - e.g rdar://68453483 - This maybe same radar as in (2)- After Power System EE approved, can be assigned to Core OS DRI approve. This can also contain actual non UI CBAT reads for CoreOS DRI to confirm at later stage (so confirm format first by System Power EE and Core OS then later on confirm actual non UI reads by Core OS).

5- CBAT System Validation (Ask System Power EE to create if they didn't, they need validate CBAT on bench by EVT) e.g rdar://68499357

6- CBAT Check to be added to QT/CT eg. rdar://72874291

Link ALL above radars to each other in Radar APP

Below Two are not POR yet, so dont create for now.

7- CBTC Key in Burnin (Apple TDL to create one) like : rdar://71999524 (J5xx) Request to check CBTC key in Burnin as part of POR coverage)
8-CBTC check radar (Apple TDL to create one) (rdar://72030596 (J5xx, CBAT, CBTC=1 Check in iCheck)

NOTE: When loading Logs/Excel, Make sure FCT/G-1 Script version and Build stage is clearly mentioned
If there is more than one vendor, need upload for both vendors

After DVT, at each overlay change, please do an Excel Self-audit yourself to see if CBAT not broken on all vendors
Do the same self check audit on each MP overlay release/update for each vendor

Add CBAT Excel Audit to Ops handover Checklist

Each of the sanity/Integrity checks must be shown as PASS/FAIL items on GUI . Some proposed names are like this:

Battcharger Syscfg_check-CBAT_sanity_check_version, Battcharger Syscfg_check-CBAT_sanity_check_readback_compare, Battcharger Syscfg_check-CBAT_sanity_fixed_string_check,
Battcharger Syscfg_check-CBAT_CBTC
Battcharger Syscfg_check-CBAT_integrity_check_5V1A9V3A

Note: A potential Verification method of CBAT on FCT is outlined in this document. However, due to cycle time concerns on FCT and System EE thinking this is not must to add as long as FCT stores CBAT with required accuracy and direct syntax. <rdar://68253867> (QN/QF,P2, FCT CBAT audit tracking)

[CBAT_Verification_flow_V00.pdf](#) (The recommended procedure in this doc. or FCT CBAT verification is not tested and may change)

CB Erase behavior of Current iPad Smt Syscfg keys

Ref: iPad_SMT_AP_syscfg_V00

Sysconfig Name	Data Format	Max Size Bytes	Calibration ?	Customer Facing Key?	Write Station Names	Write Station CB Addresses	CB Erase Station Names	CB Erase Station Addresses	iOS Customer	Diags Spec Radar	Source of Data	Upload PDCA
CBAT	Numeric	296	Yes	Yes	FCT	1	FCT	1	?	rdar://51690861 (CBAT syscfg key updates for J417/J420 projects)	FCT Meas.	?
CFG#	String	64	No	No	DFU	0	(NONE)	NONE	?		SCAN	YES
MLB#	String	17	No	No	DFU	0	(ALL)	ALL (RANGE?)	?		SCAN	YES
RFEM	Numeric	24	No	TBD	DFU	0	(ALL)	ALL (RANGE?)	?	rdar://problems/32854520 rdar://51752328 ([J4xx]) Request for a New syscfg size for RFEM)	DFU Log	?
WSKU	Numeric	16	No		DFU	0	DFU	0	?	rdar://36193866 (New SysCfg key request: WSKU for WiFi SKU)	DFU log	?

2- ACE/Minion FWDL. (Minion FWDL was used only on J21x in 2018 - Specific flow for Minion FWDL is here : [FCT_Minion_FlowChat.pdf](#))

ACE FWDL is done using Xavier SPI programmer and provided binary file

Copy
ACE/Minion MP-Binary file to Overlay
Same file loaded to Xavier memory offline
Make the file in Xavier READ ONLY

START FCT TEST

Optional

Compute SHA1 and MD5 Checksums on MP-Binary in Mac mini
Compare with Hard-Coded Checksums

Read Back Binary From Xavier Memory. Compute SHA1 and MD5
Compare with Hard-Coded Checksums

START ACE/Minion FLASH PROGRAMMING and VERIFY
Use the file in Xavier memory to do FWDL and VERIFY

Read Back Binary From ACE FLASH. Compute SHA1 and MD5
Compare with Hard-Coded Checksums
Save Read back bin file in local logs and to Insight

ACE 2 OTP will be programmed via binary file provided by ACE FW team. A reference ACE2 OTP programming flow is here (from PG/PF MP) : [ACE2_OTP_PROD_FUSED\(MP\)_V001.pdf](#)

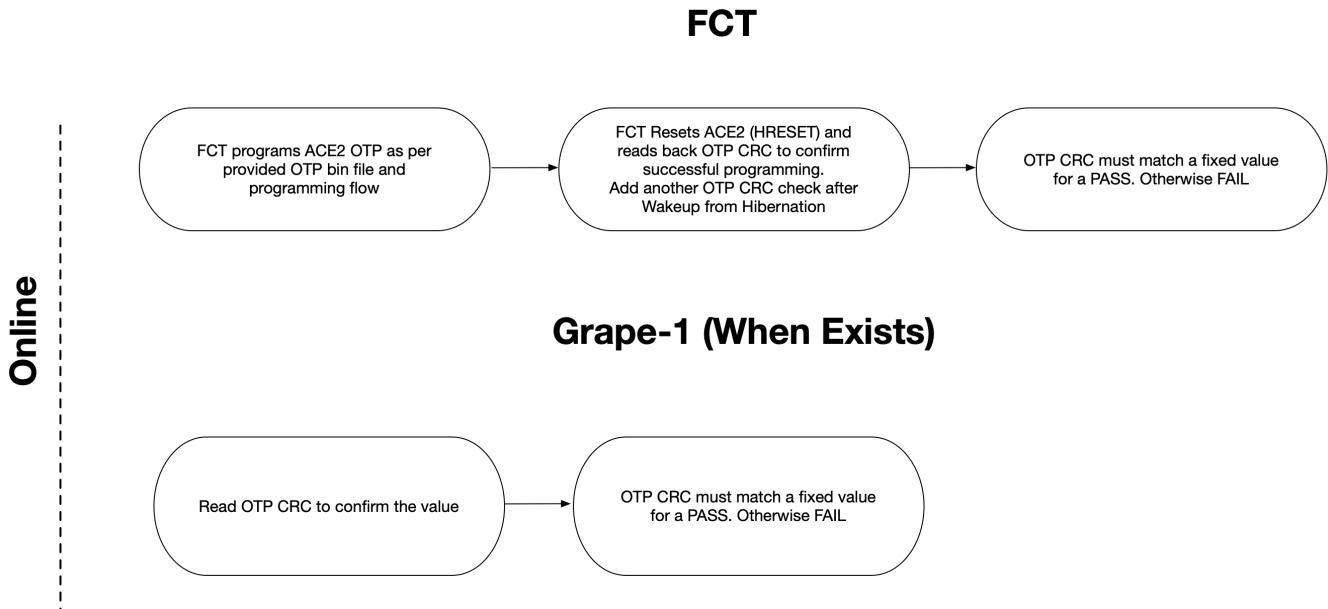
There is no formal validation needed for functionality/features of ACE2 OTP and if the CRC matches the required value, then its sufficient to conclude that OTP was programmed correctly and will do what it is intended for

<[rdar://problem/58131152](#)> PG/PF, Product Fused ACE2 OTP Validation

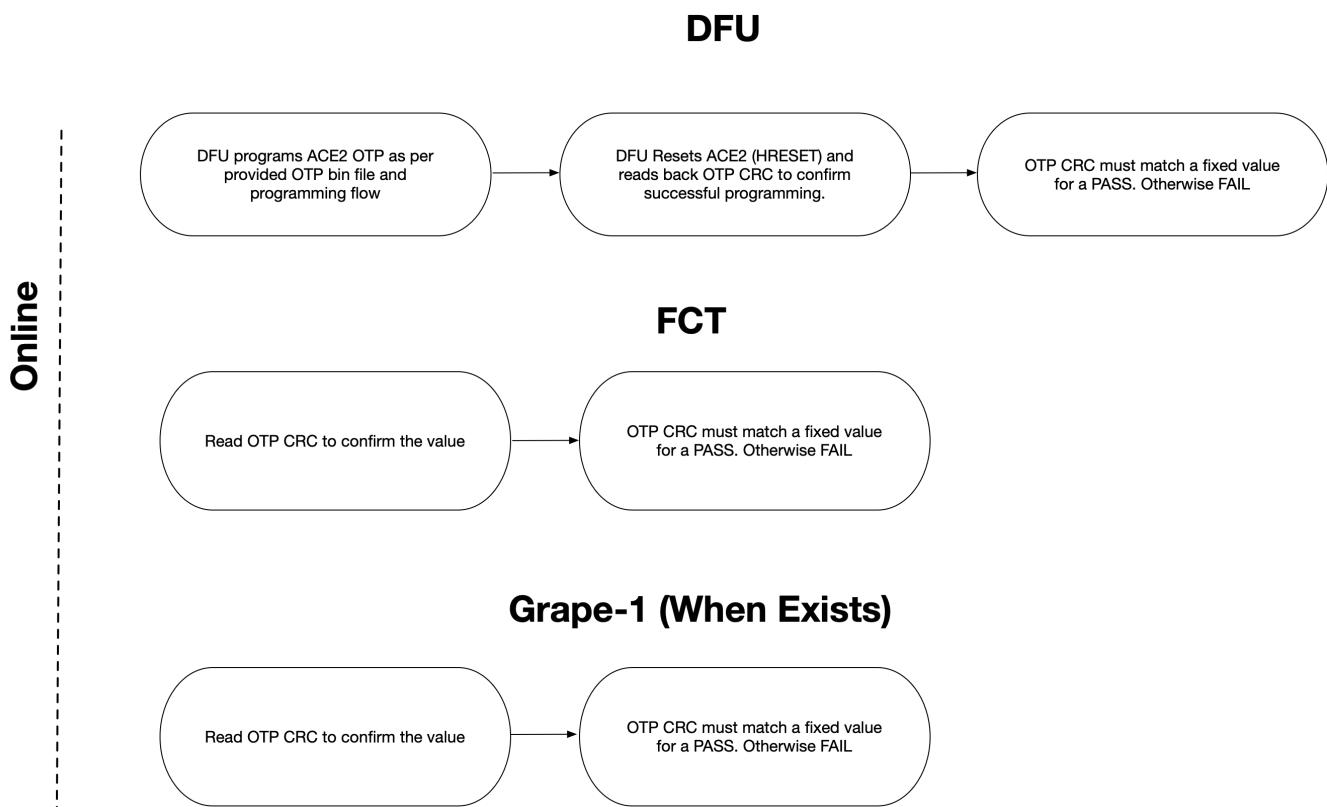
So for test line perspective, we add ACE2 OTP CRC checks after programming at the station and sanity checks on downstream stations as applicable

Below flow is based on : ACE2 OTP Verify_V03

ACE2 OTP Program and Verify (ACE FWDL and OTP @FCT)



ACE2 OTP Program and Verify (ACE FWDL and OTP @DFU)

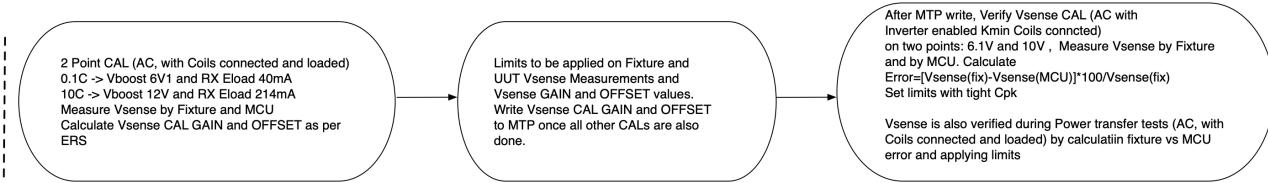


Scorpius FCT

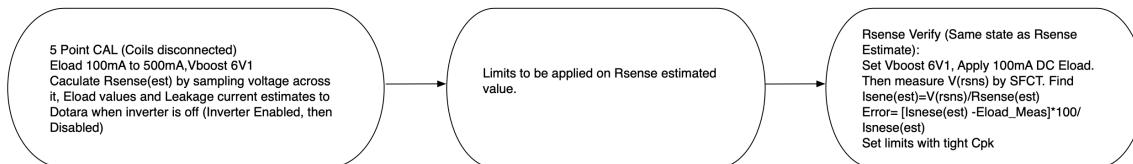
Source: SFCT_UUT_CALS_V06

SFCT UUT CAL AND VERIFY Flows (Dotara Based Platforms)

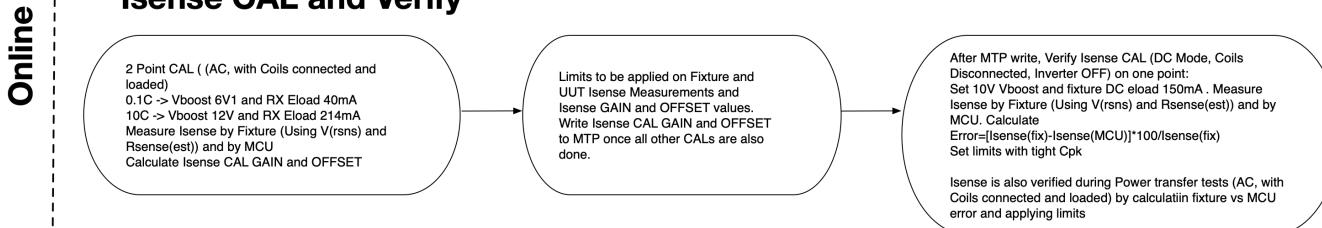
Vsense CAL and Verify



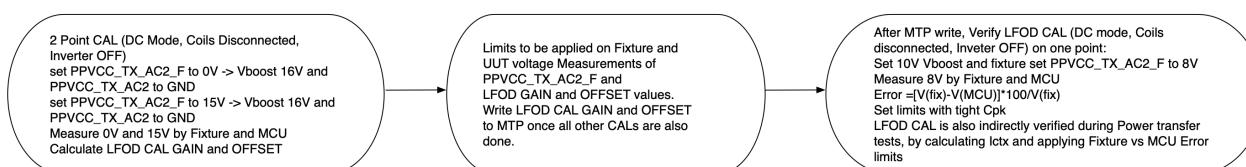
Rsense Estimate and Verify



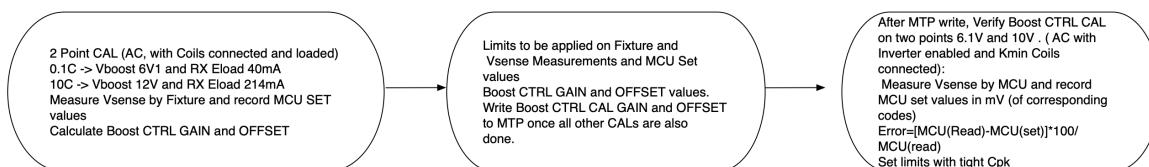
Isense CAL and Verify



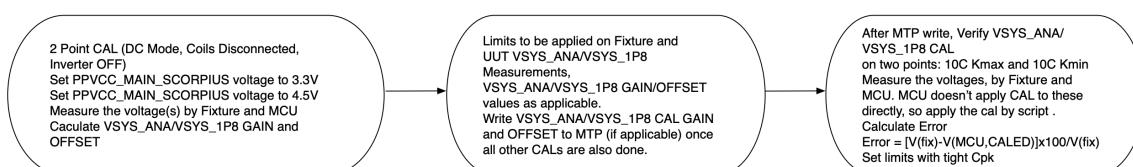
LFOD CAL and Verify



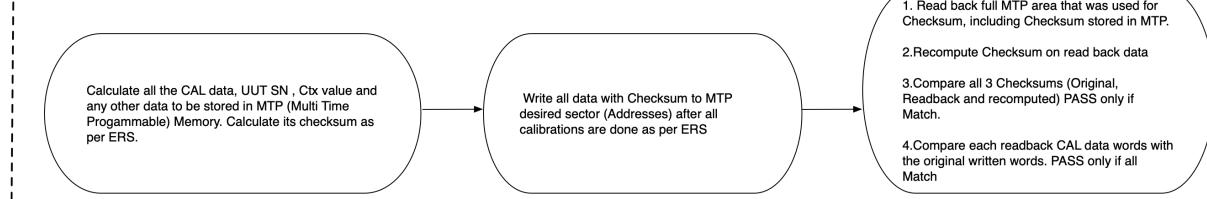
Boost CTRL CAL and Verify



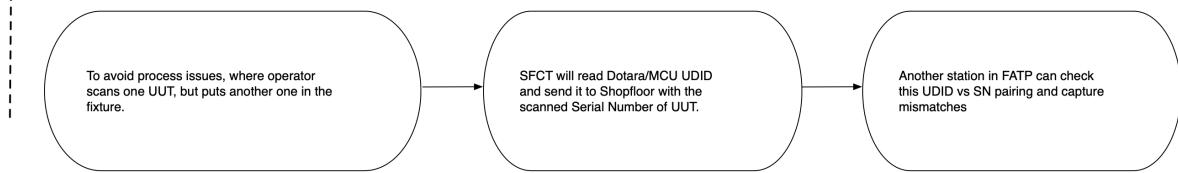
Vsys_ANA & VSYS_1P8 CTRL CAL and Verify



MTP write and checksum



Wireless UDID and UU SN Linking



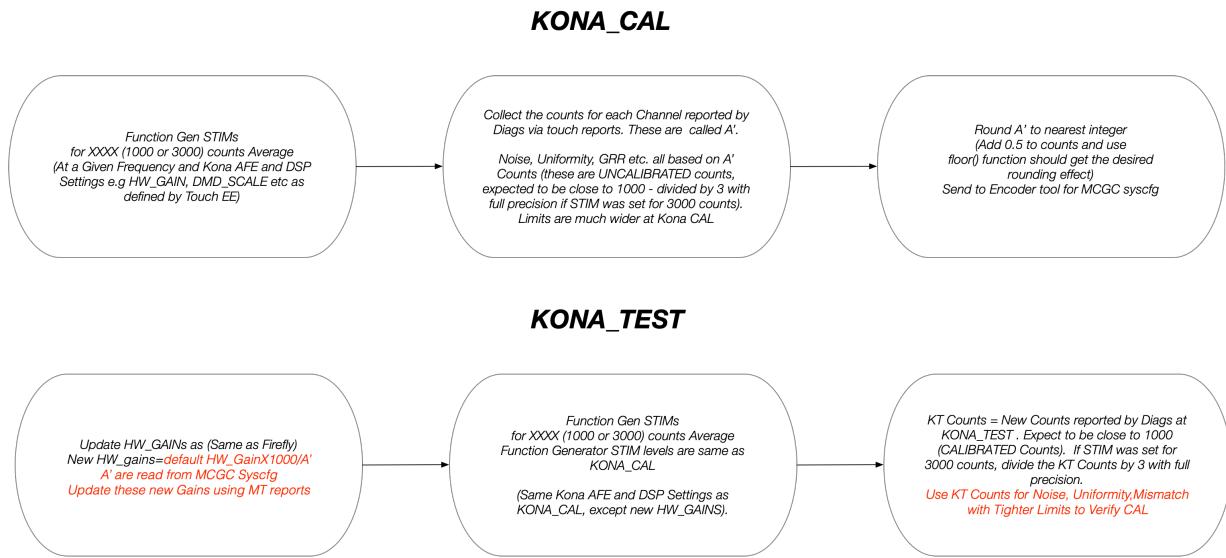
If we have more than 1 vendor, please do a cross check for MTP CAL and Verify steps. e.g , CAL on one vendor SFCT , so that it stores Checksum in MTP.

While the MTP Checksum Verify can be done on another vendor SFCT

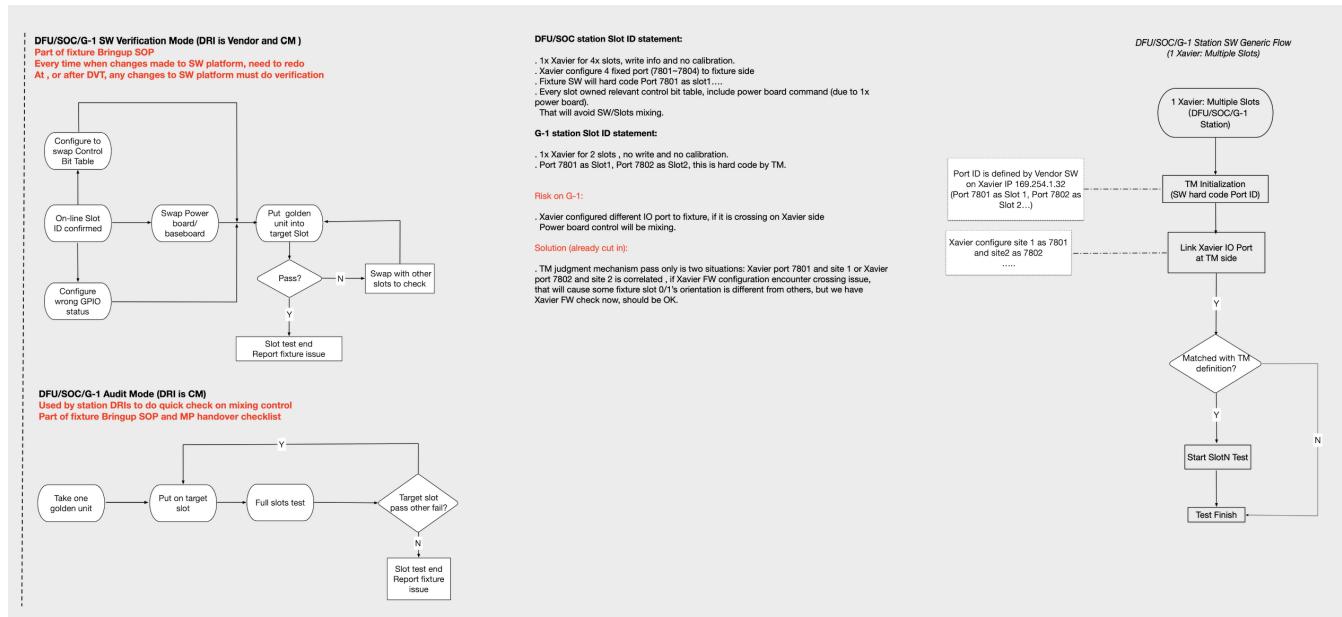
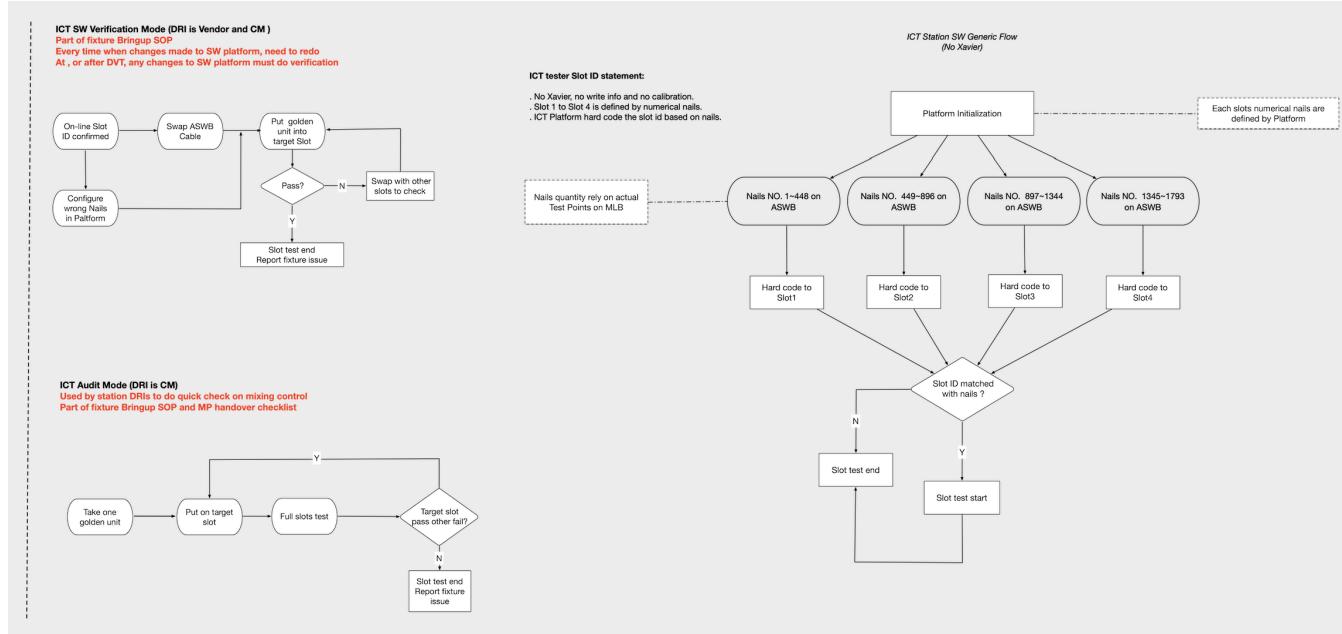
- 1- Vendor 1 does CAL , calculates Checksum and stores CAL and Checksum to MTP
- 2- vendor two uses UUTs, read the CAL data and Checksum from MTP sector 127 Recalculate the Checksum using the SAME function used on mainline script Match Checksum from TP vs New computed Checksum
- 3- Repeat for 4 UUTs
- 4- Detailed Checksum Algo (source QN/QF ERS is shown at end of page - each project maybe different)

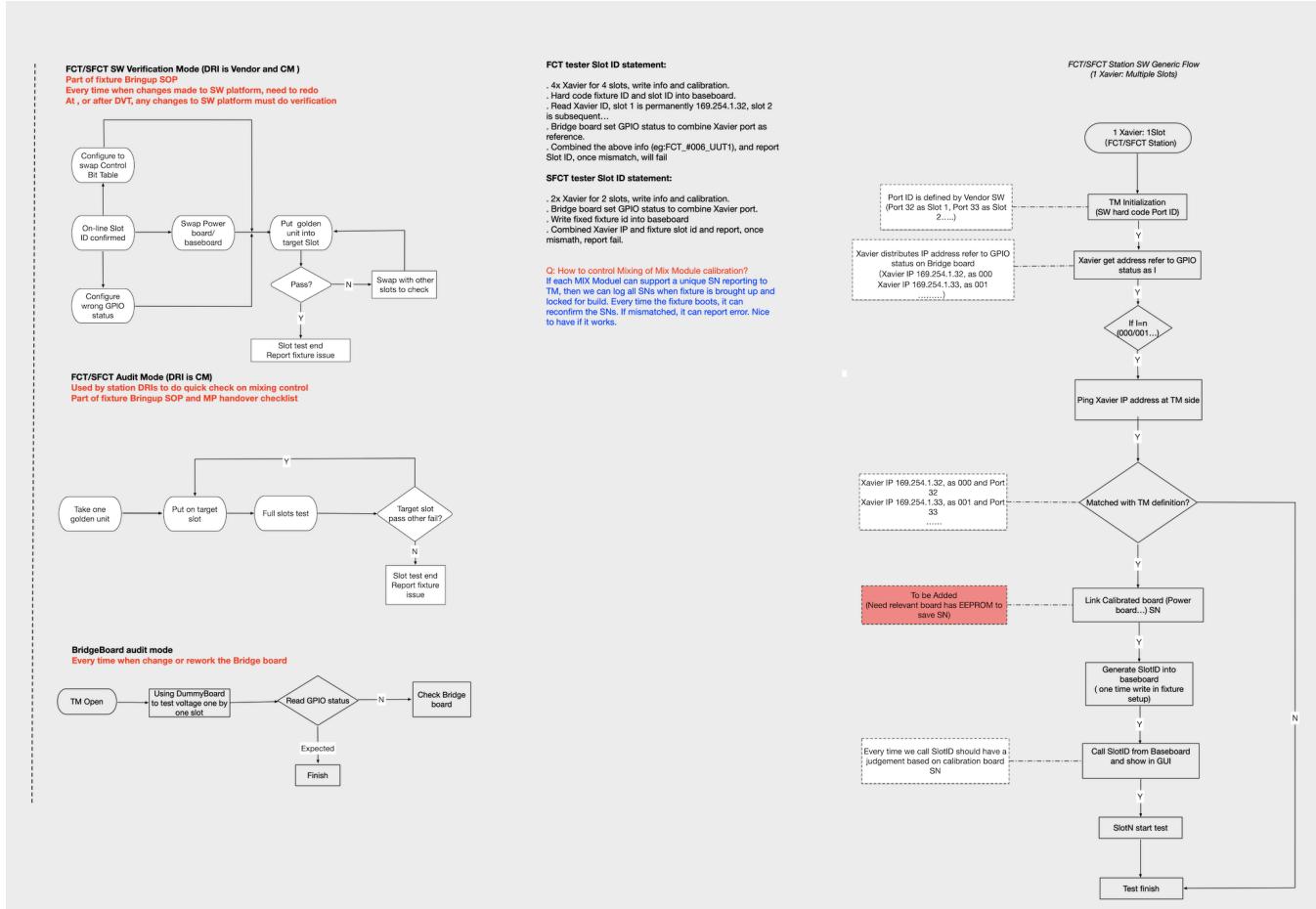
KONA-CAL

Kona-CAL is no longer used on ipad SMT line. Just putting here for sake of documentation



Both KONA_CAL and KONA_TEST can be same fixture with test running two times i.e. CAL and then Verify However, we used to run KONA_TEST sequence on a separate fixture after KONA_CAL to catch any fixture HW related issues that can cause a poorly CAL'ed channel to be masked during Test/verify sequence.





QN/QF SFCT Checksum flow (Ref : Checksum_process_V4.pdf) : [Checksum_process_V4.pdf](#)

