Information note

N° 10290AERRA

Dear customer,

With this Infineon Technologies AG information note, we would like to inform you about the following

PRO-SIL AURIX SafeTlib Release Notes Addendum V25.0 affecting products TC29x, TC27x, TC26x, TC23x, TC22x, TC21x

On 16 April 2020, Infineon acquired Cypress.
We are now in the process of merging and consolidating our tools and processes for PCN, Information Notes, Errata and Product Discontinuance.
For further details, please visit our website:
https://www.infineon.com/cms/en/about-infineon/company/cypress-acquisition/

Infineon Technologies AG
Postal Address Headquarters: Am Campeon 1-15, D-85579 Neubiberg, Phone +49 (0)89 234-0
Chairman of the Supervisory Board: Dr. Wolfgang Eder
Management Board: Dr. Reinhard Ploss (CEO), Dr. Helmut Gassel, Jochen Hanebeck, Constanze Hufenbecher, Dr. Sven Schneider
Registered Office: Neubiberg
Commercial Register: München HRB 126492

Information note

N° 10290AERRA

▶ Products affected

Please refer to attached affected product list 1_cip10290

▶ Detailed change information

Subject

PRO-SIL AURIX SafeTlib Release Notes Addendum V25.0 affecting products TC29x, TC27x, TC26x, TC23x, TC22x, TC21x

Reason

Update of the Release Notes Addendum due to new known issues

Description

Old	New
■ Release Notes Addendum Version 24.0	■ Release Notes Addendum Version 25.0

▶ Product identification

Not applicable (no change of product)

▶ Impact of change

Assessment in Application required !

▶ Attachments

1_cip10290 affected product list
3_cip10290 Release Notes Addendum V25.0

▶ Intended start of delivery

Not applicable

If you have any questions, please do not hesitate to contact your local sales office.

PRO-SIL™ AURIX™ SafeTlib

Release Notes Addendum

Version : 25.0
Date : 2021-10-29

About this document

Scope and purpose

This release notes addendum documents known issues in SafeTlib releases and the changes planned in the future releases. Only the issues in production release (PR) or maintenance release (MR) quality modules are listed.

This document also provides information on:

- Mapping between SafeTlib software test and Safety mechanisms
- Microcontroller Test Library test details

Attention:

- The customer shall be aware that the CIC61508 watchdog was not developed as per ISO26262 objectives and hence, the customer shall evaluate CIC61508 before usage in a safety system.
- The customer shall refer to the TLF35584 safety case and safety manual before usage in a safety system.
- The user manual released with Rel531 or newer maintenance releases has to be referred for the following releases: Rel513, Rel513.1, Rel514, Rel514.1 and Rel652. The differences are documented in section 4 Release-specific changes in the User Manual.
- Software Based Self-Test (SBST) is delivered as part of the SafeTlib release but not evaluated within the SafeTlib Safety Case. A separate Safety Case Report is planned for SBST.
- In Rel532 release, a single release note is delivered for TC27x-CA, TC27x-DB and TC27x-DC steps. Customer shall refer the Release notes section 2.1.1 package contents based on the hardware device steps selected for usage. Similarly for TC29x-BB and TC29x-BC steps single Release notes is delivered. Customer shall refer the Release notes section 2.1.1 package contents based on the hardware device steps selected for usage.

Reference documents

This document must be read in conjunction with the following documents:

- TC22x_TC21x_AB_Errata_Sheet_v1_7, 2020-11, Infineon Technologies AG Munich
- TC22x_TC21x_AC_Errata_Sheet_v1_4, 2020-11, Infineon Technologies AG Munich
- TC23x AB Errata sheet, V1.8, 2020-11, Infineon Technologies AG Munich
- TC23x AC Errata sheet, V1.5, 2020-11, Infineon Technologies AG Munich
- TC26x BB Errata sheet, V1.7, 2020-11, Infineon Technologies AG Munich
- TC26x BC Errata sheet, V1.4, 2020-11, Infineon Technologies AG Munich
- TC27x CA Errata sheet, V1.9, 2020-11, Infineon Technologies AG Munich
- TC27x DB Errata sheet, V1.8, 2020-11, Infineon Technologies AG Munich
- TC27x BC Errata sheet, V1.10, 2020-11, Infineon Technologies AG Munich
- TC27x DC Errata sheet, V1.4, 2020-11, Infineon Technologies AG Munich
- TC29x BB Errata sheet, V1.8, 2020-11, Infineon Technologies AG Munich
- TC29x BC Errata sheet, V1.4, 2020-11, Infineon Technologies AG Munich

Intended audience

This document is intended for anyone who is using the SafeTlib software.

Scope of Release Addendum

This release notes addendum documents known issues in following AURIX SafeTlib releases.

#	Device	Name	Date	Quality	Release
1.	TC27x BC TC27x CA	Base_PR11N	2014-12-18	PR	559.1 558.1
2.	TC27x BC TC27x CA	Base_PR11	2015-01-15	PR	557 554
3.	TC297TF TC29x	TC29x_PR2_PR2.1	2015-02-16	PR	501 502
4.	TC27x CA	TC27x_PR5	2015-07-03	PR	594
5.	TC27x CA TC27x DB	TC27x_PR5.2	2015-08-28	PR	592
6.	TC27x CA TC27x DB	TC27x_PR7	2015-08-31	PR	513
7.	TC27x DB TC27x CA TC27x BC	TC27x_PR2.3	2015-09-21	PR	503.1
8.	TC27x CA TC27x DB	TC27x_PR7.1	2015-11-26	PR	513.1
9.	TC29x BB	TC29x_PR7.2	2015-12-22	PR	514
10.	TC26x BB	TC26x_PR7.3	2015-12-22	PR	514.1
11.	TC23x AB TC22x AB TC21x AB	TC23x_TC22x_TC21x_PR5	2016-01-20	PR	652
12.	TC27x CA TC27x DB TC29xBB TC26x BB	TC27x_TC29x_TC26x_MR1	2016-05-27	MR	531
13.	TC23x AB TC22x AB TC21x AB	TC23x_TC22x_TC21x_MR1	2016-07-11	MR	631
14.	TC27x CA TC27x DB TC27xDC TC29xBB TC29xBC TC26x BB	TC27x_TC29x_TC26x_MR2	2016-08-31	MR	532
15.	TC23x AB TC23x AC TC22x AB TC21x AB	TC23x_TC22x_TC21x_MR2	2016-10-17	MR	632

Scope of Release Addendum

#	Device	Name	Date	Quality	Release
16.	TC27x CA TC27x DB TC27x DC TC29x BB TC29x BC TC26x BB TC26x BC	TC27x_TC29x_TC26x_MR3	2017-01-31	MR	533
17.	TC23x AB TC23x AC TC22x AB TC21x AB	TC23x_TC22x_TC21x_MR3	2017-05-26	MR	633
18.	TC23x AB TC23x AC TC22x AB TC22xAC TC21x AB TC21xAC	TC23x_TC22x_TC21x_MR4	2017-09-28	MR	634
19.	TC27x CA TC27x DB TC27x DC TC29x BB TC29x BC TC26x BB TC26x BC	TC27x_TC29x_TC26x_MR4	2017-10-31	MR	534
20.	TC27x CA TC27x DB TC27x DC TC29x BB TC29x BC TC26x BB TC26x BC	TC27x_TC29x_TC26x_MR5	2018-05-14	MR	535
21.	TC23x AB TC23x AC TC22x AB TC22xAC TC21x AB TC21xAC	TC23x_TC22x_TC21x_MR5	2018-07-31	MR	635
22.	TC27x CA TC27x DB TC27x DC	TC27x_MR3.1	2019-02-28	MR	533.1
23.	TC27x CA TC27x DB TC27x DC	TC27x_MR5.1	2019-02-28	MR	535.1

Scope of Release Addendum

#	Device	Name	Date	Quality	Release
24.	TC27x CA TC27x DB TC27x DC TC29x BB TC29x BC TC26x BB TC26x BC	TC27x_TC29x_TC26x_MR6	2019-06-25	MR	544
25.	TC23x AB TC23x AC TC22x AB TC22xAC TC21x AB TC21xAC	TC23x_TC22x_TC21x_MR6	2019-10-22	MR	636
26.	TC27x CA TC27x DB TC27x DC TC29x BB TC29x BC TC26x BB TC26x BC	TC27x_TC29x_TC26x_MR7	2020-07-17	MR	545
27.	TC23x AB TC23x AC TC22x AB TC22xAC TC21x AB TC21xAC	TC23x_TC22x_TC21x_MR7	2021-06-11	MR	637
28.	TC27x CA TC27x DB TC27x DC TC29x BB TC29x BC TC26x BB TC26x BC	TC27x_TC29x_TC26x_MR8	2021-06-11	MR	546

Table of Contents

1 Known issues overview

- = Not applicable, ● = Applicable

Note: Release number mentioned in the right most columns is the latest release.

Module	ID	558.1/559.1	557/554	501/502	594	592	513	503.1	513.1	514	514.1	652	531	631	532	632	533	633	634	534	535	635	533.1	535.1	544	636	545	637	546
Generic	0000050328-637	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Generic	0000050329-9	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Generic	0000050328-126	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Generic	0000050328-138	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Generic	0000050328-139	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Generic	0000050328-152	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Generic	0000050329-341	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Generic	0000050328-341	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Generic	0000050328-425	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Generic	Linker options	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Generic	0000050328-531	●	●	●	●	●	●	●	-	-	-	-			-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Generic	0000050328-567	●	●	●	●	-	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	●	-	-	-	-	-	-
Generic	0000050328-1080	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Generic	0000050328-1144	-	-	-	-	●	-	-	-	-	-	-	●	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Generic	0000050328-1193	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	●	-	-	-	-	-	-
Generic	0000050328-1196	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	●	-	-	-	-	-	-
Generic	0000050328-1213	-	-	-	-	-	-	-	-	-	-	-	●	●	●	●	●	-	-	-	-	-	●	-	-	-	-	-	-

Known issues overview

Module	ID	558.1/559.1	557/554	501/502	594	592	513	503.1	513.1	514	514.1	652	531	631	532	632	533	633	634	534	535	635	533.1	535.1	544	636	545	637	546
Generic	0000050328-1216	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	●	-	-	-	-	-	-
Generic	0000050328-1219	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	●	-	-	-	-	-	-
Generic	0000050328-1271	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	●	-	-	-	-	-	-
Generic	0000050328-1289	-	-	-	-	●	-	-	-	-	-	-	●	-	●	-	●	●	●	●	-	-	●	-	-	-	-	-	-
Generic	0000050328-1335	-	-	-	-	-	-	-	-	-	-	-	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-
Generic	0000050328-1344	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	●	●	-	●	-	-	-	-	-
Generic	0000050328-1362	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-
Generic	0000050328-1373	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-
Generic	0000050328-1409	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	●	-	-
Generic	0000050328-1428	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-
Clock Monitor Test	0000050328-427	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Clock Monitor Test	0000050328-495	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Clock Monitor Test	0000050328-520	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Clock Monitor Test	0000050328-1044	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Clock Monitor Test	0000050328-1333	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-
CPU BUS MPU Test	0000050328-1148	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-
CPU BUS MPU Test	0000050328-1234	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	●	-	-	-	-	-	-
CPU BUS MPU Test	0000050328-1284	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	●	-	-	-	-	-	-
CPU MPU Test	0000050328-525	-	-	-	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CPU MPU Test	0000050328-680	-	-	-	●	-	●	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Module	ID	558.1/559.1	557/554	501/502	594	592	513	503.1	513.1	514	514.1	652	531	631	532	632	533	633	634	534	535	635	533.1	535.1	544	636	545	637	546
CPU MPU Test	0000050328-1339	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	●	●	-	●	-	-	-	-	-
CPU Watchdog Test and Safety Watchdog Test	0000050329-387	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CPU Watchdog Test and Safety Watchdog Test	0000050328-526	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CPU Watchdog Test and Safety Watchdog Test	0000050328-1238	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	●	-	-	-	-	-	-
CPU Watchdog Test and Safety Watchdog Test	0000050328-1290	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	●	-	-	-	-	-	-
CPU Watchdog Test and Safety Watchdog Test	0000050328-1300	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	●	-	-	-	-	-	-
DMA Test	0000050328-523	-	-	-	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
DMA Test	0000050328-709	-	-	-	-	-	●	-	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
DMA Test	0000050328-846	-	-	-	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
DMA Test	0000050328-951	-	-	-	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Flexible CRC Engine (FCE) Test	0000050328-906	-	-	-	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Interrupt Router Test	0000050328-702	-	-	-	-	-	●	-	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
IOM Test	0000050328-498	-	-	-	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
IOM Test	0000050328-1073	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-
LMU Bus MPU Test	0000050328-527	-	-	-	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
LMU Register Access Protection Test	0000050328-528	-	-	-	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
LMU Bus MPU Test and LMU	0000050328-478	-	-	-	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Module	ID	558.1/559.1	557/554	501/502	594	592	513	503.1	513.1	514	514.1	652	531	631	532	632	533	633	634	534	535	635	533.1	535.1	544	636	545	637	546
Register Access Protection Test																													
Peripheral SRAM Test	0000050329-391	-	-	-	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Peripheral SRAM Test	0000050329-395	-	-	-	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Peripheral SRAM Test	0000050328-956	-	-	-	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PMU ECC and EDC Test	0000050336-1	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PMU ECC and EDC Test	0000050328-461	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PFLASH Monitor Test	0000050328-666	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PMU ECC and EDC Test / PFLASH Monitor Test	0000050328-210	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PMU ECC and EDC Test / PFLASH Monitor Test	0000050328-440	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PMU ECC and EDC Test / PFLASH Monitor Test	0000050328-692	-	-	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PMU ECC and EDC Test / PFLASH Monitor Test	0000050328-925	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PMU ECC and EDC Test / PFLASH Monitor Test	0000050328-983	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PMU ECC and EDC Test / PFLASH Monitor test	0000050329-494	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	●	-	-	-	-	-	-
PMU ECC and EDC Test / PFLASH Monitor Test	0000050328-1287	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-	●	-	-	-	-	-	-
PMU ECC and EDC Test / PFLASH Monitor Test	0000050328-1358	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-

Module	ID	558.1/559.1	557/554	501/502	594	592	513	503.1	513.1	514	514.1	652	531	631	532	632	533	633	634	534	535	635	533.1	535.1	544	636	545	637	546
PMU ECC and EDC Test	0000050328-125	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PMU ECC and EDC Test	0000050328-458	●	●	●	-	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PMU ECC and EDC Test	0000050329-476	-	-	-	-	●	-	-	-	-	-	-	●	-	●	-	●	-	-	-	-	-	●	-	-	-	-	-	-
SafeWDG Driver for External CIC61508 Watchdog	0000050328-1249	-	-	-	-	-	●	-	●	●	●	●	●	●	●	●	●	-	-	-	-	-	●	-	-	-	-	-	-
SafeWDG Driver for External TLF35584 Watchdog	0000050328-586	-	-	-	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SafeWDG Driver for External TLF35584 Watchdog	0000050328-622	-	-	-	-	-	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SafeWDG Driver for External TLF35584 Watchdog	0000050328-669	-	-	-	●	-	●	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SafeWDG Driver for External TLF35584 Watchdog	0000050328-848	-	-	-	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SafeWDG Driver for External TLF35584 Watchdog	0000050328-984	-	-	-	●	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SafeWDG Driver for External TLF35584 Watchdog	0000050328-1011	-	-	-	●	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SafeWDG Driver for External TLF35584 Watchdog	0000050328-1028	-	-	-	●	●	●	●	●	●	●	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SafeWDG Driver for External TLF35584 Watchdog	0000050328-1167	-	-	-	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SafeWDG Driver for External TLF35584 Watchdog	0000050328-1278	-	-	-	-	-	●	-	●	●	●	●	●	●	●	●	●	●	-	-	-	-	●	-	-	-	-	-	-
SafeWDG TLF35584 - QSPI	0000050328-1425	-	-	-	-	-	-	-	-	-	-	-	-	-	●	-	●	-	-	●	●	-	●	●	●	-	●	-	-

Module	ID	558.1/559.1	557/554	501/502	594	592	513	503.1	513.1	514	514.1	652	531	631	532	632	533	633	634	534	535	635	533.1	535.1	544	636	545	637	546
driver																													
SafeWDG TLF35584 - QSPI driver	0000050328-1431	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-
Software-Based Self Test (SBST)	0000050328-103	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Software-Based Self Test (SBST)	0000050328-283	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Software-Based Self Test (SBST)	0000050329-409	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Software-Based Self Test (SBST)	0000050328-937	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Software-Based Self Test (SBST)	0000050328-1111	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Software-Based Self Test (SBST)	0000050328-1120	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Software-Based Self Test (SBST)	0000050328-1246	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	●	-	-	-	-	-	-
Safety Flip-Flop (SFF) Test	0000050345-1	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SFR Test	0000050328-521	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SFR Test	0000050328-670	-	-	-	-	-	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SMU Driver	0000050328-61	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SMU Driver	0000050328-342	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SMU Driver	0000050328-410	●	●	●	●							-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SMU Driver	0000050355-5	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Module	ID	558.1/559.1	557/554	501/502	594	592	513	503.1	513.1	514	514.1	652	531	631	532	632	533	633	634	534	535	635	533.1	535.1	544	636	545	637	546
SMU Driver	0000050328-639	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SMU Driver	0000050328-656	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SMU Driver	0000050328-889	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SMU Driver	0000050328-1292	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-
SMU Driver	0000050328-1305	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	●	-	-	-	-	-	-
SMU Driver	0000050328-1341	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-
SMU Driver	0000050328-1425	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	●	●	-	-	●	●	-	●	-	-
SMU Driver	0000050328-1429	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-
SMU Driver	0000050328-1430	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-
SMU test	0000050329-451	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SMU test	0000050328-1419	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
SMU Driver/ SMU test	0000050328-1334	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-
SPB Register Access Protection Test	0000050328-683	-	-	-	●	-	●	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SPB Timeout Test	0000050328-655	-	-	-	●	-	●	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SPB Timeout Test	0000050328-774	-	-	-	-	-	●	-	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SPB Register Access Protection Test / SPB Timeout Test	0000050328-406	-	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SPB Register Access Protection Test / SPB Timeout Test	0000050328-473	-	-	-	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SPB Register Access	0000050328-530	-	-	-	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Module	ID	558.1/559.1	557/554	501/502	594	592	513	503.1	513.1	514	514.1	652	531	631	532	632	533	633	634	534	535	635	533.1	535.1	544	636	545	637	546
Protection Test / SPB Timeout Test																													
SPB Register Access Protection Test / SPB Timeout Test	0000050328-1395	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-
SRAM ECC Test	0000050329-205	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRAM ECC Test	0000050329-206	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRAM ECC Test	0000050329-216	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRAM ECC Test	0000050328-204	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRAM ECC Test	0000050329-343	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRAM ECC Test	0000050329-372	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRAM ECC Test	0000050329-385	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRAM ECC Test	0000050329-394	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRAM ECC Test	0000050328-671	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRAM ECC Test	0000050328-708	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRAM ECC Test	0000050328-1285	-	-	-	-	-	-	-	-	-	-	●	-	●	-	●	-	●	●	-	-	-	-	-	-	-	-	-	-
SRAM ECC Test	0000050328-1299	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	●	-	-	-	-	-	-
SRAM ECC Test	0000050328-1315	●	●	-	●	●	●	●	●	-	●	-	●	-	●	-	●	-	-	●	●	-	●	-	-	-	-	-	-
SRI EDC and Bus Error Test	0000050329-174	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRI EDC and Bus Error Test	0000050329-211	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRI EDC and Bus Error Test	0000050350-1	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRI EDC and Bus Error Test	0000050328-97	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Known issues overview

Module	ID	558.1/559.1	557/554	501/502	594	592	513	503.1	513.1	514	514.1	652	531	631	532	632	533	633	634	534	535	635	533.1	535.1	544	636	545	637	546
SRI EDC and Bus Error Test	0000050350-4	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRI EDC and Bus Error Test	0000050328-260	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRI EDC and Bus Error Test	0000050328-446	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRI EDC and Bus Error Test	0000050328-454	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRI EDC and Bus Error Test	0000050328-524	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRI EDC and Bus Error Test	0000050328-582	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRI EDC and Bus Error Test	0000050328-606	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRI EDC and Bus Error Test	0000050328-607	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRI EDC and Bus Error Test	0000050328-624	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRI EDC and Bus Error Test	0000050328-625	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRI EDC and Bus Error Test	0000050328-649	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRI EDC and Bus Error Test	0000050328-650	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRI EDC and Bus Error Test	0000050328-663	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRI EDC and Bus Error Test	0000050328-696	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRI EDC and Bus Error Test	0000050328-884	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRI EDC and Bus Error Test	0000050328-1067	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRI EDC and Bus Error Test	0000050328-1432	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-
Test Handler	0000050329-75	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Test Handler	0000050329-148	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Test Handler	0000050329-158	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Test Handler	0000050328-25	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Test Handler	0000050328-37	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Known issues overview

Module	ID	558.1/559.1	557/554	501/502	594	592	513	503.1	513.1	514	514.1	652	531	631	532	632	533	633	634	534	535	635	533.1	535.1	544	636	545	637	546
Test Handler	0000050328-1340	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-
Trap Test	0000050329-260	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Trap Test	0000050329-261	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Trap Test	0000050329-266	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Trap Test	0000050328-394	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Trap Test	0000050328-476	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Trap Test	0000050328-522	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Trap Test	0000050328-698	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Trap handling modules	0000050328-141	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Trap handling modules	0000050328-590	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Trap handling modules	0000050328-253	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

1.1 Known issues overview: Issues added in the current version

- = Not applicable, ● = Applicable

Note: Release number mentioned in the right most columns is the latest release.

Module	ID	558.1/559.1	557/554	501/502	594	592	513	503.1	513.1	514	514.1	652	531	631	532	632	533	633	634	534	535	635	533.1	535.1	544	636	545	637	546
Generic	0000050328-1438	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-
CPU Watchdog Test and Safety Watchdog Test	0000050328-1436	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-
PMU ECC and EDC Test / PFLASH Monitor test	0000050328-1472	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

2 Known issues: details

2.1 Generic

2.1.1 0000050328-637

`<#pragma align x>` is not working for code sections for Tasking compiler.

Impact:

- None, if the code sections are aligned properly
- If code sections are not aligned, unexpected traps can occur.

Workaround:

The required alignment shall be achieved through the linker script. The alignment requirement of the following code sections (if present) is as follows:

`DEFAULT_START_SEC_IVT_CODE`: align 4
`DEFAULT_IRTST_START_SEC_IVT_CODE`: align 4
`DEFAULT_MTL_START_SEC_TVT_CODE`: align 256
`DEFAULT_LMUBUSMPULFM_START_SEC_TVT_CODE`: align 256
`DEFAULT_LMUREGACCPROTTST_START_SEC_TVT_CODE`: align 256
`DEFAULT_PMUECCEDCTST_START_SEC_TVT_CODE`: align 256
`DEFAULT_PMUECCEDCTST_START_SEC_RAM_CODE`: align 4
`DEFAULT_PFLASHMONTST_START_SEC_RAM_CODE`: align 4
`DEFAULT_PMU_APPL_START_SEC_RAM_CODE`: align 4
`DEFAULT_START_SEC_8BYTE_ALIGN_CODE`: align 8

2.1.2 0000050329-9

Compiler options for MCAL and SafeTlib not aligned. --eabi-compliant and--integer-enumeration are not present in compiler options of SafeTlib.
Impact:
Integration of MCAL and SafeTlib is not efficient.
Workaround:
Compiler options to be updated in SafeTlib (add --eabi-compliant and--integer-enumeration) to ensure compatible options with MCAL for easy integration.

2.1.3 0000050328-126

Tasking 4.2r2 compiler bug (TCVX-38438): The Tasking compiler option **–**OP needs to be used to disable constant propagation optimization. This is required to avoid incorrect code generation for if constructs which checks for constant objects.
Impact:
There is no impact on the SafeTlib, since this scenario is not present.
Workaround:
Disable constant propagation optimization (-OP) to avoid incorrect code generation for application code.

2.1.4 0000050328-138

Tasking 4.2r2 compiler bug (TCVX-39670): The compiler erroneously optimizes function call between disable and enable intrinsics.
Impact:
All the critical sections are analyzed within `__disable()` / `__enable()` as well as `Mcal_SuspendAllInterrupts` / `Mcal_ResumeAllInterrupts` pair and no issues found. There is no impact on the SafeTlib.
Workaround:

One of the following shall be followed to avoid erroneous function call optimization in the application code:

- Use inline assembly instead of intrinsics in application code
- Disable compiler generic assembly optimization (-OG)

2.1.5 0000050328-139

Tasking 4.2r2 compiler bug (TCVX-39883): The compiler optimizes a bit store in a volatile word with st.t instruction. This is incorrect, as st.t instruction does a memory byte access instead of a word access.
Impact:
There is no impact on the SafeTlib, since st.t instruction is not generated.
Workaround:
Disable peephole optimizations –OY to avoid the impact due to compiler bug in the application code.

2.1.6 0000050328-152

For a target selection of TC27xBC, signature tool does not use reset value of CCUCON9 for Signature calculation. In TC27xBC, CCUCON9 has a value of 0x8000000 after PORST. The signature tool assumes that the value is 0x00000000 and calculates a wrong CRC.
Impact:
If a wrong value is generated, it will be found during development stage itself. There is no safety impact.
Workaround:
Customer has to manually use the reset value for CCUCON9, that is, 0x80000000 for signature calculation in TC27xBC target selection.

2.1.7 0000050329-341

Incorrect MCAL SFR Definitions: Some SFRs are word writeable only but they are not defined using Ifx_Strict_32Bit.
Impact:
SafeTlib is performing 32-bit write/read access of SFRs which are word writeable only. Therefore there is no impact on SafeTlib.
Workaround:
Ensure 32-bit access of SFRs which are word accessible in case SFR provided with SafeTlib are used in application code.

2.1.8 0000050328-341

Tasking Linker file syntax is not as per the guideline in the statements with the keyword `<select>`.
It is currently as: `select "*(<Section name>)";`

It is recommended as follows: `select "<Section name>"`
Impact:
The linker file syntax is updated as per the Tasking linker guideline. The output files generated with both the syntax (as mentioned above) are same. Hence, there is no impact.
Workaround:
Not applicable. The user is advised to use the syntax as per the Tasking linker guideline.

2.1.9 0000050328-425

MemMap Section: Inconsistent Placement of variables in sections. Init and non-init data are used together in one section. The following modules are affected:
SMU Driver, CPU MPU Test, DMA Test, Flexible CRC Engine Test, LMU Bus MPU Test, LMU Register Access Protection Test, PMU ECC and EDC Test, PFLASH Monitor Test, Peripheral SRAM Test, CPU Watchdog Test and Safety Watchdog Test, SafeWDG Driver for External CIC61508 Watchdog, SafeWDG Driver for External TLF35584 Watchdog, SafeWDG QSPI
Impact:
No consistency in memory mapping. Init and Non-Init data are used together in one section. A strict checking of section assignments by linker script may lead to failure in linking stage.
Workaround:
None

2.1.10 0000050328-531

Reserved bits in SFRs are not written correctly. There are reserved bits in the many SFRs which are read as 1, and should be written with 0. This is recommended in the User Manual. There are many instances in SafeTlib where this is violated.
Impact:
No safety impact. Analysis is performed in all the instances where there was a violation of User Manual recommendation with respect to writing into reserved bits. There is no impact on SafeTlib.
Workaround:
Not applicable

2.1.11 0000050328-567

Hightec GNU compiler `#pragma GCC reset_options` does not restore optimization level.
Impact:
No Safety impact. Functionality is not affected. All modules that include header files with a reference to a symbol within a No_Optimize code section will actually be compiled with no optimization. All functions intended to be INLINED will not be inlined afterwards.
Workaround:
To ensure the intended optimization level, explicitly specify optimization level in `Ifx_Memmap.h`.

2.1.12 0000050328-1080

The following information is missing in the User manual.
SafeTlib assumes that all masters used by the tests have access rights granted via appropriate ACCEN settings. This is given for the startup tests with the default configuration after reset and shall be ensured by application.
Impact:
If a master (used by application) is not configured in the module ACCEN settings, SafeTlib modules will not be able to write into registers protected by the ACCEN scheme
Workaround:

User shall ensure that all masters used by the tests have access rights granted through appropriate ACCEN settings. This is given for the startup tests with the default configuration after reset and shall be ensured by application.

2.1.13 0000050328-1144

Functions or Variables are not allocated in the expected memory section for WindRiver Compiler.
Impact:
Code and data of SafeTlib modules on compilation with Windriver are placed in default ".text" memory section instead of provided memory sections via Ifx_Memmap.h file.
Workaround:
User has to add "-Xpragma-section-last" to windriver compiler option and include "Mcal_Compiler.h" in following modules header files as _DIABDATA_C_TRICORE_ macro value is defined there.
SMU driver,CPU MPU test,CPU Watchdog test,PFLASH ECC test,SRAM ECC test,Pflash Monitor test,LMU Bus MPU test,DMA test,FCE test,IOM test and SRAM Peripheral Test.

2.1.14 0000050328-1193

AoU related to tresos command line code generation is missing in the user manual.
The customer using Tresos command line interface is expected to validate the input parameters (ex: .epc /.arxml) and react on the exit codes of tresos_cmd.bat file which is used for code generation. Additionally customer is expected to clear old configuration files manually before initiating the code generation.
Note: There is no problem for Tresos GUI users.
Impact:
Incorrect configuration files would be used for the build.
Workaround:
Clear the old configuration files present in the output directory then generate configuration files using tresos GUI and compare GUI generated configuration files with command line generated configuration files.

2.1.15 0000050328-1196

Global variable used in Set/ResetEndInit protection APIs for password and timeout backup not implemented as redundant/diverse.
Impact:
QM software which runs on other core may corrupt the back up variable and no mechanism provided to identify the corruption. In case of a fault in the system, the firing of the WDG will be earlier/delayed, hence impacting the safety.
Workaround:
None.

2.1.16 0000050328-1213

Tasking 5.0r2 compiler bug (Please refer TCVX-41441 for details): The register allocator may spill registers to stack and these stack locations may be re-used.Due to a register optimization problem this stack overlay optimization might fail.
Impact:
The occurrence of the bug does not depend on the structure of the code, but rather on external configuration, e.g. call may affect any software.
Workaround:
Customer shall check the impact (manifestation of compiler bug in generated assembly corresponding to production code that has been integrated) of the issue with the compiler vendor.

2.1.17 0000050328-1216

The following AoU is missing in the User manual.
DMA shall not be used during execution of SafeTlib Pre-run tests.
Impact:
SafeTlib Pre-run tests may fail, if DMA is used during execution of Pre-run tests.
Workaround:
User shall not use DMA during execution of SafeTlib Pre-run tests.

2.1.18 0000050328-1219

The following AoU is missing in the User manual.
If the user switches between Tresos command line interface and Tresos GUI, the user shall ensure that the output configuration files (Cfg.h/Cfg.c) generated through Tresos GUI and command line (ex: .epc/.arxml) are same.This can be ensured if user names the nodes of a container element in alphanumeric sorted order (a-z, 0-9) in Tresos GUI.
Note: Numbers represent smaller value than characters and it is not case sensitive.
Impact:
Difference in configuration files generated through Tresos GUI and Tresos command line interface.
Workaround:
User shall ensure to name nodes of a container element in alphanumeric sorted order (a-z,0-9) in Tresos GUI.

2.1.19 0000050328-1271

Remove any mandatory Linker Options.
Impact:
No restriction imposed by IFX on user on using any specific linker options for any compiler.
Workaround:
None

2.1.20 0000050328-1289

CRC calculation is done to verify that code copied to RAM is not corrupted and is same as flash. The linker symbol used for reading the code in flash is wrong and is pointing to RAM location instead of flash for windriver compiler. Both the symbols were pointing to same address in RAM.
Impact:
As both the linker symbols used are pointing to same address in RAM, CRC is being calculated over code in same addresses.
Workaround:
Update the Windriver Diab linker script so that the linker symbol points to correct Flash address.

2.1.21 0000050328-1335

SafeTlib optional compiler option –OY (Mitigation for compiler issue TCVX-43052) is missing in TASKING v5.0r2 and TASKING v5.0r2p3 Release Notes.
Impact:
None. SafeTlib is tested with/without -OY option and no issues found.
Workaround:
User shall analyse the compiler issue TCVX-43052 in their use case and decide whether to use the compiler option –OY or not.

2.1.22 0000050328-1344

Signature tool user manual: Version information is not updated in Front page, Footer and references.

- Title page and Footer: Signature tool user manual mentions as V2.1 ideally to must have been updated as V2.2
- Title page and Footer: Signature tool user manual date mentions as 2017-05-11ideally to must have been updated as 2018-05-11
- In Reference Section:
  TC29x User's Manual mentions as V1.2 ideally to must have been updated as V1.3
  AURIXTM Safety Manual mentions as V1.2 ideally to must have been updated as V1.5.1
  TLF35584 Safety Manual mentions as TLF35584 Preliminary Safety Manual V0.1 ideally to must have been updated as TLF35584 Safety Manual V1.0

Impact:
No functional impact. Document contents are correct. Only version is not updated.
Workaround:
None.

2.1.23 0000050328-1362

The following information is missing in the User manual.
In ClkmTst, SramEccTst and SffTst, Chip Revision Number information (CHREV) is used to activate software workaround for silicon step dependent errata. But CHIPID register does not have safety measure implemented in hardware, hence user should perform SFR test on CHIPID register before the execution of these tests.
Impact:
If there is no check implemented to verify whether the CHIP ID read out shows the correct value; then it may impact the functional safety.
Workaround:
In ClkmTst, SramEccTst and SffTst, Chip Revision Number information (CHREV) is used to activate software workaround for silicon step dependent errata. But CHIPID register does not have safety measure implemented in hardware, hence user should perform SFR test on CHIPID register before the execution of these tests.

2.1.24 0000050328-1373

Incorrect ISO 26262 part number mentioned in the safetycase, section 4.1.3
Impact:
No functional impact.
Workaround:
Name of the figure 5 in section 4.1.3 is mentioned as 'Goal: fulfillment of system development requirements according to ISO 26262 Part 6'.Integrators to consider it as 'Goal: fulfillment of system development requirements according to ISO 26262 Part 4'.

2.1.25 0000050328-1409

Incorrect name of compatible MCAL version for 29x Rel545 Release notes.
Impact:
Documentation issue. No functional impact.
Workaround:
For 29x device, V600_REL-405 shall be considered as compatible MCAL version for Rel545 instead of V700_REL-405 mentioned in some of the Release notes.

2.1.26 [0000050328-1428]

Hardware Errata CPU_TC.H019: Semaphore handling for shared memory resources.

When the paths from two CPUs to common memory resources are not the same for both CPUs, the effect of two generic stores from one CPU can appear in the opposite order to two generic loads from the other CPU if correct synchronization steps are not taken. This can happen when the master releasing the semaphore has a different access path to a shared resource than to its associated semaphore.
Impact:
It is possible for another master to observe the semaphore update prior to the final update of the guarded state.
Workaround:
None

2.1.27 [0000050328-1438]

User Manual reports following error codes which are not used in the software.
PMUECCEDC_TIMEOUTERR
PFLASHMON_TIMEOUTERR
PHLSRAM_ETR_EOVMISSING
LMUREGACC_DENY_NOSMUALRM
SMUTST_NMI_RESOURCELOCKED
SMUTST_RT_RESOURCELOCKED
Impact:
No functional impact.
Workaround:
User shall not consider the unused error codes during integration.

2.1.28 Linker Options

Mandatory and Optional linker options are not classified clearly.

Workaround:

In general, there is no restriction on the usage of the linker options imposed by SafeTlib, as the linker options do not influence the generated opcode. However the SafeTlib has been tested only with the specified Linker options.
Following linker options can be used which has been tested with SafeTlib.
Tasking linker options:

-Cmpe:vtc	Multicore option selected
-Cmpe:tc0	Single core option selected

Hightec GNU linker options:

-o "$(APPL_OUTDIR)/$(APPL_TARGET).elf" -T"$(APPL_BASE_DIR)\_mcal_pjt.ld" -Wl, -Map(APPL_OUTDIR)/$(APPL_TARGET).map" -nostartfiles -Wl,--allow-multiple-definition -Wl,--cref -Wl,

--oformat=elf32-tricore -mcpu=tc27xx -Wl,--mem-holes -Wl,--extmap="a" where APPL_OUTDIR defines the output directory for the generated elf file e.g APPL_OUTDIR = .\Application\debug\TC26x\ tasking for TC26x APPL_TARGET defines the generated elf file name e.g APPL_TARGET = RefApp_TC26x_tskg for TC26x APPL_BASE_DIR defines the Application Directory APPL_BASE_DIR= .\Application

Linker option details

-mcpu=tc27xx	For Hightec GNU v4.6.3.0, -mcpu=tc27xx is used to support TC1.6 core. This is done as there is no support –mcpu with TC26x / TC29x/ TC23x/ TC22x/ TC21x.
-o "$(APPL_OUTDIR)/$(APPL_TARGET).elf "	Specifies the output file name
-Wl,--oformat=elf32-tricore	Specify target and format of the output file
-T"$(APPL_BASE_DIR)_mcal_pjt.ld"	Specifies the linker script file
-Wl,--mem-holes	Enables the spreading of data over different memory regions
-nostartfiles	The standard system startup files are not used when linking.
-Wl,-Map="$(APPL_OUTDIR)/$(APPL_TARGET).map"	Specifies the link and name of the map file
-Wl,--allow-multiple-definition	Allows multiple definitions of symbols
-Wl,--cref –Wl	Output a cross reference table
-Wl,--extmap="a"	Generates an extended map file with additional information

2.2 Clock Monitor Test

2.2.1 0000050328-427

Implementation of workaround for EPN CCU_TC.002. Due to a hardware issue, clock monitoring for 7.5 MHz frequency is not possible.
Impact:
Unexpected SMU alarms were observed sporadically for the respective clock monitor during the execution of clock monitor test.
Workaround:
None

2.2.2 0000050328-495

AoU related to PLL setting prerequisite is missing in the User Manual. The system operating frequency shall be configured around 100 MHz during the Clock Monitor Test.
Impact:
During PLL clock monitor test, CCU switches to the back-up clock fBACK as clock source fBACK = 100 MHz. If the difference between system operating frequency and back-up clock source is higher, it will lead to sudden change in the frequency which changes the power consumption. This may lead to a power supply drop which further may lead to a reset.
Workaround:
In order to avoid the sudden change in system frequency, the system operating frequency shall be configured around 100 MHz before pre-run test of SafeTlib.

2.2.3 0000050328-520

If MCAL switches off PLLERAY to enter power save mode (this is done in applications which do not use ERAY and have fOSC below 16 MHz) before execution of SafeTlib pre-run test, clock monitor test fails.
Impact:
Clock monitor test fails.
Workaround:
MCAL shall not switch off PLLERAY if clock monitor test needs to be executed.

2.2.4 0000050328-1044

The following information is missing in the User manual.
User shall ensure that $f_{PLLERAY}$ shall be in normal mode.
Impact:
If PLLERAY is in power save mode during ClkmTst, ClkmTst will fail.
Workaround:
User shall ensure that $f_{PLLERAY}$ shall be in normal mode.

2.2.5 0000050328-1333

Enable/Disable option for CCUCON0 register ADCCLKSEL bit not applicable for devices other than 27xBC and hence this option not available in latest versions of signaure tool. However screenshot in signature tool user manual still has this option.
Impact:

Signature tool is correct and does not provide enable/disable option for CCUCON0 register ADCCLKSEL bit for the not applicable devices. Hence there is no impact.

Workaround:

User shall ignore the ADCCLKSEL bit options mentioned in the signature tool user manual for devices other than 27xBC.

2.3 CPU BUS MPU Test

2.3.1 0000050328-1148

The following information is missing in the User manual.
PSW.S bit shall not be set for the core before executing CpuBusMpuLfmTst test.

Impact:

SafeTLib test CpuBusMpuLfmTst fails, if called with PSW.S being set.

Workaround:

User shall ensure that, PSW.S bit is not set for the core before executing CpuBusMpuLfmTst test.

2.3.2 0000050328-1234

In CpuBusMpuLfmTst, when expecting the alarm for unauthorized access, there is no timeout in place to wait for the expected alarm.

Impact:

SafeTLib test CpuBusMpuLfmTst fails when SPB clock is configured to lower values (PLL divider in register SCU_CCUCON0.SPBDIV is set greater than 4).

Workaround:

User shall configure SPB clock to higher values (PLL divider in register SCU_CCUCON0.SPBDIV shall be set less than or equal to 4).

2.3.3 0000050328-1284

CpuBusMpuLfmTst test fails if any of the alarm in same group (alarm group 0 if executing this test on Core0, alarm group 1 if executing this test on Core1, alarm group 6 if executing this test on Core2) is raised before or during the execution of CpuBusMpuLfm test.
These unexpected alarms could result from correctable single bit errors in the local DSPRs as single bit correctable error alarms belong to same group.

Impact:

SafeTLib test CpuBusMpuLfmTst fails due to signaling of unexpected alarm in same group.

Workaround:

None

2.4 CPU MPU Test

2.4.1 0000050328-525

The following registers are modified by the test and not restored:
CPU0_DPWE0 and CPU0_DPRE0

Impact:

CPU_DPRE_0, CPU_DPWE_0:
After the CPU MPU Test, the protection set 0 is configured to allow read and write access potentially to the unintended data protection ranges (which depends on the value of (CPU_DPRE_0, CPU_DPWE_0).

The value of DPRE_0 and DPWE_0 after the test on one hand could be restrictive compared to the setting intended by the application. This would lead to traps, which will be very easily found during development. No safety impact would be expected.
On the other hand, if the wrong value in DPRE_0 and DPWE_0 registers leads to a weaker access protection than originally intended, there is a high probability that this could have an impact on safety. From code which is using protection set 0, unintended reads or writes outside it is assigned memory region might not be properly caught.
Workaround:
Data protection set 0 (DPRE_0 and DPWE_0) shall be reconfigured with the application intended values after the prerun test of SafeTlib.

2.4.2 0000050328-680

Global variable *CpuMpuTst_lTestData[CoreId].IncludeMemory[0]* is not initialized.
Impact:
The test succeeds even though the write access to the variable *CpuMpuTst_lTestData[CoreId].IncludeMemory[0]* fails due to hardware error.
Workaround:
None

2.4.3 0000050328-1339

Non-matching section defines in CpuMpuTst.h file. The IFX_CPUMPUTST_STOP_SEC_CODE_8BYTE_ALIGN_ASIL_B define does not match with the IFX_CPUMPUTST_START_SEC_CODE_ASIL_B definition.
Impact:
Map file shows that the code is located in correct section despite having incorrect name definition. Hence there is no functional impact.
Workaround:
None

2.5 CPU Watchdog Test and Safety Watchdog Test

2.5.1 0000050329-387

The Watchdog test fails if it is run in parallel on several cores.
Impact:
WDG test fails when it is run in parallel on all the cores. The test fails in the CPU, which tries to acquire the spin lock that is already locked by other CPU.
Workaround:
WdgTst shall not run in parallel on all the cores.

2.5.2 0000050328-526

The following registers are modified by the test and not restored:
SRC_BCUSPBSBSRC
SBCU_ECON

Impact:
SRC_BCUSPBSBSRC:
SRC_BCUSPBSBSRC.SRR and sometimes SRC_BCUSPBSBSRC.IOV bits are set during the execution of the test.

If after the test the interrupt source is enabled (by setting the SRC_BCUSPBSBSRC.SRE bit), the pending service request takes part in the interrupt arbitration and a false interrupt event will get triggered.
If in the interrupt routine the plausibility of the interrupt is checked (as required by the safety manual), no reason would be found. Therefore the wrong interrupt could be detected. A plausibility check in case of an SPB service interrupt would for example be to check SBCU_ECON.ERRCNT is greater than zero).
In any case, if the handling of the interrupt in the end would lead (ultimately and unconditionally) to a device reset, the problem would be easily detected during development (as the program would always reset during startup). This is the typical use case for SPB errors; therefore the expected probability for safety impact is very low.

SBCU_ECON:
There is a change in the FPI error counter (SBCU_ECON.ERRCNT) value. The FPI bus error counter will be changed due to the occurrence of FPI bus errors during the execution of the test. There is no impact. This value has to be ignored by the application.

Workaround:
SRC_BCUSPBSBSRC:
Clear the service request (by setting SRC_BCUSPBSBSRC.CLRR) and the interrupt overflow flag (by setting SRC_BCUSPBSBSRC.IOVCLR) after running the prerun test of SafeTlib, but before activating the service request (SRC_BCUSPBSBSRC.SRE).

SBCU_ECON:
Clear the FPI bus error counter (SBCU_ECON.ERRCNT) by performing dummy read of SBCU_ECON register after running the pre-run test of SafeTlib.

2.5.3 0000050328-1238

In Watchdog test, when expecting the alarm there is no timeout in place to wait for the expected alarm.
Impact:
CPU and Safety Watchdog test fails when SPB clock is configured to lower values (PLL divider in register SCU_CCUCON0.SPBDIV is set greater than 4).
Workaround:
User shall configure SPB clock to higher values (PLL divider in register SCU_CCUCON0.SPBDIV shall be set less than or equal to 4).

2.5.4 0000050328-1290

Watchdog Test Signature calculation always includes the complete register content from AG3 instead of considering only specific alarm bit for Watchdog test.
Impact:
SafeTLib Watchdog Test will pass with wrong signature due to signaling of unexpected alarm in same group.
Workaround:
None.

2.5.5 0000050328-1300

SafeTLib watchdog test might fail when running from non-cache locations.
Impact:
Execution of Watchdog Test from non cached memory is slower compared to Cached memory.

Due to slower execution, WDG timer reload value could increment and rollover to 0x0000 before WDG is enabled. WdgTst fails due to long expiry set by rolled over reload value.
Workaround:
Do not configure a non-cached location for SafeTLib watchdog test.

2.5.6 0000050328-1436

Calling function TRAP_UnregisterTrapHandler immediately after the illegal change of protected register to generate asynchronous trap is vulnerable to SPB delays.
Impact:
SCU writes takes more time if there is SPB traffic from another bus master and Trap may generate after the TRAP_UnregisterTrapHandler and test fails.
Workaround:
None

2.6 DMA Test

2.6.1 0000050328-523

The following registers are modified by the test and not restored:
DMA_ME1CHSR
DMA_CHCSRz
DMA_SHADR
SRC_DMAERR

Note: : z is the DMA channel number, which is configurable.

Impact:
DMA_CHCSRz, DMA_SHADR:
DMA_CHCSRz.ICH bit is set during the execution of the test. If a channel pattern detection interrupt service request or a channel source and destination wrap buffer interrupt service request is generated after execution of the DMA test, also an additional channel transfer interrupt service request will wrongly be indicated to the interrupt service routine of the user application.
This might lead to the user application wrongly assuming that new data has been copied to the destination buffer. In this case there is a risk that old data/wrong data are used.

SRC_DMAERR:
If after the test the interrupt source is enabled (by setting the SRC_DMAERR.SRE bit), the pending service request takes part in the interrupt arbitration and a false interrupt event will get triggered.
If in the interrupt routine the plausibility of the interrupt is checked (as required by the safety manual), no reason would be found. Therefore the wrong interrupt could be detected. A plausibility check in case of an DMA error service interrupt would for example be to check for the error reason in DMA_ERRSRx (x=0..1).
In any case, if the handling of the interrupt in the end would lead (ultimately and unconditionally) to a device reset, the problem would be easily detected during development (as the program would always reset during startup).

DMA_ME1CHSR:

This is a status register of DMA move engine. This contains the status of previous DMA action. This value has to be ignored by the application.

Workaround:
DMA_CHCSRz, DMA_SHADR, DMA_ME1CHSR:
The DMA channel shall be reset by setting the TSRz.RST bit after running the pre-run test of SafeTlib.

SRC_DMAERR:
Clear the service request (by setting SRC_DMAERR.CLRR) after running the pre-run test of SafeTlib, but before activating the service request (SRC_DMAERR.SRE).

2.6.2 0000050328-709

During the execution of the API `DmaTst_CRCTst()`, uint32 buffers are allocated on the ustack. If the `LSL` file is designed by the application in a way that A10 register contains a 0xD000xxxx value instead of 0x7000xxxx value then the DMA test fails. This is because 0xD000xxxx region is a reserved region for DMA.

Impact:
DMA test fails if A10 register contains a 0xD000xxxx.
Workaround:
User stack shall be allocated on Global address map (0x7000xxxx).

2.6.3 0000050328-846

The DMA Time Stamp test (DmaTst_TimestampTst) fails due to setting of the CHCSRz.ICH bit before updating timestamp data (which is the part of DMA transfer). The DMA test checks for the timestamp information after CHCSRz.ICH bit is set, assuming DMA transfer is complete. Since in some cases, timestamp information is not yet updated even though CHCSRz.ICH bit is set, DMA test fails.

Impact:

DMA test fails.

Workaround:

None

2.6.4 0000050328-951

DMA CRC test fails when cached location is used as a source address.

Impact:

DMA CRC test fails if cached location (segment 8 /9) is used as source address.

Workaround:

Do not configure a cached location as a source address.

2.7 Flexible CRC Engine (FCE) Test

2.7.1 0000050328-906

Signature Tool generates wrong Signature in case 16-bit registers are configured in the FCE test.

Impact:
Wrong signatures generated for 16-bit registers. No safety impact, since the issue of wrong signatures was found in the development stage.
Workaround:
Do not configure 16-bit registers in FCE test. OR
For only FCE test, use the value generated from the signature tool which is released with Rel531.

2.8 Interrupt Router Test

2.8.1 0000050328-702

If the current CPU priority number (CPUx_ICR.CCPN) is not set to 0, the interrupt router test fails as the interrupts requested by this test would be blocked.

Impact:
The interrupt router test fails if the current CPU priority number (ICRx.CCPN) is not 0.
Workaround:

The user must ensure that current CPU priority number (CPUx_ICR.CCPN, where *x* is core number) is set to 0 before interrupt router test is executed. This can be done by calling `MTCR(ICR, ….)` instruction before executing pre-run test of the SafeTlib.

2.9 IOM Test

2.9.1 0000050328-498

IOM test configuration parameters are mapped to section IFX_IOMTST_START_SEC_POSTBUILDCFG_ASIL_B. Since IOM test is a link time module, mapping has to be done to link time memory section
Impact:
No functional impact. Configuration parameters which are link time are mapped to post-build memory section. Memory protection cannot be selectively enabled for link time and post-build modules for SafeTlib modules. Therefore, both link time and post-build memory sections can be accessed during SafeTlib test execution.
Workaround:
Customer has to ensure that the section IFX_IOMTST_START_SEC_POSTBUILDCFG_ASIL_B is mapped to link time memory section in the `Ifx_MemMap.h` file as follows:

```
#elif defined (IFX_IOMTST_START_SEC_POSTBUILDCFG_ASIL_B)
   #undef       IFX_IOMTST_START_SEC_POSTBUILDCFG_ASIL_B
   #define DEFAULT_START_SEC_LINKTIMECFG
#elif defined (IFX_IOMTST_STOP_SEC_POSTBUILDCFG_ASIL_B)
   #undef       IFX_IOMTST_STOP_SEC_POSTBUILDCFG_ASIL_B
   #define DEFAULT_STOP_SEC_POSTBUILDCFG
```

2.9.2 0000050328-1073

The following information is missing in the User manual.
GPIO pin used for IomTst shall not be connected to external circuitry which forces the pin to a permanent high/low state during the test.

Impact:
If any external signal forces the GPIO pin used for IomTst to a permanent low/high against the internal weak pulls during the test, IOM test will fail.
Workaround:
User shall ensure that GPIO pin used for IomTst shall not be connected to external circuitry which forces the pin to a permanent high/low state during the test.

2.10 LMU Bus MPU Test

2.10.1 0000050328-527

The SRC_XBARSRC following register is modified by the test and not restored:

Impact:
SRC_XBARSRC:
SRC_XBARSRC.SRR and sometimes SRC_XBARSRC.IOV bits are set during the execution of the test.
If after the test, the interrupt source is enabled (by setting the SRC_XBARSRC.SREbit), the pending service request takes part in the interrupt arbitration and a false interrupt event will get triggered.
Workaround:
SRC_XBARSRC:
Clear the service requests (by setting SRC_XBARSRC.CLRR) and the interrupt overflow flag (by setting SRC_XBARSRC.IOVCLR) after running the pre-run test of SafeTlib, but before activating the service request (SRC_XBARSRC.SRE).

2.11 LMU Register Access Protection Test

2.11.1 0000050328-528

The following registers are modified by the test and not restored:

- SRC_XBARSRC
- SBCU_ECON

Impact:
SRC_XBARSRC:
SRC_XBARSRC.SRR and sometimes SRC_XBARSRC.IOV bits are set during the execution of the test.
If after the test the interrupt source is enabled (by setting the SRC_XBARSRC.SRE bit), the pending service request takes part in the interrupt arbitration and a false interrupt event will get triggered.
SBCU_ECON:
There is a change in the FPI error counter (SBCU_ECON.ERRCNT) value. The FPI bus error counter will be changed due to the occurrence of FPI bus errors during the execution of the test. There is no impact. This value has to be ignored by the application.
Workaround:
SRC_XBARSRC:
Clear the service requests (by setting SRC_XBARSRC.CLRR) and the interrupt overflow flag (by setting SRC_XBARSRC.IOVCLR) after running the pre-run test of SafeTlib, but before activating the service request (SRC_XBARSRC.SRE).
SBCU_ECON:
Clear the FPI bus error counter (SBCU_ECON.ERRCNT) by performing dummy read of SBCU_ECON register after running the pre-run test of SafeTlib.

2.12 LMU Bus MPU Test / LMU Register Access Protection Test

2.12.1 0000050328-478

The following registers are modified by the test and not restored:

- XBAR_ARBCON4
- XBAR_INTSAT

Impact:
XBAR_ARBCON4:
Interrupt acknowledgement bit field (XBAR_ARBCON4.INTACK) is set during the execution of the test, indicating that the interrupt acknowledgment of SRI arbiter error is pending.
If the handling of this SRI arbiter error (by interrupt routine or through polling the INTACK bit) will lead (ultimately and unconditionally) to a device reset, the problem will be easily detected during development (as the program would always reset during startup). This is the typical use case for SRI error handling, therefore, the expected probability for safety impact is very low.
In case the SRI arbiter error is ignored, the only effect would be that the ERRADDR and ERR registered stay locked and are not updated with the information of later error on the same arbiter. This should not have any safety impact (if an arbiter error can have safety impact, the application can anyway not simply ignore it).

XBAR_INTSAT:
Bit field PRSCI4 is set in the SRI arbiter interrupt status register (XBAR_INTSAT.PRSCI4). It is set during the execution of the LMU Bus MPU Test / LMU Register Access Protection Test, indicating a protocol error from SCI4 (Slave connection interface 4) has occurred and is yet to be handled/acknowledged. This does not have any impact. The bit field PRSCI4 has to be ignored.

Workaround:
XBAR_ARBCON4:
Clear the XBAR_ARBCON4.INTACK bit by writing 1 to it after running the prerun test of SafeTlib.

XBAR_INTSAT:
Clear the XBAR_INTSAT.PRSCI4 bit by writing 1 to it after running the prerun test of SafeTlib.

2.13 Peripheral SRAM Test

2.13.1 0000050329-391

CAN communication fails after execution of peripheral SRAM test for CAN SRAM.
Impact:
Peripheral memory is always getting cleared whereas it should be getting cleared under the check of user configuration.
Workaround:
The user shall perform kernel reset for the CAN after the pre-run tests of SafeTLib. Refer to the chapter Controller Area Network Controller in the microcontroller user manual to know how to perform the kernel reset.

2.13.2 0000050329-395

The following registers are modified by the test and not restored:
DMA_MEMCON.INTERR
Impact:

DMA_MEMCON.INTERR is the bit that indicates the occurrence of internal ECC error during the execution of the test. There is no impact due to this since it does not block the further ECC errors.
Workaround:
User shall clear DMA_MEMCON.INTERR bit after the execution of pre-run tests of the SafeTlib.

2.13.3 0000050328-956

The following Ethermac and EMEM clock registers are modified by the test and not restored.
ETH_CLC.DISR
EMEM_CLC.DISR
Impact:
No functional impact. The Ethermac and EMEM clock remains enabled after the completion of Peripheral SRAM test.
Workaround:
The values of ETH_CLC.DISR and EMEM_CLC.DISR shall be backed up before the execution of SafeTlib pre-run test and shall be restored after the execution of SafeTlib pre-run tests.

2.14 PMU ECC and EDC Test

2.14.1 0000050336-1

A non-correctable multi-bit / addressing error occurred because of reading a page that was not programmed. This faulty read operation was occurring when unprogrammed memory area that was beyond the boundary of a helper function (local functions) was accessed for copying from PFlash to RAM.
Impact:
Unexpected trap class 4 (data synchronous trap) occurs when unprogrammed memory is accessed.
Workaround:
None

2.14.2 0000050328-461

The following information is missing in the User manual.
The integrator can either use the service function `Sl_ChkProgErrorPattern` of the SafeTlib or a Flash tool to program the appropriate test pattern. It is recommended to use the `Sl_ChkProgErrorPattern` API function only inside the production line.
Using the `Sl_ChkProgErrorPattern` API in the field would prevent end-of-line locking of the Flash, which introduces security issues with respect to the tuning protection.
Impact:
This information is missing in the User Manual. If the customer uses the `Sl_ChkProgErrorPattern` API in the field, it would prevent end-of-line locking of the Flash, which introduces security issues with respect to the tuning protection.
Workaround:
None

2.14.3 0000050329-476

Windriver Diab compiler optimizes the assembly code in function INVOKE_APPL_TRAP_HANDLER by removing forwarding of Trap_Id to Application trap handler.
Impact:
Correct value of Trap_Id is not provided to the application trap handler.
Workaround:
None.

2.15 PFLASH Monitor test

2.15.1 0000050328-666

Unintended clearing of SMU alarms in SMU_AG2 register. All Alarms except alarms 2, 3, 4 and 6 of group 2 which are not supposed to be cleared are also getting cleared.

Impact:

If group 2 alarms that are not related to PFlash Monitor test are already raised before executing the `PFlashMonTst`, they will be cleared. If these alarms are set due to genuine reason during the execution of other SafeTlib tests or application code before the execution of PFLASH Monitor test, the user will miss the alarms.

Workaround:

User should ensure that none of the alarms in SMU alarm group 2 are set before execution of PFLASH Monitor Test by running `PFlashMonTst` in a dedicated group.

2.16 PMU ECC and EDC Test / PFLASH Monitor test

2.16.1 0000050328-210

If `PMUECCEDC_SMU_SETALMACTIONERR` error occurs, the error code `PMUECCEDC_GENERALHWFAILURE` is returned.

Impact:

Wrong error reporting.

Workaround:

None

2.16.2 0000050328-440

Writing to the FSR register by a bitwise operation in Hightec GNU is updating other bits. Hightec GNU implements the SFR bit update operation as a read-modify-write operation. When this is done on a status register, such as FLASH0_FSR, unintended bit flips can occur.

Impact:

Unintended bit flips can occur in the FSR register.

Workaround:

None

2.16.3 0000050328-692

The run address of a function (`Sl_lReadFlashPage`) in the code section `PFLASH_MON_ECCEDC_RAMCODE` is being set to PFlash instead of RAM by the Windriver Diab compiler due to inlining of small-sized functions. This function must be executed from RAM.

Impact:

A function that was expected to be executed from RAM runs from PFlash, resulting in reading of wrong ECC value due to concurrent instruction fetches from PFLASH. This will lead to the failure of the PMU ECC and EDC Test / PFLASH Monitor test.

Workaround:

The `Sl_lReadFlashPage` function shall not be made inline. This has to be done explicitly for Windriver Diab compliler - Use #pragma noinline Sl_lReadFlashPage

2.16.4 0000050328-125

If the RAM reserved for PMU ECC and EDC /PFLASH Monitor Test execution corresponds to DSPR/PSPR of a different core, the test could fail because invalidation of PCACHE is not performed. This happens only if the PCACHE already contains an entry corresponding to the reserved address.

Impact:

Class 1 Memory protection NULL Address trap will occur if the PCACHE already contains an entry corresponding to the reserved address.

Workaround:

User should reserve the RAM in the DSPR/PSPR of the core which is used to program the error patterns.

2.16.5 0000050328-458

The array reserved for PMU ECC and EDC /PFLASH Monitor Test execution was getting allocated in LMU Ram instead of the intended PSPR.

Impact:

There is no functional impact. The use of farnoclear for the section `MTL_PMU_ECC_EDC_TST_SPRAM` without #pragma noclear lead to the linker assigning the `PmuEccEdcTst_Spram to LMU RAM` array instead of PSPR. `#pragma noclear` could not be used because of the use of "--eabi-compliant" in the compiler options.

Workaround:

Use `farbss` instead of `farnoclear` in the `Ifx_MemMap.h` file for the section `MTL_PMU_ECC_EDC_TST_SPRAM`.

```
#pragma section farbss="MTL_PMU_ECC_EDC_TST_SPRAM
```

2.16.6 0000050328-925

Flash access during pattern generation shall always use non-cached addressing.

Impact:

If cacheable area is used for storing the patterns and additionally Dcache is enabled, it can lead to a wrong ECC value getting programmed. This can lead to a failure of PMU ECC and EDC Test / PFLASH Monitor test.

Workaround:

Workaround 1: Patterns shall be allocated to a non-cached address through linker file as mentioned in the SafeTlib User Manual linker script section.

Workaround 2: User shall disable Dcache before execution of PMU ECC and EDC Test / PFLASH Monitor tests.

Workaround 3:

The user code shall not access the test patterns PmuEccEdcTst_TestPatternBx. After pattern programming, user shall verify in production that PMU test in SafeTlib does not fail.

2.16.7 0000050328-983

Devices with three PFlash banks are not supported.

Impact:

On selecting a device with three PFlash banks using the `ResourceSubDerivative` parameter in the `Resource xdm` file, the compilation fails. This is due to missing support for three PFlash bank devices.

Workaround:

None

2.16.8 0000050329-494

Mismatch in version number between source code and delivered text file for PMU ECC and EDC and PFLASH Monitor Test header files.

Impact:

Delivered text file contains incorrect version number for PMU ECC and EDC and PFLASH Monitor Test header files.
Workaround:
None

2.16.9 0000050328-1287

The following AoU is missing in the User manual.
Ensure interrupts must be disabled and FLASH access by any Hardware modules prevented during programing of error pattern.
Impact:
Accessing DFLASH or PFLASH during programing of error pattern might lead wrong programing of error pattern.
Workaround:
Ensure interrupts must be disabled and FLASH access by any Hardware modules prevented during programing of error pattern.

2.16.10 0000050328-1358

The "Sl_FlsErrPtrn " checks only multi bit error pattern available in at least one pattern set or not to verify whether the bit errors are already programmed. This does not ensure all the patterns are already programmed.
Impact:
PFLASH Monitor Test could not work if a power interruption happens during ECC error injection.
Workaround:
User shall ensure that there is no power interruption happens during ECC error injection.

2.16.11 0000050328-1472

Following information missing in the user manual.
Sl_ChkProgErrorPattern API may read PFLASH pages with uncorrectable ECC errors. It is expected that this API is executed before the SMU is put into RUN state. If this API is used when the SMU is already in RUN state, the caller shall be prepared to receive SMU alarms due to PFLASH ECC errors. In this case, caller shall clear received SMU alarms.
Impact:
PFLASH ECC error SMU alarms may be reported if the SMU is already in Run state while executing Sl_ChkProgErrorPattern API.
Workaround:
Execute Sl_ChkProgErrorPattern API before the SMU is put into RUN state.

2.17 SafeWDG Driver for External CIC61508 Watchdog

2.17.1 0000050328-1249

The following AoU is missing in the User manual.
User shall use user commands in secure SPI mode only to read Dflash and not to do any flash write operation in secure SPI mode (See User Manual SM_AURIX_STL_188).
Impact:
CIC61508 Dflash write is not allowed using CIC61508 driver.

Workaround:
User shall use user commands in secure SPI mode only to read Dflash and not to do any flash write operation in secure SPI mode.

2.18 SafeWDG Driver for External TLF35584 Watchdog

2.18.1 0000050328-586

Tx/Rx buffers (`SafeWdgExtTlf_TxBuf[]`, `SafeWdgExtTlf_RxBuf[]`) for DMA transfers not safely aligned.
In the tasking compiler, the required 4-byte alignment is ensured due to mandatory -eabi-compliant option.
In the Hightec GNU /Windriver Diab compilers, the required 4-byte alignment happens as the data type of the Tx/Rx buffers is unit32, even though these variables are declared in the middle of the TLF data structure.
Impact:
None, if the alignment is achieved. Otherwise it shall lead to error in the DMA transfer.
Workaround:
None

2.18.2 0000050328-622

The Tresos configuration for `DEVCTRL.VREFEN` is wrongly generated.
Impact:
If the application wants to enable `DEVCRTL.VREFREN` through the configuration, it will remain in the disabled state due to the wrongly generated configuration file.
Workaround:
To enable `VREFREN`, users are required to manually modify `SAFEWDGEXTTLF_VREF_EN` in the `SafeWdgExtTlf_Cfg.h` configuration file:

```
#define SAFEWDGEXTTLF_VREF_EN (0x8U)
```

2.18.3 0000050328-669

In the `SafeWdgExtTlf_Activate()` API, during the first write operation to DEVCTRL register, the default values are written instead of customer configured values for TRK1EN, TRK2EN, COMEN, VREFEN bit fields.
Impact:
No safety impact since the customer configured values are written in to the DEVCTRL register in the `SafeWdgExtTlf_Activate()` API during the second write.
Workaround:
Not applicable

2.18.4 0000050328-848

Synchronization between SafeWdg driver and External TLF35584 is lost during servicing of functional watchdog, when SPI communication is missed (due to external noise factors).
Impact:
Due to loss of synchronization, error counter starts incrementing in external TLF watchdog and shall lead to a reset.
Workaround:
None

2.18.5 0000050328-984

When QSPI communication error occurs, the `SafeWdgExtTlf_GetJobResult()` API does not return expected error code (expected `SWDG_JOB_COM_ERR`)

Impact:
The application wrongly indicates that communication is still going on (API `SafeWdgExtTlf_GetJobResult returns SWDG_JOB_ACCEPTED`API) in case there is QSPI communication error.
Workaround:
In order to recover from such situation, where `SafeWdgExtTlf_GetJobResult()` returns `SWDG_JOB_ACCEPTED` for longer than expected, this situation can be considered as an error on TLF side and user shall take appropriate action for that.

2.18.6 0000050328-1011

Configuration parameter `SafeWdgExtTlfStbyEn` does not have any effect. Irrespective of the configuration, standby power is enabled.
Impact:
The standby power cannot be disabled even though it is configured to be disabled using the configuration parameter `SafeWdgExtTlfStbyEn`.
Workaround:
User can configure settings for Standby LDO using `SafeWdgExtTlf_UserRequest()` API After execution of `SafeWdgExtTlf_Activate` sequence is over.

2.18.7 0000050328-1028

The following AoU is missing in the User manual.
In case TLF driver is configured to use DMA, user has to ensure, that safe state is triggered on SMU alarms 17, 18, 19 of alarm group 4.
Impact:
In case of DMA RAM error, TLF driver fails to handle this error and proceeds to transferring wrong data to the external TLF chip. This may lead to unexpected behavior of the external TLF device.
Workaround:
Safe state is triggered on SMU alarms 17, 18, 19 of the alarm group 4.

2.18.8 0000050328-1167

SafeWdgExtTlf_Activate() fails when SYSPCFG1 – SS2DEL, ERRSLPEN, ERRRECEN, ERRREC are configured for non-default values.
Impact:
TLF35584 activation fails with SYSPCFG1–SS2DEL/ERRSLPEN/ERRRECEN/ERRREC configuration.
Workaround:
1. User shall not change values of SYSPCFG1–SS2DEL/ERRSLPEN/ERRRECEN/ERRREC in configuration tool.
2. User shall call SafeWdgExtTlf_UserRequest () API for desired values to be written to SYSPCFG1 and SYSPCFG0 registers.

2.18.9 0000050328-1278

During startup initialization of External TLF35584 Watchdog standby LDO supply is switched off.
Impact:
SafeWDG Driver for External TLF35584 Watchdog creates glitch on standby LDO supply.
Workaround:
None

2.19 SafeWDG TLF35584 - QSPI driver

2.19.1 0000050328-1425

SafeWdgQspi.bmd files contains two <MIN> tags instead of one <MAX> and one <MIN> tag for the parameter SafeWdgQspiModule for few devices.

Impact:

Possible error during configuration of SafeWdgQspiModule.

Workaround:

None

2.19.2 0000050328-1431

An automatic variable "Fifo_Level" is define as static within the function definition of "SafeWdgQspi_lDMAChRst_QspiRxFIFOClear ()" and no memory section defined for this variable.

Impact:

The variable will not be stored in a defined region.

Workaround:

Allocate this variable in a proper section.

2.20 Software-Based Self Test (SBST)

2.20.1 0000050328-103

The value of test ID in the test result from the SBST tests on TC1.6E core and TC1.6P core (APIs `CpuTst_CpuSbstETst` and `CpuTst_CpuSbstPTst` respectively) is wrong.

Impact:

Interpretation of the SBST test result will be misleading as wrong test Id is returned as part of result. Hence, it can mislead the user that wrong test has been executed.

Workaround:

The Test Id part of the SBST test result should not be considered for any verification. The lower 16 bits should be evaluated for confirming a test pass/fail.

2.20.2 0000050328-283

SBST is not backward compatible towards `EXPECTED_SBST_CRC_VALUE`.

Impact:

When SBST configuration file, `CpuSbstTst_Cfg.h`, of Rel594 and later is used with earlier SBST implementation file, `CpuSbstTst.c` then it leads to compilation warning.

Workaround:

None

2.20.3 0000050329-409

Description regarding SBST execution requirements is not clear in the User Manual. SBST shall be run on all non-lockstep cores used by the Safety applications.

Impact:

It is mandatory to run the test on all non-lockstep cores used by the Safety applications.

There is no impact if SBST is run on a lockstep core.
Workaround:
Run SBST on all non-lockstep cores used by the Safety applications.

2.20.4 0000050328-937

SBST Wrapper test signature calculation is not correct. Calculated SBST CRC result (`sbst_crc_result`) should be a part of the test signature calculation for SBST Wrapper. Currently SBST wrapper considers only the seed value and the final result for the test signature calculations.
Impact:
If the instruction which is used to compare calculated CRC and the expected CRC is corrupted and always return TRUE, the function returns `SBST_P_SUCCESS` and it will also return the correct signature. Hence, the wrong calculated SBST CRC result is not detected. So watchdog does not put the system in safe state in case of wrong CRC from SBST test.
Workaround:
User can use CRC generated by SBST at location `__SBST_CRC_RESULT` from the linker allocation to know if CRC generated by SBST is as per the expected value. In case of wrong value reported at this location from SBST, user need to implement measure to go to safe state.

2.20.5 0000050328-1111

Missing AoUs in UM

- User shall ensure that SBST is configured in run time tests.
- User shall ensure that all application traps are terminated to trap handlers.
- User shall ensure that all unused traps are terminated in endless loops(no recovery is possible and watchdog will time out).
- User shall test the plausibility of interrupts and the interrupt rate at application level. This includes the checking of valid peripheral notification when in the interrupt handler, and using an independent software counter in each ISR, that is checked for plausibility in a lower rate periodic task (to detect too few or too many interrupts).

Impact:
If Non lockstep core is used by safety Applications, not configuring SBST test in run time tests is a safety issue.
Unpredictable behaviour due to Application Traps/unused traps if AoU is not fulfilled
Safety Mechanism [SAS_SBST_AoU_008] of Aurix safety manual not fulfilled if user does not test the plausibility of interrupts and the interrupt rate at application level.
Workaround:
All AoUs shall be fulfilled by the user.

2.20.6 0000050328-1120

Support functions SchM_Enter_SafeTcore (void) and SchM_Exit_SafeTcore (void) are called from SBST with CALLA instruction. This instruction only allows for absolute addresses in the lower 2MB range(where SBST code is located).Therefore the sections containing these two functions need to be allocated in the same group as all other SBST sections.

Impact:
Call back functions SchM_Enter_SafeTcore (void) and SchM_Exit_SafeTcore (void) will not execute if it's not allocated in lower 2MB range.

Workaround:
The required allocation shall be achieved through the customer memmap and linker script.

2.20.7 0000050328-1246

The following information is missing in the User manual.
The task which executes the SBST must have write access to its own CSA area. This is required because SBST explicitly writes to CSA.
Impact:
SBST test shall fail if the task which executes the SBST does not have write access to its own CSA area.
Workaround:
User shall ensure that the task which executes the SBST has write access to its own CSA area.

2.21 SFF test

2.21.1 0000050345-1

Register monitor mechanism reports a failure in the SMU when SFF test is executed. This issue is applicable in TC27xBC step device only (Errata SMU_TC.002). This error shows when PLL clock is selected as a clock source. With backup clock as clock source, the error does not exist.
Impact:
Safety flip-flops of the SMU module cannot be monitored using the SFF test.
Workaround:
Do not activate SFF test. To monitor the registers of SMU, periodical readback of the registers can be done.

2.22 SFR test

2.22.1 0000050328-521

SFR test (Crc and Cmp tests) does not support 16-bit registers.
Impact:
If any 16-bit register is configured, then it will lead to trap generation.
Workaround:
None

2.22.2 0000050328-670

Due to the errata MTU_TC.005, user has to enable the MBIST mode or enable the module kernel clock to access MCx_ECCD to execute SFR test with 16 bits register support. This information should be added in SafeTlib UM as AoU to the user (See User Manual SM_AURIX_STL_082).
Impact:
When MBIST is disabled, there can be sporadic cases during access of ECCD register wherein value read/written from/to MCx_ECCD registers are wrong.
Workaround:
None

2.23 SMU Driver

2.23.1 0000050328-61

Setup of the FSP timings is incorrect in case SMU configuration files are generated using the delivered Tresos plug-ins. The configured value in the parameter `<SmuFSPFaultStateDuration>` is wrongly written to `SMU_FSP.TFSP_LOW` bit field (which is read only) instead of `SMU_FSP.TFSP_HIGH` bit field.

Impact:

FSP signal is not generated correctly, that is `TFSP_HIGH` time is 0 instead of the user-defined duration.

Workaround:

User is required to modify the generated configuration file with the desired FSP signal duration.
Member FSPCfg in the corresponding Configset of the structure Smu_ConfigRoot, which contains the value written into the SMU_FSP register has to be modified in Smu_PBCfg.c.
Bits[31-22] which corresponds to TFSP_HIGH contain the value `<SmuFSPFaultStateDuration>`.
Bits[21-16] which corresponds to TFSP_LOW should be set to 0.

2.23.2 0000050328-342

In the EP devices (TC21x, TC22x, TC23x), during configuration of the `<SmuIdleRequest>` parameter in the `<SmuAlarmGlobalConfig>` container, invalid options appear in the drop-down list.
The following invalid options are visible: SMU_SELECT_IDLE_CPU1, SMU_SELECT_IDLE_CPU2, SMU_SELECT_IDLE_CPU0_CPU1, SMU_SELECT_IDLE_CPU0_CPU2, SMU_SELECT_IDLE_CPU1_CPU2, SMU_SELECT_IDLE_CPU0_CPU1_CPU2. These options are not valid for EP devices because the EP devices have only one core.

Impact:

Few invalid options appear in the drop-down list during the configuration of parameter <SmuIdleRequest> in the container <SmuAlarmGlobalConfig>. Selecting invalid options will have no effect during execution of SafeTlib.

Workaround:

User shall configure only valid options. The valid options for the <SmuIdleRequest> parameter are: `SMU_SELECT_IDLE_NONE` and `SMU_SELECT_IDLE_CPU0`.

2.23.3 0000050328-410

Explicit check for the 'Unspecified SMU state' not done in the `Smu_SetAlarmStatus` and `Smu_ReleaseFSP` APIs.
In the `Smu_ClearAlarmStatus` API, read back of Alarm status register is not required to confirm the clearing of alarm status when spinlock is not acquired for SMU driver before read-back operation.

Impact:

The `Smu_SetAlarmStatus` API sets the alarm status and `Smu_ReleaseFSP` API releases the FSP even when the SSM field in the `SMU_DBG` register indicates that the SMUis in the unspecified state.
Unnecessary read-back of the SMU alarm status register is performed when there is a failure in acquiring spinlock for SMU driver.

Workaround:

None

2.23.4 0000050355-5

In the case of standalone MCAL (without SafeTlib), `MemMap.h` header file has to be included instead of `Ifx_MemMap.h` file in the SMU driver module files to comply with MCAL requirement (AUTOSAR compliance).

Impact:

None. There is no impact on SafeTlib. There is no change in the hex output of SafeTlib as the code change is within the MCAL-specific compiler switch.
Workaround:
Not applicable

2.23.5 0000050328-639

In the hardware User Manual, for the `SMU_AFCNT` register, the size and position of ACNT (Alarm Counter) field is incorrectly mentioned as `SMU_AFCNT[15:8]` and bits `SMU_AFCNT[7:4]` are incorrectly mentioned as Reserved; read as 0 in the hardware User Manual.
The correct size and position of field ACNT field in register `SMU_AFCNT` is `SMU_AFCNT[15:4]`.
Impact:
None. SafeTlib is not using field ACNT of the register `SMU_AFCNT`.
Workaround:
Customer has to consider `SMU_AFCNT[15:8]` to get the actual value of ACNT in their application software.
Bit field access of ACNT shall not be used in case the application is using SFR files provied by SafeTlib.

2.23.6 0000050328-656

AoU related to GPIO P33.8 (error pin) is missing in the User Manual. When SMU is disabled, port pin P33.8 (error pin) cannot be used as GPIO or alternate function input/output.

This is due to an application hint – PORTS_TC.H006.

Impact:
When SMU is disabled, changes in the configuration of port pin P33.8 will not become effective.
Workaround:

If P33.8 shall be used as GPIO or alternate function input/output, any one of the following points has to be taken care.

- Do not disable the SMU, i.e. keep SMU_CLC = 0x0 (default after reset). In this case, the configuration of P33.8 may be changed by software at any time.
- Configure P33.8 before the SMU is disabled by software (SMU_CLC.DISR = 1B).
- After the SMU is disabled, the configuration of P33.8 can no longer be modified by software.
- If the SMU is disabled by software (SMU_CLC.DISR = 1B, i.e. not clocked), clear bit position 8 of P33_PCSR once after any reset (Application, System Reset, PORST) before configuring P33.8. Controlling P33.8 as FSP by SMU is possible only once after a reset.

Note: Write access to the register P33_PCSR is Safety ENDINIT protected.

2.23.7 0000050328-889

Explicit check of permanent lock in the `SMU_KEYS` register is not done.
Impact:
In case of failure of permanent lock of SMU (which is done `by SMU_KEYS.U = SMU_CFG_PERLCK`),
1. Temporary lock of SMU gets tested instead of permanent lock
2. `Smu_LockConfigRegs()` return success
Workaround:
None

2.23.8 0000050328-1292

SMU de-init does not initialize some of the SFRs to default reset value.
Impact:
Due to wrongly configured values during SMU deinit, there will be no reaction of Recovery Timer 0, CPU0, 1, 2, safety Watchdog Timeouts. If customer has configured some reaction for these timeouts (e.g reset, NMI), the reaction will not take place. This is only applicable if SMU could not be initialized successfully by SafeTlib through API Sl_SwitchTstPhase (). In case of successful execution of API Sl_SwitchTstPhase (), SMU de-init is followed by SMU init which configures all the registers correctly.
Workaround:
None

2.23.9 0000050328-1305

SmuFSPFaultStateDuration range starts with 1 instead of 0 in Xdm and user manual.
Impact:
User Can not configure SmuFSPFaultStateDuration to 0 using Tresos.
Workaround:
Do manual configuration for SmuFSPFaultStateDuration.
File: Smu_PBCfg.c,
Parameter: Smu_ConfigType->FSPCfg = 0

2.23.10 0000050328-1341

In Smu.c file, macro UINT8_MAX is defined as 0xFF. Standard header file stdint.h could also define the same which result in redefinition.
Impact:
If the user includes stdint.h, this redefinition generates compiler warning. The value for UINT8_MAX in SMU driver and stdint.h would be same. Hence there is no functional impact.
Workaround:
None

2.23.11 0000050328-1425

Smu.bmd contains an empty line at the beginning of the file for few devices.
Impact:
Causes invalid schema error in configuration tool like daVinci Configurator 5.
Workaround:
None

2.23.12 0000050328-1429

The following AoU is missing in the User manual.
Due to Hardware Errata SMU_TC.H013, It is recommented to use FSP Time-switching protocol only.In case FSP protocol is not used at all or used in FSP bi-stable mode, the [Application SW] shall read periodically, once per FTTI, the SMU_CLC register to react on unintended disabled SMU.
Impact:
Transient faults can possibly affect the SMU_CLC register and lead to disabling the SMU_core.
Workaround:

Use FSP Time-switching protocol only.In case FSP protocol is not used at all or used in FSP bi-stable mode, the [Application SW] shall read periodically, once per FTTI, the SMU_CLC register to react on unintended disabled SMU.

2.23.13 0000050328-1430

The following AoU is missing in the User manual.
Due to Hardware Errata SMU_TC.H014, If FSP configured for Time switching mode, during transition from START to RUN state,Enable FSP by Smu_SetupErrorPin() 10 SPB clock cycles (or more) after Smu_ReleaseFSP().
Impact:
Due to an internal synchronization issue, an unintended short pulse of a duration of around 80 ns on the FSP pins during transition from START to RUN state.
Workaround:
If FSP configured for Time switching mode, during transition from START to RUN state, Enable FSP by Smu_SetupErrorPin () 10 SPB clock cycles (or more) after Smu_ReleaseFSP ().

2.24 SMU test

2.24.1 0000050329-451

The following information is missing in the User manual.
Configuration parameter SmuFSPPrescaler1 in Smu Driver has to be configured between the values SMU_REF_CLK_FRQ_DIV_2 and SMU_REF_CLK_FRQ_DIV_64 in the SmuConfigSet used for SafeTlib pre-run test
Impact:
If the configuration parameter `SmuFSPPrescaler1` is configured with the values `SMU_REF_CLK_FRQ_DIV_128/SMU_REF_CLK_FRQ_DIV_256`, the `SmuTst_RtTst` fails with the error code `SMUTST_RT_TIMEOUT_ALMNOTSETERR`.
Workaround:
User shall ensure that the configuration parameter `SmuFSPPrescaler1` in the SMU driver is configured between the values `SMU_REF_CLK_FRQ_DIV_2` and `SMU_REF_CLK_FRQ_DIV_64` in the `SmuConfigSet` used for SafeTlib pre-run test.

2.24.2 0000050328-1419

SMU test fails for lower frequencies (Ex: fsri=10 MHz and fspb=10 MHz)
Impact:
SMU test fails with `SMUTST_RT_TIMEOUT_RTRUNERR` error code.
Workaround:
Change the SMU configuration by adjusting the FSMU_FS (Ex: prescalar1 (PRE1) value 1 or above) for the SafeTlib pre run test.

2.25 SMU Driver/ SMU test

2.25.1 0000050328-1334

Due to a synchronization issue, ALM3 [27] is sporadically triggered if the PRE2 field of register FSP is written while the SMU is configured in Time Switching protocol (FSP.MODE = 10B) and FSP [0] is toggling with a defined TSMU_FFS period. Also, ALM3[27] is sporadically triggered if the PRE1 or TFSP_HIGH fields of register FSP are written while the SMU is in the Fault State and TFSP_FS has not yet been reached (STS.FSTS=0B) (regardless of the FSP.MODE configuration).In addition, an unexpected ALM2[29] or ALM2[30] is sporadically triggered if field FSP.PRE1 or RTC.RTD is written, and at least one recovery timer is running based on a defined TSMU_FS period (regardless of the FSP.MODE configuration).The alarms can only be cleared with cold or warm Power-On reset. (Errata SMU_TC.012)

Impact:

Unexpected SMU alarms while writing to SMU_FSP or SMU_RTC register.

Workaround:

None

2.26 SPB Register Access Protection Test

2.26.1 0000050328-683

Memory sections are not applied on the following configuration variables: <Rmc_Not_Available> and <Rmc_Available>.

These variables should have been placed in the memory section, `DEFAULT_RAM_32BIT_NONZERO_INIT`.

Impact:

The Linker in the customer build environment can place the variables <Rmc_Not_Available> and <Rmc_Available> outside the SafeTlib used memory sections.

If memory protection is enabled for memory sections not used by SafeTlib, then SPB Register Access Protection Test leads to trap.

Note: All SafeTlib memory sections are defined in the `Ifx_MemMap.h` file.

Workaround:

Customer shall place the variables `Rmc_Not_Available`' and `Rmc_Available` in the `DEFAULT_RAM_32BIT_NONZERO_INIT` memory section.

2.27 SPB Timeout Test

2.27.1 0000050328-655

Due to errata MTU_TC.005, MBIST mode or module kernel clock has to be enabled for the access to MCx_ECCD.

Impact:

When MBIST is disabled or module kernel clock is not enabled, there can be sporadic cases wherein value read/written from/to MCx_ECCD registers are wrong or sporadic traps can be observed.

Workaround:

Run SPB test only from core 0.

2.27.2 0000050328-774

Error capture registers, `SBCU_EADD` and `SBCU_EDAT`, are not checked for the expected values when an SPB timeout error is triggered during the test.

Impact:

SPB timeout error triggered during the test is validated by checking one of the error capture register `SBCU_ECON` and checking occurrence of SMU alarm. Hence, there is no safety impact with respect to SPB timeout error report mechanism.

The data captured in the `SBCU_EADD` and `SBSU_EDAT` registers during SPB timeout error are not validated.

Workaround:

None

2.28 SPB Register Access Protection Test / SPB Timeout Test

2.28.1 0000050328-406

When configuration of SMU alarm action results in a failure in the restore function, unregistering of trap handler for trap class 4 is not done.

Impact:

Unexpected class 4 traps occurring after the execution of SPB test are not forwarded to the application trap handler.

Workaround:

None

2.28.2 0000050328-473 / 0000050328-530

The following registers are modified by the test and not restored:

- `XBAR_ARBCON6`
- `XBAR_ARBCOND`
- `XBAR_INTSAT`
- `SRC_BCUSPBSBSRC`
- `SBCU_ECON`
- `SRC_XBARSRC`

Impact:

XBAR_ARBCON6, XBAR_ARBCOND:

Interrupt acknowledgement bit fields (XBAR_ARBCON6.INTACK and XBAR_ARBCOND.INTACK) are set during the execution of the SPB Register Access Protection Test / SPB Timeout Test indicating that the interrupt acknowledgment of SRI arbiter error is pending.
If the handling of this SRI arbiter error (by interrupt routine or through polling the INTACK bit) will lead (ultimately and unconditionally) to a device reset, the problem will be easily detected during development (as the program would always reset during startup). This is the typical use case for SRI error handling, therefore the expected probability for safety impact is very low.
In case the SRI arbiter error is ignored, the only effect would be that the ERRADDR and ERR registered stay locked and are not updated with the information of later error on the same arbiter. This should not have any safety impact (if an arbiter error can have safety impact, the application can anyway not simply ignore it).

XBAR_INTSAT:
Bit fields PRSCI6 and PRSCID are set in the SRI arbiter interrupt status register (XBAR_INTSAT). It is set during the execution of the SPB Register Access Protection Test / SPB Timeout Test indicating a protocol error from SCI6 and SCID (sSlave connection interfaces) has occurred and is yet to be handled/acknowledged. This does not have any impact. The bit fields PRSCI6 and PRSCID have to be ignored.

SRC_BCUSPBSBSRC:
SRC_BCUSPBSBSRC.SRR and sometimes SRC_BCUSPBSBSRC.IOV bits are set during the execution of the test. If after the test the interrupt source is enabled (by setting the SRC_BCUSPBSBSRC.SRE bit), the pending service request takes part in the interrupt arbitration and a false interrupt event will get triggered.
If in the interrupt routine the plausibility of the interrupt is checked (as required by the safety manual), no reason would be found. Therefore the wrong interrupt could be detected. A plausibility check in case of an SPB service interrupt would for example be to check SBCU_ECON.ERRCNT is greater than zero).

In any case, if the handling of the interrupt in the end would lead (ultimately and unconditionally) to a device reset, the problem would be easily detected during development (as the program would always reset during startup). This is the typical use case for SPB errors, therefore the expected probability for safety impact is very low.

SRC_XBARSRC:
SRC_XBARSRC.SRR and sometimes SRC_XBARSRC.IOV bits are set during the execution of the test.
If after the test the interrupt source is enabled (by setting SRC_XBARSRC.SRE bit), the pending service request takes part in the interrupt arbitration and a false interrupt event will get triggered.

Workaround:
XBAR_ARBCON6, XBAR_ARBCOND:
Clear the XBAR_ARBCON6.INTACK and XBAR_ARBCON6.INTACK bits by writing 1 to it after running the pre-run test of SafeTlib.

XBAR_INTSAT:
Clear XBAR_INTSAT.PRSCI6 and XBAR_INTSAT.PRSCID bit fields by writing 1 to it after running the pre-run test of SafeTlib.

SRC_BCUSPBSBSRC:
Clear the service request (by setting SRC_BCUSPBSBSRC.CLRR) and the interrupt overflow flag (by setting SRC_BCUSPBSBSRC.IOVCLR) after running the prerun test of SafeTlib, but before activating the service request (SRC_BCUSPBSBSRC.SRE)

SRC_XBARSRC:
Clear the service requests (by setting SRC_XBARSRC.CLRR) and the interrupt overflow flag (by setting SRC_XBARSRC.IOVCLR) after running the test, but before activating the service request (SRC_XBARSRC.SRE)

2.28.3 0000050328-1395

The following AoU is missing in the User manual.User must ensure that there are no concurrent SPB accesses by other SPB bus masters [Eg: CPU, DMA, HSM, Debugger, CAN, HSSL...].
Impact:
Concurrent SPB accesses by other SPB bus masters during SPB test can lead to Bus error.
Workaround:
User must ensure that there are no concurrent SPB accesses by other SPB bus masters [Eg: CPU, DMA, HSM, Debugger, CAN, HSSL...].

2.29 SRAM ECC Test

2.29.1 0000050329-205

During the initialization of the memory tests for the SRAM, if the read operation fails during the memory backup procedure, the backed up data will not get restored.
Impact:
The memory selected to be tested will get corrupted by the SRAM ECC test.
Workaround:
The tested memory areas (as per the configuration) need to be backed up by the user before the SafeTlib tests and is to be restored after the pre-run tests.

2.29.2 0000050329-206

The RANGE register is being used to run the memory tests on dedicated parts of the RAM. In SRAM ECC test, backup and restore of the MBIST RANGE register is not being done.
Impact:
Subsequent MBIST tests performed by the application may not test the full SRAM memory as the RANGE register is not restored.
Workaround:
The RANGE register needs to be backed up by the user before the pre-run tests of SafeTlib and has to be restored its completion.

2.29.3 0000050329-216

Inconsistent error codes are returned in case of correctable, uncorrectable, EOV errors.
Impact:
No functional impact. It will only lead to wrong error code reporting. The error detection would still work.
Workaround:
Since the error codes are inconistent, the user should not use the error codes to make decisions. The user should only take into consideration that an error has resulted.

2.29.4 0000050328-204

Optimization is not correctly restored for the Hightec GNU and the Windriver Diab compilers. IFX_SRAM_STOP_SEC_CODE_NO_OPTIMIZED_ASIL_B is not defined.
Impact:
This issue can have an impact in the optimization settings of the compilation process. The optimization of the Hightec GNU and Windriver Diab compilers will be affected (only for the TC21x, TC22x, TC23x and TC24x devices). The optimization of the code remains turned off at the end of the file and will affect the optimization of the next files getting compiled, unless overridden.
Workaround:
The impact is for the code which is getting compiled after the SramEccTst.c file. So, if specific optimization is required for the code which follows SRAM code, user shall take care. If not, the optimizations would be turned off.

2.29.5 0000050329-343

The following recommendation from the safety manual is not implemented:
It is recommended to write the RDBFL registers of each SRAM instance with an invalid data pattern of data and ECC, for example All-0, before starting the safety related application software. This is to avoid the silent corruption of safety related data during application runtime, due to a failure mode of the MBIST for which the content of the RDBFL registers is written accidentally in memory.
Impact:
It can result in the corruption of safety related data during application runtime, due to a failure mode of the MBIST.
Workaround:
After the execution of SafeTLib prerun tests, the user should set the RDBFL register with data having incorrect ECC bits. Refer the microcontroller user manual for details.

2.29.6 0000050329-372

PIETR.IED bit remains set after the execution of SRAM ECC test.
Impact:

No functional impact for SafeTlib Test, but will impact the subsequent error detections.
The PIETR->IED bit indicates whether a program integrity error is detected or not. This bit gets set when an integrity error is detected. But, another integrity error can be logged if this bit is not set. In this issue, this bit remains set at the end of the SRAM tests. So, the recording of error information (in PIETR and PIEAR registers) for next occurring program integrity error is inhibited for application. The corresponding traps also would be disabled.
Workaround:
The user shall clear PIETR.IED bit after the execution of pre-run test of SafeTlib.

2.29.7 0000050329-385

ECCD register flags are not restored after SRAM test.
Impact:
Further ECC errors will not be detected. Memories affected are: CPU 1/2 DSPRs and CPU1 PTAG.
Workaround:
The ECCD flags need to be cleared by the user.
For CPU1 and CPU2 DSPRs, the ECCD flags can be cleared in the user code.
For CPU1 PTAG, enable auto RAM init in FLASH0_PROCOND register in the user code. Refer to the microcontroller User Manual for more details.

2.29.8 0000050329-394

The following registers are modified by the test and are not restored:
XBAR_ARBCON4.INTACK
XBAR_INTSAT.PRSCI4
LMU_MEMCON.RMWERR

Impact:
XBAR_ARBCON4.INTACK:
Interrupt acknowledgement bit field (XBAR_ARBCON4.INTACK) is set during the execution of the test, indicating that the interrupt acknowledgment of SRI arbiter error is pending.
If the handling of this SRI arbiter error (by interrupt routine or through polling the INTACK bit) will lead (ultimately and unconditionally) to a device reset, the problem will be easily detected during development (as the program would always reset during startup). This is the typical use case for SRI error handling, therefore the expected probability for safety impact is very low.
In case the SRI arbiter error is ignored, the only effect would be that the ERRADDR and ERR registered stay locked and are not updated with the information of later error on the same arbiter. This should not have any safety impact (if an arbiter error can have safety impact, the application can anyway not simply ignore it).

XBAR_INTSAT.PRSCI4:
Bit field PRSCI4 is set in the SRI arbiter interrupt status register (XBAR_INTSAT.PRSCI4). It is set during the execution of the LMU Bus MPU Test / LMU Register Access Protection Test, indicating a protocol error from SCI4 (Slave connection interface 4) has occurred and is yet to be handled/acknowledged. This does not have any impact. The bit field PRSCI4 has to be ignored.

LMU_MEMCON.RMWERR:
No impact Indicates Internal Read Modify Write Error which is set during the execution of the test.

Workaround:
XBAR_ARBCON4.INTACK:

Clear the XBAR_ARBCON4.INTACK bit by writing 1 to it after running the prerun test of the SafeTlib.

XBAR_INTSAT.PRSCI4:

Clear the XBAR_INTSAT.PRSCI4 bit by writing 1 to it after running the pre-run test of the SafeTlib.

LMU_MEMCON.RMWERR:
Clear the LMU_MEMCON.RMWERR bit.

2.29.9 0000050328-671

During the SRAM clearing process, the MBIST Enable status is not checked before accessing MBIST registers.
Impact:
If there is a hardware error (which causes a failure in MBIST Enable Logic), an unexpected Class 4 Tin3 trap is raised while clearing SRAM before exiting the SRAM test.
Workaround:
None. Peripheral SRAM test is supported only on devices without functional deviation, MTU_TC.008, as mentioned in the User manual.

2.29.10 0000050328-708

After execution of memory test for CPU memory, the original status has to be restored. As a part of this restoration, the MBIST mode is also disabled which ensures further CPU RAM accesses are possible. In case MBIST mode disabling fails, an access to CPU memory will cause software to hang (undefined state) since CPU gets stalled.

Impact:
The software cannot go ahead as the CPU is stalled. But, this is not a Safety critical issue because the system enters the safe state at the startup itself. The test is executed in Pre-run phase and the RUN phase is activated after the successful execution of all Pre-run tests only.
Workaround:
None

2.29.11 0000050328-1285

Inside SramEccTst_lInitSingleCore (), SramEccTst_lResetSafetyENDINIT () is called under a condition while SramEccTst_lSetSafetyENDINIT () called unconditionally.

Impact:
Uninitialised access to Structure WdtContext when the result of "SramEccTst_lInitSmuAlarms (SRAM_CPU0_DSPR, BackupData);" is not SRAMECC_SUCCESS could lead to trap.
Workaround:
None

2.29.12 0000050328-1299

Refer errata MTU_TC.005: Access to MCx_ECCD and MCx_ETTRi while MBIST disabled.On multicore systems, During SramEccTst all CPUs except the one executing the test must be in HALT or IDLE state. After reset, For CPU memories the clock is enabled until the CPU is explicitly put to IDLE mode by software.

Impact:

When MBIST is disabled and CPU is in IDLE state [clock disabled], upon concurrent SPB access by other Bus masters there can be sporadic cases where read/write access of MCx_ECCD can cause a Trap.

Workaround:

User must ensure that there are no concurrent SPB accesses by other SPB bus masters [Eg: CPU, DMA, HSM, Debugger, CAN, HSSL...].

2.29.13 0000050328-1315

The following AoU is missing in the User manual.

User must ensure that Address Error Notification ECCD.AENE should be enabled during SRAM ECC test. [Can be disabled after test. See Errata workaround CPU_TC.125]

Impact:

SRAM ECC test fails if Address Error Notification ECCD.AENE is disabled during the test.

Workaround:

User must ensure that Address Error Notification ECCD.AENE should be enabled during SRAM ECC test. [Can be disabled after test. See Errata workaround CPU_TC.125]

2.30 SRI EDC and Bus Error Test

2.30.1 0000050329-174

Test crashes at unknown program location when SMU alarm action is configured to issue a reset for any alarm.

Impact:

Sporadic and unexpected resets were observed and SRI EDC and Bus error test fails.

Workaround:

The user shall ensure that the SMU alarms are not configured to issue a reset during the SafeTlib execution.

2.30.2 0000050329-211

XBAR_ARBCONx, PIETR and DIETR registers are not restored at the end of the SRI EDC and Bus Error test.

Impact:

Logging of further errors in PIETR, DIETR and XBAR_ARBCONx is not possible.

Workaround:

PIETR, DIETR, XBAR_ARBCONx registers of respective core should be cleared by the user after the execution of SafeTlib pre-run tests.

2.30.3 0000050350-1

DSTR register not cleared at the end of the test.

Impact:

Logging of further errors in DSTR, DEADD is not possible.

Workaround:

DSTR registers of respective core should be cleared by the user after the execution of SafeTlib pre-run tests.

2.30.4 0000050328-97

Issue1:

SriTst uses the Overlay registers but assumes that they contain reset values when setting up the overlay for use within the test.

Impact:

In case overlay is configured by the user application before the SRI test (OVC_CON.CSEL, RABARx, OMASKx, OTARx), it will interfere with the test possibly leading to its failure.
Workaround:
Customer should not use overlay feature till end of pre-run phase of SafeTlib (OVC_CON.CSELx, RABRx, OMASKx, OTARx should contain RESET values)

Issue 2:

If LMUTestEnCpuX, PMUPFlashTestEnCpuX, PMUDFlashTestEnCpuX and DMATestEnCpuX[X=0,1,2,] are disabled in the Tresos and only XBARTestEnCpuX configuration is enabled, SRI EDC and Bus Error Test will not test XBAR.

Impact:

XBAR slave decoders will not be tested even though it is enabled in the configuration.
Workaround:
Customer should enable at least one of the configurations LMUTestEnCpuXPMUPFlashTestEnCpuX, PMUDFlashTestEnCpuX, or DMATestEnCpuX along with XBARTestEnCpuX[X=0, 1, 2].

2.30.5 0000050350-4

SRI EDC and Bus Error test does not release SPB error capture lock in the local function SriTst_lUser1AccTst. The test sets SBCU_EDAT, SBCU_EADD and SBCU_ECON registers. SBCU_ECON registers is not read back, thereby the recording of next application related real SBCU bus error is inhibited.
Impact:
If more than one FPI Bus transactions generate a bus error, only the first bus error information is captured for application relevant data [SBCU_EDAT, SBCU_EADD].
Workaround:
Read SBCU_ECON after the SafeTlib pre-run test execution to release the error logging lock.

2.30.6 0000050328-260

Backup of DSPR Memory used for DMA destination address is not correctly implemented.
Comment: Dedicated area used for SRI test.
Impact:
Intended backup does not work as expected. No functional impact as the used data area for the SRI test shall be reserved anyway and a backup would not be required.
Workaround:
None.

2.30.7 0000050328-446

LMU_MEMCON is not cleared at the end of SRI EDC and Bus Error test.
Impact:
ADDERR, DATAERR and INTERR flags will prevent the subsequent errors from getting logged.
Workaround:
User shall clear LMU_MEMCON [ADDERR, DATAERR and INTERR] bits after SriTst execution.

2.30.8 0000050328-454

ECCD register error log is not cleared after SRI test.
Impact:
ECCD will not log subsequent errors.

Workaround:
SRI EDC and Bus Error test must be executed after SRAM ECC test because SRAM ECC test clears ECCD error flags.

2.30.9 0000050328-524

The following registers are not restored:
DMA_ME1CHSR
DMA_CHCSR000
SRC_XBARSRC
DMA_SDCRCR
DMA_RDCRCR
SRC_BCUSPBSBSRC

Impact:
DMA_CHCSR000:
DMA_CHCSRz.ICH bit is set during the execution of the test. If a channel pattern detection interrupt service request or a channel source and destination wrap buffer interrupt service request is generated after execution of the SRI EDC and Bus Error Test, also an additional channel transfer interrupt service request will wrongly be indicated interrupt service routine of the user application.
This might lead to the user application wrongly assuming that new data has been copied to the destination buffer. In this case there is a risk that old data/wrong data are used.

SRC_XBARSRC:
SRC_XBARSRC.SRR and sometimes SRC_XBARSRC.IOV bits are set during the execution of the test.
If after the test the interrupt source is enabled (by setting the SRC_XBARSRC.SRE bit), the pending service request takes part in the interrupt arbitration and a false interrupt event will get triggered.

DMA_SDCRCR, DMA_RDCRCR:
If the CRC registers are used afterwards without (re-)initializing them properly, a wrong checksum will be calculated. This will be easily detected (also during development) and should lead to a proper error handling by the application like for any CRC error. Therefore no safety impact is to be expected.

SRC_BCUSPBSBSRC:
SRC_BCUSPBSBSRC.SRR and sometimes SRC_BCUSPBSBSRC.IOV bits are set during the execution of the test.If after the test the interrupt source is enabled (by setting the SRC_BCUSPBSBSRC.SRE bit), the pending service request takes part in the interrupt arbitration and a false interrupt event will get triggered.
If in the interrupt routine the plausibility of the interrupt is checked (as required by the safety manual), no reason would be found. Therefore the wrong interrupt could be detected. A plausibility check in case of an SPB service interrupt would for example be to check SBCU_ECON.ERRCNT is greater than zero).
In any case, if the handling of the interrupt in the end would lead (ultimately and unconditionally) to a device reset, the problem would be easily detected during development (as the program would always reset during startup). This is the typical use case for SPB errors, therefore the expected probability for safety impact is very low.

DMA_ME1CHSR:
This is a status register of DMA move engine. This contains the status of previous DMA action. This value has to be ignored by the application.

Workaround:
DMA_CHCSRxxx, DMA_ME1CHSR:

The DMA channel should be reset by setting the TSRz.RST bit after running the prerun test of SafeTlib.

SRC_XBARSRC:
Clear the service requests (by setting SRC_XBARSRC.CLRR) and the interrupt overflow flag (by setting SRC_XBARSRC.IOVCLR) after running the prerun test of SafeTlib, but before activating the service request (SRC_XBARSRC.SRE)

DMA_SDCRCR, DMA_RDCRCR:
(Re-)Initialize DMA_SDCRCRz and DMA_RDCRCRz properly after running the prerun test of SafeTlib.

SRC_BCUSPBSBSRC:
Clear the service request (by setting SRC_BCUSPBSBSRC.CLRR) and the interrupt overflow flag (by setting SRC_BCUSPBSBSRC.IOVCLR) after running the prerun test of SafeTlib, but before activating the service request (SRC_BCUSPBSBSRC.SRE).

2.30.10 0000050328-582

In the Hightec GNU compiler, Inline attribute does not force function to be inlined
Impact:
Unexpected traps during SRI EDC and Bus Error test execution are not redirected to application trap handler. On occurrence of unexpected trap, control goes in infinite unexpected trap handling loop because of CTYP context management trap and will not exit until external intervention.
Workaround:
None

2.30.11 0000050328-606

Incorrect register (SCU_ACCEN1) is used to generate bus error during CPU user-mode and Invalid opcode test. SCU module is SPB slave and unprivileged access generates SPB bus error.
Impact:
SRI bus error for user mode access protection and unsupported opcode is not tested.
Workaround:
None

2.30.12 0000050328-607

XBAR_ARBCON9-10 [INTACK bit] is not cleared, if PFlash bank 2 and 3 memory addresses are used for PMUPFlashTestEnCpuX[].
Impact:
Application will not get updated information in error logging registers [XBAR_ERRADDR, XBAR_ERRx] for PFlash2/3 memory addresses SRI faults.
Workaround:
- Do not configure PF2 and PF3 addresses to validate Pflash memory test for Xbar faults.
- Clear XBAR_ARBCON[9/10].INTACK bit by writing '1' into it after the execution of SafeTlib pre-run tests.

2.30.13 0000050328-624

SRI EDC and Error test is using SFRs connected over SPB causing SPB bus alarm (ALM3[31]) to be set as consequence of error injection. This alarm should be in cleared state to begin the validation of error injection.
Impact:
If there is any pending SPB alarm ALM3[31], SRI test will fail.

Workaround:
Clear SMU Alarm (ALM3 [31]) after SMU initialization and before pre-run test execution of SafeTLib to ensure it will not impact the execution of SRI EDC and error test. User has to take necessary action if alarm3[31] is already raised.

2.30.14 0000050328-625

Save/restore of DSPR memory locations when executing the DMA test (which is enabled by configuration parameter DMATestEnCPU0/1/2) is incorrect.
Impact:
The address used for save/restore of DSPR memory is not correct. Unintended DSPR memory location is backed up and is restored. DSPR memory location of uint32 size at [SriTstDSPRTstAdrCpuX + 4 bytes(X=0-2)] is changed during the execution and is not restored.
Workaround:
Use DSPR location of uint32 size at [SriTstDSPRTstAdrCpuX + 4 bytes(X=0-2)] only after SRI EDC and Bus Error test execution.

2.30.15 0000050328-649

Due to errata MTU_TC.005, MBIST mode or module kernel clock has to be enabled for the access to MCx_ECCD.
Impact:
When MBIST is disabled or module kernel clock is not enabled, there can be sporadic cases wherein value read/written from/to MCx_ECCD registers are wrong or sporadic traps can be observed.
Workaround:
None

2.30.16 0000050328-650

PMUPFlashDecoderTstX is testing only the PF0 Address range due to the limitation in tresos configuration. (Selection of other flash bank addresses is not supported in configuration.)
Impact:
PFlash tests for all the PFlash banks cannot be performed.
Workaround:
None

2.30.17 0000050328-663

SMU alarm error code (SRI_ECC_ADDRESS) is overwritten by Trap error code (SRI_ECC_ADRBTRAP_NG) during SRI decoder tests.
Impact:
Application software will receive trap error code (SRI_ECC_ADRBTRAP_NG) even if the SriTst fails due to no SMU alarm is not raised for address decoder ECC errors.
Workaround:
None

2.30.18 0000050328-696

SRC_XBARSRC register is not restored correctly after Sri EDC and Bus error Test execution due to HW [X-BAR] behavior deviation.

Impact:
If DFLASH module PMUDFlashTestEnCpuX and XBAR module XBARTestEnCpuX is disabled in Sri EDC and Bus error test configuration, SRC_XBARSRC.SRR and sometimes SRC_XBARSRC.IOV bits are set during the execution

of the test. If after the test the interrupt source is enabled (by setting the SRC_XBARSRC.SRE bit), the pending service request takes part in the interrupt arbitration and a false interrupt event will get triggered.
Workaround:
Clear the service request (by setting SRC_XBARSRC.CLRR) and the interrupt overflow flag (by setting SRC_XBARSRC.IOVCLR) after running the prerun test of SafeTlib, but before activating the service request (SRC_XBARSRC.SRE).

2.30.19 0000050328-884

The SRI EDC and Bus Error test fails, if MCAL_WdgLib data variables under section DEFAULT_START_SEC_VAR_NONZERO_INIT_32BIT/DEFAULT_STOP_SEC_VAR_NONZERO_INIT_32BIT in Ifx_MemMap.h are located in PSPR.
Impact:
The location of data variables under the section DEFAULT_START_SEC_VAR_NONZERO_INIT_32BIT to PSPR causes the system to issue Data Asynchronous Trap.
Workaround:
The variables marked by the following sections in Ifx_MemMap.h shall be placed in DSPR-DEFAULT_START_SEC_VAR_NONZERO_INIT_32BIT / DEFAULT_STOP_SEC_VAR_NONZERO_INIT_32BIT.

2.30.20 0000050328-1067

SRI test relies on default value of code access cacheability (code access cacheability off) for DSPR
Impact:
SRI test fails if code access cacheability is enabled for DSPR.
Workaround:
Disable Code access cacheability for DSPR by setting bit field PMA1.CAC bits 5,6,7 and 13 to 0

2.30.21 0000050328-1432

If the SRI EDC and Bus Error test configuration contains the default reference to test the SRI EDC decoders related to DFLASH, any uncorrectable errors at the configured location (Ex: 0xAF000000) will cause a trap, and the SafeTlib test will fail. This is particularly true if EEPROM emulation is running on DFLASH 0 and a power loss happens during an erase operation.
Impact:
A DSE Trap will occur if SafeTlib reads from a location with uncorrectable errors in DFLASH. The FLS driver disables traps for reading uncorrectable DFLASH errors by setting MARD.TRAPDIS, but most applications execute SafeTlib before FLS.
Workaround:
Implement one of these two workarounds:
#1: Remove the SRI EDC decoders test related to DFLASH from the SafeTlib Tresos configuration
#2: Set MARD.TRAPDIS before calling the SRI EDC and Bus Error test

2.31 Test Handler

2.31.1 0000050329-75

Tresos does not report an error if a wrong paramset index of the SafeTlib test is specified in TestHandler configuration.
Impact:

There is no safety impact due to this issue, since the respective SafeTlib test reports failure if wrong paramset (out of range) index is used.
Workaround:
Not applicable.

2.31.2 0000050329-148

Tresos does not report an error if tests pertaining to a different core is used in the TestHandler.
Impact:
There is no impact due to this issue, since an intented SafeTlib test will be called eventhough the SafeTlib tests pertaining to other cores is configured.
Workaround:
Not applicable.

2.31.3 0000050329-158

ErrorPin Handling: Error Pin (FSP) initialization is missing in test handler.
Impact:
Safety path is not enabled.
Workaround:
Call the `Smu_ReleaseFSP()` as shown below after the `Sl_PreInit()` in the application code.

```
********
Sl_PreInit(CfgPtr);
Smu_StateType SmuState = Smu_GetSmuState();
if (SmuState == SMU_START_STATE)
{
  Result = Smu_ReleaseFSP();
}
Sl_Init(CfgPtr);
********
```

2.31.4 0000050328-25

Missing AoU that Sl_PreInit should be called only on the master core.
Impact:
Possibility of corruption of spinlocks if Sl_PreInit is invoked on the slave cores leading to incorrect test result.
Workaround:
Sl_PreInit should be called only by the master core.

2.31.5 0000050328-37

Missing AoU that Sl_Init should be called only after all the early pre-run tests are finished.
Impact:
No impact. If Sl_Init is called before the initialization of early pre-run test, it returns a failure.
Workaround:
Customer shall ensure that Sl_Init is called after all early prerun test in the master core are completed.

2.31.6 0000050328-1340

In TstHandler_PBCfg.c, IFX_TSTHANDLER_START_SEC_POSTBUILDCFG_ASIL_B is defined before inclusion of "TstHandler.h". This creates nested START Sections since TstHandler adds another START section.
Impact:

Map file shows that the variable is located in correct section despite having nested START definitions. Hence there is no functional impact.
Workaround:
None

2.32 Trap test

2.32.1 0000050329-260

In case of an unexpected trap during the execution of the Trap Test, the software jumps to an undefined address. This is caused because of wrong address calculation.
Impact:
Invalid program flow.
Workaround:
None

2.32.2 0000050329-261

In trap 4 class handler, the return address from the trap is not updated in A11 register when there is a mismatch in the stored and calculated CRC.
Note: To ensure the integrity of trapcounters, CRC is maintained for the trapcounters of traps of all class.
Impact:
If there is CRC mismatch in the trap 4 class handler, it will result in the infinite loop of trap occurrences.
Workaround:
None

2.32.3 0000050329-266

DSTR register not cleared at the end of the test.
Impact:
If DSTR register is not cleared, DSTR register is not updated for the further occurring traps. Therefore for the new traps, trap source information to aid the localization of faults is not possible. Since the traps will continue occur as expected for the error scenarios, there will be no safety impact.
Workaround:
DSTR registers of all the cores shall be cleared by the user after execution of prerun test of SafeTlib.

2.32.4 0000050328-394

TrapTst multicore parallel execution fails due to optimization of trap_TrapHandlers variable access by HighTec GNU Compiler.
This issue is not observed when TrapTst is executed on a core with other cores halted.
Impact:
Trap test fails when the other cores are running.
Workaround:
TrapTst shall be executed on a core with other cores halted.

2.32.5 0000050328-476 / 0000050328-522

The following registers are modified by the test and not restored:
XBAR_ARBCOND
XBAR_INTSAT

SRC_XBARSRC

Impact:
XBAR_ARBCOND:
Interrupt acknowledgement bit field (XBAR_ARBCOND.INTACK) is set during the execution of the Trap Test, indicating that the interrupt acknowledgment of SRI arbiter error is pending.
If the handling of this SRI arbiter error (by interrupt routine or through polling the INTACK bit) will lead (ultimately and unconditionally) to a device reset, the problem will be easily detected during development (as the program would always reset during startup). This is the typical use case for SRI error handling, therefore the expected probability for safety impact is very low.
In case the SRI arbiter error is ignored, the only effect would be that the ERRADDR and ERR registered stay locked and are not updated with the information of later error on the same arbiter. This should not have any safety impact (if an arbiter error can have safety impact, the application can anyway not simply ignore it).

XBAR_INTSAT:
Bit fields PRSCID is set in the SRI arbiter interrupt status register (XBAR_INTSAT). It is set during the execution of the SPB Register Access Protection Test / SPB Timeout Test, indicating a protocol error from SCID (Slave connection interfaces) has occurred and is yet to be handled/acknowledged. This does not have any impact. The bit field PRSCID has to be ignored

SRC_XBARSRC:
SRC_XBARSRC.SRR and sometimes SRC_XBARSRC.IOV bits are set during the execution of the test.
If after the test the interrupt source is enabled (by setting SRC_XBARSRC.SRE bit), the pending service request takes part in the interrupt arbitration and a false interrupt event will get triggered.

Workaround:
XBAR_ARBCOND:
Clear the XBAR_ARBCOND.INTACK bit by writing 1 to it after running the test.

XBAR_INTSAT:
Clear XBAR_INTSAT.PRSCID bit field by writing 1 to it after running the prerun test of SafeTlib.

SRC_XBARSRC:
Clear the service requests (by setting SRC_XBARSRC.CLRR) and the interrupt overflow flag (by setting SRC_XBARSRC.IOVCLR) after running the prerun test of SafeTlib, but before activating the service request (SRC_XBARSRC.SRE)

2.32.6 0000050328-698

Trap test not designed to run in parallel on multiple cores. There is no synchronization mechanism to acess the register SCU_TRAPDIS and few local variables in TrapTst.c
Impact:
Trap test fails when it is called concurrently from all the cores.
Workaround:
TrapTst shall be executed on a core with other cores halted.

2.33 Trap handling modules

Following modules have trap handling:

CPU MPU Test

LMU Register Access Protection Test
SMU Test
SPB Timeout Test
SPB Register Access Protection Test
SRI EDC and Bus Error Test
Trap Test
CPU Watchdog test and Safety Watchdog test

2.33.1 0000050328-141

Trap handlers do not properly support following corner case with unexpected traps.

- The occurrence of an unexpected trap of same TIN and Class will be counted as expected trap.
- If there is unexpected trap of the same class but different TIN, then they are ignored.

Impact:
Unexpected traps may not be forwarded to the application trap handler. Moreover the error clearing of trap related registers like DIETR, PIETR for class 4 traps is not consistent across all modules.
Workaround:
Customer is recommended to clear trap related registers DIETR, PIETR, PSTR, DSTR, DATR after the execution of SafeTlib prerun test by calling MTCR(DIETR, 0x00), MTCR (PIETR, 0x00) and so on..

2.33.2 0000050328-590

For DEFAULT_MTL_START_SEC_TVT_CODE memory section, pre-compiler check is done with

1) _TASKING_C_TRICORE_ instead of __TASKING__
2) _GNU_C_TRICORE_ instead of __GNUC__

Impact:
No safety impact. It is not possible to set _TASKING_C_TRICORE_ and _GNU_C_TRICORE_ at the same time.
Workaround:
Not applicable

2.33.3 0000050328-253

Alignment of Trap Table - Stl_TrapVectors is not done for Hightec GNU / Windriver Diab compilers.
Impact:
The software jumps to an undefined address since trap table is not aligned.
Workaround:
Application Software will have to ensure that Trap vector table is aligned to 256 bytes by specifying alignment in MemMap as shown below

```
#elif defined (DEFAULT_MTL_START_SEC_TVT_CODE)
   #undef       DEFAULT_MTL_START_SEC_TVT_CODE
   #ifdef __TASKING__
   #pragma section code "Mtl_TrapTab.Stl_TrapVectors"
   #elif defined __GNUC__
   #pragma section ".text.Mtl_TrapTab.Stl_TrapVectors" ax 256
   #elif defined _DIABDATA_C_TRICORE_
   #if (_DIABDATA_C_TRICORE_ == 1U)
   #pragma section CODE ".text.Mtl_TrapTab.Stl_TrapVectors" X
   #pragma align_functions 256
   #endif /* #if (_DIABDATA_C_TRICORE_ == 1U) */
```

```
#endif
```

3 Changes planned in future releases

This section is not applicable for this release.

4 Release-specific changes in the User Manual

The following table summarizes the updates made to the User Manual for the existing issues against respective releases. For more information on the affected release numbers, refer to section 4.1 Related Issue Overview.

Description	Applicable releases	Related issue IDs
Added SM_AURIX_STL_159	513 513.1 514 514.1 652	0000050328-675
Return values for the following APIs is updated: • `SafeWdgInt_Init` • `SafeWdgInt_Activate` • `SafeWdgInt_Service` • `SafeWdgInt_GetSeed` The return value `E_OK` is replaced by `SWDG_JOB_SUCCESS` and `E_NOT_OK` is replaced by `SWDG_JOB_INV_PARAM` / `SWDG_JOB_INV_STATE`.	513 513.1 514 514.1 652	0000050328-771 0000050328-741
Return values for the following APIs is updated: • `SafeWdgExtTlf_Init` • `SafeWdgExtTlf_DeInit` • `SafeWdgExtTlf_Activate` • `SafeWdgExtTlf_Service` • `SafeWdgExtTlf_GetWdgInfo` • `SafeWdgExtTlf_GetSeed` • `SafeWdgExtTlf_GetErrCntr` • `SafeWdgExtTlf_UserRequest` • `SafeWdgExtTlf_GetJobResult` Additonal error codes are returned to indicate failures as against earlier releases where single error code was returned for multiple failures (`SWDG_JOB_FAILED`).	513 513.1 514 514.1 652	0000050328-771 0000050328-741
The return value, `SAFEWDG_TLF_ACTIVE`, of the following APIs: • `SafeWdgExtTlf_GetState` • `SafeWdgIf_GetState` is removed, which was one of the possible states returned in earlier releases. `SAFEWDG_TLF_ACTIVE` is also removed as one of the possible values for type definition `SafeWdgExt_StateType`.	513 513.1 514 514.1 652	0000050328-771 0000050328-741
The return values of the following APIs is updated: • `SafeWdgExtCic_Init` • `SafeWdgExtCic_DeInit` • `SafeWdgExtCic_Activate` • `SafeWdgExtCic_Service`	513 513.1 514 514.1 652	0000050328-771 0000050328-741

Description	Applicable releases	Related issue IDs
• `SafeWdgExtCic_GetWdgInfo` • `SafeWdgExtCic_GetSeed` • `SafeWdgExtCic_GetErrCntr` • `SafeWdgExtCic_UserRequest` • `SafeWdgExtCic_GetJobResult` Additonal error codes are returned to indicate failures as against earlier releases where single error code was returned for multiple failures (`SWDG_JOB_FAILED`).		
The return values for the following APIs is updated: • `SafeWdgIf_Init` • `SafeWdgIf_DeIni` • `SafeWdgIf_Activate` • `SafeWdgIf_Service` • `SafeWdgIf_GetWdgInfo` • `SafeWdgIf_GetSeed` • `SafeWdgIf_GetErrCntr` • `SafeWdgIf_UserRequest` • `SafeWdgIf_GetJobResult` Additonal error codes are returned to indicate failures as against earlier releases where single error code was returned for multiple failures (`SWDG_JOB_FAILED`).	513 513.1 514 514.1 652	0000050328-771 0000050328-741
The following information is added/updated in the User Manual: • Description of ALM2[6] • Section 12.5.1.1.7 • Section 12.6.1.2.3 • Updated Figure 10 for additional Flash pages (Page 18-27) shown in Green color in the User Manual. • In Section 6.4.2.1, added Code Listting 14, 17 and 20: Allocation of test patterns to respective Flash banks when the Address Buffer Full SMU Alarm test is enabled.	513 513.1 514 514.1 652	0000050355-4
The following information is added to the User Manual: • Figure 49 • Section 12.8.1.1.5 Configuration parameter in configuration tools is renamed to STL_BIT_16 and STL_BIT_32 from BIT_16 and BIT_32, respectively.	513.1 514 514.1 652	0000050328-936
Chapter 12.8(SFR Compare Test)-> Parameter SfrTstRegisterBitWidth is added to support the selection of 16-, 32-bit register.	513.1 514 514.1 652	0000050328-521
The following sections are after/removed after the introduction of variant devices selection through configuration • Added Section 4.6 • In Section 12.21, removed configuration container for `PhlSramTst PhlSramTstADASdevice` and	513.1 514 514.1 652	0000050328-559 0000050328-282

Description	Applicable releases	Related issue IDs
`PhlSramTstEDdevice`. • In Section 12.15, removed configuration container for `SramEccTstADASdevice`.		
In Section 5.1, updated Example 1, Example 2 and Example 3. In this release, the batch file name and commands are updated. Single batch file is used with compiler as parameter as against earlier releases where different batch files were used for each compiler. Commands for old releases are: • `Build_SafeTlib_RefApp_Tasking.bat cleanbuild 27` • `Build_SafeTlib_RefApp_GNU.bat cleanbuild 27` • `Build_SafeTlib_RefApp_Windriver.bat cleanbuild 27`	513 513.1 514 514.1 652	0000050328-441
In Section 12.5 following configuration parameters are added: • `PmuEccEdcTstSecurityDisableCustCallback` • `PmuEccEdcTstSecurityEnableCustCallback`	513.1 514 514.1 652	0000050328-428
Updated Table 60. The `IomTst` configuration class is changed from post-build to link time.	513.1 514 514.1 652	0000050328-498
In Table 52, added parameter entries (No. 36, 37, 38, 39, 40 and 45).	513.1 514 514.1 652	0000050328-604

4.1 Related Issue Overview

Module	Description	ID
SafeWDG Driver for External CIC61508 Watchdog	Support Secure SPI mode of CIC.	0000050328-675
SafeWDG Driver for internal/External CIC61508/ External TLF35584 Watchdogs	Introduce Detailed error codes are in Internal/External CIC61508/TLF35584 Watchdog driver instead of single generic error code SWDG_JOB_FAILED to improve debugging.	0000050328-741 0000050328-771
PFLASH Monitor Test	Monitor SMU alarm for CBAB overflow	0000050355-4
SFR Test	Rename defines for BIT_16; BIT_32 to STL_BIT_16 and STL_BIT_32.	0000050328-936

Module	Description	ID
SFR Test	SFR Test (SFR CRC and SFR Compare tests) doesn't support 16bit registers. Parameter SfrTstRegisterBitWidth is added to support the selection of 16-, 32-bit register.	0000050328-521
Generic(for addition of marking Option devices)	Support for TC27x Marking options devices	0000050328-559 0000050328-282
Generic(for supporting additional compilers)	Added Tasking 5.0r2 support for SafeTlib package	0000050328-441
PMU ECC and EDC Test / PFLASH Monitor Test	Add support to enable/disable Flash protection for pattern programming. Update: User configurable call-back functions provided for enabling and disabling Flash write protection.	0000050328-428
SPB Register Access Protection Test	Missing port configuration for SPB test – Register access protection test Impact: SPB register access protection mechanism is not verified for the following ports: P24,P25,P26,P30,P31,P41	0000050328-604
IOM Test	IOM test configuration parameters are mapped to section IFX_IOMTST_START_SEC_POSTBUILDCFG_ASIL_B. As IOM test is a link time module, IomTst configuration class is changed from post-build to link time.	0000050328-498

5 Appendix

5.1 SafeTlib test mapping to Safety Mechanism

SafeTlib test name	Safety mechanisms covered by SafeTlib	Comments
SMU Driver	SM2[AoU].SMU.LOCK:TEST	This gets covered by an SMU Driver API which is called by an Test Handler API - Sl_SwitchTstPhase
CPU MPU Test	SM2[AoU].CPU.MPU:TEST	-
CPU Bus MPU Test	SM2[AoU].CPU.BUS.MPU:TEST	Contribute to SM2[AoU].SMU.ALARM:TEST
SRI EDC and Bus Error Test	SM2[AoU].SRI:EDC SM2[AoU].SRI:EH	Contribute to SM2[AoU].SMU.ALARM:TEST
PMU ECC and EDC Test	SM2[AoU].PFLASH:ECC SM2[AoU].PFLASH:ADDRMON SM2[AoU].PFLASH.EDC:CMP	Contribute to SM2[AoU].SMU.ALARM:TEST
PFLASH Monitor Test	SM2[AoU].PFLASH:Monitor	-
CPU Register Access Protection Test	SM2[AoU].CPU.AP:TEST	Contribute to SM2[AoU].SMU.ALARM:TEST
Trap Test	SM2[AoU].CPU.TRAP:TEST	NMI is not tested as part of this SafeTlib test. This needs to be checked by the user.
Clock Monitor Test	SM2[AoU].CLK:CLKMON	Contribute to SM2[AoU].SMU.ALARM:TEST
Voltage Monitor Test	SM2[AoU].EVR:MON, SM2[AoU].EVR33:MON, SM2[AoU].EVR13:MON, SM2[AoU].ExtVREG:MON	Contribute to SM2[AoU].SMU.ALARM:TEST
CPU Watchdog test	SM2[AoU].WDT:TEST	Contribute to SM2[AoU].SMU.ALARM:TEST
Safety Watchdog test	SM2[AoU].WDT:TEST	Contribute to SM2[AoU].SMU.ALARM:TEST
SRAM ECC Test	SM2[AoU].SRAM:ECC SM2[AoU].LMU.SRAM.ECC:MONITOR SM2[AoU].SRAM:ADDRMON SM2[AoU].SRAM:Monitor	Contribute to SM2[AoU].SMU.ALARM:TEST
Lockstep Test	SM2[AoU].CPU:LOCKSTEP.ALARM_TEST	Contribute to SM2[AoU].SMU.ALARM:TEST
Software Based Self-Test (SBST)	SM1[AoU].CPU:SBST	-
SPB Register Access Protection Test	SM2[AoU].Register:AP SM2[AoU].STM:AP SM2[AoU].DMA:AP SM2[AoU].IR:AP	SafeTlib checks the forbidden access only. Contribute to SM2[AoU].SMU.ALARM:TEST

SafeTlib test name	Safety mechanisms covered by SafeTlib	Comments
	SM2[AoU].SBCU:AP SM2[AoU].SMU:AP SM2[AoU].FCE:AP SM2[AoU].IOM:AP SM2[AoU].PORT:AP SM2[AoU].QSPI:AP SM2[AoU].MSC:AP SM2[AoU].CAN:AP SM2[AoU].SENT:AP SM2[AoU].ERAY:AP SM2[AoU].GTM:AP SM2[AoU].CCU6:AP SM2[AoU].VADC:AP SM2[AoU].DSADC:AP SM2[AoU].MTU:AP SM2[AoU].SCU:AP SM2[AoU].Ethernet:AP SM2[AoU].HSSL:AP SM2[AoU].PSI5:AP SM2[AoU].SRI:AP SM2[AoU].I2C:AP SM2[AoU].PMU:AP	
SPB Timeout Test	SM2[AoU].SPB:TIMEOUT	Contribute to SM2[AoU].SMU.ALARM:TEST
Safety Flip-Flop (SFF) Test	SM2[AoU].Register:SFF	Contribute to SM2[AoU].SMU.ALARM:TEST
PHL SRAM Test	SM2[AoU].PSRAM:ECC SM2[AoU].PSRAM:ADDRMON SM2[AoU].PSRAM:Monitor	Contribute to SM2[AoU].SMU.ALARM:TEST
DMA Test	SM2[AoU].DMA:TIMESTAMP SM2[AoU].DMA.DATA:CRC32, SM2[AoU].DMA.ADDR:CRC32	-
Interrupt Router Test	SM2[AoU].IR:EDC	Contribute to SM2[AoU].SMU.ALARM:TEST
Input Output Monitor (IOM) Test	SM2[AoU].IOM:TEST	Contribute to SM2[AoU].SMU.ALARM:TEST
Flexible CRC Engine(FCE) Test	SM2[AoU].FCE:TEST	-
LMU Bus MPU Test	SM2[AoU].LMU.MPU:TEST	Contribute to SM2[AoU].SMU.ALARM:TEST
LMU Register Access Protection Test	SM2[AoU].LMU.AP:TEST	Contribute to SM2[AoU].SMU.ALARM:TEST

5.2 Microcontroller Test Library tests details

The integrity tests of the Microcontroller Test Library have certain requirements

- Whether the test has to executed on All CPUs or only on Any one CPU
- Whether the other CPUs shall be in HALT or IDLE mode during the execution. In some cases, other CPUs may be running provided some conditions specified by the test are met. Such conditions if applicable are given in the chapter for each test in User Manual.
- Whether a configuration is necessary
- Whether interrupts that are generated by the application are allowed.

Test	TestID	Execution CPU	Other CPUs in HALT or IDLE mode	Configuration Required	Interrupts Allowed
CPU MPU Test	0x01	All CPUs	No	Yes	No
SFR Compare Test	0x02	Any CPU	No	Yes	Yes
SFR CRC Test	0x03	Any CPU	No	Yes	Yes
SRI EDC and Bus Error Test	0x04	All CPUs	Yes	Yes	No
CPU Bus MPU Test	0x06	All CPUs	No	Yes	No
Trap Test	0x07	All CPUs	Yes	No	No
CPU Register Access Protection Test	0x09	All CPUs	No	No	No
PMU ECC and EDC Test	0x0A	Any CPU	Yes	Yes	No
CPU Watchdog Test	0x0B	All CPUs	Yes	Yes	Yes
Safety Watchdog Test	0x0C	Any CPU	Yes	Yes	Yes
Voltage Monitor Test	0x0D	Any CPU	No	No	No
Clock Monitor Test	0x0E	Any CPU	No	No	Yes
Software Based Self Test (SBST) for TC1.6E CPU	0x0F	All CPUs**	No	Yes	Yes
Software Based Self Test (SBST) for TC1.6P CPU	0x10	All CPUs**	No	Yes	Yes
Smu Recovery Timer Test	0x11	Any CPU	Yes	No	Yes
Smu Interrupt Request Test	0x12	All CPUs	Yes	No	Yes
Smu Nmi Test	0x14	Any CPU	No	No	Yes
SRAM ECC Test	0x16	All CPUs*	Yes	Yes	No
Interrupt Router Test	0x17	All CPU	No	Yes	No
Lockstep Test	0x18	All CPUs	No	No	No
LMU RegAcc Prot Test	0x19	All CPU	Yes	No	Yes
LMU Bus Mpu Lfm Test	0x1A	All CPU	Yes	Yes	Yes
SPB Register Access Protection Test	0x1C	All CPU	Yes	Yes	No
SPB Timeout Test	0x1D	Any CPU	Yes	No	No
Flexible CRC Engine (FCE) Test	0x1E	Any CPU	No	Yes	Yes
DMA CRC Test	0x1F	Any CPU	No	Yes	Yes

Test	TestID	Execution CPU	Other CPUs in HALT or IDLE mode	Configuration Required	Interrupts Allowed
DMA Time Stamp Test	0x20	Any CPU	No	Yes	Yes
DMA Safe Linked List Test	0x21	Any CPU	No	Yes	Yes
Input Output Monitor(IOM) Test	0x22	Any CPU	Yes	Yes	Yes
PFLASH Monitor Test	0x24	Any CPU	Yes	Yes	Yes
Safety Flip-Flop (SFF) Test	0x25	Any CPU	No	No	Yes
Peripheral SRAM Test	0x26	Any CPU	No	Yes	No

Note:

** At least two CPUs are required on multi-CPU devices.*

*** All non-lockstep CPUs used by Safety applications.*

Revision History

Date	Version	Description of change
2021-10-29	V25.0	Release addendum 2021-10 (Issues till 2021-10-25 are considered) - Old Issues which are added in v24.0 are moved to the table in Section 1 New issues are added in Section 1.1
2021-01-28	V24.0	Release addendum 2021-01 (Issues till 2021-01-27 are considered) - Old Issues which are added in v23.0 are moved to the table in Section 1 New issues are added in Section 1.1 Moved 0000050328-1334 from SMU Driver to SMU Driver/ SMU test section as the issue applicable for both SMU Driver and SMU test.
2020-03-17	V23.0	Release addendum 2020-03 (Issues till 2020-03-16 are considered) - Old Issues which are added in v22.0 are moved to the table in Section 1 New issues are added in Section 1.1
2019-05-29	V22.0	Release addendum 2019-05 (Issues till 2019-05-28 are considered) - Old Issues which are added in v21.0 are moved to the table in Section 1 New issues are added in Section 1.1
2019-03-08	V21.0	Release addendum 2019-03 (Issues till 2019-03-07 are considered) - Old Issues which are added in v20.0 are moved to the table in Section 1 New issues are added in Section 1.1
2018-11-22	V20.0	Release addendum 2018-11 (Issues till 2018-10-17 are considered) - Old Issues which are added in v19.0 are moved to the table in Section 1 New issues are added in Section 1.1
2018-03-22	V19.0	Release addendum 2018-03 (Issues till 2018-03-21 are considered) - Old Issues which are added in v18.0 are moved to the table in Section 1 - New issues are added in Section 1.1
2018-01-29	V18.0	Release addendum 2018-01 (Issues till 2018-01-28 are considered) - Old Issues which are added in v17.0 are moved to the table in Section 1 - New issues are added in Section 1.1 - Issue details updated for 0000050328-1284 - Added following release EP MR4 (Rel634) – TC23x, TC22x, TC21x HE MR4 (Rel534) – TC27x, TC29x, TC26x

2017-08-28	V17.0	Release addendum 2017-08 (Issues till 2017-08-18 are considered) - Old Issues which are added in v16.0 are moved to the table in Section 1 - New issues are added in Section 1.1 - HW Errata sheet reference version is updated for TC26x BB, TC26x BC, TC23x AB, TC22x AB, TC21x AB and added newly for TC22x AC, TC21x AC
2017-06-14	V16.0	Release addendum 2017-06 (Issues till 2017-05-22 are considered) - Old Issues which are added in v15.0 are moved to the table in Section 1 - New issues are added in Section 1.1 - HW Errata sheet reference version is updated for TC29x BB, TC29x BC, TC26x BB, TC27x CA, TC27x DB, TC27x BC, TC27x DC, TC23x AC and added newly for TC26x BC - Added following release EP MR3 – TC23x, TC22x, TC21x (PRO-SIL_Aurix_SafeTlib_TC23x_TC22x_TC21x_MR3)
2017-05-03	V15.0	Release addendum 2017-05 (Issues till 2017-04-28 are considered) - Old Issues which are added in v14.0 are moved to the table in Section 1 - New issues are added in Section 1.1 - Added following release HE MR3 – TC27x, TC29x, TC26x (PRO-SIL_Aurix_SafeTlib_TC27x_MR3, PRO-SIL_Aurix_SafeTlib_TC29x_MR3, PRO-SIL_Aurix_SafeTlib_TC26x_MR3)
2017-01-12	V14.0	Release addendum 2017-01 (Issues till 2016-12-14 are considered) - Old Issues which are added in v13.0 are moved to the table in Section 1 - New issues are added in Section 1.1 - Added following release EP MR2 – TC23x, TC22x, TC21x (PRO-SIL_Aurix_SafeTlib_TC23x_TC22x_TC21x_MR2) - Issue 0000050329-13 removed since the Register File Definition qualified for Safety. - HW Errata sheet reference version is updated for TC23x AB and added newly for TC23x AC. - RCs are de-scoped in Release Addendum: RC releases Rel521, Rel522, Rel661 and Rel521.1 are removed. Known issues overview table in Section 1 and Section 1.1 are updated accordingly.

2016-10-13	V13.0	Release addendum 2016-10 (Issues till 2016-10-03 are considered) - Old Issues which are added in v12.0 are moved to the table in Section 1 - New issues are added in Section 1.1 - Added following release HE MR2 – TC27x, TC29x, TC26x (PRO-SIL_Aurix_SafeTlib_TC27x_MR2, PRO-SIL_Aurix_SafeTlib_TC29x_MR2, PRO-SIL_Aurix_SafeTlib_TC26x_MR2) - Added section “Release-specific changes in the User Manual”. - Issue 0000050329-13 removed for releases where the Register File Definition qualified for Safety.
2016-07-20	V12.0	Release addendum 2016-07 (Issues till 2016-06-30 are considered) - Old Issues which are added in v11.0 are moved to the table in Section 1 - New issues are added in Section 1.1 - Added following release EP MR1 – TC23x, TC22x, TC21x (PRO-SIL_Aurix_SafeTlib_TC23x_TC22x_TC21x_MR1) - Issue details is updated for 0000050328-846, 0000050328-666, 0000050328-670, 0000050328-655, 0000050328-708, 0000050328-624, 0000050328-649, 0000050328-696, 0000050328-925
2016-05-27	V11.0	Release addendum 2016-05 (Issues till 2016-05-20 are considered) - Notes added in Scope and Purpose section in the front page. - Description of the issues is enhanced. - Old Issues which are added in v10.0 are moved to the table in Section 1 - New issues are added in Section 1.1 - Added following release HE MR1 – TC27x, TC29x, TC26x (PRO-SIL_Aurix_SafeTlib_TC27x_MR1, PRO-SIL_Aurix_SafeTlib_TC29x_MR1, PRO-SIL_Aurix_SafeTlib_TC26x_MR1) - Section 3 is updated with new Jiras which will be taken up for upcoming releases - Section 4.1 Table is updated. - Removed Jira: 0000050328-444, 0000050328-629, 0000050329-363, 0000050328-513, 0000050328-592, 0000050329-283, 0000050329-405, 0000050329-384, 0000050328-353, 0000050329-353, 0000050328-415, 0000050329-170, CPU_TC.125_EPN / CPU_TC.126_EPN, 0000050329-183, 0000050329-244, 0000050329-404 (since these were duplicated/No issues/Internal issues)

2016-03-16	V10.0	Release addendum 2016-03 (Issues till 2016-03-09 are considered) – Editorial changes (Feature re-ordering of the issues, tabular form coverted to free text – New issues are added in Section 1.1 – HW errata's which are analysed are added in Front page – Added following releases PR5 – TC23x, TC22x, TC21x (PRO-SIL_Aurix_SafeTlib_ TC23x_TC22x_TC21x_PR5) RC6 – TC23x, TC22x, TC21x (PRO-SIL_Aurix_SafeTlib_ TC23x_TC22x_TC21x_RC6) – Removed following releases RC5 – TC23x (PRO-SIL_Aurix_SafeTlib_TC23x_RC5) RC5.1 – TC22x, TC21x (PRO-SIL_Aurix_SafeTlib_TC22x_TC21x_RC5.1) – Removed Jira: 0000050329-164, 0000050330-4, 0000050328-431, 0000050328-549, 0000050328-653, 0000050328-659, 0000050328-604, 0000050329-351, 0000050328-460, 0000050329-74, 0000050328-419, 0000050328-697, 0000050328-349, 0000050328-88
2016-01-08	V9.0	Release addendum 2016-01 (Issues till 2016-01-08 are considered) – Removed following releases RC4 - TC29x (PRO-SIL_AURIX_SafeTlib_TC29x_RC4) RC4.1 - TC26x (PRO-SIL_Aurix_SafeTlib_TC26x_RC4.1) RC7.1 - TC26x (PRO-SIL_Aurix_SafeTlib_TC26x_RC7.1) RC7.2 - TC26x (PRO-SIL_Aurix_SafeTlib_TC29x_RC7.2) RC5.3 – TC26x Since there is no request for PR (PRO-SIL_Aurix_SafeTlib_TC26x_RC5.3) – Added following releases PR7.2 - TC29x (PRO-SIL_Aurix_SafeTlib_TC29x_PR7.2)) RC8 - TC29x (PRO-SIL_Aurix_SafeTlib_TC29x_RC8) PR7.3 - TC26x (PRO-SIL_Aurix_SafeTlib_TC26x_PR7.3) RC8.1 - TC26x (PRO-SIL_Aurix_SafeTlib_TC26x_RC8.1) – Added Section 5.4 and 6.4 – Modified Jira description : 0000050328-476, 0000050328-478, 0000050328-495, 0000050328-522, 0000050328-523, 0000050328-524, 0000050328-525, 0000050328-526, 0000050328-527, 0000050328-528, 0000050328-530, 0000050328-531, 0000050328-567, 0000050328-590, 0000050328-592, 0000050328-606

2015-11-26	V8.0	Release addendum 2015-11 – Removed RC3 release PRO-SIL_Aurix_SafeTlib_TC23x_TC22x_TC21x_RC3. – Removed RC7 release PRO-SIL_Aurix_SafeTlib_TC27x_RC7 (Rel504). – Added RC5 (PRO-SIL_Aurix_SafeTlib_TC23x_RC5) and RC5.1 (PRO-SIL_Aurix_SafeTlib_TC22x_TC21x_RC5.1) releases. – Added Sections 5.2 and 6.3 – Added Section 8 – Workaround added for 0000050328-141. – Added PRO-SIL_Aurix_SafeTlib_TC27x_PR7.1(Rel513.1).
2015-09-30	V7.0	Release addendum 2015-09 – Streamlined Issues applicable to all releases. – Added new issues for releases based on its applicability. – Removed PRO-SIL_Aurix_SafeTlib_TC27x_RC5. – Added PR5.2, RC7, PR7, RC5.3(Windriver), RC7.1, RC7.2 and PR2.3 (Rel503.1) releases – Added Section 6 List of known Issues: 32 modules scope* – Issue '0000050336-3' removed as it not applicable for releases without copytable functionality. – Issue '0000050329-316' removed as it was a User manual topic. – Section 2 updated for clarity about obsolete releases. – Removed following information from Section 10. Changes required for PRO-SIL_AURIX_SafeTlib_Base_TC22x_TC21x_RC2', 'Changes required for PRO-SIL_AURIX_SafeTlib_TC29x_RC4'
2015-08-03	V6.0	Release addendum 2015-08 – Streamlined Issues applicable to all releases – Removed PRO-SIL_AURIX_SafeTlib_TC23x_RC1.1 – Removed PRO-SIL_AURIX_SafeTlib_Base_TC22x_TC21x_RC2 – Removed PRO-SIL_AURIX_SafeTlib_TC27x_RC5.1 – Added Section 5.7 PRO-SIL_Aurix_SafeTlib_TC27x_RC5 – Added Section 5.8 PRO-SIL_Aurix_SafeTlib_TC23x_TC22x_TC21x_RC3 – Added Section 5.9 PRO-SIL_Aurix_SafeTlib_TC27x_PR5 – Added Section 6.4 PRO-SIL_Aurix_SafeTlib_TC27x_RC5 – Added Section 6.5 PRO-SIL_Aurix_SafeTlib_TC23x_TC22x_TC21x_RC3 – Added Section 6.6 PRO-SIL_Aurix_SafeTlib_TC27x_PR5

Revision History

2015-05-20	V5.0	Release addendum 2015-05 – Streamlined Issues applicable to all releases – Removed PRO-SIL_AURIX_SafeTlib_TC26x_RC2.2 – Added Section 5.7 PRO-SIL_AURIX_SafeTlib_TC27x_RC5.1 – Added Section 5.8 PRO-SIL_AURIX_SafeTlib_TC29x_RC4 – Added Section 5.9 PRO-SIL_Aurix_SafeTlib_TC26x_RC4.1 – Added Section 8.2 Changes required for PRO-SIL_AURIX_SafeTlib_Base_TC22x_TC21x_RC2: Marking option specific changes for SAK-TC222S-12F133F AB – Added Section 8.3 Changes required for PRO-SIL_AURIX_SafeTlib_TC29x_RC4: Marking option specific changes for SAK-TC297TA-64F300S BB
2015-04-16	V4.0	Release addendum 2015-04
2015-03-23	V3.0	Release addendum 2015-03
2015-02-19	V2.0	Release addendum 2015-02
2015-01-23	V1.0	Release addendum for PRO-SIL_AURIX_SafeTlib_Base_PR11N release

Edition 2021-10-29

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about this document?

Email: erratum@infineon.com

Document reference
10290AERRA

Information note N°10290AERRA

PRO-SIL AURIX SafeTlib Release Notes Addendum V25.0 affecting products TC29x, TC27x, TC26x, TC23x, TC22x, TC21x

Sales name	SP number	OPN	Package
SAK-TC212L-8F133F AB	SP001268334	TC212L8F133FABKXUMA1	PG-TQFP-80-7
SAK-TC212L-8F133F AC	SP001658192	TC212L8F133FACKXUMA1	PG-TQFP-80-7
SAK-TC212L-8F133N AC	SP001592690	TC212L8F133NACKXUMA1	PG-TQFP-80-7
SAK-TC212S-8F133F AB	SP001287988	TC212S8F133FABKXUMA1	PG-TQFP-80-7
SAK-TC212S-8F133F AC	SP001658194	TC212S8F133FACKXUMA1	PG-TQFP-80-7
SAK-TC212S-8F133N AC	SP001658200	TC212S8F133NACKXUMA1	PG-TQFP-80-7
SAK-TC212S-8F133SC AC	SP001936928	TC212S8F133SCACKXUMA1	PG-TQFP-80-7
SAK-TC213L-8F133F AB	SP001268338	TC213L8F133FABKXUMA1	PG-TQFP-100-23
SAK-TC213L-8F133F AC	SP001658222	TC213L8F133FACKXUMA1	PG-TQFP-100-23
SAK-TC213L-8F133N AC	SP001641922	TC213L8F133NACKXUMA1	PG-TQFP-100-23
SAK-TC214L-8F133F AB	SP001268342	TC214L8F133FABKXUMA1	PG-TQFP-144-27
SAK-TC214L-8F133N AC	SP001641930	TC214L8F133NACKXUMA1	PG-TQFP-144-27
SAK-TC214S-8F133F AB	SP001287990	TC214S8F133FABKXUMA1	PG-TQFP-144-27
SAK-TC222L-16F133F AB	SP001268336	TC222L16F133FABKXUMA1	PG-TQFP-80-7
SAK-TC222L-16F133F AC	SP001658208	TC222L16F133FACKXUMA1	PG-TQFP-80-7
SAK-TC222L-16F133N AC	SP001584424	TC222L16F133NACKXUMA1	PG-TQFP-80-7
SAK-TC222S-12F133F AB	SP001329964	TC222S12F133FABKXUMA1	PG-TQFP-80-7
SAK-TC222S-12F133F AB	SP001641678	TC222S12F133FABKXQMA1	PG-TQFP-80-7
SAK-TC222S-12F133F AC	SP001721584	TC222S12F133FACKXUMA1	PG-TQFP-80-7
SAK-TC222S-16F133F AB	SP001288596	TC222S16F133FABKXUMA1	PG-TQFP-80-7
SAK-TC222S-16F133F AC	SP001658212	TC222S16F133FACKXUMA1	PG-TQFP-80-7
SAK-TC222S-16F133N AC	SP001644376	TC222S16F133NACKXUMA1	PG-TQFP-80-7
SAK-TC223L-16F133F AB	SP001268340	TC223L16F133FABKXUMA1	PG-TQFP-100-23
SAK-TC223L-16F133F AC	SP001658232	TC223L16F133FACKXUMA1	PG-TQFP-100-23
SAK-TC223L-16F133N AC	SP001592698	TC223L16F133NACKXUMA1	PG-TQFP-100-23
SAK-TC223S-16F133F AB	SP001288594	TC223S16F133FABKXUMA1	PG-TQFP-100-23
SAK-TC223S-16F133F AC	SP001658234	TC223S16F133FACKXUMA1	PG-TQFP-100-23
SAK-TC224L-16F133F AB	SP001268344	TC224L16F133FABKXUMA1	PG-TQFP-144-27
SAK-TC224L-16F133N AC	SP001641938	TC224L16F133NACKXUMA1	PG-TQFP-144-27
SAK-TC233L-32F200F AB	SP001268346	TC233L32F200FABKXUMA1	PG-TQFP-100-23
SAK-TC233L-32F200F AB	SP004396818	TC233L32F200FABKXUMA2	PG-TQFP-100-23
SAK-TC233L-32F200F AC	SP001674224	TC233L32F200FACKXUMA1	PG-TQFP-100-23
SAK-TC233L-32F200F AC	SP001700652	TC233L32F200FACKXQMA1	PG-TQFP-100-23
SAK-TC233L-32F200N AC	SP001408138	TC233L32F200NACKXUMA1	PG-TQFP-100-23
SAK-TC233LC-24F133F AB	SP001270740	TC233LC24F133FABKXUMA1	PG-TQFP-100-23
SAK-TC233LC-24F133F AC	SP001584028	TC233LC24F133FACKXUMA1	PG-TQFP-100-23
SAK-TC233LC-24F133N AC	SP002725546	TC233LC24F133NACKXUMA1	PG-TQFP-100-23
SAK-TC233LP-16F200F AB	SP001270730	TC233LP16F200FABKXUMA1	PG-TQFP-100-23
SAK-TC233LP-16F200F AC	SP001682994	TC233LP16F200FACKXUMA1	PG-TQFP-100-23
SAK-TC233LP-16F200N AC	SP001627900	TC233LP16F200NACKXUMA1	PG-TQFP-100-23
SAK-TC233LP-24F200N AC	SP002944554	TC233LP24F200NACKXUMA1	PG-TQFP-100-23

Sales name	SP number	OPN	Package
SAK-TC233LP-32F200F AB	SP001268348	TC233LP32F200FABKXUMA1	PG-TQFP-100-23
SAK-TC233LP-32F200F AC	SP001399222	TC233LP32F200FACKXUMA1	PG-TQFP-100-23
SAK-TC233LP-32F200F AC	SP001692512	TC233LP32F200FACKXQMA1	PG-TQFP-100-23
SAK-TC233LP-32F200N AC	SP001397720	TC233LP32F200NACKXUMA1	PG-TQFP-100-23
SAK-TC233LP-32F200N AC	SP001641894	TC233LP32F200NACKXQMA1	PG-TQFP-100-23
SAK-TC233S-32F200N AC	SP001658270	TC233S32F200NACKXUMA1	PG-TQFP-100-23
SAK-TC233S-32F200N AC	SP001720744	TC233S32F200NACKXQMA1	PG-TQFP-100-23
SAK-TC233SP-32F200N AC	SP001689232	TC233SP32F200NACKXUMA1	PG-TQFP-100-23
SAK-TC233SP-32F200N AC	SP001708334	TC233SP32F200NACKXQMA1	PG-TQFP-100-23
SAK-TC234L-24F200F AC	SP001658274	TC234L24F200FACKXUMA1	PG-TQFP-144-27
SAK-TC234L-24F200N AC	SP001624402	TC234L24F200NACKXUMA1	PG-TQFP-144-27
SAK-TC234L-32F200F AB	SP001268350	TC234L32F200FABKXUMA1	PG-TQFP-144-27
SAK-TC234L-32F200F AC	SP001658280	TC234L32F200FACKXUMA1	PG-TQFP-144-27
SAK-TC234L-32F200N AC	SP001408144	TC234L32F200NACKXUMA1	PG-TQFP-144-27
SAK-TC234LA-32F200F AB	SP001276932	TC234LA32F200FABKXUMA1	PG-TQFP-144-27
SAK-TC234LC-24F133F AC	SP001584026	TC234LC24F133FACKXUMA1	PG-TQFP-144-27
SAK-TC234LH-32F200F AB	SP005184614	TC234LH32F200FABKXUMA1	PG-TQFP-144-27
SAK-TC234LP-24F200F AC	SP001658278	TC234LP24F200FACKXUMA1	PG-TQFP-144-27
SAK-TC234LP-32F200F AB	SP001268352	TC234LP32F200FABKXUMA1	PG-TQFP-144-27
SAK-TC234LP-32F200F AC	SP001399224	TC234LP32F200FACKXUMA1	PG-TQFP-144-27
SAK-TC234LP-32F200F AC	SP002574662	TC234LP32F200FACKXQMA1	PG-TQFP-144-27
SAK-TC234LP-32F200N AC	SP001397714	TC234LP32F200NACKXUMA1	PG-TQFP-144-27
SAK-TC234LP-32F200N AC	SP001642890	TC234LP32F200NACKXQMA1	PG-TQFP-144-27
SAK-TC234LX-32F200F AB	SP001350934	TC234LX32F200FABKXUMA1	PG-TQFP-144-27
SAK-TC234LX-32F200F AB	SP001689442	TC234LX32F200FABKXQMA1	PG-TQFP-144-27
SAK-TC237L-32F200S AB	SP001268354	TC237L32F200SABKXUMA1	PG-LFBGA-292-6
SAK-TC237L-32F200S AC	SP001658282	TC237L32F200SACKXUMA1	PG-LFBGA-292-6
SAK-TC237LP-32F200N AC	SP001397698	TC237LP32F200NACKXUMA1	PG-LFBGA-292-6
SAK-TC237LP-32F200S AB	SP001268356	TC237LP32F200SABKXUMA1	PG-LFBGA-292-6
SAK-TC237LP-32F200S AC	SP001399220	TC237LP32F200SACKXUMA1	PG-LFBGA-292-6
SAK-TC264D-40F200N BC	SP001676268	TC264D40F200NBCKXUMA2	PG-LQFP-144-22
SAK-TC264D-40F200W BB	SP001257822	TC264D40F200WBBKXUMA1	PG-LQFP-144-22
SAK-TC264D-40F200W BC	SP001399214	TC264D40F200WBCKXUMA1	PG-LQFP-144-22
SAK-TC264DA-40F200N BC	SP001676270	TC264DA40F200NBCKXUMA2	PG-LQFP-144-22
SAK-TC264DA-40F200N BC	SP004374874	TC264DA40F200NBCKXUMA3	PG-LQFP-144-22
SAK-TC264DA-40F200W BB	SP001257824	TC264DA40F200WBBKXUMA1	PG-LQFP-144-22
SAK-TC264DA-40F200W BB	SP005562408	TC264DA40F200WBBKXUMA3	PG-LQFP-144-22
SAK-TC264DA-40F200W BC	SP001399218	TC264DA40F200WBCKXUMA1	PG-LQFP-144-22
SAK-TC264DC-40F200W BC	SP001611462	TC264DC40F200WBCKXUMA1	PG-LQFP-144-22
SAK-TC265D-40F200N BC	SP001397662	TC265D40F200NBCKXUMA1	PG-LQFP-176-22
SAK-TC265D-40F200W BB	SP001257830	TC265D40F200WBBKXUMA1	PG-LQFP-176-22

Information note N°10290AERRA

PRO-SIL AURIX SafeTlib Release Notes Addendum V25.0 affecting products TC29x, TC27x, TC26x, TC23x, TC22x, TC21x

Sales name	SP number	OPN	Package
SAK-TC265D-40F200W BB	SP005337974	TC265D40F200WBBKXUMA2	PG-LQFP-176-22
SAK-TC265D-40F200W BC	SP001399216	TC265D40F200WBCKXUMA1	PG-LQFP-176-22
SAK-TC265DC-40F200W BB	SP001363114	TC265DC40F200WBBKXUMA1	PG-LQFP-176-22
SAK-TC265DC-40F200W BC	SP001611464	TC265DC40F200WBCKXUMA1	PG-LQFP-176-22
SAK-TC267D-40F200N BC	SP001592934	TC267D40F200NBCKXUMA1	PG-LFBGA-292-6
SAK-TC267D-40F200S BB	SP001270092	TC267D40F200SBBKXUMA1	PG-LFBGA-292-6
SAK-TC267D-40F200S BC	SP001662612	TC267D40F200SBCKXUMA1	PG-LFBGA-292-6
SAK-TC275T-64F200N DC	SP001651570	TC275T64F200NDCKXUMA1	PG-LQFP-176-22
SAK-TC275T-64F200W CA	SP001106358	TC275T64F200WCAKXUMA1	PG-LQFP-176-22
SAK-TC275T-64F200W DB	SP001217592	TC275T64F200WDBKXUMA1	PG-LQFP-176-22
SAK-TC275T-64F200W DC	SP001662592	TC275T64F200WDCKXUMA1	PG-LQFP-176-22
SAK-TC275TC-64F200N DC	SP001667658	TC275TC64F200NDCKXUMA1	PG-LQFP-176-22
SAK-TC275TC-64F200W CA	SP001217626	TC275TC64F200WCAKXUMA1	PG-LQFP-176-22
SAK-TC275TC-64F200W DB	SP001217602	TC275TC64F200WDBKXUMA1	PG-LQFP-176-22
SAK-TC275TC-64F200W DC	SP001611466	TC275TC64F200WDCKXUMA1	PG-LQFP-176-22
SAK-TC275TP-64F200N DC	SP001397622	TC275TP64F200NDCKXUMA1	PG-LQFP-176-22
SAK-TC275TP-64F200N DC	SP001630432	TC275TP64F200NDCKXQMA1	PG-LQFP-176-22
SAK-TC275TP-64F200W CA	SP001105592	TC275TP64F200WCAKXUMA1	PG-LQFP-176-22
SAK-TC275TP-64F200W CA	SP001332056	TC275TP64F200WCAKXQMA1	PG-LQFP-176-22
SAK-TC275TP-64F200W DB	SP001217586	TC275TP64F200WDBKXUMA1	PG-LQFP-176-22
SAK-TC275TP-64F200W DC	SP001366138	TC275TP64F200WDCKXUMA1	PG-LQFP-176-22
SAK-TC277T-64F200N DC	SP001665660	TC277T64F200NDCKXUMA1	PG-LFBGA-292-6
SAK-TC277T-64F200N DC	SP005337796	TC277T64F200NDCKXUMA2	PG-LFBGA-292-10
SAK-TC277T-64F200S CA	SP001146094	TC277T64F200SCAKXUMA1	PG-LFBGA-292-6
SAK-TC277T-64F200S DB	SP001217632	TC277T64F200SDBKXUMA1	PG-LFBGA-292-6
SAK-TC277T-64F200S DB	SP005337792	TC277T64F200SDBKXUMA2	PG-LFBGA-292-10
SAK-TC277T-64F200S DC	SP001662590	TC277T64F200SDCKXUMA1	PG-LFBGA-292-6
SAK-TC277T-64F200S DC	SP005337794	TC277T64F200SDCKXUMA2	PG-LFBGA-292-10
SAK-TC277TC-64F200N DC	SP001667664	TC277TC64F200NDCKXUMA1	PG-LFBGA-292-6
SAK-TC277TC-64F200S CA	SP001217636	TC277TC64F200SCAKXUMA1	PG-LFBGA-292-6
SAK-TC277TC-64F200S DB	SP001217638	TC277TC64F200SDBKXUMA1	PG-LFBGA-292-6
SAK-TC277TC-64F200S DC	SP001611468	TC277TC64F200SDCKXUMA1	PG-LFBGA-292-6
SAK-TC277TC-64F200S DC	SP005337800	TC277TC64F200SDCKXUMA3	PG-LFBGA-292-10
SAK-TC277TC-64F200S DC	SP005345022	TC277TC64F200SDCKXUMA2	PG-LFBGA-292-10
SAK-TC277TP-64F200N DC	SP001397616	TC277TP64F200NDCKXUMA1	PG-LFBGA-292-6
SAK-TC277TP-64F200N DC	SP003475204	TC277TP64F200NDCKXQMA1	PG-LFBGA-292-6
SAK-TC277TP-64F200N DC	SP005337810	TC277TP64F200NDCKXUMA3	PG-LFBGA-292-10
SAK-TC277TP-64F200N DC	SP005345024	TC277TP64F200NDCKXUMA2	PG-LFBGA-292-10
SAK-TC277TP-64F200S BC	SP001103396	TC277TP64F200SBCKXUMA1	PG-LFBGA-292-6
SAK-TC277TP-64F200S CA	SP001146098	TC277TP64F200SCAKXUMA1	PG-LFBGA-292-6
SAK-TC277TP-64F200S DB	SP001217634	TC277TP64F200SDBKXUMA1	PG-LFBGA-292-6

Information note N°10290AERRA

PRO-SIL AURIX SafeTlib Release Notes Addendum V25.0 affecting products TC29x, TC27x, TC26x, TC23x, TC22x, TC21x

Sales name	SP number	OPN	Package
SAK-TC277TP-64F200S DC	SP001366136	TC277TP64F200SDCKXUMA1	PG-LFBGA-292-6
SAK-TC277TP-64F200S DC	SP005337808	TC277TP64F200SDCKXUMA3	PG-LFBGA-292-10
SAK-TC297T-96F300N BC	SP001684616	TC297T96F300NBCKXUMA1	PG-LFBGA-292-6
SAK-TC297T-96F300N BC	SP005411081	TC297T96F300NBCKXUMA2	PG-LFBGA-292-10
SAK-TC297T-96F300S BB	SP001265616	TC297T96F300SBBKXUMA1	PG-LFBGA-292-6
SAK-TC297TA-128F300N BC	SP001397668	TC297TA128F300NBCKXUMA1	PG-LFBGA-292-6
SAK-TC297TA-128F300N BC	SP005337844	TC297TA128F300NBCKXUMA2	PG-LFBGA-292-10
SAK-TC297TA-128F300S BB	SP001130460	TC297TA128F300SBBKXUMA1	PG-LFBGA-292-6
SAK-TC297TA-128F300S BB	SP005337840	TC297TA128F300SBBKXUMA2	PG-LFBGA-292-10
SAK-TC297TA-128F300S BC	SP001662208	TC297TA128F300SBCKXUMA1	PG-LFBGA-292-6
SAK-TC297TA-128F300S BC	SP005337846	TC297TA128F300SBCKXUMA2	PG-LFBGA-292-10
SAK-TC297TA-64F300S BB	SP001265584	TC297TA64F300SBBKXUMA1	PG-LFBGA-292-6