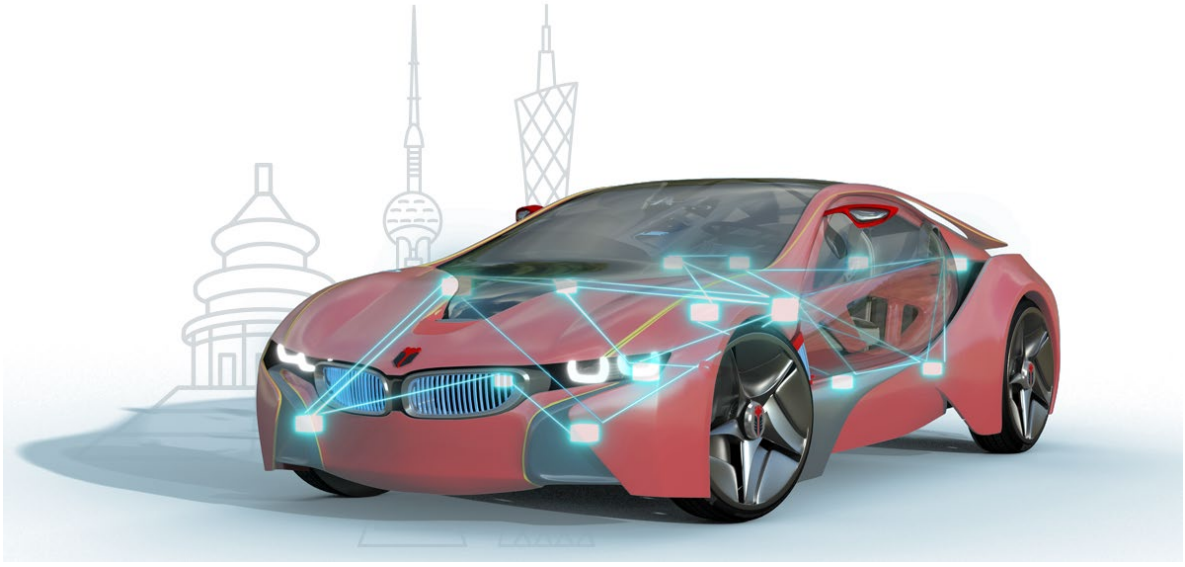
英飞凌 TC3XX_MTU 模块介绍以及使用方法

INTRODUCTION AND USAGE METHODS OF INFINEON TC3XX MTU MODULE



英飞凌 TC3XX_MTU 模块介绍以及使用方法

INTRODUCTION AND USAGE METHODS OF INFINEON TC3XX MTU MODULE

1 概述 OVERVIEW

MTU （Memory Test Unit）控制和监控设备内各种存储器的测试、初始化和数据完整性检查功能。该平台中的每个 SRAM 周围都有一些数字逻辑，称为 SRAM 支持硬件（SSH）。SSH 是一个硬件块，它控制内部存储器的错误检测和纠正，内存内置自检（MBIST）。每个 SSH 块提供一个统一的接口来控制其各种功能。有多个这样的 SSH 实例，每个实例控制一个或多个不同的内部内存。AurixPlus 平台中的内存测试单元（Memory Test Unit， MTU）提供了寄存器接口来配置和控制这些不同的内存控制器。

MTU (Memory Test Unit) controls and monitors the testing, initialization, and data integrity check functions of various memories in the device. Each SRAM in the platform has some digital logic around it called SRAM Supporting Hardware (SSH). SSH is a hardware block that controls error detection and correction of internal memory, in-memory self-test (MBIST). Each SSH block provides a unified interface to control its various functions. There are multiple such SSH instances, each of which controls one or more different internal memory. The Memory Test Unit (MTU) in the AurixPlus platform provides a register interface to configure and control these different memory controllers.

2 MTU 系统视图 SYSTEM VIEW

系统中不同的 IP 模块（如 CPU、LMU、CAN 等）内部可能有一个或多个 SRAM。每个 SRAM 周围都有自己的 SSH 模块，有自己的寄存器（SRAM 支持硬件（SSH）寄存器）和 MBIST 逻辑。每个 SSH 通过内部接口分别连接到 MTU。MTU 本身连接到 SPB 总线，并且每个 SSH 的寄存器可以通过这个 SPB 接口访问。每个 SPB 访问都由 MTU 在内部转发（和返回）到（或从）相应的 SSH。

Different IP modules in the system (such as CPU, LMU, CAN, etc.) may have one or more Srams inside. Each SRAM has its own SSH module around it, with its own registers (SRAM supports hardware (SSH) registers) and MBIST logic. Each SSH is connected to the MTU separately through an internal interface. The MTU itself is connected to the SPB bus, and each SSH register can be accessed through this SPB interface. Each SPB access is internally forwarded (and returned) by the MTU to (and from) the corresponding SSH.

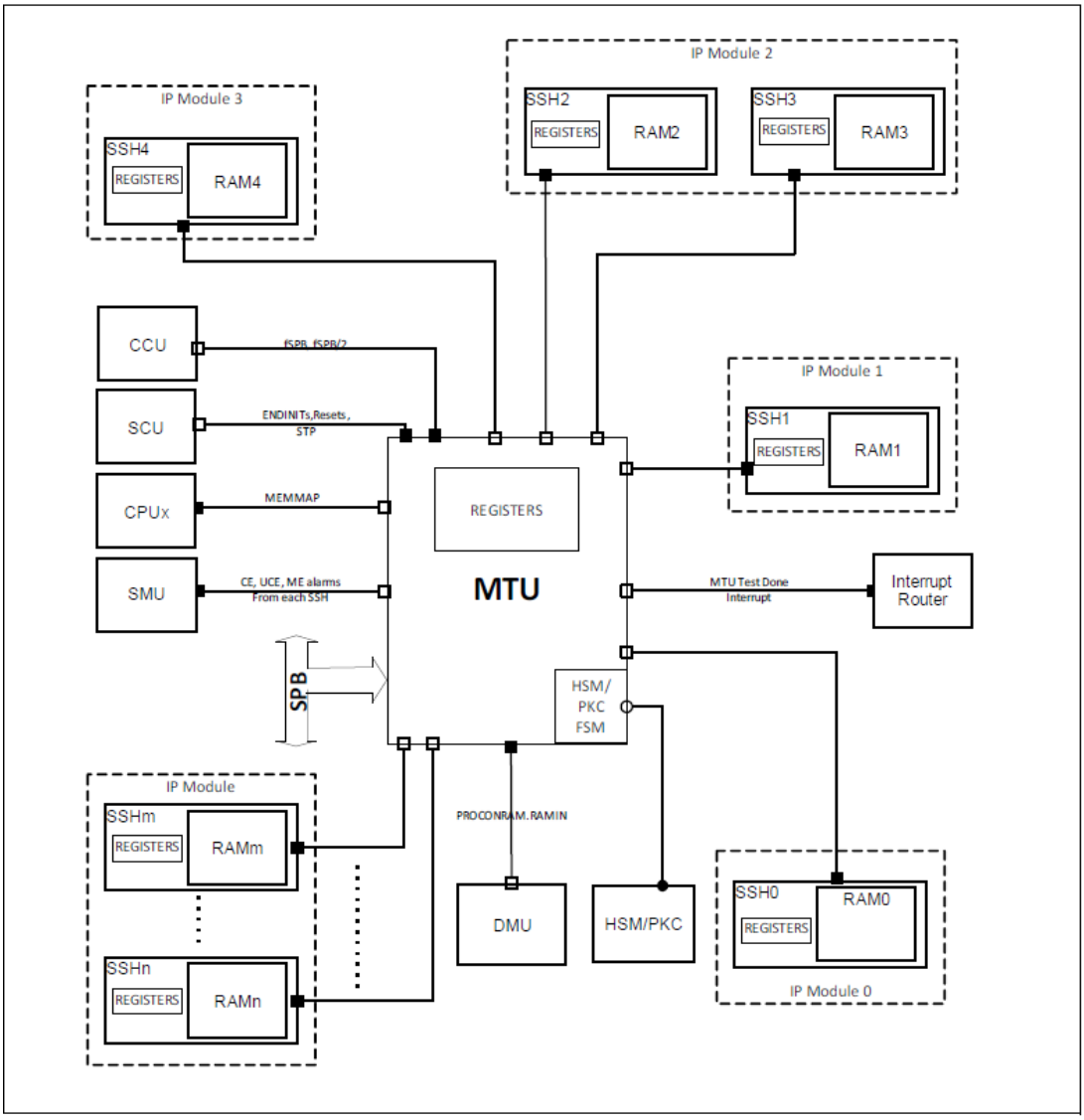


Figure 154 System View: MTU, SSHs and SRAMs in IPs

3 MTU 功能特性 FUNCTIONAL FEATURES

1.提供到内部 SSH 实例的统一接口。

– MTU 提供一个统一的寄存器接口用于控制每个内部 SSH 实例的操作和功能。

– 系统中每个 SRAM 的各种可配置的测试类型可以通过 MTU 控制。

2.数据初始化

– 系统中每个 SRAM 块都能通过 MTU 进行硬件初始化。

– Security-Sensitive 存储器可以自动初始化，防止通过 MTU 将数据读出。

3.内存错误纠正和检测

– 系统中每个 SRAM 的内存错误纠正和检测都能通过 MTU 配置。

– 可纠正的和不可纠正的错误检测。

– 地址错误检测。

4.SMU 警告通知

– 来自 SRAM/SSH 的 3 个警告（CE alarm, UCE alarm and ME alarm）被发送到 MTU，然后被 MTU 转发到 SMU。

1. Provide a unified interface to the internal SSH instance.

- MTU provides a unified register interface to control the operations and functions of each internal SSH instance.

- Various configurable test types for each SRAM in the system can be controlled via MTU.

2. Data initialization

- Each SRAM block in the system can be hardware initialized using MTU.

- Security-Sensitive storage can be automatically initialized to prevent data from being read out through the MTU.

3. Memory error correction and detection

- Memory error correction and detection of each SRAM in the system can be configured by MTU.

- Correctable and uncorrectable error detection.

- Address error detection.

4. SMU warning notification

- Three alarms (CE alarm, UCE alarm and ME alarm) from SRAM/SSH are sent to MTU and then forwarded to SMU by MTU.

4 功能描述 FUNCTIONAL DESCRIPTION

4.1 SSH 使能(Enabling the SRAM Support Hardware)

AurixPlus 平台中的每个 SSH 在 MTU_MEMTESTx (x=0-2)寄存器中都有一个与之相关的启用位。在 AurixPlus 平台中实现的 SSH 启用位的 MEMTESTi (i=0-2)寄存器。只有当 SSH 启用时，每个 SSH 的控制寄存器才可访问(MTU_MEMTEST.MEMx_EN = 1)。唯一的例外是寄存器 MCi_ECCS (i=0-95)， MCi_ECCD (i=0-95)和 MCi_ETRRx (i=0-95;x=0-4)， MCi_ERRINFOx (i=0-95;x=0-4)， MCi_ALMSRCS (i=0-95)和 MCi_FAULTSTS (i=0-95)(在每个 SSH 实例中)，它们在任何时候都可以允许容易的运行时访问 ECC 特性，即这些特定的寄存器(即。ECCS、ECCD、ETRRx、ERRINFOx、ALMSRCS 和 FAULTSTS)可以读写，唯一的前提条件是 MTU 时钟通过 MTU_CLC 寄存器使能。只有通过 MEMTEST 寄存器使能 SSH 时，才能访问 SSH 的所有其他寄存器，否则 MTU 将返回 SPB 错误。对于某些 SSH，通过 MEMTEST 寄存器启用或禁用 SSH 可能会根据配置导致整个或部分内存被自动初始化。这个自动初始化可能需要数百个时钟周期来完成。在此期间，不能访问 SSH 寄存器(甚至不能访问通常不需要启用 SSH 的寄存器)，并且任何对运行自动初始化的 SSH 的寄存器访问都将导致 SPB 总线错误。注意，在通过 MTU 访问任何 SSH 寄存器之前，必须通过 CCU 寄存器启用包含 SRAM 的模块的时钟输入。除了这个 CCU 时钟配置之外，不需要启用模块时钟(例如，使用 CLC 寄存器)来访问 SSH 寄存器。

Each SSH in the AurixPlus platform has an enabled bit associated with it in the MTU_MEMTESTx (x=0-2) register. SSH enabled MEMTESTi (i=0-2) register implemented in the AurixPlus platform. Each SSH's control register is accessible only when SSH is enabled (MTU_MEMTEST.MEMx_EN = 1). The only exceptions are the registers MCi_ECCS (i=0-95), MCi_ECCD (i=0-95), and MCi_ETRRx (i=0-95; x=0-4), MCi_ERRINFOx (i=0-95; x=0-4), MCi_ALMSRCS (i=0-95), and MCi_FAULTSTS (i=0-95)(in each SSH instance), which at any time allow easy run-time access to ECC features, namely these specific registers (i.e. ECCS, ECCD, ETRRx, ERRINFOx, ALMSRCS, and FAULTSTS) can read and write only if the MTU clock is enabled through the MTU_CLC register. All other registers of SSH can be accessed only if SSH is enabled through the MEMTEST register, otherwise MTU will return an SPB error. For some SSHS, enabling or disabling SSH through the MEMTEST register may cause all or part of memory to be automatically initialized, depending on the configuration. This automatic initialization can take hundreds of clock cycles to complete. During this period, SSH registers cannot be accessed (even registers that are not normally required to enable SSH), and any register access to running auto-initialized SSH will result in an SPB bus error. Note that the clock input of the module containing the SRAM must be enabled through the CCU register before any SSH register can be accessed through the MTU. Apart from this CCU clock configuration, there is no need to enable the module clock (for example, using the CLC register) to access the SSH register.

Note:

1. 当启用 SSH 时，对内存的功能访问暂时不可用。当一个内存正在被测试时，软件应该阻止对该内存的功能性访问(例如，在测试 CPU 的本地内存时，CPU 应该处于空闲状态，或从其他内存执行)。
   When SSH is enabled, memory access is temporarily unavailable. When a memory is being tested, the software should block functional access to that memory (for example, when testing the CPU's local memory, the CPU should be idle, or executing from other memory).
2. 正确执行 SSH 操作(例如 MBIST 测试)要求 MTU 和包含待测内存的模块在 SSH 操作期间都是可操作的(例如:测试)。当启用 SSH 时，包含待测内存的模块的重置可能会导致意外行为。软件必须注意，当启用 SSH 时，诸如应用程序重置、模块重置、备用条目、时钟频率改变等事件不会被触发。例如，模块复位可能会导致模块在复位后尝试正常访问 SRAM，而 SSH 仍然是启用的(因此功能访问是禁用的)，这样的场景必须通过软件来避免。
   Properly performing an SSH operation (e.g. a MBIST test) requires that both the MTU and the module containing the memory under test are operable during an SSH operation (e.g. a test). When SSH is enabled, the reset of the module containing the memory under test can cause unexpected behavior. Software must be aware that when SSH is enabled, events such as application resets, module resets, alternate entries, clock frequency changes, etc. are not triggered. For example, a module reset may cause the module to try to access SRAM normally after the reset while SSH is still enabled (and therefore functional access is disabled), such a scenario must be avoided with software.
3. 当模块复位时，如果 MTU 和 SSH 之间的通信正在进行(例如 SSH 寄存器读或写)，通信将被中止，并在几个周期内立即在 SPB 上触发总线错误，没有任何超时。只有正在进行的读写可能被破坏，未来的通信不会受到影响。
   When the module is reset, if the communication between MTU and SSH is in progress (such as SSH register read or write), the communication is aborted and a bus error is triggered immediately on the SPB within a few cycles without any timeouts. Only ongoing reads and writes may be disrupted, and future communications will not be affected.
4. 如果访问了无效的 SSH 寄存器(例如，在一个 SSH 实例中不存在的寄存器，或一个不存在的 SSH 实例)，这将导致 SPB 上的总线错误。
   If an invalid SSH register is accessed (for example, a register that does not exist in an SSH instance, or an SSH instance that does not exist), this will result in a bus error on the SPB.

4.2 Security-Sensitive 存储器和自动初始化(Security-Sensitive Memories and AutoInitialization)

对于 TC3XX 平台，以下内存属于 Security-Sensitive 内存。

For the TC3XX platform, the following memory is considered Security-Sensitive memory.

1. CPUx_DMEM(x = 0 - 5) – 仅内存的缓存部分属于 Security-Sensitive 内存
   CPUx_DMEM(x = 0 - 5) - Only the cache part of the memory is considered security sensitive.
2. CPUx_DTAG(x = 0 - 5)
3. CPUx_PMEM(x = 0 - 5) – 仅内存的缓存部分属于 Security-Sensitive 内存
   CPUx_PMEM(x = 0 - 5) - Only the cache part of the memory is considered security sensitive.
4. CPUx_PTAG(x = 0 - 5)

对于以上 Security-Sensitive 内存，MTU 必须阻止通过 SSH 去读写这些内存。对于安全应用程序，Security-Sensitive 内存可以通过自动初始化以擦除现有内容。这是由 DMU 中的 PROCONRAM 寄存器控制的。

For the Security-Sensitive memory, the MTU must prevent the memory from being read or written over SSH. For secure applications, Security-Sensitive memory can be automatically initialized to erase existing content. This is controlled by the PROCONRAM register in the DMU.

如果 PROCONRAM.RAMIN = 00、01 或 10，则启用自动内存内容初始化。PROCONRAM 配置初始化是由冷复位触发，还是由热复位触发，或者两者都触发。在这些模式中，只要对应的 MTU_MEMTEST，也会触发 Security-Sensitive 内存的自动初始化。MEMx_EN 或 MTU_MEMMAP 被更改（即，如果内存映射模式被更改，则相应的 SSH 被启用或禁用）。MEMSTATi （i=0-2）寄存器位表示 MEMTEST 状态的改变是否触发了内存 x 的自动数据初始化。MEMx_EN 或 MTU_MEMMAP 和初始化序列尚未完成。如果 MEMSTAT 寄存器中的 MEMx_AIU 位被设置，这意味着该内存的自动初始化仍在进行中。在启动 Auto-initialization 后，SSH 才会启用，在启动 Auto-initialization 前，SSH 才会禁用。软件可以通过轮询这个位来等待自动初始化完成。

If PROCONRAM.RAMIN = 00, 01 or 10 then automatic memory content initialization is enabled. PROCONRAM. configures whether the initialization is triggered by cold resets, warm resets or both.In these modes, an automatic initialization of security-sensitive memories is also triggered whenever the corresponding MTU_MEMTEST.MEMx_EN or MTU_MEMMAP is changed (i.e. The corresponding SSH is enabled or disabled or if memory map mode is changed). The MEMSTATi (i=0-2) register bits indicate whether an automatic data initialization of Memory x has been triggered by a change of state of MEMTEST.MEMx_EN or MTU_MEMMAP and the initialization sequence has not yet completed. If a MEMx_AIU bit in a MEMSTAT register is set,

this means that Autoinitialization for that memory is still underway. The SSH is enabled only after, and disabled before the Auto-initialization starts. The software can wait for the Auto-initialization to be completed by polling this bit.

如果 PROCONRAM.RAMIN=11，则复位时不自动初始化 RAM 内容，当 SSH 模块启用或禁用 MTU_MEMTEST 时，不自动初始化 RAM 内容。MEMx_EN 或 MTU_MEMMAP 寄存器位。这允许应用程序使用 SSH（例如，用于错误注入，数据修改或运行时内存测试），而不会损坏内存内容。

If PROCONRAM.RAMIN=11 then no automatic initialization of RAM content is performed on a reset and no automatic initialization of RAM content is performed when SSH modules are enabled or disabled with MTU_MEMTEST.MEMx_EN or MTU_MEMMAP register bits. This permits the use of the SSH by an application (E.g. for error injection, data modification or runtime memory testing) without unwanted corruption of memory content.

4.3 SSH 操作(SRAM Support Hardware (SSH) Operation)

4.3.1 内存测试启动(STARTING A MEMORY TEST SEQUENCE)

每个内存测试的启动是通过写入对应 SSH 寄存器 MCONTROL.START 位来启动。在启用 MCONTROL.START 位之前，需要正确配置配置寄存器 MCi_CONFIG0 (i=0-95), MCi_CONFIG1 (i=0-95)。

Each memory test is started by writing the corresponding SSH register MCONTROL.START bit. Before enabling the MCONTROL.START bit, the configuration registers MCi_CONFIG0 (i=0-95) and MCi_CONFIG1 (i=0-95) need to be configured correctly.

1. CONFIG0.ACCSTYPE 指定将在当前内容的每个单一地址上执行的访问类型（read/write）。
   CONFIG0.ACCSTYPE specifies the type of access (read/write) that will be performed on each single address of the current content.
2. CONFIG0.NUMACCS 指定当前内容中单个地址的访问总数。
   CONFIG0.NUMACCS Specifies the total number of accesses to a single address in the current content.
3. CONFIG1.ACCSPAT 指定访问模式。
   CONFIG1.ACCSPAT Specifies the access mode.
4. CONFIG1.AG_MOD 指定 SSH 使用复杂的寻址方案。
   CONFIG1.AG_MOD specifies that SSH uses a complex addressing scheme.
5. CONFIG1.AG_MOD 指定配置测试为无创测试。当使用无创测试时，MSTATUS.FAIL 是不固定的，此时软件必须检测 ECCD, ETRR , ERRINFO 寄存器是否在测试过程中检测到任何错误。
   CONFIG1.AG_MOD specifies that the configuration test is noninvasive. When using noninvasive tests, MSTATUS.FAIL is not fixed, and the software must check whether the ECCD, ETRR, and ERRINFO registers detect any errors during the test.

内存测试完成由 MSTATUS.DONE=1 标识。如果测试失败，则 MSTATUS.FAIL 位会被置起。Note：如果测试序列由于中间故障而停止测试，那么 MSTATUS.DONE 不会置 1。

Memory test completion is identified by MSTATUS.DONE=1. If the test fails, the MSTATUS.FAIL bit is raised. Note: If the test sequence stops testing due to an intermediate fault, MSTATUS.DONE will not be set to 1.

4.3.2 内存测试完成中断 (MEMORY TEST DONE INTERRUPT)

MTU 为中断路由器（IR）提供一个中断。中断表示所有正在运行的测试已经完成。该信号复位值高。当对任何 SRAM 的测试开始时，该信号变为低电平。在所有正在进行的测试完成后，这个信号再次变高，并且这个上升沿触发中断。

The MTU provides an interrupt to the interrupt router (IR). The interrupt signifies the completion of all running tests. The reset value of this signal is high. When a test on any SRAM is started, this signal goes to low. On completion of all on-going tests, this signal again goes high, and this rising edge triggers the interrupt.

4.3.3 获取详细的内存测试结果(GETTING DETAILED MEMORY TEST RESULTS)

MSTATUS.FAIL 和 MSTATUS.DONE (可以通过 MEMDONEi (i=0-2) 寄存器从 MTU 本身轮询)位提供了测试通过/测试失败的信息和测试完成状态。

The MSTATUS.FAIL and MSTATUS.DONE (can be polled from the MTU itself via MEMDONEi (i=0-2) register) bits provide a general pass/fail information and test completion status.

如果 MCONTROL.FAILDMP = '1'，测试在失败后停止，故障信息立即用于转存。MSTATUS.FDA 用于失败转存。任何转存信息只能在 RDBFL 和 ETRR(0)查询。RDBFL 包含失败位映射，ETRR 包含失败地址。

If MCONTROL.FAILDMP = '1', the test stops after a failure and the fail information is immediately available for dump. MSTATUS.FDA (fail dump available) is set in this case. Any dump information has to be polled from registers RDBFL and ETRR(0). RDBFL contains the fail bit map and ETRR the failed address.

读取 MSTATUS 和 RDBFL(n-1)寄存器，当 MSTATUS.FDA = '1'时，将会复位 MSTATUS.FDA 成 0。MCONTROL.RESUME 后续设置将会恢复中断测试序列。

Reading MSTATUS and the RDBFL(n-1) registers with MSTATUS.FDA = '1' will reset MSTATUS.FDA back to '0'. A consequent setting of MCONTROL.RESUME will resume the interrupted test sequence.

范围寄存器可用于对不断移动的内存范围运行连续测试，以便最终分析完整的内存。

The RANGE register can be used to run consecutive tests on constantly shifted memory ranges so that in the end the complete memory has been analyzed.

内存测试期间的故障注入。
Error Injection During Memory Tests.

- 非破坏性测试：通过 ECCMAP 编程一个带有错误 ECC 的 Word 来产生数据错误。
  For a non-destructive test, an error in the data can be introduced by programming a word with wrong ecc before the test, via the ECCMAP bits.

- 破坏性测试：通过RANGE.INJERR位来注入故障，RANGE.INJERR=1 和 RANGE.RAEN=1时，将RANGE.ADDR字段作为一个指向SRAM位置的指针，在测试期间不会对该地址进行写访问。在测试之前，用软件写一个特定的值（使用RDBFL寄存器使用单个SRAM写访问），然后使用RANGE.INJERR&&RANGE.RAEN=1进行测试。
  During a destructive march test, it is possible to inject data errors via the RANGE.INJERR bit. When this bit is set (and RANGE.RAEN = 1), then the RANGE.ADDR field is taken as a pointer to a physical SRAM location, to which write accesses during the test are not executed. Software can then write a particular value (i.e. using single SRAM write access using the RDBFL register) before a march test, and then run the test with RANGE.INJERR and RAEN = 1.

- 数据错误：
  1. 测试在全地址运行，RANGE.ADDR指定的地址不执行写访问，导致测试过程中与其数据不匹配导致测试失败。
     The test then runs over the full memory, and on the address corresponding to RANGE.ADDR, writes are not executed. This results in mismatch of the expected data during the test, resulting in a FAIL.

  2. 注入错误后，所有正常的诊断和通知都能被测试，Alarm能够被触发，在非破坏性测试期间，能够在 ETRR/ERRINFO中跟中错误。
     With errors injected, all normal diagnostics and notifications can be tested- that is, alarms are triggered, errors are tracked in the ETRR/ERRINFO during a non-destructive test.
  3. 在测试中，FAIL位能够被置位，当FAILDUMP = 1时，能够获得 fail bitmap同时FDA会被置位。
     And during a march test, FAIL bit is set, and when FAILDUMP = 1, the fail bitmap is obtained and FDA is set.
  4. fail bitmap是简单的(预期数据模式)XOR(实际数据模式)。
     The fail bitmap is simply (Expected data pattern) XOR (Actual data pattern).

- 地址错误：通过简单地将SFLE位设置为1，在测试期间触发地址错误。注意，这将触发来自每个地址的地址错误，并且在非破坏性测试期间，导致ETRR/ERRINFO被地址错误填充。

  Address errors are triggered during the test by simply setting SFLE bit to 1. Note that this will trigger an address error from each address, and during a non-destructive test, result in the ETRR/ERRINFO getting filled with address errors.

4.3.4 用定义的内容填充内存(FILLING A MEMORY WITH DEFINED CONTENTS)

SSH 可以用定义的模式快速的对部分内存或者整个内存进行填充。例如使用 MCONTROL.DINIT 位，每个周期有一次写访问，具有完整的内存数据宽度。因此，必须确保 MCONTROL.SRAM_CLR=0。

The SSH can be used to fill a memory range or a complete memory with a defined pattern very fast, i.e. one write access per cycle with the full memory data width, using the MCONTROL.DINIT bit. For this, it has to be ensured that MCONTROL.SRAM_CLR = 0.

首先，在设置 MCONTROL.DINIT 之前，软件需要用所需的位模式填充 RDBFL 寄存器（具体参考 MCi_RDBFLy (i=0-95;y=0-66)）。在此模式下，不强制要求具有有效的 ECC 码。

Before setting the MCONTROL.DINIT, the software should first fill the RDBFL register with the desired bit pattern (please refer to the MCi_RDBFLy (i=0-95;y=0-66)). It is not mandatory to have a valid ECC code in this pattern.

其次，设置 RANGE 寄存器，设置为需要填充模式的内存范围。

Next, the RANGE register needs to be set with the memory range into which the pattern needs to be filled.

接下来，将 MCONTROL.DINIT 和 MCONTROL.START 设置为开始初始化 RAM。然后软件需要清除 MCONTROL.START 位。当 MSTATUS.DONE 被硬件设置为 1 时，表示内存填充操作完成。

Next, the MCONTROL.DINIT needs to be set, and then the MCONTROL.START bit should be set to start initializing the RAM. The software should then clear the MCONTROL.START. When MSTATUS.DONE bit is set by the hardware, the memory filling operation is complete.

注意：对某些 SRAM，不支持这种使用 DINIT 位的 SRAM 初始化的方法。

This method of SRAM initialization using the DINIT bit is not supported for certain SRAMs. These exceptions are mentioned in the product specific appendix chapter.

4.3.5 初始化 RAM(INITIALIZING SRAMS)

使用 MCONTROL.SRAM_CLR 位可以初始化整个 SRAM，所有的 SSH 和 SRAMS 都支持初始化。

Using the MCONTROL.SRAM_CLR bit, it is possible to initialize the complete SRAM. This is supported for all SSHs and SRAMs.

- 使能 SSH，MEMTESTx.MEMxEN = 1。注意：使能动作会触发 UCE 告警，需要在使能前通过设置 ALMSRCS.OPENE 来关闭告警。
  For this operation, enable the SSH (MEMTESTx.MEMxEN = 1), and: (Note that this will trigger an UCE alarm. Therefore, the alarm reaction may need to be disabled before. Software can set ALMSRCS.OPENE before)
- 设置 MCONTROL.SRAM_CLR 位。
  Set the MCONTROL.SRAM_CLR.
- 使用 MCONTROL.START 来开启初始化。
  Start the initialization using MCONTROL.START.
- 等待 MSTATUS.DONE 复位和清除 MCONTROL.START 位。
  Wait for MSTATUS.DONE to be reset and clear the MCONTROL.START.
- 通过轮询 MSTATUS.DONE 位来等待初始化结束。
  Wait for the end of the initialization by polling the MSTATUS.DONE bit.
- 清除 MCONTROL.SRAM_CLR,并离开测试模式。
  Clear the MCONTROL.SRAM_CLR and leave the test mode (MEMTESTx.MEMxEN = 0).

通过初始化，整个 SRAM 被 ECC 正确的零值填充。清除初始化引起的 UCEE 和 OPERR 故障标志，重新启用 ALMSRCS.OPENE。

With this initialization, the complete SRAM will be filled with ECC-correct zero value. Clear the UCERR and OPERR flags set due to this operation. Re-enable ALMSRCS.OPENE if it was disabled before the test.

4.3.6 读取单个内存位置(READING A SINGLE MEMORY LOCATION)

SSH 还可以用于读取单个单词的内容。RDBFL 寄存器保存完整内存字的内容，因此可以读取所有内存位，包括 ECC 或奇偶校验位。

必要的步骤是：

1. 进入内存测试模式。
2. 初始化寄存器 RANGE.RAEN=0，RAEN 指定地址访问模式，0 表示单地址访问模式且单个内存地址可读可写；CONFIG0.NUMACCS = 1，NUMACCS = 1 表示每个地址的访问次数；CONFIG0.ACCSTYPE =1，ACCSTYPE 表示访问类型，1 表示读访问；CONFIG1 = 0，AG_MOD 表示地址生成器模式，0 代表使用线性地址生成运行测试，SELFASTB 指定选择快速位，0 表示正常寻址序列位 0 在正常位置，ACCSPAT=0 指定访问模式。

MCi_RANGE (i=0-95)
Range Register, single address mode (1008_H+i*100_H) **Application Reset Value: $XXXX_H$**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAEN	INJERR	ADDR													
rw	rw	rw													

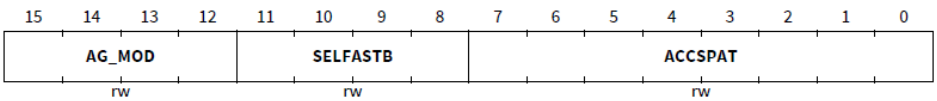
Field	Bits	Type	Description
ADDR	13:0	rw	**Address** When RAEN = 0,This field specifies the address of a single memory location. Reads and writes to this location are possible. When RAEN=1, this field is interpreted as 2 different fields. ADDR[13:7] is interpreted as Upper Range Limit. ADDR[6:0] is interpreted as Lower Range Limit. For smaller SRAMs which require lesser number of address bits, the MSB bits are reserved. Writes to these bits are ignored, and reads return '0'.
INJERR	14	rw	**Inject Error** Enables Error-Injection during march tests. This is supported only for linear march tests. 0_B Do not mask any writes during march tests. RANGE.ADDR and RAEN used normally. 1_B Use RANGE.ADDR as a pointer to a physical SRAM address to which write accesses during a march test will not be executed. This bit helps in error injection during a march test over the whole SRAM. This bit has an effect only when RAEN is also set. With INJERR and RAEN = '1', the test is by default run over the entire SRAM.

Field	Bits	Type	Description
RAEN	15	rw	**Range Enable** 0 Disabled, single address mode. In this case a single word can be addressed for read or write. Config registers have to be set as follows CONFIG.NUMACCS:= "0001" (single access) CONFIG.AG_MOD := "0000" (linear) MCONTROL.DIR :=1 (up) For read just the value in this location will be delivered. No check against expected values is made; i.e. MSTATUS.FAIL will not be set. 1 Enabled. ADDR[13:7] is interpreted as Upper Range Limit. ADDR[6:0] is interpreted as Lower Range Limit.

MCi_CONFIG0 (i=0-95)
Configuration Registers (1000_H+i*100_H) **Application Reset Value: 2002_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NUMACCS				R8				ACCSTYPE							
rw				r				rw							

MCi_CONFIG1 (i=0-95)
Configuration Register 1 (1002_H+i*100_H) Application Reset Value: 0000_H



3. MCONTROL = 0x4009 (FAILDMP = 0, direction up, start):开始读操作。MSTATUS.DONE 将被清除。
4. MCONTROL = 0x4008,清除启动位。
5. 等待 MSTATUS.DONE 再次置位（也可以轮询 MEMDONEi (i=0-2)寄存器状态）
6. 读取 RDBFL 寄存器数据
7. 脱离内存测试模式
8. 清除 UCERR 和 OPERR 故障标志（开启测试前需要关闭故障告警）

1. Enter memory test mode
2. Initialize registers, RANGE := RAEN = 0 (range disabled = single address) & address to be read, CONFIG0 := 1001H (NUMACCS = 1H, ACCSTYPE = 01H (read))CONFIG1 := 0000H (linear mode, non inverted pattern)
3. MCONTROL := 4009H (FAILDMP = 0, direction up, start): Start read operation. MSTATUS.DONE will be cleared now.
4. MCONTROL := 4008H (clear START)
5. Wait for MSTATUS.DONE to be set again (Poll the corresponding bit in the **MEMDONEi (i=0-2)** register).
6. Read RDBFL register
7. Leave memory test mode
8. Clear the UCERR and OPERR flags set due to this operation.

4.3.7 写入单个内存位置(WRITING TO A SINGLE MEMORY LOCATION)

SSH 还可以用于写入单个单词的内容。RDBFL 寄存器保存完整内存字的内容，因此可以读取所有内存位，包括 ECC 或奇偶校验位。

必要的步骤是:

1. 进入内存测试模式。
2. 初始化寄存器 RANGE.RAEN=0，RAEN 指定地址访问模式，0 表示单地址访问模式且单个内存地址可读可写；CONFIG0.NUMACCS = 1，NUMACCS = 1 表示每个地址的访问次数；CONFIG0.ACCSTYPE =0，ACCSTYPE 表示访问类型，0 表示写访问；CONFIG1 = 0，AG_MOD 表示地址生成器模式，0 代表使用线性地址生成运行测试，SELFASTB 指定选择快速位，0 表示正常寻址序列位 0 在正常位置，ACCSPAT=0 指定访问模式。
3. MCONTROL = 0x4009 (FAILDMP = 0, direction up, start):开始读操作 MSTATUS.DONE 将被清除。
4. MCONTROL = 0x4008,清除启动位。
5. 等待 MSTATUS.DONE 再次置位（也可以轮询 MEMDONEi (i=0-2)寄存器状态）
6. 脱离内存测试模式
7. 清除 UCERR 和 OPERR 故障标志（开启测试前需要关闭故障告警）

The SSH can also be used to write the contents of RDBFL register to a single memory location. RDBFL holds the contents of a complete memory word and thus it is possible to write to all memory bits, including ECC or parity bits. The necessary steps are:(Note that this will trigger the UCE alarm. Therefore, the alarm reaction may need to be disabled before)

1. Enter memory test mode
2. Initialize registers
3. RDBFL := write data， RANGE := RAEN= 0 (range disabled = single address) & address to be written to ， CONFIG0 := 1000H (NUMACCS = 1H, ACCSTYPE = 00H (write)), CONFIG1 := 0000H (linear mode, non inverted pattern)

3. MCONTROL := 4009H (FAILDMP = 0, direction up, start): Start write operation. MSTATUS.DONE will be clearednow.
4. MCONTROL := 4008H (clear START)
5. Wait for MSTATUS.DONE to be set again (Poll the corresponding bit in the **MEMDONEi (i=0-2)** register).
6. Leave memory test mode
7. Clear the UCERR and OPERR flags set due to this operation.

4.3.8 复位后的告警处理(ALARM HANDLING AFTER RESET)

当一个告警产生时，系统可能会进行复位。ECCD 寄存器中的告警状态位(ECCD.CERR,UCERR 和 MERR)在复位后被清除，但错误状态位(ETRR,ERRINFO 和 FAULTSTST 寄存器)仍然有用，错误状态位只能通过软件清除，并只能通过热 PORST 复位。

When an alarm occurs, the system may perform a reset. The Alarm status bits in the ECCD register (i.e. ECCD. CERR, UCERR and MERR) are cleared after an application reset. However, the error status bits (i.e. ETRR, ERRINFO and FAULTSTS registers) are still available until a power-on reset for diagnosis purposes - they can only be cleared by software and are reset only with a warm PORST.

通过应用程序复位，告警本身也会被清除。这可以防止单个警报复位循环。

The alarm itself is also cleared with an application reset. This prevents a single alarm creating a reset loop.

4.4 安全特性(Safety Features)

SRAM、SSH 和 MTU 实现的安全特性：

1. 错误检测和纠正码/逻辑。
2. 地址错误监控。
3. SRAM 多路复用因子。
4. 错误跟踪寄存器。
5. 临界寄存器位的安全触发器。

The safety features implemented in the SRAMs, SSH instances and the MTU are the following:

1. Error Detection and Correction Codes / Logic
2. Address Error Monitor
3. SRAM Mux Factor
4. Error Tracking Registers
5. Safety Flip-Flop implementation for critical register bits.

5 MBIST 检测应用示例(EXAMPLE OF MBIST DETECTION)

编程步骤(Program Steps):

1. 使能 MTU 接口：CLC. DISR = 0，CLC.EDIS = 1。
   Enable MTU interfaces: CLC. DISR = 0, CLC.EDIS = 1.

CLC
Clock Control Register **(0000$_H$)** **Application Reset Value: 0000 0003$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								0				EDIS	Resvd	DISS	DISR
r								r				rw	r	rh	rw

Field	Bits	Type	Description
DISR	0	rw	**Module Disable Request Bit** Used for enable/disable control of the module. 0_B Module disable is not requested 1_B Module disable is requested
DISS	1	rh	**Module Disable Status Bit** Bit indicates the current status of the module If the RMC field is implemented and if it is 0, DISS is set automatically. 0_B Module is enabled 1_B Module is disabled
Resvd	2	r	**Resvd** Read as 0. Must be written with 0_H
EDIS	3	rw	**Sleep Mode Enable Control** Used for module Sleep Mode control. 0_B Sleep Mode request is regarded. Module is enabled to go into Sleep Mode on a request. 1_B Sleep Mode request is disregarded: Sleep Mode cannot be entered on a request.

2. 清除 SSH 故障信息：MCi_ECCD.TRC = 1(i:0-95)。
   Clear SSH fault information: MCi_ECCD.TRC = 1(i:0-95).

Field	Bits	Type	Description
TRC	4	w	**Tracking Clear** Writing this bit with '1' clears the EOV, VAL bits plus the ETRR and ERRINFO registers, depending on the PERMERR settings. This bit will always read 0. 0_B No effect. 1_B Clear the ETRR, ERRINFO and ECCD.VAL & EOV bits. If a PERMERR bit is set, then the corresponding entries are not cleared. Note: If PERMERR and TRC are written at the same time, the clearing due to TRC takes place with the previous PERMERR settings, and the new settings take effect only after.

3. 清除 SMU 故障信息：Smu_ClearAlarmStatus。
   Clear the fault information about the SMU: Smu_ClearAlarmStatus.

Table 241 Alarm Mapping related to ALM0 group (cont'd)

Alarm Index	Module	Safety Mechanism & Alarm Indication
ALM0[3]	Reserved	Reserved
ALM0[4]	MTU	Safety Mechanism: SRAM Monitor Alarm: CPU0 PCACHE TAG Uncorrectable critical error detection Alarm Type: Level
ALM0[5]	MTU	Safety Mechanism: SRAM Monitor Alarm: CPU0 PCACHE TAG Miscellaneous error detection Alarm Type: Level
ALM0[6]	MTU	Safety Mechanism: SRAM Monitor Alarm: CPU0 PSPR/PCACHE Single bit error correction Alarm Type: Level
ALM0[7]	MTU	Safety Mechanism: SRAM Monitor Alarm: CPU0 PSPR/PCACHE Uncorrectable critical error detection Alarm Type: Level
ALM0[8]	MTU	Safety Mechanism: SRAM Monitor Alarm: CPU0 PSPR/PCACHE Miscellaneous critical error detection Alarm Type: Level
ALM0[9]	Page 22	Safety Mechanism: SRAM Monitor Alarm: CPU0 DSPR/DCACHE/DLMU Single bit error correction Alarm Type: Level
ALM0[10]	Page 22	Safety Mechanism: SRAM Monitor Alarm: CPU0 DSPR/DCACHE/DLMU Uncorrectable critical error detection Alarm Type: Level
ALM0[11]	Page 22	Safety Mechanism(s): SRAM Monitor Alarm: CPU0 DSPR/DCACHE/DLMU Miscellaneous error detection Alarm Type: Level

4. 关闭 SMU 告警动作：SMU_AGiCFj = 0。
   Disable the SMU alarm action: SMU AGiCFj = 0.

SMU_AGiCFj (i=0-3;j=0-2)
Alarm Configuration Register **(100_H+i*12+j*4)** **Application Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	CF24	CF23	CF22	0	0	0	0	0	0
r	r	r	r	r	r	r	rw	rw	rw	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	CF14	CF13	CF12	CF11	CF10	CF9	CF8	CF7	CF6	CF5	CF4	0	CF2	CF1	CF0
r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	rw	rw	rw

Field	Bits	Type	Description
CFz (z=0-2,4-14,22-24)	z	rw	**Configuration flag x (x=0-2) for alarm z belonging to alarm group i.** The configuration flags 0, 1 and 2 must be used together to define the behavior of the SMU_core when a fault state is reported by the alarm n belonging to this group. 0_B Configuration flag x (x=0-2) is set to 0 1_B Configuration flag x (x=0-2) is set to 1
0	31, 30, 29, 28, 27, 26, 25, 21, 20, 19, 18, 17, 16, 15, 3	r	**Reserved** Read as 0; should be written with 0.

5. 初始化 SSH(Initializing SSH):
   1. 关闭 ALMSRCS. MISCE 和 OPENE 故障检测。
      Disable ALMSRCS. MISCE and OPENE fault detection.

MCi_ALMSRCS (i=0-95)
Alarm Sources Configuration Register ($10EE_H+i*100_H$) Application Reset Value: $003F_H$

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R6										MISCE	OPENE	OVFE	ADDRE	DBE	SBE
r										rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
SBE	0	rw	**Single Bit Error Notification & Tracking Enable** This bit enables ECC Single Bit Detection/Correction event to be tracked forwarded to the CE or UCE alarm. If ECCS.ECE bit is '1', then SBE errors are forwarded to CE alarm. Otherwise to UCE alarm. The error status can be read from the ERRINFO registers (ERRINFO[x].SBERR) 0_B SBE errors are neither tracked in the ETRR, nor notified via an alarm. 1_B SBE errors are tracked in the ETRR & ERRINFO, and notified via an alarm (CE if ECE = 1, UCE if ECE = 0).
DBE	1	rw	**Double Bit Error Notification and Tracking Enable** This bit enables ECC Double Bit Errors in the SRAM to be tracked and forwarded as an UCE alarm. The error status can be read from the ERRINFO registers (ERRINFO.DBERR). 0_B DBE errors are neither tracked in the ETRR, nor notified via an alarm. 1_B DBE errors are tracked in the ETRR & ERRINFO, and notified via a UCE alarm.

Field	Bits	Type	Description
ADDRE	2	rw	**Address Error Notification Enable** This bit enables the detection and tracking of Address Faults in the SRAM, and forward them as a source of UCE alarm. The error status can be read from the ERRINFO registers (ERRINFO.ADDRERR). 0_B Address Faults in the SRAM are neither tracked in the ETRR, nor notified via an alarm. 1_B Address Faults in the SRAM are tracked in the ETRR & ERRINFO, and notified via a UCE alarm.
OVFE	3	rw	**ETRR Overflow notification enable- OVFE** This bit enables the forwarding of the ETRR Overflow event as an alarm source to the UCE alarm. The Error information can be obtained via the ECCD.VALID bits and the EOV bit. 0_B Do not report Error Tracking (ETRR) Buffer Overflow Error. 1_B Report Error Tracking (ETRR) Buffer Overflow Error via the UCE alarm
OPENE	4	rw	**SSH Operational Error Notification Enable** This bit enables the forwarding of many errors which are critical to the operation of the SRAM or SSH. These errors are forwarded as one of the sources of the UCE alarm. The error status can be read from FAULTSTS.OPERR bits. This bit is enabled by default. 0_B Do not enable the detection and forwarding SSH/SRAM operation critical errors as a source to the UCE alarm. 1_B Enable the detection and forwarding SSH/SRAM operation critical errors as a source to the UCE alarm
MISCE	5	rw	**SSH Misc. Errors Notification Enable** This bit enables the forwarding of many errors which may be critical to the operation of the SRAM or SSH in the future. These errors are forwarded as one of the sources of the ME alarm. The error status can be read from FAULSTS.MISCERR. This bit is enabled by default. 0_B Do not enable the detection and forwarding of misc. SSH/SRAM errors as a source to the ME alarm. 1_B Enable the detection and forwarding of misc. SSH/SRAM errors as a source to the ME alarm
R6	15:6	r	**Reserved - Res** Reads return 0

2. 配置 MEMTESTi(i:0-2),使能对应 SSH。
   Configure MEMTESTi(i:0-2) to enable SSH.
3. 轮询 MEMSTAT0 寄存器，等待 SSH 的 SRAM 自动初始化完成。
   Poll the MEMSTAT0 register and wait for the SRAM automatic initialization of SSH to complete.

4. 设置 CONFIG0 = 0x4005(NUMACCS = 4，ACCSTYPE = 5), CONFIG1 = 0x5008(ACCSPAT = 8, AG_MOD = 5)。
   Set CONFIG0 = 0x4005(NUMACCS = 4，ACCSTYPE = 5), CONFIG1 = 0x5008(ACCSPAT = 8, AG_MOD = 5)
5. 开启 SRAM 测试，MCONTROL. START = 1。
   Start SRAM Test, MCONTROL. START = 1.
6. 清除开启标志位，MCONTROL. START = 0。
   Clear Start Flag，MCONTROL. START = 0。
7. 轮询 MSTATUS.DONE，等待 SRAM 测试完成。
   Poll MSTATUS.DONE and wait for the SRAM test to complete.
8. 查看 ECCD 寄存器的 SERR，CERR，MERR，EOV 故障。
   Check the SERR, CERR, MERR, and EOV faults of the ECCD register.
9. 查看 ERRINFO 和 ETRR 寄存器获取故障信息。
   Check the ERRINFO and ETRR registers to obtain fault information.
10. 清除故障信息，ECCD.TRC = 1。
    Clear the fault information. ECCD.TRC = 1.

6 ZC TC3XX SAFETYFRAME 产品介绍(PRODUCT INTRODUCTION)

汽车电控系统的电气化、智能化发展日趋复杂，对于电子电气架构的安全性要求也越来越高。通过对道路车辆应用场景的 HARA 分析，为了使安全目标被降级分解、保持危害发生可能性低于风险的受限值，汽车功能安全越来越受到重视。近年来，在功能安全标准上参考 ISO 26262；在软件架安全架构上参考 E-GAS 分层。在电子电气系统中，对于通用的 Element 通常采用 SEooC(safety element out of context)方法进行设计开发。

The electrification and intelligent development of automobile electronic control system is becoming more and more complex, and the safety requirements of electronic and electrical architecture are becoming higher and higher. Through HARA analysis of road vehicle application scenarios, more and more attention is paid to vehicle functional safety in order to degrade and decompose safety objectives and keep the possibility of hazard occurrence lower than the risk limit. In recent years, reference has been made to ISO 26262 for functional safety standards; Refer to E-GAS layering for the software shelf security architecture. In electronic and electrical systems, For the common Element, the SEooC (safety element out of context) approach is usually adopted for its design and development.

知从科技推出 SAFETY FRAME 为各车载控制器客户提供 ASIL 等级分解咨询、FMEDA 分析过程支持、芯片级自检安全机制开发、SafetyFrame 配置与软件集成等全流程功能安全服务。

ZC launched SAFETY FRAME to provide customers with ASIL level decomposition consultation, FMEDA analysis process support, chip-level self-check safety mechanism development, SafetyFrame configuration and software integration and other full-process functional safety services.

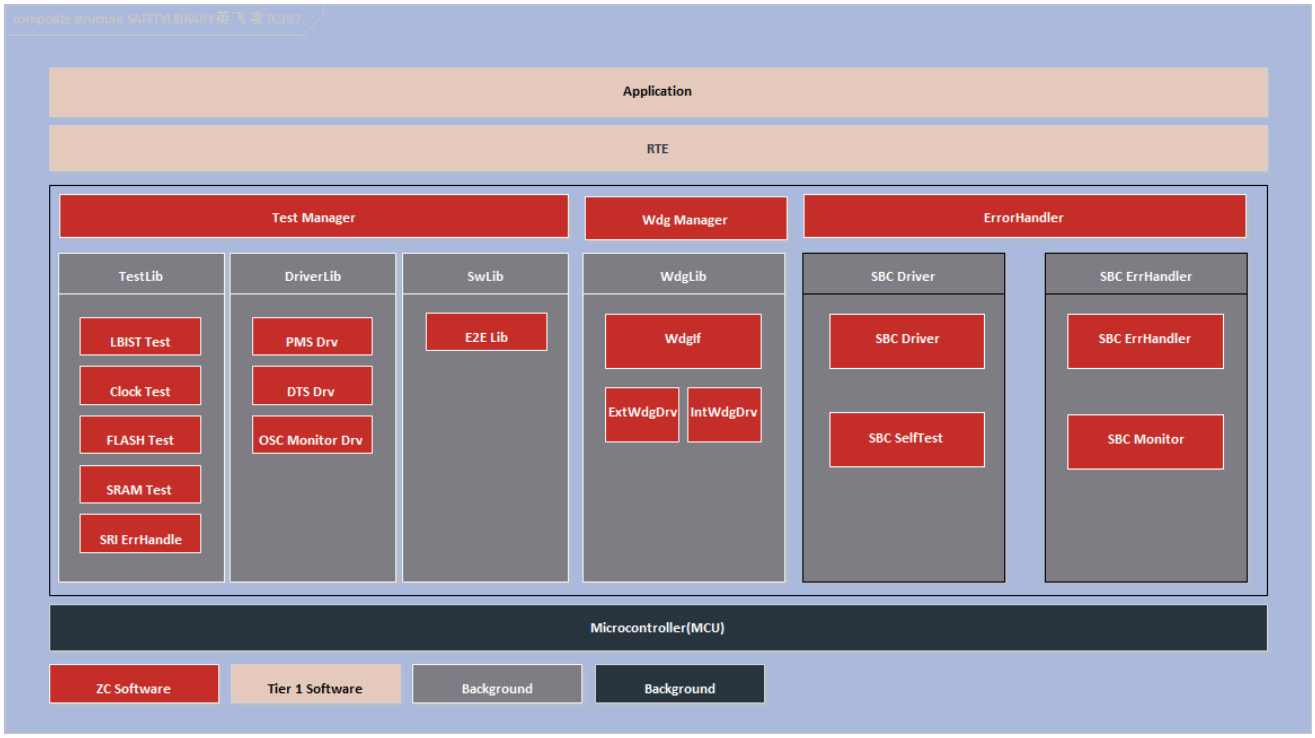
SAFETY FRAME 包括 3 个组件：MCU 内部模块自检测试组件（即 SF.MCU）、SBC 硬件安全机制的驱动组件(即 SF.SBC)、安全架构组件(即 SF.Architecture)。SF.Architecture 的核心模块为 Test Manager，用于 MCU&SBC 的 Safety Library 调度管理，包括 Safety Wdgm、Safety SBC/ASIC 驱动模块调度、与应用层 PFC(Program Flow Check)接口等，SF.MCU 包含 3 大模块：

SAFETY FRAME consists of 3 components: the internal module self-checking test component of MCU (i.e. SF.MCU), the driver component of SBC's hardware security mechanism (i.e. SF.SBC), and the safety architecture component (i.e. SF.Architecture). The core module of SF.Architecture is Test Manager, which is used for the scheduling management of Safety Library for MCU and SBC, including Safety Wdgm, scheduling of Safety SBC/ASIC driver modules, and interfaces with application layer PFC (Program Flow Check), etc. SF.MCU contains 3 major modules:

- TestLib--实现 MCU 芯片模块的检测。
  TestLib-- Implementation of MCU chip module inspection.
- DriverLib--实现 MCU 芯片模块的驱动。
  DriverLib-- Implements the MCU chip module driver.
- SwLib--用户常用的数字签名库、端到端保护库等接口。
  SwLib-- Interfaces such as digital signature database and end-to-end protection database are commonly used by users.

SAFETY FRAME 在软件模块化分层原则上，将 Function Controller 和 Monitoring Controller 分别由 SF.MCU 和 SF.SBC 实现，并部署在 EGAS Level2 和 Level3 层级，充分考虑了程序流监控和关断路径设计的应用需求。

In the principle of software modular layering, Function Controller and Monitoring Controller are implemented by SF.MCU and SF.SBC, respectively. It is also deployed at EGAS Level2 and Level3 levels, taking into account the application requirements of program flow monitoring and shutdown path design.



软件架构 / Software Architecture

知从 SafetyFrame 产品实现的功能安全模块包括：Test Manager 模块、LBIST Test 模块、MBIST Test 模块、PFlash Test 模块、MCU Firmware Test 模块、Register Test 模块、DMA Test 模块、SRI Error Handling 模块、MONBIST Test 模块、Mcu Register Monitor 模块、Register Monitor Test 模块、Evadc Test 模块、Interrupt monitor Test 模块、Clock Plausibility Test 模块、DAM Test 模块、Convctrl Test 模块、CPU Internal BUS Test 模块、STM Test 模块、GTM TIM Clock Test 模块、Gtm IOM Alarm Test 模块、Gtm Tom Tim Test 模块、Port Test 模块、GptTst 模块、PMS configuration 模块、DTS Configuration 模块、OSC Clock Monitor 模块、SMU Error Handler 模块、SMU Software Alarm Drv 模块、IR FFI Control 模块、GTM IOM Configuration 模块、ERU Configuration 模块、TLF35584 Driver 模块、TLF35584 Error Handler 模块、E2E 保护模块、Safe Watchdog Manager 模块、Safe Watchdog Interface 模块、Safe Internal Watchdog 模块、Safe SBC Watchdog 模块。

The functional safety modules implemented by SafetyFrame products include: Test Manager module, LBIST Test module, MBIST Test module, PFlash Test module, MCU Firmware Test module, Register Test module, DMA Test module, SRI Error Handling module, MONBIST Test module, Mcu Register Monitor module, Register Monitor Test module, Evadc Test module, Interrupt monitor Test module, Clock Plausibility Test module, DAM Test module, Convctrl Test module, CPU Internal BUS Test module, STM Test module, GTM TIM Clock Test module, Gtm IOM Alarm Test module, Gtm Tom Tim Test module, Port Test module, GptTst module, PMS configuration module, DTS Configuration module, OSC Clock Monitor module, SMU Error Handler module, SMU Software Alarm Drv module, IR FFI Control module, GTM IOM Configuration module, ERU Configuration module, TLF35584 Driver module, TLF35584 Error Handler module, E2E protection module, Safe Watchdog Manager module, Safe Watchdog Interface module, Safe Internal Watchdog module, Safe SBC Watchdog module.

知从 SafetyFrame 产品针对本文中提及的 TC3XX MTU 模块实现了内存自检 MBIST 模块，以及寄存器自检 RegMonTst 模块，模块实现了以下安全机制。

SafetyFrame products for the TC3XX MTU module mentioned in this article to implement the memory self-check MBIST module, as well as register self-check RegMonTst module, the module implements the following safety mechanism.

安全机制(Safety Mechanism)
ESM[SW]:AMU.LMU_DAM:REG_MONITOR_TEST
ESM[SW]:CIF.RAM:REG_MONITOR_TEST
ESM[SW]:CPU.DCACHE:REG_MONITOR_TEST
ESM[SW]:CPU.DLMU:REG_MONITOR_TEST
ESM[SW]:CPU.DSPR:REG_MONITOR_TEST
ESM[SW]:CPU.DTAG:REG_MONITOR_TEST
ESM[SW]:CPU.PCACHE:REG_MONITOR_TEST
ESM[SW]:CPU.PSPR:REG_MONITOR_TEST
ESM[SW]:CPU.PTAG:REG_MONITOR_TEST
ESM[SW]:DMA.RAM:REG_MONITOR_TEST
ESM[SW]:EMEM.RAM:REG_MONITOR_TEST
ESM[SW]:ERAY.RAM:REG_MONITOR_TEST
ESM[SW]:GETH.RAM:REG_MONITOR_TEST
ESM[SW]:GTM.RAM:REG_MONITOR_TEST
ESM[SW]:HSPDM.RAM:REG_MONITOR_TEST
ESM[SW]:LMU.RAM:REG_MONITOR_TEST
ESM[SW]:MCMCAN.RAM:REG_MONITOR_TEST
ESM[SW]:PSI5.RAM:REG_MONITOR_TEST
ESM[SW]:SCR.RAM:REG_MONITOR_TEST
ESM[SW]:SDMMC.RAM:REG_MONITOR_TEST
ESM[SW]:SPU.BUFFER:REG_MONITOR_TEST
ESM[SW]:SPU.CONFIG:REG_MONITOR_TEST
ESM[SW]:SPU.FFT:REG_MONITOR_TEST
ESM[SW]:TRACE.TRAM:REG_MONITOR_TEST
ESM[SW]:VMT:MBIST
SMC[SW]:VMT:MBIST