AN14721

How to Understand and Use TRDC in i.MX Devices

Rev. 1.0 — 30 June 2025

Application note

Document information

Information	Content
Keywords	AN14721, i.MX, TRDC, resource isolation, security
Abstract	Resource isolation plays an important role in functional safety and security. In terms of functional safety, resource isolation can reduce the failure impact between different domains. In terms of security, resource isolation can protect sensitive data.

1 Introduction

Resource isolation plays an important role in functional safety and security. In terms of functional safety, resource isolation can reduce the failure impact between different domains. In terms of security, resource isolation can protect sensitive data. Starting from NXP's i.MX 8ULP and i.MX 9 series chips, there are two resource-isolation mechanisms: one is the MIX hardware design method, and the other is the Trusted Resource Domain Controller (TRDC) logic-isolation method. In the hardware design of the SoC, i.MX 9 chips are divided into multiple MIXs. For example, i.MX 95 contains AONMIX, ANAMIX, WAKEUPMIX, and others. All MIXs are separated from one another on the die because they are designed as separate modules and integrated at the SoC level. A failure in a non-safety-relevant MIX does not directly affect the safety-relevant MIX. It helps to achieve functional safety. However, the design of the MIX is fixed in terms of hardware and the hardware resources within each MIX cannot be adjusted. The TRDC provides a more flexible method of resource isolation, which can be customized and configured by developers to implement any resource-access policies.

2 Rationale

The Trusted Resource Domain Controller (TRDC) consists of three parts: Domain Assignment Controller (DAC), Memory Block Checker (MBC), and Memory Region Checker (MRC). The roles and functions of these three parts in the resource access process are shown in Figure 1.

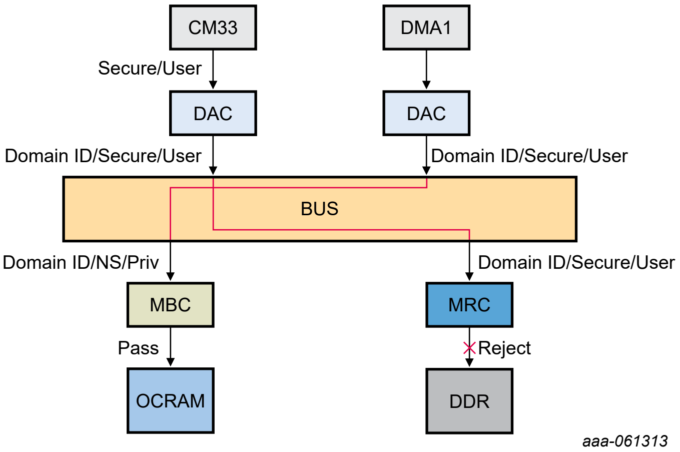


Figure 1. TRDC access process

When a specific master accesses a certain resource, the access process can be divided into the following steps:

1. The Domain Assignment Controller (DAC) assigns attributes to the master, including the domain ID (DID), privileged mode, and secure status.
2. The master access signal reaches the Memory Block Checker (MBC) or Memory Region Checker (MRC) via the system bus.
3. The MBC or MRC checks whether the access permissions are in line with the configuration based on the master attributes and access types (read, write, execute).
4. If the permissions are granted by the configuration, the access is successful. Otherwise, the access is denied.

2.1 DAC

The Domain Assignment Controller (DAC) is mainly used to assign attributes to the master. A master refers to a bus master that can issue data transactions, which can be classified into processor and non-processor, such as Arm Cortex A55 (CA55), Arm Cortex M33 (CM33), DMA, and so on.

Three attributes are assigned:

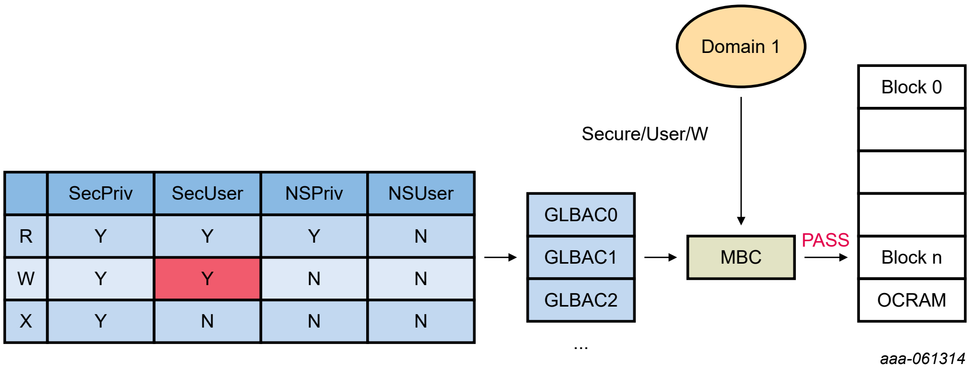
1. **DID (domain ID)**
   The DID is an attribute for dividing logical domains. Masters with the same DID are masters within the same domain. The range of the DID is from 0 to 15. Every master has a default DID, which can be obtained from the respective SoC reference manuals.
2. **Privileged mode**
   In the Arm system, all modes except the user mode are privileged modes. The DAC can reconfigure this attribute of the master, such as setting the attribute to "User" or "Privileged", or directly use the master's attribute.
3. **Secure status**

The secure status originates from the Arm TrustZone technology, including secure and non-secure states. The DAC can also configure this attribute of the master such as secure or non-secure or directly use the master's attribute.

When the master access signal is processed by the DAC and the DID, the privilege mode and secure status for the master are determined.

2.2 MBC

The Memory Block Checker (MBC) is mainly used for checking the access rights of internal resources. Internal resources include the memory and peripherals, such as AIPS, OCRAM, and so on. Each resource is divided into multiple resource blocks according to a fixed granularity. The MBC mechanism is shown in Figure 2.



	SecPriv	SecUser	NSPriv	NSUser
R	Y	Y	Y	N
W	Y	Y	N	N
X	Y	N	N	N

Figure 2. MBC access control

The principle of access checking is as follows:

1. Configure a set of the Global Access Control (GLBAC).
   Eight GLBACs are available. Each GLBAC contains permission settings for 12 different combinations of access modes, including: secure/non-secure, privileged/user, read/write/execute.
2. For the block of a certain resource, select a certain GLBAC for each DID.
3. According to the master attributes and GLBAC, the MBC checks the master access permission.
4. If the permission is granted by the configuration, the access is successful. Otherwise, the access is denied.

2.3 MRC

The Memory Region Checker (MRC) is used for checking the access rights of external resources. External resources are usually the external memory, such as DRAM, FlexSPI, and others. These resources can be divided into multiple regions of different sizes. The MRC mechanism is shown in Figure 3.

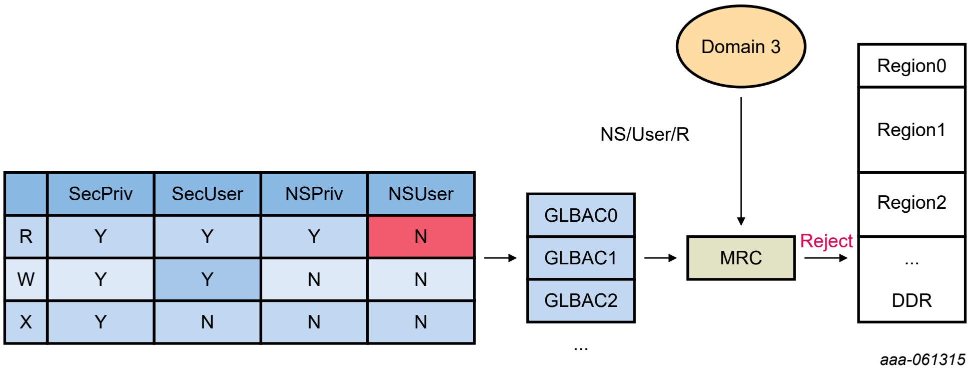


Figure 3. MRC access control

The principle of access checking is as follows:

1. Configure a set of Global Access Control (GLBAC).
2. Divide a certain resource into regions and select a certain GLBAC for each DID.
   The setting of the region size is completely determined by the user and has no fixed value. Usually, you must set the start address and the end address.
3. According to the master attributes and GLBAC, the MRC checks the master access permission.
4. If the permission is granted by the configuration, the access is successful. Otherwise, the access is denied.

The principle of the MRC is similar to that of the MBC. The difference is that the MRC manages permissions by region and the size of the region is configurable. The MBC is divided into blocks and the block size is fixed.

3 TRDC usage

The following section describes how to use the TRDC for three aspects: registers, configuration software, and configuration tools.

3.1 Registers

Registers are the most direct configuration method. In the TRDC, due to the large number of configuration registers, you must learn how to determine the corresponding register positions.

3.1.1 DAC

The DAC is used to assign the DID and other attributes for masters. Each master has a default DID value. For example, in i.MX 93, the default DID value allocation is shown in Table 1.

Table 1. Default DID in i.MX 93

Default DID	Masters
0	EdgeLock Secure Enclave-AP
1	MTR_MSTR
2	CM33__I, CM33_S

Table 1. Default DID in i.MX 93*...continued*

Default DID	Masters
3	CA55, GIC600

If you want to change the default DID or other attributes, such as Privileged/User and Secure/non-secure, find the DAC address of the required master in the chip and write the corresponding register value.

For example, if you want to change the DID of the CA55 in i.MX 93 to 4, the privilege mode attribute follows the master, and the secure status is fixed as secure, then perform the following steps:

1. Find the MDAC location of the CA55:
   The following information is in the MDAC configuration table, in the TRDC chapter:

Table 2. CA55 MDAC information in i.MX 93

Master	MIX	Master index	Number of DAC registers
CA55 read channel	NICMIX	0	4
CA55 write channel		1	4

2. Find the register location:
   The NICMIX TRDC base address is 0x49010000.
   The corresponding DAC register is MDA_W(r)_(m)_DFMT(n), where r is the number of registers, m is the master index, and n is the master type.

Table 3. CA55 DAC registers in i.MX 93

Master	DAC register	Offset
CA55 read channel	MDA_W0_0_DFMT0	0x800
	MDA_W1_0_DFMT0	0x804
	MDA_W2_0_DFMT0	0x808
	MDA_W3_0_DFMT0	0x80C
CA55 write channel	MDA_W0_1_DFMT0	0x820
	MDA_W1_1_DFMT0	0x824
	MDA_W2_1_DFMT0	0x828
	MDA_W3_1_DFMT0	0x82C

3. Write the register:

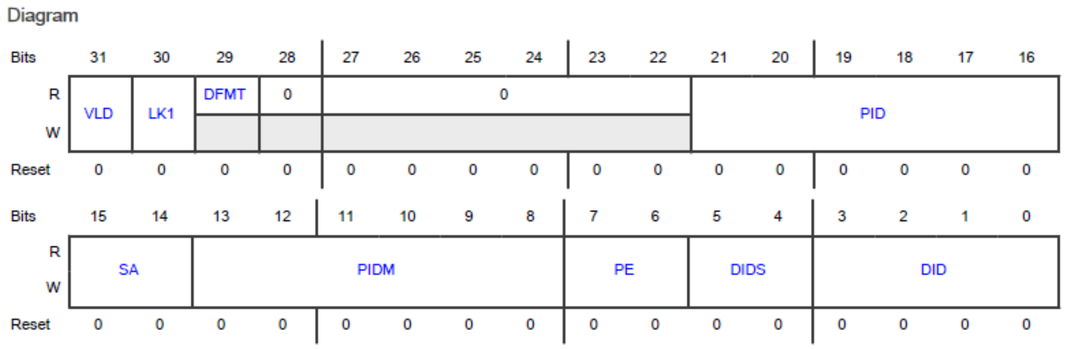


Figure 4. MDA_Wr_m_DFMT0 register

The MDAC register description is in the memory map of the TRDC chapter of the reference manual.

If you want the DID of the CA55 to be 4, the privilege mode attribute follows the host, and the secure status is fixed as secure, then: DID=4, SA=0, VLD=1.

In the remaining bit domains, PE, PIDM, and PID are used to dynamically configure the DID. When these functions are not in use, you can configure them all to 0. For a detailed explanation, see the functional description in the TRDC chapter. LK1 can lock the register to prevent subsequent changes and set it to 0 when not in use.

Therefore, the register value to write is 0x80000004.

Since both channels of the CA55 require the DID configuration and only one register configuration is needed for each, the following applies:

CA55 read channel:

```
MDA_W0_0_DFMT0: write 0x80000004 to address 0x49010800
```

CA55 write channel:

```
MDA_W0_1_DFMT0: write 0x80000004 to address 0x49010820
```

3.1.2 MBC

The MBC checks the access of the internal resources of the chip. After the DAC configuration completes, the master identifier is replaced by the DID.

For example, in i.MX 93, you must implement the following:

The master with DID=3 only in the SP (secure privilege) and SU (secure non-privilege) states has the read/write/execute permissions for the OCRAM segment from 0x20500000 to 0x2050FFFF.

The master with DID=5 has all access permissions for this segment of OCRAM.

Then:

1. Find the OCRAM information in the MBC configuration table.

   Table 4. OCRAM MBC information in i.MX 93

MIX	MBC instance	Port number	Peripherals	Block number	Block size
NICMIX	3	0	OCRAM	40	16 kB
		1	OCRAM	40	16 kB

   There are two OCRAM ports because the OCRAM is accessed via the AXI bus, which has two separate read and write access channels. Therefore, SLV0 corresponds to the OCRAM read channel, and SLV1 corresponds to the OCRAM write channel.
2. Find and configure a set of GLBAC:
   The NICMIX TRDC base address is 0x49010000.

   Table 5. GLBAC

Register	Offset
MBC3_MEMN_GLBAC0	0x16020
MBC3_MEMN_GLBAC1	0x16024
MBC3_MEMN_GLBAC2	0x16028
MBC3_MEMN_GLBAC7	0x1603C

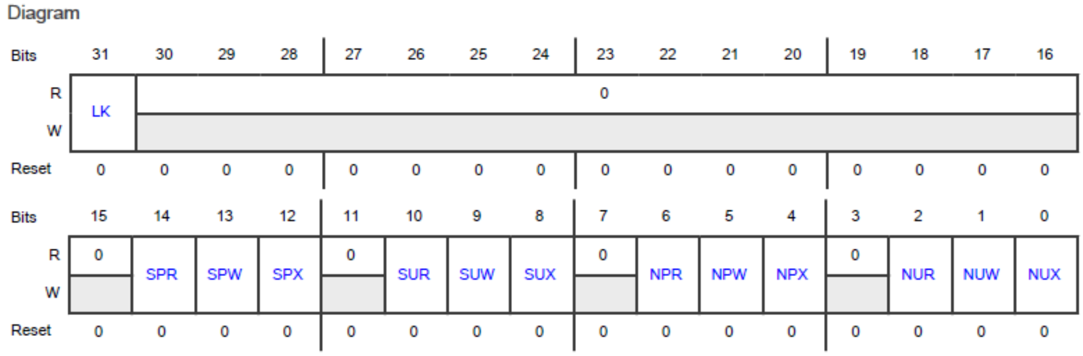


Figure 5. MBC_MEMN_GLBAC register

Since one domain must have the read/write/execute permissions only in the SP (secure privilege) and SU (secure non-privilege) states while the other domain has all access permissions, set the GLBAC as follows:

GLBAC0:

```
Write 0x7700 to address 0x49026020: SPR=1, SPW=1, SPX=1, SUR=1, SUW=1, SUX=1.
```

GLBAC1:

```
Write 0x7777 to address 0x49026024: All access granted.
```

3. Find and write the MBC control register where the OCRAM block is located:
   a. OCRAM configuration segment: 0x20500000~0x2050FFFF.
   b. OCRAM in i.MX 93: 0x20480000~0x2051FFFF.
   c. Block number: 40.
   d. Block size: 16 kB (0x400).
4. Corresponding OCRAM blocks:
   a. Start block: (0x20500000-0x20480000)/0x4000=32.
   b. End block: (0x2050FFFF-0x20480000)/0x4000=35
5. The MBC configuration registers are MBC[m]_DOM[d]_MEM[s]_BLK_CFG_W[w]:
   a. m is the instance number of MBC, and OCRAM corresponds to 3.
   b. d is the DID. You must configure DID=3 and DID=5.
   c. s is the memory port number. OCRAM corresponds to 0 and 1.
   d. w is the configuration word number. Each word is used for eight memory blocks.

Since the OCRAM configuration block is [32:35], w1=32/8=4 with a remainder of 0, and w2=35/8=4 with a remainder of 3. Therefore, you must configure block [0:3] in word4, which corresponds to bit [0:15].

Table 6. MBC OCRAM registers

DID	Register	Address	Value [0:15]
3	MBC3_DOM3_MEM0_BLK_CFG_W4	0x49026650	0x0000[1]
	MBC3_DOM3_MEM1_BLK_CFG_W4	0x49026790	
5	MBC3_DOM5_MEM0_BLK_CFG_W4	0x49026A50	0x9999[2]
	MBC3_DOM5_MEM1_BLK_CFG_W4	0x49026B90	

[1] 0x0000: Use GLBAC0 and NSE=0, which means no access for the non-secure status.

[2] 0x9999: Use GLBAC1 and NSE=1, which means that the access of the non-secure status is granted.

3.1.3 MRC

MRC is used for the access control of external resources.

For example, in i.MX 93, a master with DID=3 only in the secure state (SP/SU) can perform the read/write/execute accesses to the DDR region from 0x80000000 to 0x9FFFFFFF.

Then:

1. Find the DDR information in the MRC table:

Table 7. DRAM MRC information

MIX	MRC instance	Slave memory	MRC descriptors
NICMIX	0	DRAM	16

2. Find and configure the GLBAC:
   The base address of the NICMIX TRDC is 0x49010000.

Table 8. GLBAC

Register	Offset
MRC0_MEMN_GLBAC0	0x18020
MRC0_MEMN_GLBAC1	0x18024
MRC0_MEMN_GLBAC2	0x18028
MRC0_MEMN_GLBAC7	0x1803C

   Since one domain is required to have the read/write/execute access only in the SP (secure privilege) and SU (secure non-privilege) states, set the GLBAC0 as follows:
   Write 0x7700 to address 0x49028020: SPR=1, SPW=1, SPX=1, SUR=1, SUW=1, SUX=1, others are 0.
3. Write the MRC configuration registers:
   The MRC configuration register is MRC[m]_DOM[d]_RGD[r]_W[w], where:
   a. m is the instance number, 0.
   b. d is DID, DID=3.
   c. r is the required region number. Configure one area, r=0.
   d. w is the configuration word number. Word0 is used to specify the start address and select the GLBAC, while word1 is used to specify the end address and perform other functions.

Table 9. MRC DRAM configuration

DID	Register	Offset	Address	Value [0:15]
3	MRC0_DOM3_RGD0_W0	0x18340	0x49028340	0x80000000[1]
	MRC0_DOM3_RGD0_w1	0x18344	0x49028344	0x9FFFC011[2]

[1] 0x80000000: Start address=0x80000000, use GLBAC0.
[2] 0x9FFFC011: End address=0x9FFFFFFF, NSE=1, no access for non-secure, VLD=1.

3.2 Configuration software

If the chip does not use the System Manager software, such as i.MX 93 and i.MX 91, the TRDC can be configured by the Arm Trusted Firmware (ATF) software.

If the chip uses the System Manager software, such as i.MX 95 and i.MX 943, the TRDC can be configured by the System Manager software.

3.2.1 ATF

ATF provides the reference implementation of the secure world software. It contains the TRDC configurations.

In ATF, the TRDC is configured by the `plat/imx/{SOC name}/trdc_config.h` header file. This file lists the MBC and MRC settings in each MIX, including the GLBAC, MBC, and MRC configuration tables.

3.2.1.1 GLBAC

The GLBAC structure is defined as follows:

```
struct trdc_glbac_config {
        uint8_t mbc_mrc_id; //MBC or MRC instance index
        uint8_t glbac_id;   //GLBAC index
        uint32_t glbac_val; //GLBAC configuration
};
```

For example:

```
/* aonmix */
struct trdc_glbac_config trdc_a_mbc_glbac[] = {
        /* MBC1 */
        { 1, 0, SP(RW)  | SU(RW)   | NP(RW)  | NU(RW) },
};
```

It means that the GLBAC0 in AONMIX MBC1 is set to read/write for all statuses.

If you want other GLBACs, just append them in the corresponding variable array.

3.2.1.2 MBC

The MBC configuration table structure is defined as follows:

```
struct trdc_mbc_config {
        uint8_t mbc_id;        //MBC instance index
        uint8_t dom_id;        //DID
        uint8_t mem_id;        //Memory port index
        uint8_t blk_id;        //Block number
        uint8_t glbac_id       //GLBAC index
        bool secure;           //If set, NSE=0 which means no access for non-
secure status
};
```

For example:

```
/* wakeupmix */
struct trdc_mbc_config trdc_w_mbc[] = {
        { 1, 2, 3, MBC_BLK_ALL, 0, true },
};
```

It means that in the access from DID2 to WAKEUPMIX MBC1 port3, all peripherals are set to GLBAC0. Besides, the non-secure access is not allowed. According to the i.MX 93 TRDC MBC table, it refers to GPIO4.

Note that in this configuration, for the memory block number, a special macro is defined:

```
#define MBC BLK ALL 255
```

If the block number is set to this macro, it indicates that all blocks of this memory port are set to the same access permission. If there are other special memory block configurations in this memory port, you can append them in the configuration table, which overwrites the settings in the MBC BLK ALL. When adding or modifying the MBC configuration, see Section 3.1.2 in the register level to find the corresponding memory block information in the reference manual and append it to the existing configuration list.

3.2.1.3 MRC

The MRC configuration table structure is defined as follows:

```
struct trdc_mrc_config {
        uint8_t mrc_id;  //MRC instance index
        uint8_t dom_id;  //DID
        uint8_t region_id;          //Region number
        uint32_t region_start;      //Region start address
        uint32_t region_size;       //Region end address
        uint8_t glbac_id;           //GLBAC index
        bool secure;                //If set, NSE=0 which means no access for
 non-secure status
};
```

For example:

```
/* wakeupmix */
struct trdc_mrc_config trdc_w_mrc[] = {
        { 1, 2, 0, 0x28000000, 0x08000000, 0, true  },
};
```

The DID2 access to the WAKEUPMIX MRC1 region 0 (0x28000000~0x08000000) is set to GLBAC0 and the non-secure status access is not allowed.

When adding or modifying the MRC configuration, see Section 3.1.3 in the register level to find the corresponding memory area information in the reference manual and append it to the existing configuration list.

3.2.1.4 DAC

The DAC is configured by the ATF API interfaces, such as:

```
/* Set MTR to DID1 */
trdc_mda_set_noncpu(0x44270000, 4, 0, false, 0x2, 0x2, 0x1);
```

For non-processor masters, the input parameters are, respectively: TRDC base address, master number, MDAC register number, DIDB, SA, PA, and DID.

```
/* Set M33 to DID2*/
trdc_mda_set_cpu(0x44270000, 1, 0, 0x2, 0x0, 0x2, 0x0, 0x0, 0x0);
```

For the processor masters, the input parameters are, respectively: TRDC base address, master number, MDAC register number, SA, DIDS, DID, PE, PIDM, and PID.

3.2.2 System Manager

The System Manager (SM) is a low-level system function which runs on a System Control Processor (SCP) to support the isolation and management of power domains, clocks, resets, sensors, pins, and others on complex application processors. It often runs on a Cortex-M processor. The SM is supported on processors like i.MX 943, i.MX 95, and others.

The configuration file of the System Manager is located at `configs/{platform}.cfg`. It includes the configuration of the TRDC. This configuration file is not written in the C language but parsed in Perl. The parsing script is `configs/configtool.pl`. Running "`make config={platform} cfg`" parses the configuration file and generates the corresponding C language configuration header file and other files. The TRDC header file is generated at `configs/{platform}/config_trdc.h`. This configuration file is divided using domains and Logical Machines (LM). Each domain and logical machine contains the corresponding masters and resources that correspond to the configuration of the TRDC.

For example, in the ELE domain:

```
#==========================================================================#
# ELE Domain                                                               #
#==========================================================================#

#allocate domain ID 0 to ELE
DOM0                    did=0

#Configure one GLBAC named DATA, with the permissions being read and write.
DATA:                   perm=rw

# Resources

# Memory

#ELE has the read/write access to the below resources.
M33_TCM_SYS             DATA, begin=0x020200000, size=256K
OCRAM                   DATA, begin=0x020480000, size=352K
DDR                     DATA, begin=0x080000000, end=0x87FFFFFFF
```

The master DAC is configured in the `did=` field.

The GLBAC is defined in the `perm=` field, similar to macro definitions in the C language. The preset permission types are in the `sm/doc/config.md {Configtool Resources}` section. For example, `sec_rw` is equivalent to `GLBAC=0x6600`.

The `# Resources` and `# Memory` sections (respectively) specify the access permission settings of domains or logical machines for specific resources, that is, the MBC and MRC configurations.

For example, if the ELE needs access to the OCRAM on i.MX 95, add the following configuration to the ELE domain of `configs/mx95evk.cfg`:

```
 M33_TCM_SYS            DATA, begin=0x020200000, size=256K
 OCRAM                  DATA, begin=0x020480000, size=352K
 DDR                    DATA, begin=0x080000000, end=0x87FFFFFFF
+OCRAM                  DATA, begin=0x020480000, size=352K
```

The OCRAM is defined in `devices/MIMX95/configtool/nocmix.cfg`.

```
#==========================================================================#
# Memories                                                                 #
#==========================================================================#

OCRAM:                  PD_NOC, MBC_N2=0, MBC_N2=1, origin=0x20480000, \
                        nblks=22, blksize=16K
```

3.3 Configuration tool

NXP also provides the MCUXpresso Config Tools to configure the TRDC with a GUI interface for some chips. The TEE tool, which is used to generate the TRDC configuration, is a submodule in the MCUXpresso Config Tools. In the TEE tool, each module of the TRDC can be configured in the GUI interface. After the configuration is completed, the configuration header file can be exported to replace the TRDC configuration header file in the ATF or the TRDC source code file in the M-core SDK. The detailed user manual is in the installation location of the TEE tool.

4 Acronyms and abbreviations

Table 10. Acronyms and abbreviations

Acronym	Definition
ELE-AP	EdgeLock Secure Enclave, also called EdgeLock Secure Enclave (Advanced Profile) (ELE-AP)
TRDC	Trusted Resource Domain Controller
MBC	Memory Block Checker
MRC	Memory Region Checker
GLBAC	Global Access Control
ATF	Arm Trusted Firmware
SM	System Manager
TEE	Trusted Execution Environment

5 References

- *i.MX 9x Reference Manual* (available at www.nxp.com)
- *System Manager Document* (available at https://github.com/nxp-imx/imx-sm)

6 Note about the source code in the document

Example code shown in this document has the following copyright and BSD-3-Clause license:

Copyright 2025 NXP Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN

7 Revision history

Table 11. Revision history

Document ID	Release date	Description
AN14721 v.1.0	30 June 2025	• Initial version

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Contents

1 Introduction 2
2 Rationale 2
2.1 DAC 3
2.2 MBC 3
2.3 MRC 4
3 TRDC usage 4
3.1 Registers 4
3.1.1 DAC 4
3.1.2 MBC 6
3.1.3 MRC 8
3.2 Configuration software 8
3.2.1 ATF 9
3.2.1.1 GLBAC 9
3.2.1.2 MBC 9
3.2.1.3 MRC 10
3.2.1.4 DAC 10
3.2.2 System Manager 10
3.3 Configuration tool 12
4 Acronyms and abbreviations 12
5 References 12
6 Note about the source code in the document 12
7 Revision history 13
Legal information 14