恩智浦半导体
应用笔记

文档编号：AN12077
第 2 版，2019 年 9 月

使用 i.MX RT FlexRAM

1. 简介

本文档介绍了 i.MX RT MCU 提供的灵活存储器阵列。本文档的第一部分概述了 FlexRAM 存储器的所有特性，包括：

- 存储库阵列配置。
- 存储器类型大小定义。
- 可用存储器控制器。
- 电源域和时钟。
- 中断请求生成。

本文档的第二部分演示了在特定应用用例上的 FlexRAM 配置使用。该部分阐明了在应用中充分利用 i.MX RT1050 MCU 的 FlexRAM 存储器所需考虑的注意事项。它重点介绍在应用正常运行时从性能角度考虑的应用存储器能力，以及低功耗特性的实现。

目录

2. FlexRAM 存储器

FlexRAM 是高度可配置且极其灵活的 RAM 存储器阵列。该存储器阵列包含可以独立配置为通过不同类型的接口进行访问的存储库，例如 I-TCM（指令紧密耦合存储器）、D-TCM（数据紧密耦合存储器）或 AXI（系统）。存储库可以用作 ITCM、DTCM 或 OCRAM 存储器。可将多达三个不同的电源域分配给专用存储库或存储库组（PDRET、PDRAM0 和 PDRAM1），这可以降低低功耗模式下的功耗。

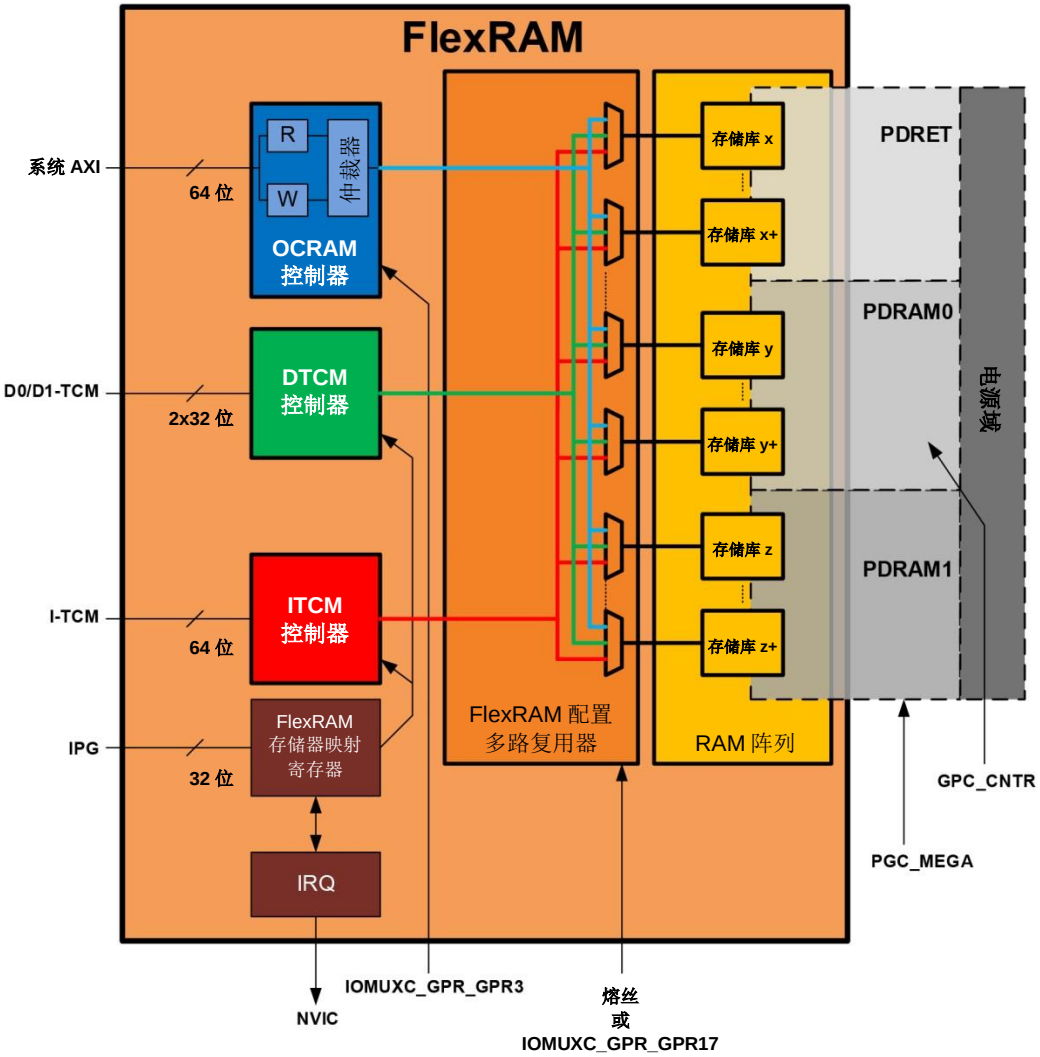

图 1. FlexRAM 的通用框图

备注

在某些 IMXRTxxxx 器件中，会有一个附加 OCRAM（它不是 FlexRAM 的一部分）。它用来增加总的片上存储器大小。由于 FlexRAM 阵列中不包含这种存储器，因此在本文档中未考虑该存储器。

2.1. FlexRAM 配置

FlexRAM 是一个可配置存储器 RAM 阵列，其中包含多个存储库。

2.1.1. FlexRAM 存储库配置

FlexRAM 阵列中的每个存储库均可以配置用作：

- 由 64 位 I-TCM 接口访问的 I-TCM（指令紧密耦合存储器）。
- 由两个 32 位（D0 和 D1）TCM 接口以交错方式访问的 D-TCM（数据紧密耦合存储器）。
- 由 64 位系统 AXI 总线访问的 OC RAM（片上 RAM 存储器）。

备注

所有 TCM 接口的运行频率与 Arm® Cortex®-M7 内核相同，并且彼此同步。

OCRAM 控制器通过 64 位 AXI 总线连接到互连总线结构(NIC)的一个从机端口。该从机端口的频率限定值为内核频率的¼。例如，如果 Arm Cortex-M7 内核的运行频率为 528 MHz，则连接到 OCRAM 控制器的 AXI 总线频率限定为 132 MHz。与 xTCM 存储器比较，预期对 OCRAM 的数据访问性能会降低。L1 缓存存储器可帮助解决这一问题。

FlexRAM 存储库的配置有两种可选来源：

- FUSE FlexRAM 配置值（默认值）。
- IOMUXC_GPR_GPR17 寄存器中定义的 FLEXRAM_BANK_CFG 字段值。

IOMUXC_GPR_PGR16 寄存器中定义的 FLEXRAM_BANK_CFG_SEL 位的值可进行这两个来源之间的选择。该位默认设置为 0，并将熔丝值用于 FlexRAM 配置。

2.1.1.1. 静态配置

FUSE FlexRAM 存储库配置值表示 FlexRAM 存储库的静态配置，因为在器件启动之后此项无法更改。FUSE FlexRAM 配置值使用熔丝图中位于 0x6D0 地址第[16-19]位位置的熔丝（熔丝被称为 Default_FlexRAM_Part）。表 1 中的示例显示了基于相应器件熔丝设置的可用 FlexRAM 存储库配置。空白器件的值设置为 0000，表示默认 FlexRAM 配置 0。

备注

OCRAM 的最低配置为 64 KB（参见表 1）。这是必不可少的，因为 ROM 代码的执行至少需要 64 KB RAM。OCRAM 的最低要求取决于器件。

表 1. 由 RT1010 中的熔丝定义的静态 FlexRAM 配置

	FUSE FlexRAM 配置值	IOMUXC_GPR_GPR17 (FLEXRAM_BANK_CF) （二进制）	存储库				OCRAM [kB]	DTCM [kB]	ITCM [kB]
	0x6D0 [19:16]		0	1	2	3			
0	0b0000	11100101	*O*	*O*	*D*	*I*	64	32	32
1	0b0001	11101001	*O*	*D*	*D*	*I*	32	64	32
2	0b0010	10100101	*O*	*O*	*D*	*D*	64	64	0
3	0b0011	10101001	*O*	*D*	*D*	*D*	32	96	0
4	0b0100	11111001	*O*	*D*	*I*	*I*	32	32	64
5	0b0101	01100101	*O*	*O*	*D*	*O*	96	32	0
6	0b0110	11111101	*O*	*I*	*I*	*I*	32	0	96
7	0b0111	11110101	*O*	*O*	*I*	*I*	64	0	64
8	0b1000	01110101	*O*	*O*	*I*	*O*	96	0	32
9	0b1111	01010101	*O*	*O*	*O*	*O*	128	0	0
	O - OCRAM、D - DTCM、I - ITCM								

表 2. 由 RT1020 中的熔丝定义的静态 FlexRAM 配置

	FUSE FlexRAM 配置值	IOMUXC_GPR_GPR17 (FLEXRAM_BANK_CFG) （二进制）	存储库								OCRAM [kB]	DTCM [kB]	ITCM [kB]
	0x6D0 [19:16]		0	1	2	3	4	5	6	7			
0	0b0000	0101111110100101	*O*	*O*	*D*	*D*	*I*	*I*	*O*	*O*	128	64	64
1	0b0001	1111101010100101	*O*	*O*	*D*	*D*	*D*	*D*	*I*	*I*	64	128	64
2	0b0010	0101101010100101	*O*	*O*	*D*	*D*	*D*	*D*	*O*	*O*	128	128	0
3	0b0011	1110101010010101	*O*	*O*	*O*	*D*	*D*	*D*	*D*	*I*	96	128	32
4	0b0100	1111111110100101	*O*	*O*	*D*	*D*	*I*	*I*	*I*	*I*	64	64	128
5	0b0101	1010101010100101	*O*	*O*	*D*	*D*	*D*	*D*	*D*	*D*	64	192	0
6	0b0110	0101011110100101	*O*	*O*	*D*	*D*	*I*	*O*	*O*	*O*	160	64	32
7	0b0111	0101010110100101	*O*	*O*	*D*	*D*	*O*	*O*	*O*	*O*	192	64	0
8	0b1000	0101111101100101	*O*	*O*	*D*	*O*	*I*	*I*	*O*	*O*	160	32	64
9	0b1001	1111111101100101	*O*	*O*	*D*	*O*	*I*	*I*	*I*	*I*	96	32	128
10	0b1010	1111111111100101	*O*	*O*	*D*	*I*	*I*	*I*	*I*	*I*	64	32	160
11	0b1011	0101010101100101	*O*	*O*	*D*	*O*	*O*	*O*	*O*	*O*	224	32	0
12	0b1100	1111111101010101	*O*	*O*	*O*	*O*	*I*	*I*	*I*	*I*	128	0	128
13	0b1101	0101011101100101	*O*	*O*	*D*	*O*	*I*	*O*	*O*	*O*	192	32	32
14	0b1110	1111111111110101	*O*	*O*	*I*	*I*	*I*	*I*	*I*	*I*	64	0	192
15	0b1111	0101010101010101	*O*	*O*	*O*	*O*	*O*	*O*	*O*	*O*	256	0	0
			O - OCRAM、D - DTCM、I - ITCM										

表 3. 由 RT1050 / RT106x 中的熔丝定义的静态 FlexRAM 配置

	FUSE FlexRAM 配置值	IOMUXC_GPR_GPR17 (FLEXRAM_BANK_CFG) （二进制）	存储库																OCRAM [kB]	DTCM [kB]	ITCM [kB]
	0x6D0 [19:16]		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15			
0	0b0000	01010101101011111111101001010101	O	O	O	O	D	D	I	I	I	I	D	D	O	O	O	O	256	128	128
1	0b0001	01010101010110101111101001010101	O	O	O	O	D	D	I	I	D	D	O	O	O	O	O	O	320	128	64
2	0b0010	01011010111111111111111110100101	O	O	D	D	I	I	I	I	I	I	I	I	D	D	O	O	128	128	256
3	0b0011	01010101010101011110101010010101	O	O	O	D	D	D	D	I	O	O	O	O	O	O	O	O	352	128	32
4	0b0100	01010101010111111111101001010101	O	O	O	O	D	D	I	I	I	I	O	O	O	O	O	O	320	64	128
5	0b0101	01010101010101011111101001010101	O	O	O	O	D	D	I	I	O	O	O	O	O	O	O	O	384	64	64
6	0b0110	01010101111111111111111110100101	O	O	D	D	I	I	I	I	I	I	I	I	O	O	O	O	192	64	256
7	0b0111	11111111111111111111111111110101	O	O	I	I	I	I	I	I	I	I	I	I	I	I	I	I	64	0	448
8	0b1000	01011010101011111111101010100101	O	O	D	D	D	D	I	I	I	I	D	D	D	D	O	O	128	256	128
9	0b1001	01010101101010101111101010100101	O	O	D	D	D	D	I	I	D	D	D	D	O	O	O	O	192	256	64
10	0b1010	10101010111111111111111110100101	O	O	D	D	I	I	I	I	I	I	I	I	D	D	D	D	64	192	256
11	0b1011	10101010101010101010101010100101	O	O	D	D	D	D	D	D	D	D	D	D	D	D	D	D	64	448	0
12	0b1100	01010101010101011111111101010101	O	O	O	O	I	I	I	I	O	O	O	O	O	O	O	O	384	0	128
13	0b1101	01010101010101010101011110010101	O	O	O	D	I	O	O	O	O	O	O	O	O	O	O	O	448	32	32
14	0b1110	01010101010111111111111111110101	O	O	I	I	I	I	I	I	I	I	O	O	O	O	O	O	256	0	256
15	0b1111	01010101010101010101010101010101	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	512	0	0
			O - OCRAM、D - DTCM、I - ITCM																		

2.1.1.2. 运行时配置

通过更改 IOMUXC_GPR_GPR17 寄存器中定义的 FLEXRAM_BANK_CFG 字段的值，还可以在运行时配置 FlexRAM 存储库。将 IOMUXC_GPR_GPR16 寄存器中定义的 FLEXRAM_BANK_CFG_SEL 位设置为 1 时，可以实现此功能。在这种情况下，FlexRAM 存储库的配置直接取决于 IOMUXC_GPR_GPR17 寄存器中定义的 FLEXRAM_BANK_CFG 字段的值，并且可在应用运行时进行更改。

FlexRAM 存储库配置值(FLEXRAM_BANK_CFG)是一个 32 位值。该 32 位值包含每个 FlexRAM 存储库的配置值。每个 FlexRAM 存储库使用两位进行配置：

- 00b—存储库未使用。
- 01b—存储库配置为 OCRAM。
- 10b—存储库配置为 DTCM。
- 11b—存储库配置为 ITCM。

表 1、表 2 和表 3 中的第三列显示对于由 eFUSE 完成的静态配置 FLEXRAM_BANK_CFG 字段可能的配置值。

备注

建议在从熔丝定义存储库配置进行切换之前，即在 FLEXRAM_BANK_CFG_SEL 位设置为 1 之前，将相应的 FlexRAM 存储库配置值(FLEXRAM_BANK_CFG)写入 IOMUXC_GPR_GPR17 寄存器。

这种存储器配置方式可能会影响应用的安全性（堆栈溢出、无效指令执行、访问超出范围等）。应适当考虑代码/数据边界位置，以避免应用崩溃。另外，重新配置中的存储库正在更改其访问接口，但仍包含相同的数据。OCRAM 配置至少需要 64 KB，因为 ROM 代码需要这一部分的 RAM 用于执行（堆栈/静态数据）。

使用建议：

- 强烈建议从与 ITCM/DTCM/FlexRAM 专用 OCRAM 不同的存储器类型执行重新配置 FlexRAM 的代码。例如，从 FlexSPI 串行 NOR 闪存执行代码，并从 SDRAM/非 FlexRAM 专用 OCRAM 访问数据。因此，要求在重新配置期间无需访问 FlexRAM 阵列。有关更多详细信息，请参见第 3.1.3 节“软件实现”。
- 要求在应用启动初始阶段，在首次访问堆栈之前，先执行重新配置 FlexRAM 的代码。
- 请避免在 FlexRAM 重新配置期间启用中断。进入/退出中断服务例程需要从堆栈自动推出/弹出，并从专用存储器中获取向量。这有可能导致访问未定义的存储器空间，虽然它不一定会在重新配置期间导致问题，但之后会在应用运行中导致问题。
- 应用程序流程不再需要重新配置的存储库中的数据/代码。
- 在访问之前，应使用初始化器数据/代码重新填充重新配置的存储库（预期其中没有堆栈/堆）。
- 可寻址存储器空间已改变（避免潜在的应用指针访问该空间）。
- 要求在这部分代码执行过程中不能访问 FlexRAM。有关更多详细信息，请参见第 3.1.3 节“软件实现”。

2.1.2. 存储器类型大小的定义

FlexRAM 阵列中的每个存储库具有相同的存储器大小，可以通过下式计算：

存储库大小 =（总 Flex RAM 大小）/（FlexRAM 阵列中的存储库数量）

例如，iMXRT1050 MCU 具有：

- FlexRAM 总大小为 512 KB。
- FlexRAM 阵列中的 16 个存储库（block0-block15）。
- 存储库大小为 512 KB / 16 = 32 KB。

通过下式很容易计算出专用存储器类型(ITCM/DTCM/OCRAM)的大小：

存储器类型大小 =（配置为专用存储器类型的存储库数量）（存储库大小）*

存储器类型大小严格取决于应用需求，其总量变化范围为 0 B 到 FlexRAM 总大小（根据配置）。例如，在 iMXRT1050 上，它的范围可以从 0 到 512 KB。指定的存储器类型(ITCM/DTCM/OCRAM)无需在 FlexRAM 存储库的连续块中严格配置。

表 4. DTCM/OCRAM 存储库的非连续块配置示例(i.MX RT1050)

IOMUXC_GPR_GPR17 (FLEXRAM_BANK_CFG)	存储库																OCRAM	D-TCM	I-TCM
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15			
01011010111111111111111110100101	O	O	D	D	I	I	I	I	I	I	I	I	D	D	O	O	128	128	256

但是，它仍然在器件存储器映射中表示为连续地址空间（参见表 5）。

表 5. 存储器类型地址空间示例

存储器类型	大小 [kB]	地址空间	
		起始	结束
OCRAM	128	0x20200000	0x20220000
D-TCM	128	0x20000000	0x20020000
I-TCM	256	0x00000000	0x00040000

此特性让您可以根据应用的功耗需求配置存储器类型及其大小（更多详细信息请参见下一部分）。

备注

由于启动 ROM 代码的要求，无法将 OCRAM 配置为 0 kB。64 KB OCRAM 表示最低系统要求。ITCM 或 DTCM 可以配置为 0 KB（另请参见表 1 所示的可能静态配置）。

Arm Cortex-M7 规范要求 ITCM/DTCM 的大小为 2 的幂数，这会与 FlexRAM 配置功能冲突（参见表 1 中的配置 7、10、11）。请避免访问由相应 FlexRAM 配置进行配置的空 RAM 空间，或使用将 TCM 的大小定义为 2 的幂数这种建议方式，并通过将其写入 IOMUXC_GPR_GPR14->CM7_CFGxTCMSZ 中的相应字段以反映出来（这将相应更新 CM7_xTCMCR）。

如果请求的 ITCM/DTCM 大小为 0 字节，则在 IOMUXC_GPR_GPR14->CM7_CFGxTCMSZ 中将大小配置为 0 字节之前，先在 IOMUXC_GPR_GPR16->INIT_xTCM_EN 中禁用相应的 TCM。

2.2. FlexRAM 存储器控制器

FlexRAM 包含的存储器控制器负责将 AXI (OCRAM)或 TCM（I-TCM、D-TCM）接口信号转换为 RAM 阵列接口信号。每个存储器控制器和每个 RAM 阵列存储库之间都配备了多路复用器（参见图 1 中的 FlexRAM 配置多路复用器模块），它们负责根据 FlexRAM 配置正确连接存储器控制器及其 RAM 存储库。这些存储器控制器还控制对存储器的访问权限。

2.2.1. TCM 存储控制器

TCM 控制器将 TCM（64 位 I-TCM 总线或 2x32 位 D-TCM 总线）接口信号转换为 RAM 阵列信号。TCM 接口与 Cortex-M7 内核同步，运行频率相同。TCM 控制器还可以控制对 RAM 存储库的访问，并可影响存储器数据访问时间（获取 I-TCM 上的指令或访问 D-TCM 上的数据）。可以为读取和写入访问选择两种模式：

- 快速访问模式（默认）—预期在一个周期内完成访问。
- 等待访问模式—预期在两个周期内完成访问。

这可以通过启用/禁用 FlexRAM 存储器映射中 TCM 控制寄存器(TCM_CTRL)中相应的访问位来设置：

- TCM_RWAIT_EN：
  - 0—选择快速模式。
  - 1—选择等待模式。
- TCM_WWAIT_EN：
  - 0—选择快速模式。
  - 1—选择等待模式。

TCM 控制器还包含动态时钟门控制，可降低无访问数据时的功耗。

2.2.2. OCRAM 存储器控制器

OCRAM 存储器用作连接至 64 位系统 AXI 总线的从机端口模块。它包含 OCRAM 控制器，该控制器根据其配置处理 FlexRAM 存储库。OCRAM 控制器将 AXI 接口信号转换为 RAM 阵列信号。读写事务由两个独立的读写控制模块处理。控制器还包含一个仲裁器，当读写模块同时发出两个请求时，该仲裁器进行控制。仲裁器采用轮询机制工作。当目标存储库不同时，读写事务可同时进行。当目标存储库相同时，读访问优先级更高。

OCRAM 控制器还具有在存储器以较高频率运行时避免时序问题的功能。它支持向存储器访问中添加等待状态/管道：

- 读/写地址管道：
  - 此功能启用后，可将读取/写入来自 AXI 主机的地址延迟一个周期，然后再由存储器接受。

- 写入数据管道：
  - 此功能启用后，可将写入来自 AXI 主机的数据延迟一个周期，然后再由存储器接受。
- 读取数据等待状态：
  - 此功能启用后，每次读取访问都需要两个周期。

所有上述功能可以通过通用寄存器(IOMUXC_GPR_GPR3)中相应的 OCRAM 控制位(OCRAM_CTL[3:0])来启用/禁用。

在 IOMUXC_GPR_GPR3 寄存器中的 OCRAM 控制字段(OCRAM_CTL[3:0])中执行任何更改后，建议进行等待，直至相应的 OCRAM 状态位(OCRAM_STATUS[19:16])从 1 变为 0（1 表示配置已更改，但尚未应用）。

备注

OCRAM 控制器从机端口上的 64 位 AXI 总线运行频率为 Cortex-M7 内核频率的四分之一。互连总线结构(NIC)的运行频率与之相同。例如，如果 Cortex-M7 内核的运行频率为 528 MHz，则 OCRAM 接口的运行频率为 132 MHz。预期正在更改并随后用于检查相应 OCRAM 状态位的代码无法从 OCRAM 执行。

2.3. 与 FlexRAM 模块相关的时钟和时钟门

下表总结了与 FlexRAM 模块相关的所有时钟：

表 6. FlexRAM 模块

时钟名称	时钟源			外设接口/寄存器访问时钟			时钟门
	名称	默认频率	最大频率	名称	默认频率	最大频率	
flexram_clk	模块时钟	*6 MHz*	*600 MHz*	ipg_clk_root	*3 MHz*	*150 MHz*	CCM_CCGR3[CG9] - flexram_clk_en
ocram_clk	ocram_exsc_aclk_exsc	*3 MHz*	*150 MHz*	ipg_clk_root	*3 MHz*	*150 MHz*	CCM_CCGR3[CG14] - ocram_clk_en

FlexRAM 时钟(flexram_clk)是对模块进行时控的主时钟，源自 Arm 内核时钟(core_clk)。这些 TCM 由相同的源时钟提供时控。片上 RAM 控制器与系统 AXI 接口完全同步，OCRAM 时钟(ocram_exsc_clk)源自 axi_clk，并将其除以 4。最后是与 FlexRAM 模块相关的外围寄存器，由外设总线（IPG 接口）访问。该器件由外设总线时钟(ipg_clk)进行时控。

备注

互连总线结构(NIC)是一个总线矩阵，用于将总线主机（如 ARM AXIM、DMA、USB、ENET、uSDHC）与总线从机（OCRAM 控制器、ARM AHBS、FlexSPI 模块、SEMC 等）相互连接。它的运行频率为 Arm Cortex-M7 内核频率的四分之一。OCRAM 的运行频率与互连总线结构相同。

2.4. FlexRAM 电源域

FlexRAM 存储库阵列最多可以划分为三个不同的电源子域：

- PDRET 域。
- PDRAM0 域。
- PDRAM1 域。

表 7 显示了不同低功耗模式下的专用电源域状态。

表 7. 不同低功耗模式下的 FlexRAM 阵列电源域

	系统空闲	低功耗空闲	挂起	SNVS
ARM 内核	**WFI**	**WFI**	**掉电**	**OFF**
FlexRAM (PDRET)	*ON*	*ON*	*ON*	*OFF*
FlexRAM (PDRAM0)	*ON*	*ON*	*ON/OFF*	*OFF*
FlexRAM (PDRAM1)	*ON/OFF*	*ON/OFF*	*掉电*	*OFF*

将 FlexRAM 存储库划分为不同的电源域的主要目的是为了在不同电源模式下运行时降低功耗。

表 8 总结了对应各 RT 器件的电源域存储库的分配。

表 8. 各 RT 器件的电源域存储库分配

RT 器件	电源域存储库分配		
	PDRET	PDRAM0	PDRAM1
RT1060/RT1064	-	Bank0-Bank15	-
RT1050	Bank0	Bank1-Bank7	Bank8-Bank15
RT1020	Bank0-Bank7	-	-
RT1010	Bank0-Bank3	-	-

2.4.1. PDRET 电源域

该电源域始终上电。这意味着它已上电并转至挂起模式。PDRET 域断电时的唯一例外为 SNVS，它是一个完全独立的低电源域，通常由单独的电源（锂离子电池）供电。

2.4.2. PDRAM0 电源域

Arm 内核断电后，该域既可以处于上电状态以保留数据，也可以与内核一起断电。

此功能可通用电源控制器接口控制(GPC_CNTR)寄存器中的 FlexRAM PDRAM0 电源门启用(PDRAM0_PGE)位来控制。当此位置位（默认）后，即使 Arm 内核掉电，分配至该域的 FlexRAM 存储库也会保留其内容。当此位清零时，一旦 Arm 内核掉电，PDRAM0 电源域也将断电。

2.4.3. PDRAM1 电源域

该域的电源门控通过电源门控 mega(PGC_MEGA)寄存器控制：

- 上电序列时序控制：
  - 掉电序列时间(PGC_MEGA_PDNSCR)：
    - 在掉电请求和置位隔离之间，由 ISO 值中的 IPG 周期数定义。
    - 在置位隔离和电源切换信号反向之间，由 SW2ISO 值中的 IPG 周期数定义。
  - 上电序列时间(PGC_MEGA_PUPSCR)：
    - 在上电请求和置位电源切换信号之间，由 ISO 值中的 IPG 周期数定义。
    - 在置位电源切换信号和隔离反向之间，由 SW2ISO 值中的 IPG 周期数定义。
- 电源控制(PGC_MEGA_CTRL)：
  - 控制掉电请求信号应将该域的电源打开还是关闭。

备注

必须将即使在挂起模式下仍需保留的关键数据放入分配至 PDRET 电源域的 FlexRAM 存储库中（如果有）。

2.5. FlexRAM 中断

FlexRAM 模块可以基于两个不同的事件生成中断请求：

- 未分配地址访问（地址超出范围）。
- 魔术地址读/写访问命中（并非所有 RT 器件均支持，详情请参见相应的参考手册）。

中断请求信号根据其在中断启用寄存器(INT_SIG_EN)中的配置生成。每种存储器类型(OCRAM/DTCM/ITCM)在此寄存器中都具有各自的专用位，用于启用未分配地址访问或魔术地址访问：

- OCRAM_ERR_SIG_EN/OCRAM_MAM_SIG_EN - 置位时，启用生成 OCRAM 超出地址范围/魔术地址访问中断信号。
- DTCM_ERR_SIG_EN/DTCM_MAM_SIG_EN - 置位时，启用生成 DTCM 超出地址范围/魔术地址访问中断信号。
- ITCM_ERR_SIG_EN/ITCM_MAM_SIG_EN - 置位时，启用生成 ITCM 超出地址范围/魔术地址访问中断信号。

所有这些错误信号都经过 ORed 运算（如果启用），它们会向 NVIC 生成一个编号为 38 的中断请求。

由于所有这些事件只有一个中断向量，因此需要通过读取中断状态寄存器 (INT_STATUS)中的相应位来识别中断服务例程中的专用事件源：

- OCRAM_ERR_STATUS/OCRAM_MAM_STATUS - 置位时，表示生成 OCRAM 超出地址范围/魔术地址访问范围。
- DTCM_ERR_STATUS DTCM_MAM_STATUS - 置位时，表示生成 DTCM 超出地址范围/魔术地址访问范围的 DTCM。
- ITCM_ERR_STATUS/ITCM_MAM_STATUS - 置位时，表示生成 ITCM 超出地址范围/魔术地址访问范围的 ITCM。

每个状态位均可通过对其写入一个 log.1 来清除。

通过启用中断状态启用寄存器(INT_STAT_EN)中的专用存储器类型位，可以置位状态位：

- OCRAM_ERR_STAT_EN/OCRAM_MAM_STAT_EN - 置位时，表示允许 OCRAM 超出地址范围/魔术地址访问中断状态。
- DTCM_ERR_STAT_EN/DTCM_MAM_STAT_EN - 置位时，表示允许 DTCM 超出地址范围/魔术地址访问中断状态。
- ITCM_ERR_STAT_EN/ITCM_MAM_STAT_ENN - 置位时，表示允许 ITCM 超出地址范围/魔术地址访问中断状态。

当存在特定软件定义地址访问命中时，某些 iMX RT 器件还可以生成 FlexRAM 专用中断。

地址和访问类型由用户在魔术地址寄存器中定义。专用存储器类型(ITCM/DTCM/OCRAM)魔术地址寄存器包含两个字段；一个用于定义魔术地址，另一个用于选择访问类型（读取或写入）：

- OCRAM_MAGIC_ADDR：
  - OCRAM_MAGIC_ADDR 字段表示 OCRAM 存储器映射地址空间内的 16 位地址。
  - OCRAM_WR_RD_SEL：
    - 0 - 置位读取访问的中断生成。
    - 1 - 置位写入访问的中断生成。
- DTCM_MAGIC_ADDR：
  - DTCM_MAGIC_ADDR 字段表示 DTCM 存储器映射地址空间内的 16 位地址。
  - DTCM _WR_RD_SEL：
    - 0 - 置位读取访问的中断生成。
    - 1 - 置位写入访问的中断生成。
- ITCM_MAGIC_ADDR：
  - ITCM_MAGIC_ADDR 字段表示 ITCM 存储器映射地址空间内的 16 位地址。

- ITCM_WR_RD_SEL：
  - 0 - 置位读取访问的中断生成。
  - 1 - 置位写入访问的中断生成。

3. 在应用中使用 FlexRAM 功能

在 i.MX RT 器件上实现的 FlexRAM 存储器相比具有单个片上存储器的器件，具有明显的优势。根据应用存储器占用空间（代码/恒定数据/静态数据/堆栈等），可以重新配置片上存储器，使器件更适合于应用。这一功能非常有用，例如对于在相同硬件平台上构建的不同应用。如今，这种方法极具吸引力，因为它可以节省开发时间/成本/生产时间，并可快速缩短上市时间。

下面的小节将从应用角度介绍 FlexRAM 的灵活性。

从时间角度来看，该存储器专注于一种特定的配置技术。它展示了 FlexRAM 存储器的静态和动态配置组合。不能将其称为静态配置，因为它不使用熔丝默认配置，该默认配置在器件启动后立即生效。由于应用用例所需熔丝表（参见表 1）中的配置不可用，因此无法使用此方法。该存储器也不是真正的动态配置，因为它无法在应用运行时更改配置。它在应用启动时（在调用静态数据和读写代码初始化之前）配置 FlexRAM 存储器，并在此之后保持启动设置。可以根据在应用构建期间标识的存储器占用空间要求做出 FlexRAM 配置决策。

3.1. iMX RT1050 器件的 FlexRAM 配置演示

此处所述的 FlexRAM 配置的应用用例表示的情形是：根据应用代码编译器/链接器输出更精确地标识存储器分区大小的定义。ITCM/DTCM/OCRAM 存储器的大小取决于应用所需代码/恒定数据/静态数据/堆栈存储器的大小。它类似于本文档开头介绍的静态配置。

3.1.1. 外部存储器与 FlexRAM 存储器访问注意事项

i.MX RT10xx 器件没有嵌入式闪存（RT1064 含有串行 SPI 闪存作为 SIP）。加载到外部存储器中的代码/数据大小可以不受限制。它仅受 Arm 定义的外部 RAM 区域(1 GB)限制(0x6000 0000 - 0x9FFF FFFF)。在 i.MX RT 器件中，外部存储器可通过 FlexSPI/SEMC 接口（及其他接口）访问。这些接口的运行速度不足以在最大频率下运行时执行代码/访问数据，而不会在等待状态下造成任何损失（考虑内核运行频率为 600MHz、FlexSPI 最高频率为 166MHz（DDR 模式）或 SEMC 工作频率为 166MHz、L1 I/D 缓存无法涵盖所有的代码/数据）。另一方面，从 TCM 中执行代码/访问数据可以视为单个周期（参见上述各节中的例外情况）。

在配置 FlexRAM 之前，考虑将哪些代码/数据放入外部存储器也很重要。需要尽可能快地执行的最关键代码（没有等待状态，可充分利用 Cortex-M7 的超大规模管道性质）必须放在 ITCM 中（在此考量中，应优先考虑哈佛总线架构性质）。不太重要的数据（偶尔访问）应放在外部存储器中。只有内核才能访问的数据（堆栈、静态数据等）必须放在 DTCM 中。Cortex-M7 内核还支持通过 AHBS 接口直接访问 TCM。DMA 主机仍然可以通过 AHBS 访问 TCM。但是，该访问速度不如内核访问速度快。它应当在内核休眠/掉电时使用。

由多个总线主机（例如：内核和 DMA）访问的数据应当放在 OCRAM 中，尤其是在低功耗模式下由 DMA 访问时。

在配置 FlexRAM 之前，要考虑应用存储器占用空间中的上述所有特性。

3.1.2. FlexRAM 配置

FlexRAM 配置必须考虑链接器定义的存储器分区大小。分区大小的定义可以在应用开发期间根据应用需求加以调整。FlexRAM 配置还可以反映链接器分区的定义。

以下是一些应用示例：

- 在 5-6-5 RGB 模式下，以 320x240 的分辨率从摄像头检测图像（使用 CSI 模块）。
- 使用 DMA 模块将两个原始数据图像存储在数据缓冲区中，并使用 LCD 模块上的 DMA 进行动态显示。
- 使用存储在 30-KB 数据缓冲区中的图像结果处理图像（仅基于软件）。
- 图像处理必须尽快完成。
- 存储最后四个图像的处理数据结果（4 x 30-KB 数据缓冲区）。
- 从停止模式唤醒后，最后处理的图像数据结果必须在存储器中可用。
- 另外三个数据缓冲区采用相同的原始图像数据格式，它们只需偶尔显示（应用空闲状态）。

假设：在项目开发期间，应用软件要求：

表 9. 要求

代码	数据
120 KB	889 KB

3.1.2.1. 代码存储器占用空间

中断向量表和几个关键的中断服务例程必须放在 ITCM 中（64 位单周期访问存储器可以预取 64 位、4 x 16 位或 2 x 32 位指令）以加快其执行时间。中断向量和相应的中断服务例程占用 46 kB 的存储器。

3.1.2.2. 静态数据存储器占用空间

提供两个 150 KB 的数据缓冲区，其中包含来自图像传感器的原始数据（320x240、RGB 5-6-5 格式）。

备注

从处理角度考虑，这些缓冲区视为关键数据。这类数据通常存储在外部 SDRAM 存储器中。OCRAM 的访问时间明显快于 SDRAM，并从客户角度考虑了成本效益。建议充分利用片上存储器（如果可能）。

这些缓冲区由 DMA 通道填充，并由摄像头传感器接口模块触发（以乒乓方式）。它们也由内核进行读取并处理。这些数据缓冲区非常方便放入由 64 位系统 AXI 总线访问的 OCRAM 中。

备注

OCRAM 是一个可缓存的存储器。必须注意两个主机访问相同的可缓存存储器区域的情况。在这种情况下，建议禁用专用缓存区域。如果启用了缓存，则由软件负责保证两个主机的访问同步。

额外的四个数据缓冲区大小为 30 KB，其中包含由内核完成的图像处理结果数据。只有内核才会访问这些缓冲区，建议将这些缓冲区放入由内核数据请求直接访问的 DTCM 存储器中。余下的应用静态数据（初始化、未初始化或初始化为零）仅由内核进行处理，大小为 15 KB。此类数据的最佳位置是 DTCM。

根据应用中的待机模式调用，还偶尔会显示三个 150 KB 的恒定数据缓冲区（例如，静态 LCD 图像）。这些缓冲区可以存储在外部存储器类型中。

通过堆栈使用量分析（例如，www.iar.com/support/resources/articles/mastering-stack-and-heap-for-system-reliability），估计堆栈大小为 4 Kb（包含 10 %的附加用量）。建议将堆栈放入 DTCM 中，因为只有内核才能访问它。

3.1.2.3. 总存储器占用空间(i.MX RT1050)

使用 i.MX RT1050 器件时，若考虑上述假设，可总结如下：

表 10. 应用存储器分区要求

	代码		数据			
	RO	RW	RO	RW	RW	RW
			恒定	静态（仅 CPU）	堆栈	静态（所有总线主机）
存储器大小要求	*74*	*46*	*150*	*30*	*4*	*150*
			150	*30*		*150*
			150	*30*		
				30		
				15		
总计	**74**	**46**	**450**	**135**	**4**	**300**

RAM 存储器总量要求（在本示例中）为 483 KB。它适合具有 512 KB 存储器的 i.MX RT1050 器件。但是，当以 32 KB 存储库大小重新计算时，则为 544 kB，这意味着还需要增加一个存储库（参见表 11）。根据 Arm TCM 大小配置规范，TCM 的大小可以是 2 的幂数（32 KB、64 KB、128 KB、256 KB 或 512 KB）。

表 11. 基于上表的应用存储器分区分配

	外部存储器	ITCM	外部存储器	DTCM	OCRAM	存储器/存储库总容量
存储器大小要求	74	46	450	139	300	485
所需 FlexRAM 存储库的数量		2		5	10	17
存储器的总大小		64		160	320	544

在这种情况下，可以考虑将 15 KB 静态数据从 DTCM 移至 OCRAM（或 ITCM）。这取决于：

- OCRAM：通过系统 AXI 总线加载 OCRAM 控制器写入(CSI)/读取(LCD)的 DMA 通道。
- ITCM：在低功耗模式下无需访问静态数据。
- 对整体性能的影响。

如果将数据从 DTCM 移至 OCRAM 的方法对整体性能没有明显的影响，则可以这样做，并且能够适合 FlexRAM 存储器配置（参见表 5）。如果因性能下降而无法采用该方法，则可以采用将数据移至 ITCM 的解决方案。

表 12 显示了将应用静态数据从 DTCM 存储器移至 OCRAM 存储器的情况，前提是对性能没有明显的影响。

表 12. 将应用静态数据移至 OCRAM 时的应用存储器分区要求

	外部存储器	ITCM	外部存储器	DTCM	OCRAM	存储器/存储库总容量
存储器大小要求	74	46	450	139-15	300+15	485
所需 FlexRAM 存储库的数量		2		4	10	16
存储器的总大小		64		128	320	512

两种存储器重新配置（表 12 和表 13）都适合 i.MX RT1050 FlexRAM，并且所有 16 个存储库均被用到。

表 13. 将应用静态数据移至 ITCM 时的应用存储器分区要求

	外部存储器	ITCM	外部存储器	DTCM	OCRAM	存储器/存储库总容量
存储器大小要求	74	46+15	450	139-15	300	485
所需 FlexRAM 存储库的数量		2		4	10	16
存储器的总大小		64		128	320	512

每个存储器类型配置的存储库数量已知。但是，还不清楚各个存储库编号使用何种配置。从低功耗角度看，这取决于应用需求，因为有三个不同的电源域用于对应的存储库/存储库组。i.MX RT1050 FlexRAM 电源分配（电源子域）功能可以在应用的低功耗模式下使用。在本示例中，应用需要保留最后处理的成像数据缓冲区的数据。该缓冲区的大小(30 KB)适合一个存储库的大小(32 KB)。FlexRAM 存储库 0 在 PDRET 电源域中，并可保持数据内容直至挂起模式。FlexRAM 存储库 0 可用于存储最后的图像处理结果数据缓冲区内容。其余存储器仅在正常运行电源模式下使用。

表 14 显示了用于此示例的 FlexRAM 存储库的详细配置。图 2-4 还显示了各个存储器配置的起始地址和内容。

表 14. FlexRAM 存储库配置的应用示例

FlexRAM 存储库	配置	大小	电源域
Bank0	DTCM	128 KB	PDRET
Bank1	DTCM		PDRAM0
Bank2	DTCM		
Bank3	DTCM		
Bank4	ITCM	64 KB	
Bank5	ITCM		
Bank6	OCRAM	320 KB	
Bank7	OCRAM		
Bank8	OCRAM		PDRAM1
Bank9	OCRAM		
Bank10	OCRAM		
Bank11	OCRAM		
Bank12	OCRAM		
Bank13	OCRAM		
Bank14	OCRAM		
Bank15	OCRAM		

图 2. DTCM 存储器地址、配置、内容和相应的电源域

图 3. ITCM 存储器地址、配置、内容和相应的电源域

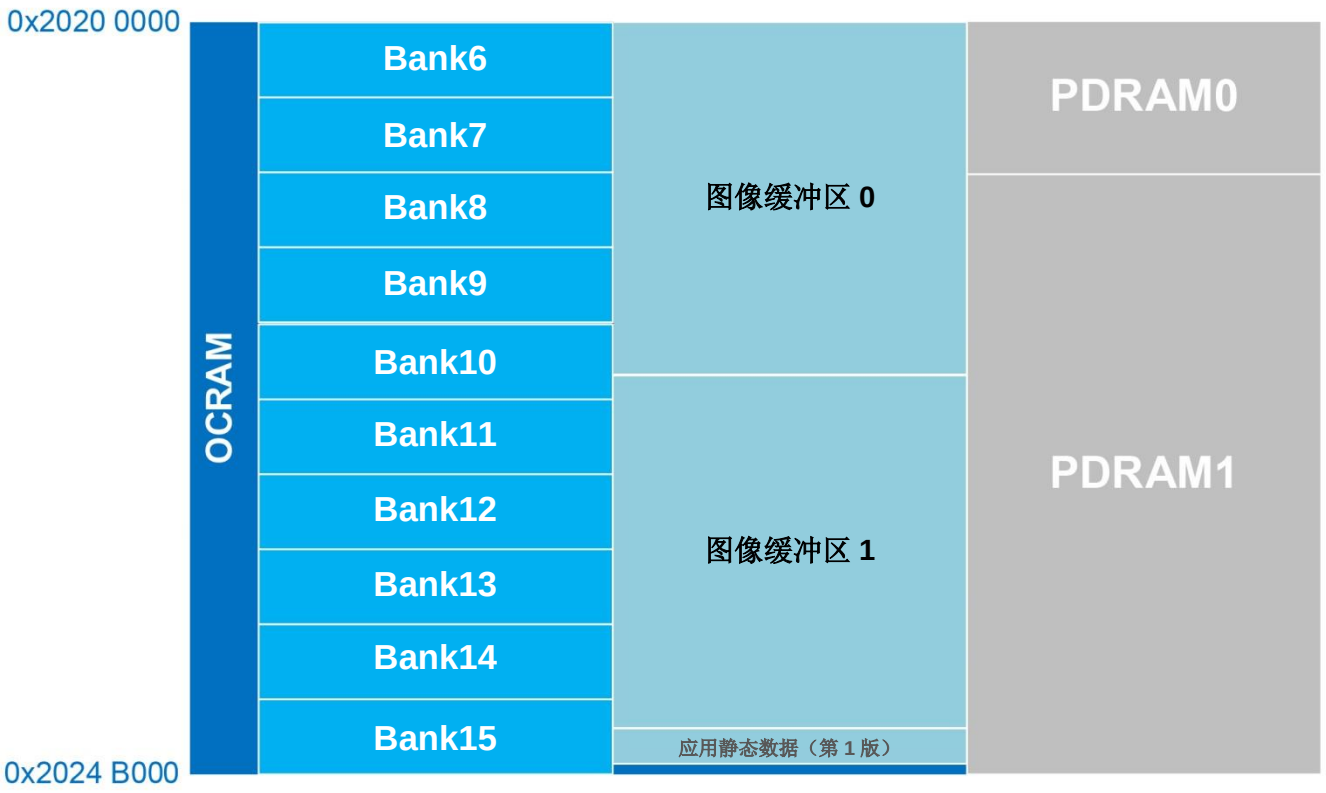

图 4. OCRAM 存储器地址、配置、内容和相应的电源域

如表 1 所示，在熔丝默认设置中没有与该用例所需的配置一致的有效 FlexRAM 配置。此时无法使用静态配置。但是，可以通过第 2.1 节“FlexRAM 配置”中定义的运行时配置方法来配置 FlexRAM。此配置必须在启动代码调用静态数据和读/写代码初始化之前完成。

FlexRAM 配置最终值将为 0x55555FAA：

IOMUXC_GPR_GPR17 (FLEXRAM_BANK_CFG)	存储库																OCRAM	D-TCM	I-TCM
	0	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1			
01010101010101010101111111010	D	D	D	D	I	I	O	O	O	O	O	O	O	O	O	O	320	128	64

备注

编译时请考虑应用项目的发布版本（无调试）目标配置。

3.1.3. 软件实现

在应用软件中 FlexRAM 配置的最佳位置是启动代码，其执行先于静态变量初始化、RW 代码初始化（如果考虑在 TCM/OCRAM 中）、向量表重定位、任何堆栈访问（推出/弹出）等。启动代码应从不同类型的存储器中执行；例如，外部串行闪存。在执行 FlexRAM 配置代码期间，不能访问 ITCM/DTCM/OCRAM 存储器，并且应禁用中断（以避免入栈/出栈）。

备注

可以在应用运行时处理 FlexRAM 重新配置；例如，位于外部串行闪存(FlexSPI)中的代码或包括外部 SDRAM(SEMC)中的堆栈/堆在内的所有数据。保证在 FlexRAM 释放后可避免数据重叠、访问不匹配等。不建议使用 DCD（器件配置数据）重新配置 FlexRAM 存储器，因为这可能与 ROM 代码存储器分配相冲突。

3.1.3.1. 考虑 Cortex-M7 TCM 大小的限制

根据 Arm 规范要求，TCM 的大小以 2 的幂数为单位（在 iMXRTxxxx FlexRAM 中，即为 0 k/32 k/64 k/128 k/256 k/512 k）。上面提到的应用用例考虑了这些限制。在此例中，上电复位事件后的应用软件将执行以下步骤：

1. 确保在重新配置代码执行期间未对 FlexRAM 阵列中的任何存储库进行访问：
   - 从 FlexRAM 阵列以外的存储器（未包含在 FlexRAM 中的 OCRAM、QSPI、SDRAM 等）中执行代码。
   - 禁用中断。
2. 根据应用需求配置 FlexRAM 存储库阵列：
   - 使用 IOMUXC_GPR_GPR17 寄存器中的 FLEXRAM_BANK_CFG 字段。
3. 从 eFuse 定义的 FlexRAM 配置切换到用户定义的 FlexRAM 配置：
   - 使用 IOMUXC_GPR_GPR16 寄存器中的 FLEXRAM_BANK_CFG_SEL 字段。
4. 如果需要任何 TCM 存储器为 0 kB，务必在将大小设置为 0 kB 之前禁用相应的 TCM。否则，请忽略此步骤：
   - 在将其配置为 0 kB 之前，务必使用 IOMUXC_GPR_GPR16 寄存器中的 INT_xTCM_EN 字段禁用相应的 xTCM 存储器。
5. 根据 Arm 规范，将专用 TCM 存储器大小按 2 的幂数配置为应用所需的大小。
   - 使用 IOMUXC_GPR_GPR14 寄存器中的 CM7_CFGxTCMSZ 字段，将其大小配置为 0 k/32 k/64 k/128 k/256 k/512 k。

   备注

   TCM 大小可以配置得较小(4 k/8 k/16 k)。FlexRAM 存储库则较大(32 k)。在这种情况下，FlexRAM 无法得到有效利用，因为内核无法访问可用存储器的一部分。访问（读/写）未分配的存储器空间时会产生总线故障。根据该设置更新 Cortex-M7 基址寄存器 CM7_xTCMCR。
6. 运行/跳转至完整的存储器定义应用代码。

```
__iomux_gpr14_adr     EQU   0x400AC038                                          1.
__iomux_gpr16_adr     EQU   0x400AC040
__iomux_gpr17_adr     EQU   0x400AC044
__flexram_bank_cfg    EQU   0x55555FAA
__flexram_itcm_size   EQU   0x7          ; 64kB
__flexram_dtcm_size   EQU   0x8          ; 128k

Reset_Handler
    CPSID  I                             ; Mask interrupts
#ifdef FLEXRAM_CFG_ENABLE
    LDR    R0,=__iomux_gpr17_adr         ; load IOMUXC_GPR17 register address to R0   2.
    MOV32  R1,__flexram_bank_cfg         ; move FlexRAM configuration value to R1
    STR    R1,[R0]
    LDR    R0,=__iomux_gpr16_adr         ; load IOMUXC_GPR16 register address to R0   3.
    LDR    R1,[R0]                       ; load IOMUXC_GPR16 register value to R0
    ORR    R1, R1, #4                    ; set corresponding FLEXRAM_BANK_CFG_SEL bit
    STR    R1,[R0]                       ; store the value to IOMUXC_GPR16 (user defined FlexRAM cfg enabled)
#ifdef FLEXRAM_ITCM_ZERO_SIZE
    LDR    R0,=__iomux_gpr16_adr         ; load IOMUXC_GPR16 register address to R0   4.
    LDR    R1,[R0]                       ; load IOMUXC_GPR16 register value to R0
    AND    R1, R1, #0xFFFFFFFE           ; clear corresponding INIT_ITCM_EN bit
    STR    R1,[R0]                       ; store the value to IOMUXC_GPR16 (disable ITCM)
#endif
#ifdef FLEXRAM_DTCM_ZERO_SIZE
    LDR    R0,=__iomux_gpr16_adr         ; load IOMUXC_GPR16 register address to R0
    LDR    R1,[R0]                       ; load IOMUXC_GPR16 register value to R0
    AND    R1, R1, #0xFFFFFFFD           ; clear corresponding INIT_DTCM_EN bit
    STR    R1,[R0]                       ; store the value to IOMUXC_GPR16 (disable DTCM)
#endif
    LDR    R0,=__iomux_gpr14_adr         ; load IOMUXC_GPR16 register address to R0   5.
    LDR    R1,[R0]                       ; load IOMUXC_GPR16 register value to R0
    MOVT   R1, #0x0000                   ; clear upper halfword of IOMUXC_GPR16 register
    MOV    R2, #__flexram_itcm_size
    MOV    R3, #__flexram_dtcm_size
    LSL    R2, R2, #16
    LSL    R3, R3, #20
    ORR    R1, R2, R3
    STR    R1,[R0]                       ; store the value to IOMUXC_GPR16 (disable DTCM)
#endif
    LDR    R0, =0xE000ED08                                                       6.
    LDR    R1, =__vector_table
    STR    R1, [R0]
    LDR    R2, [R1]
    MSR    MSP, R2
    LDR    R0, =SystemInit
    BLX    R0
    CPSIE  I                             ; Unmask interrupts
    LDR    R0, =__iar_program_start
    BX     R0
```

图 5. 遵循上述规则的汇编代码示例

3.1.3.2. 忽略 Cortex-M7 TCM 大小限制

这种方法认为应用软件清楚未分配的地址空间，并会避免对该空间的访问。该方法的优点是，不需要按 2 的幂数来考虑 TCM 存储器大小。

考虑因素：

- 需要在 IOMUXC_GPR_GPR14 寄存器的 CM7_CFGxTCMSZ 字段中保留默认的 TCM 大小设置（它考虑 FlexRAM 总大小的值）。
- 软件必须避免对未分配地址空间的访问。

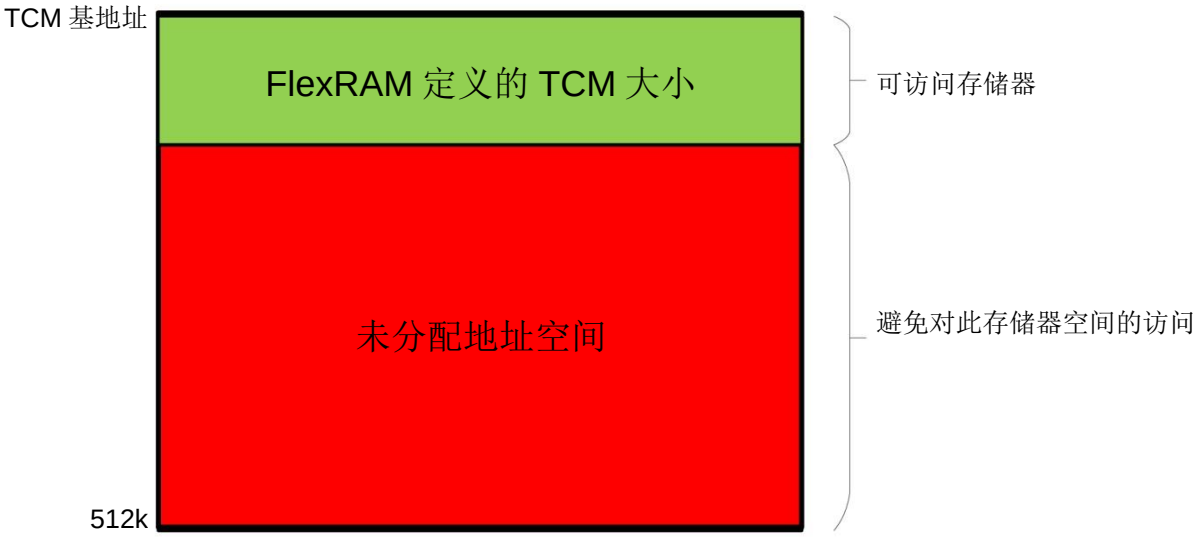

图 6. 可访问和不可访问地址空间的示例

除第 5 点和 2 的幂数考虑因素外，该软件的实现与之前的情况类似。

4. 修订记录

表中总结了本文档自初版以来所做的更改。

表 15. 修订记录

修订版本号	日期	重要变化
0	2017 年 10 月	初版
1	2018 年 8 月	新增了第 3.1.3 节“软件实现”。新增了对额外 RT10xx 器件的支持。
2	2019 年 9 月	新增了基于 RT1010 的特性。

如何联系我们：

主页：
www.nxp.com

网络支持：
www.nxp.com/support

文档编号：AN12077
第 2 版
2019 年 9 月