AN14377

MCXA15x上的连续SRAM地址的使用

第1.0版—2024年7月26日 应用笔记

文档信息

信息	内容
关键词	AN14377、MCXA、连续SRAM、重映射
摘要	本应用笔记介绍了如何配置和使用SRAM X0别名区来形成连续的SRAM地址。

1 介绍

可配置的连续SRAM地址是MCXA系列中的一项新的有益功能。连续SRAM地址具有以下优点：

- 利用连续SRAM地址映射，方便进行DMA操作。
- 适用于需要较大容量的连续SRAM区域的应用，如图形显示等。

本应用笔记以MCXA156为例，演示了如何配置和使用连续SRAM地址。

2 内存映射和架构

本节介绍了MCXA156的内存映射和内存架构。

2.1 内存映射

表1所示为MCXA156的内存映射，详情也可参见参考手册附件。SRAM X0别名区的地址紧跟SRAM B2之后，与SRAM A0、A1、A2、A3、B0、B1和B2形成了一个连续的地址空间。因此，共有128KB的连续地址SRAM。

表1. 内存映射

起始地址(十六进制)	结束地址(十六进制)	大小(KB)	说明
代码总线内存			
04000000	04001FFF	8	SRAM X0 (从端口0)
04002000	04002FFF	4	SRAM X1 (从端口0)
系统RAM			
20000000	20001FFF	8	SRAM A0 (从端口1)
20002000	20005FFF	16	SRAM A1 (从端口1)
20006000	20007FFF	8	SRAM A2 (从端口1)
20008000	2000FFFF	32	SRAM A3 (从端口1)
20010000	20017FFF	32	SRAM B0 (从端口3)
20018000	2001BFFF	16	SRAM B1 (从端口3)
2001C000	2001DFFF	8	SRAM B2 (从端口3)
2001E000	**2001FFFF**	**8**	**SRAM X0 Alias**

注：*默认情况下SRAM X0别名区域是保留的。*

2.2 内存架构

图1所示为MCXA156的内存架构。要访问SRAM X0别名区域地址，需启用SYSCON模块中的AHB矩阵重映射控制（重映射）寄存器的相应CPU0_SBUS、DMA0和USB0位。因此，CM33系统总线、DMA和USB FS最多可以访问128KB的连续地址。在启用重映射位后，SRAM X0（0x04000000-0x04001FFF）和SRAM X0别名（0x2001E000-0x2001FFFF）区域就都可以访问。当访问SRAM X0别名区地址时，该地址会被转换为相应的SRAM X0地址。然后，用户就可以访问 SRAM X0了。

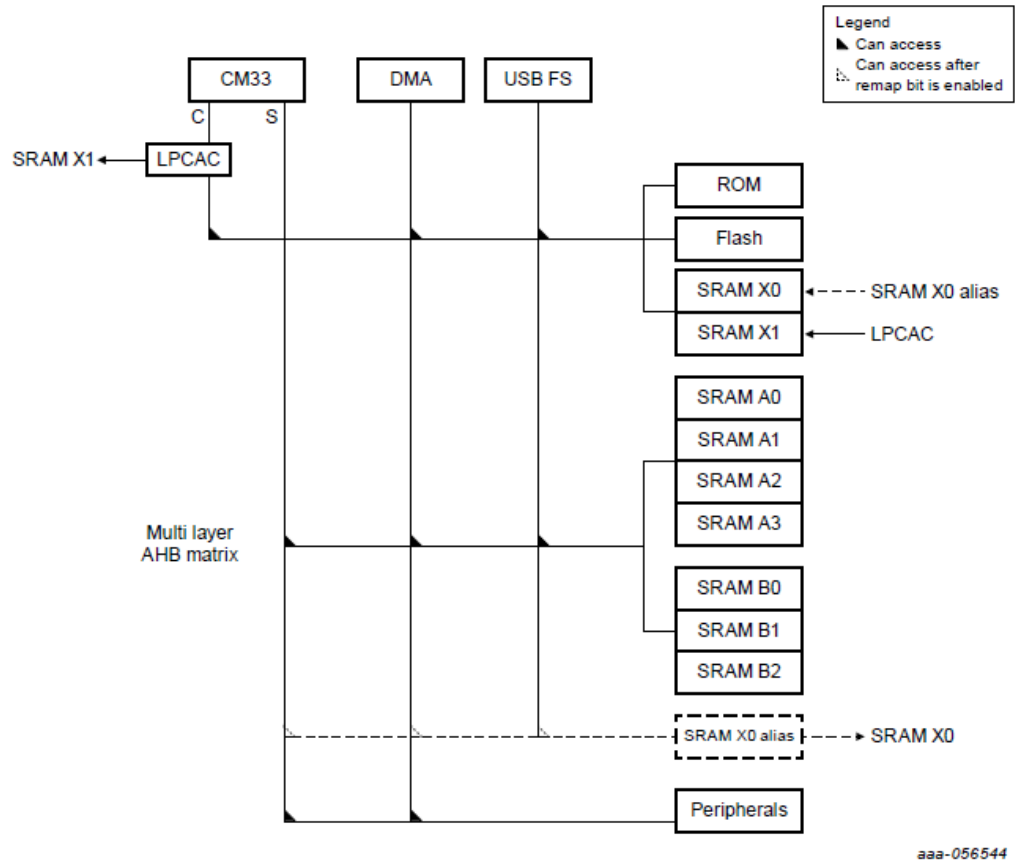

图1. 内存架构

注：*CM33通过系统总线访问SRAM X0别名区域，并通过总线代码访问SRAM X0区域。*

3 SRAM X0别名区的使用限制

本节列出了SRAM X0别名区的使用限制，如下所示：

- 在访问SRAM X0别名区前，启用相应的重映射位。
- 考虑堆栈指针（SP）的值。

用户通常希望在链接文件中定义一个连续的SRAM地址，并将堆栈区域分配在SRAM的末尾。堆栈指针是32位的，堆栈区域必须是32位对齐的，因此堆栈指针的初始地址通常是定义的SRAM末端地址加1。

如果用户要使用SRAM X0别名区域，初始堆栈指针必须为0x20020000。在跳转到扩展引导加载程序之前，ROM会检查堆栈指针。

对于MCXA156扩展引导加载程序，有效的堆栈指针区域为0x20000000至0x2001FFFF和0x04000000至0x04002FFF，因此0x20020000为无效地址。这是使用SRAM X0别名区时堆栈指针的限制。

4 解决方法

本节介绍了配置连续SRAM地址的两种解决方法。

4.1 分配堆栈空间以形成连续SRAM地址的解决方法

要在有效堆栈指针区域的末尾分配堆栈空间并启用重映射位，请执行以下步骤：

1. 将初始堆栈指针设置为0x2001FFFC，这是ROM验证的有效堆栈指针地址，并且它是32位对齐的。
2. 从0x2001FFFD到0x2001FFFF的SRAM空间可用于检查堆栈指针安全性的数据。
3. 然后，在使用堆栈前启用相应的重映射位。

4.2 骗过ROM来成功验证SP值，以形成一个连续SRAM地址的解决方法

要骗过ROM并形成一个连续的SRAM地址，请执行以下步骤：

1. 将向量表的第一个字设置为有效的堆栈指针值。
2. 然后启用相应的重映射位。
3. 在使用堆栈之前，将SP寄存器设置为所需的值。

注：*向量表中的SP值仅用于ROM检查，而非实际SP值。因此，在应用程序代码中使用向量表中的SP值时要注意。*

以下各章节将详细介绍在不同IDE中使用此解决方法的步骤。

4.2.1 MCUXpresso IDE

要修改工程代码以在MCUXpresso IDE中实现连续SRAM地址，请执行以下步骤：

1. 修改SRAM的大小，以使其包含SRAM X0别名区域。同时，修改SRAMX的位置和大小，保留SRAM X0区域，以确保数据安全，如图2所示。

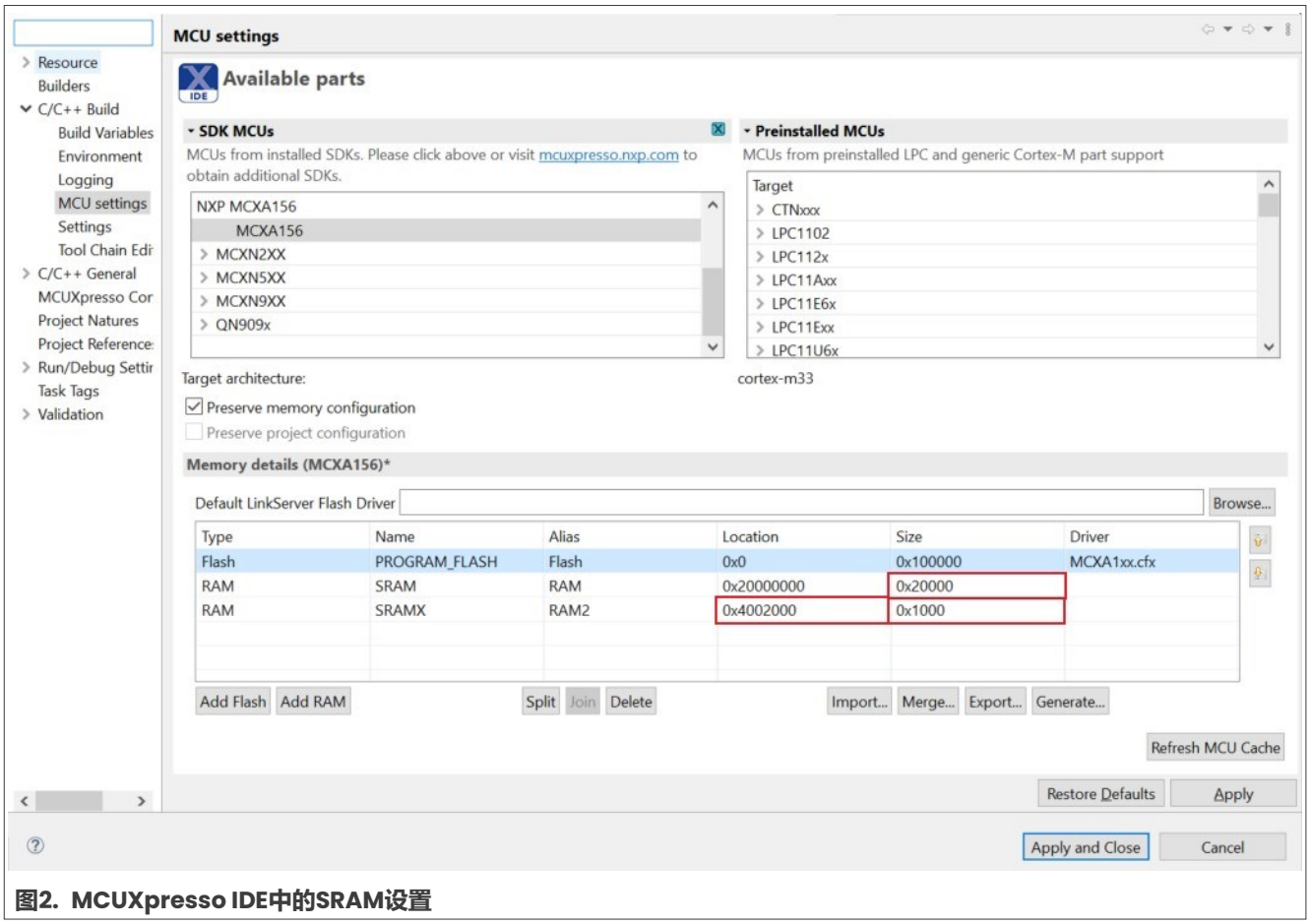

图2. MCUXpresso IDE中的SRAM设置

2. 将初始堆栈指针设置为ROM的有效值，如图3所示。

```
__attribute__ ((used, section(".isr_vector")))
void (* const g_pfnVectors[])(void) = {
    // Core Level - CM33
    &_vStackBase,                   // Set initial stack pointer to a valid value for ROM
    ResetISR,                       // The reset handler
    NMI_Handler,                    // NMI Handler
    HardFault_Handler,              // Hard Fault Handler
```

图3. 在startup_mcxa156.c中设置向量表的第一个字

3. 在使用堆栈之前，启用相应的重映射位并将SP寄存器设置为所需的值，如图4所示。

```
__attribute__ ((naked, section(".after_vectors.reset")))
void ResetISR(void) {
    // Disable interrupts
    __asm volatile ("cpsid i");
    // Config VTOR & MSPLIM register
    __asm volatile ("LDR  R0, =0xE000ED08  \n"
                    "STR  %0, [R0]         \n"
                    "LDR  R1, [%0]         \n"
                    "MOVS R0, #21          \n"// Enable Remap
                    "LDR  R1, =0x40091200 \n"
                    "STR  R0, [R1]         \n"
                    "MSR  MSP, %2          \n"//Configure SP register to actual value of stack region
                    "MSR  MSPLIM, %1       \n"
                    :
                    : "r"(g_pfnVectors), "r"(_vStackBase), "r"(_vStackTop)
                    : "r0", "r1", "r2");
```

图4. 在startup_mcxa156.c中启用重映射并配置SP寄存器

4.2.2 IAR IDE

要修改工程代码以在IAR IDE中实现连续SRAM地址，请执行以下步骤：

1. 修改m_data_end以包含SRAM X0别名区域。同时，修改DATA_region，保留SRAM X0区域，以确保数据安全，如图5所示。

```
define symbol m_interrupts_start              = 0x00000000;
define symbol m_interrupts_end                = 0x000001FF;

define symbol m_text_start                    = 0x00000200;
define symbol m_text_end                      = 0x000FFFFF;

define symbol m_data_start                    = 0x20000000;
define symbol m_data_end                      = 0x2001FFFF;

define symbol m_sramx0_start                  = 0x04000000;
define symbol m_sramx0_end                    = 0x04001FFF;


define memory mem with size = 4G;

define region TEXT_region            = mem:[from m_interrupts_start to m_interrupts_end]
                                     | mem:[from m_text_start to m_text_end];
define region DATA_region = mem:[from m_data_start to m_data_end-__size_cstack__];

define region CSTACK_region = mem:[from m_data_end-__size_cstack__+1 to m_data_end];
```

图5. 链接文件中的SRAM设置

2. 将初始堆栈指针设置为ROM的有效值，如图6所示。

```
__vector_table
        DCD     sfb(CSTACK)                     ;Set initial stack pointer to a valid value for ROM
        DCD     Reset_Handler

        DCD     NMI_Handler                     ;NMI Handler
        DCD     HardFault_Handler               ;Hard Fault Handler
        DCD     MemManage_Handler               ;MPU Fault Handler
        DCD     BusFault_Handler                ;Bus Fault Handler
        DCD     UsageFault_Handler              ;Usage Fault Handler
```

图6. 在startup_MCXA156.s中设置向量表的第一个字

3. 在使用堆栈之前，启用相应的重映射位并将SP寄存器设置为所需的值，如图7所示。

```
Reset_Handler
        CPSID   I               ; Mask interrupts
        LDR     R0, =0xE000ED08
        LDR     R1, =__vector_table
        STR     R1, [R0]
        MOVS    R0, #21          ;Enable Remap
        LDR     R1, =0x40091200
        STR     R0, [R1]
        LDR     R0, =sfe(CSTACK) ;Configure SP register to actual value of stack region
        MSR     MSP, R0
        LDR     R0, =sfb(CSTACK)
        MSR     MSPLIM, R0
        CPSIE   I               ; Unmask interrupts
        LDR     R0, =SystemInit
        BLX     R0
        LDR     R0, =__iar_program_start
        BX      R0
```

图7. 在startup_MCXA156.s中启用重映射并配置SP寄存器

4.2.3 Keil IDE

要修改工程代码以在Keil IDE中实现连续SRAM地址，请执行以下步骤：

1. 修改m_data_size以包含SRAM X0别名区域，如图8所示。

```
#define  m_interrupts_start          0x00000000
#define  m_interrupts_size           0x00000200

#define  m_text_start                0x00000200
#define  m_text_size                 0x000FFE00

#define  m_data_start                0x20000000
#define  m_data_size                 0x00020000
```

图8. 链接文件中的SRAM设置

2. 将初始堆栈指针设置为ROM的有效值，如图9所示。

```
__Vectors:
    .long   Image$$ARM_LIB_STACK$$ZI$$Base          /* Set initial stack pointer to a valid value for ROM */
    .long   Reset_Handler                           /* Reset Handler */
    .long   NMI_Handler                             /* NMI Handler*/
    .long   HardFault_Handler                       /* Hard Fault Handler*/
```

图9. 在startup_MCXA156.S中设置向量表的第一个字

3. 在使用堆栈之前，启用相应的重映射位并将SP寄存器设置为所需的值，如图10所示。

```
Reset_Handler:
    cpsid   i               /* Mask interrupts */
    .equ    VTOR, 0xE000ED08
    ldr     r0, =VTOR
    ldr     r1, =__Vectors
    str     r1, [r0]
    movs    r0, #21          /* Enable Remap */
    ldr     r1, =0x40091200
    str     r0, [r1]
    ldr     r2, =Image$$ARM_LIB_STACK$$ZI$$Limit /* Configure SP register to actual value of stack region */
    msr     msp, r2
    ldr     r0, =Image$$ARM_LIB_STACK$$ZI$$Base
    msr     msplim, r0
    ldr     r0,=SystemInit
    blx     r0
    cpsie   i               /* Unmask interrupts */
    ldr     r0,=__main
    bx      r0
```

图10. 在startup_MCXA156.S中启用重映射并配置SP寄存器

5 演示验证

本节提供了一个演示工程，用于验证SP寄存器、边界未对齐地址的读取和写入以及对连续SRAM地址的DMA访问。

5.1 硬件和软件要求

表2介绍了硬件和软件的要求。

表2. 硬件和软件的详情

类别	描述
硬件	• 开发板：FRDM-MCXA156 • 线缆：一根Type-C USB
软件	所支持的IDE： • MCUXpresso 11.9.0或更高版本 • IAR Embedded Workbench 9.50.1或更高版本 • Keil MDK 5.38或更高版本

5.2 设置

要设置此演示，请执行以下步骤：

1. 使用Type-C USB线连接FRDM-MCXA156开发板的J21和电脑的USB端口。
2. 在此处下载工程：https://github.com/nxp-appcodehub/an-continuous-sram-address-mcxa15x或参考本应用笔记附带的软件。
3. 构建工程并将工程下载到FRDM-MCXA156开发板。

5.3 SP寄存器验证

如图11所示，向量表中的第一个字，即初始堆栈指针，其值为0x2001f000，也是ROM的有效值。

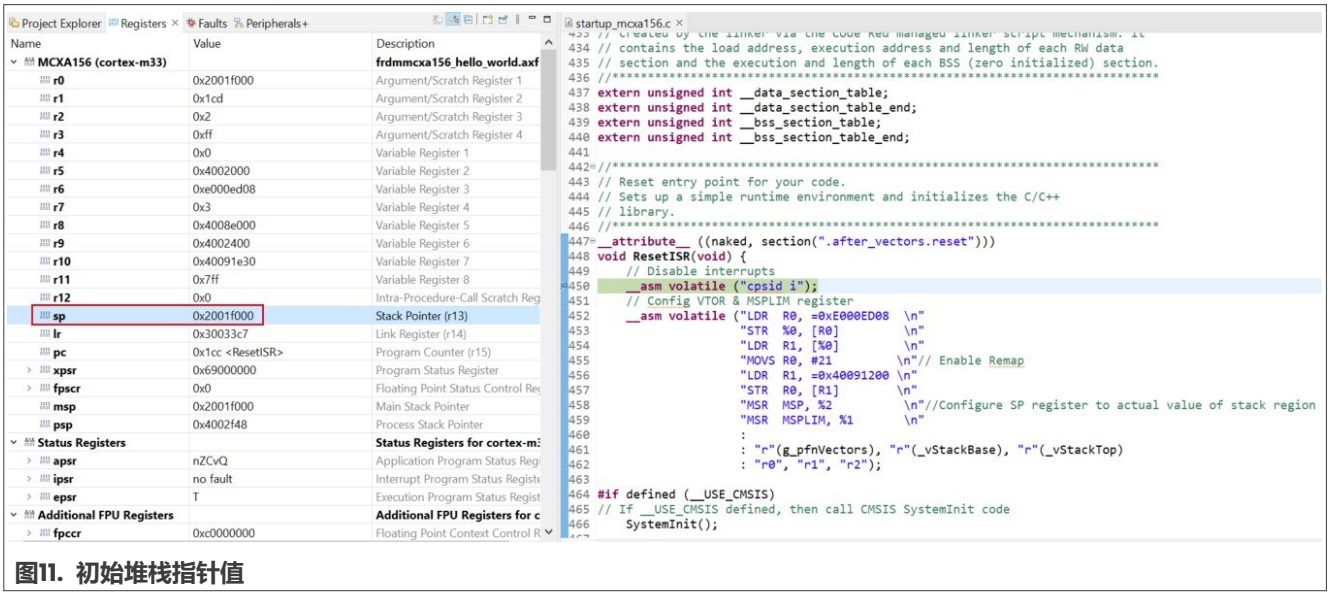

图11. 初始堆栈指针值

相应的重映射位已启用，SP寄存器的值为SRAM X0别名区的结束地址加一，如图12所示。完成上述操作后才能使用堆栈。

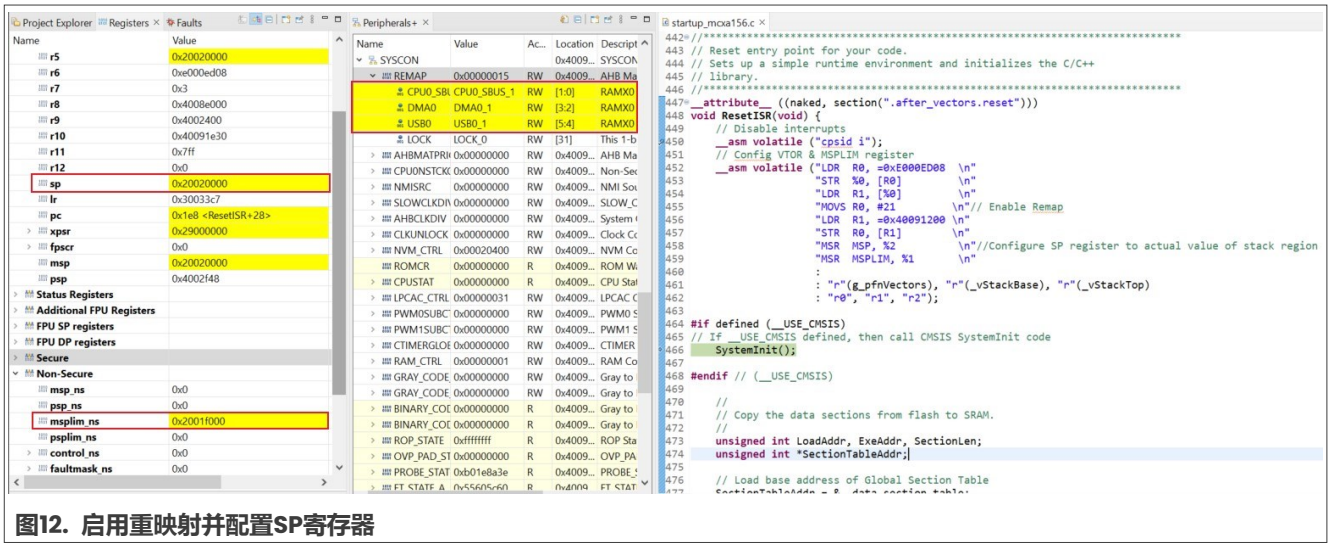

图12. 启用重映射并配置SP寄存器

5.4 边界未对齐地址的读写测试

图13所示为边界未对齐地址的读写测试，如下所示：

- 32位*p_test指针地址为0x2001DFFF（SRAM B2的最后一个字节），*p_test的初始值为0x0。
- 然后，将0xFFFFFFFF写入*p_test。
- 最后，*p_test的值被读取为0xffffffff。

32位*p_test地址不是4字节对齐的，且跨越SRAM B2和SRAM X0别名区，但读写操作正常。

```
********Read and write test for contiguous SRAM address boundary unaligned address********
32bit *p_test point address : 0x2001DFFF
Initial *p_test = 0x0
Write 0xFFFFFFFF to *p_test
Updated *p_test = 0xffffffff
```

图13. 对边界未对齐地址的访问

5.5 对连续SRAM地址的DMA访问测试

该演示工程还测试了在活动和低功耗模式下对该连续SRAM地址的DMA访问，且结果运行正常。

此目标缓冲区跨越RAM B2和RAM X0别名区，DMA可以正常传输，如图14中的日志信息所示。在掉电模式下，SYSCON寄存器处于保留状态，仍然可以使用DMA部分唤醒来传输连续SRAM地址上的数据。

```
**********************DMA access to contiguous SRAM address test************************

******************Active Mode******************

Source Buffer 1:
1       2       3       4
Source Buffer 2:
0       0       0       0
Source Buffer 3:
0       0       0       0
Destination Buffer:
0       0       0       0
Destination Buffer Address:
0x2001dff8      0x2001dffc      0x2001e000      0x2001e004

EDMA Source Buffer 1 to Destination Buffer.
Destination Buffer:
1       2       3       4

EDMA Destination Buffer to Source Buffer 2.
Source Buffer 2:
1       2       3       4

******************Power Down Mode******************
EDMA will transport Destination Buffer to Source Buffer 3.
Input any key to enter Power Down.
Entering Power Down mode...
Please Press SW2(WAKE UP) button to wake MCU and check Source Buffer 3.

MCU woken up
Source Buffer 3:
1       2       3       4
```

图14. 对连续SRAM地址的DMA访问

图15显示MCU处于掉电模式，并使用DMA部分唤醒来传输连续SRAM地址上的数据。

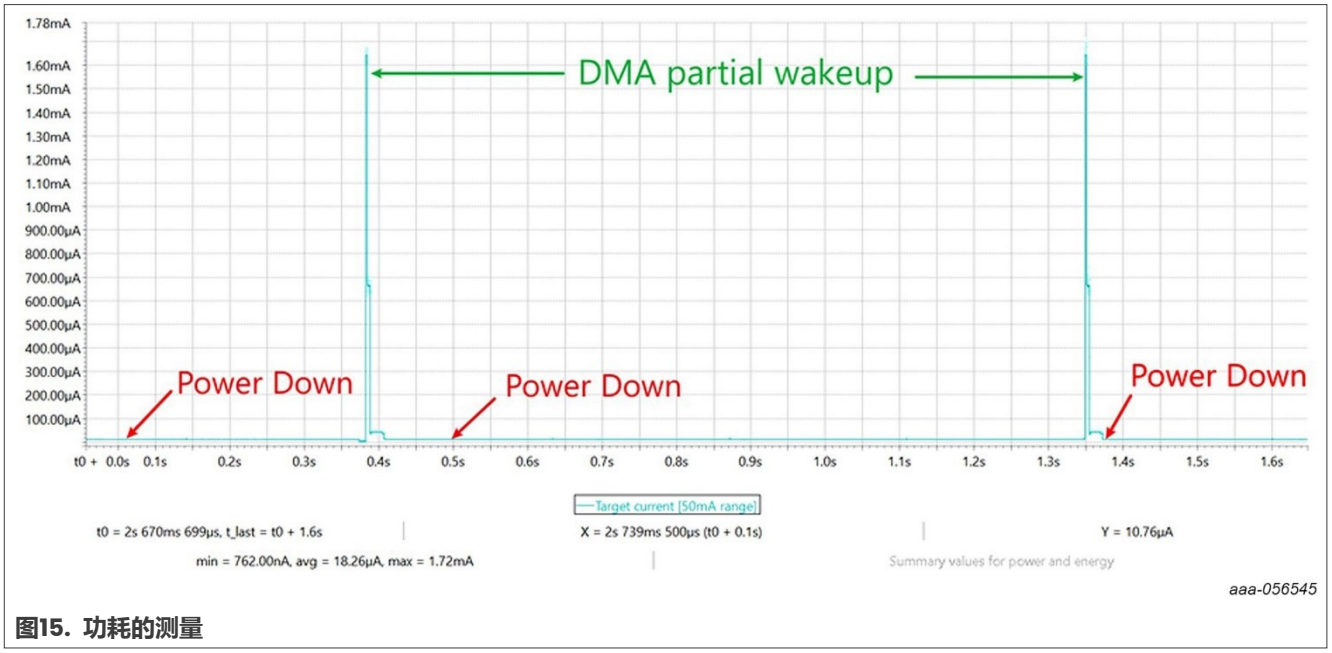

图15. 功耗的测量

6 总结

本应用笔记介绍了如何配置和使用SRAM X0别名区来形成连续的SRAM地址，并提供了一个演示工程，验证了连续SRAM地址的可行性。

7 关于本文中源代码的说明

本文中所示的示例代码具有以下版权和BSD-3-Clause许可：

2024年恩智浦版权所有；在满足以下条件的情况下，可以源代码和二进制文件的形式重新分发和使用本源代码（无论是否经过修改）：

1. 重新分发源代码必须保留上述版权声明、这些条件和以下免责声明。
2. 以二进制文件形式重新分发时，必须在文档和/或随分发提供的其他材料中复制上述版权声明、这些条件和以下免责声明。
3. 未经事先书面许可，不得使用版权所有者的姓名或参与者的姓名为本软件的衍生产品进行背书或推广。

本软件由版权所有者和参与者“按原样”提供，不承担任何明示或暗示的担保责任，包括但不限于对适销性和特定用途适用性的暗示保证。在任何情况下，无论因何种原因或根据何种法律条例，版权所有者或参与者均不对因使用本软件而导致的任何直接、间接、偶然、特殊、惩戒性或后果性损害（包括但不限于采购替代商品或服务；使用损失、数据损失或利润损失或业务中断）承担责任，无论是因合同、严格责任还是侵权行为（包括疏忽或其他原因）造成的，即使事先被告知有此类损害的可能性也不例外。

8 修订历史

表3汇总了本文的修订情况。

表3. 修订历史

文档ID	发布日期	说明
AN14377 v.1	2024年7月26日	首次公开发布

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.cn/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.

目录