Retain RAM data in reset procedure and low-power mode transition in XMC7000 MCU family

About this document

Scope and purpose

AN234800 explains the procedure to ensure RAM retention in XMC7000 MCU family when a software reset or low-power mode transition occurs.

Intended audience

This document is for anyone who uses the XMC7000 MCU family to ensure RAM retention when a software reset or low-power mode transition occurs.

Associated part family

XMC7000 family XMC7100/XMC7200 series

Table of contents

About this document **1**
Table of contents **1**
1 Introduction **2**
2 RAM retention overview **3**
2.1 Reset procedure 3
2.2 Low-power mode (DeepSleep mode) transition 4
2.3 Low-power mode (Hibernate mode) transition 5
3 RAM retention in reset **6**
3.1 Reset using LVD interrupt 6
3.2 Reset using external reset 8
4 RAM retention procedure in low-power mode **10**
4.1 Using DeepSleep mode 10
4.2 Using Hibernate mode 12
5 Glossary **15**
References **16**
Revision history **17**

1 Introduction

In Infineon XMC7000 family XMC7100/XMC7200 MCU series devices, a reset is performed asynchronously regardless of the RAM access status. Thus, if a reset occurs during operation, the RAM data might be lost. In addition, if the device power mode transitions from active to low power, you should follow the appropriate procedure for RAM retention.

Note that in Hibernate mode, the RAM data cannot be retained. Therefore, the RAM data should be transferred to the application flash once. After returning to active mode, return the RAM data from the application flash to the RAM. In this case, the transition data is defined as backup memory data.

See also the “SRAM interface” and “Work flash” chapters in the **architecture technical reference manual (TRM)**.

2 RAM retention overview

2.1 Reset procedure

Figure 1 shows the flow of RAM retention when a reset occurs. This example shows the case of retaining RAM0 data.

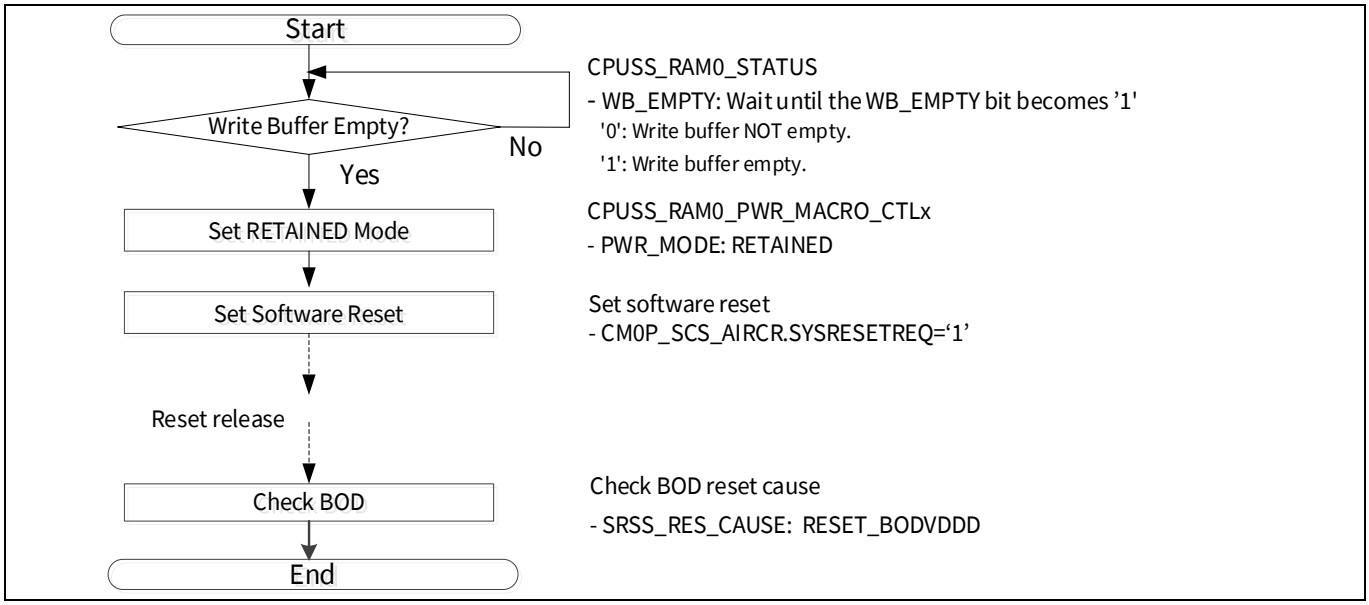

Figure 1 RAM0 retention procedure

1. Check the write buffer status of the WB_EMPTY bit of the CPUSS_RAM0_STATUS register. The WB_EMPTY bit indicates whether there is any data in the write buffer or it is empty.

   XMC7000 MCUs have the AXI bus interface. The ECC of the AXI bus interface is added to the 64-bit data. Therefore, when a partial writing (8-bit/16-bit/32-bit) occurs to the RAM, the missing data is read from the RAM. Then, the missing data and partial write data are merged to generate the 64-bit complete data. ECC is calculated over the 64-bit complete data. The write buffer is used in this operation; it may have data that is not yet written to the RAM. To prevent missing the unwritten data in the write buffer, the status of the write buffer must be checked. If there is valid data, wait until it is written to RAM0.

2. When there is no valid data in the write buffer, set the PWR_MODE bit of the CPUSS_RAM0_PWR_MACRO_CTLx register to RETAINED mode.
3. Generate a software reset. By performing such procedure, you can retain the RAM0 data and execute a reset. However, note that the RAM0 data cannot be retained if the voltage is lower than the brown-out detection (BOD: 2.7 V) level. Therefore, you should confirm that BOD has not occurred after returning from a reset.

Table 1 shows the RAM0 status register. Confirm that the WB_EMPTY bit is set to '1' before initiating a software reset.

Table 1 RAM0 status register

Register	Bit field	Bit value	Description
CPUSS_RAM0_STATUS	WB_EMPTY [0]	0	Write buffer not empty
		1	Write buffer empty

RAM retention overview

Table 2 shows the power control register, which controls system RAM0 power states with a single macro.

Table 2 Power control register

Register	Bit field	Bit value	Description
CPUSS_RAM0_PWR_MACRO_CTLx	*PWR_MODE [1:0]	0	OFF mode: Turn OFF the SRAM. This will turn OFF both the array and periphery power of the SRAM; SRAM memory contents are lost.
		1	Reserved
		2	RETAINED mode: Keep the SRAM in Retained mode. This will turn OFF the SRAM periphery power, but array power is ON to retain memory contents. SRAM contents will be retained in DeepSleep system power mode.
		3	ENABLE mode: Enable the SRAM for regular operation. SRAM contents will be retained in DeepSleep system power mode. (Default)

* To set the PWR_MODE bit field, use word access in the CPUSS_RAM0_PWR_MACRO_CTLx register. See the Registers TRM for details.

2.2 Low-power mode (DeepSleep mode) transition

Figure 2 shows the flow of RAM retention for low-power mode transition. In this method, when transitioning to a low-power mode, settings for RAM retention are performed. The procedure to check the status of the write buffer and the RETAINED mode setting of the RAM is the same. When the MCU enters a low-power mode, the main program execution stops. If a wakeup interrupt is generated, the MCU returns to Active mode.

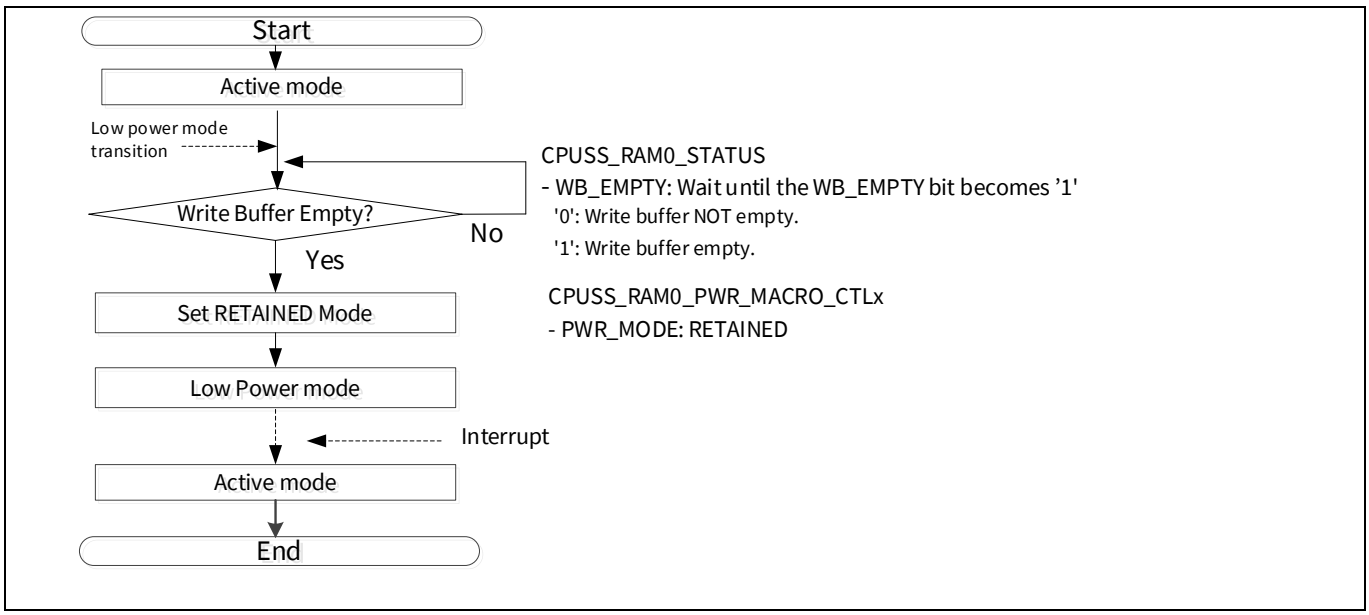

Figure 2 RAM0 retention for low-power mode

2.3 Low-power mode (Hibernate mode) transition

Figure 3 shows how the backup memory data is backed up for Hibernate mode transition. In this method, when transitioning to Hibernate mode, the backup memory data of the RAM is transferred to the application flash. When the MCU enters Hibernate mode, the main program execution stops. If a Hibernate wakeup reset is generated, the MCU returns to Active mode, and the backup memory data is transferred to the RAM from the application flash. The transfer of data from the RAM to application flash and back is handled by the user software.

To prevent unintentional overwriting of the backup data, you should consider not allowing other programs to access the backup data area of the application flash and RAM during data backup. In addition, you should ensure that the data transition time between the RAM and application flash meets your system requirements.

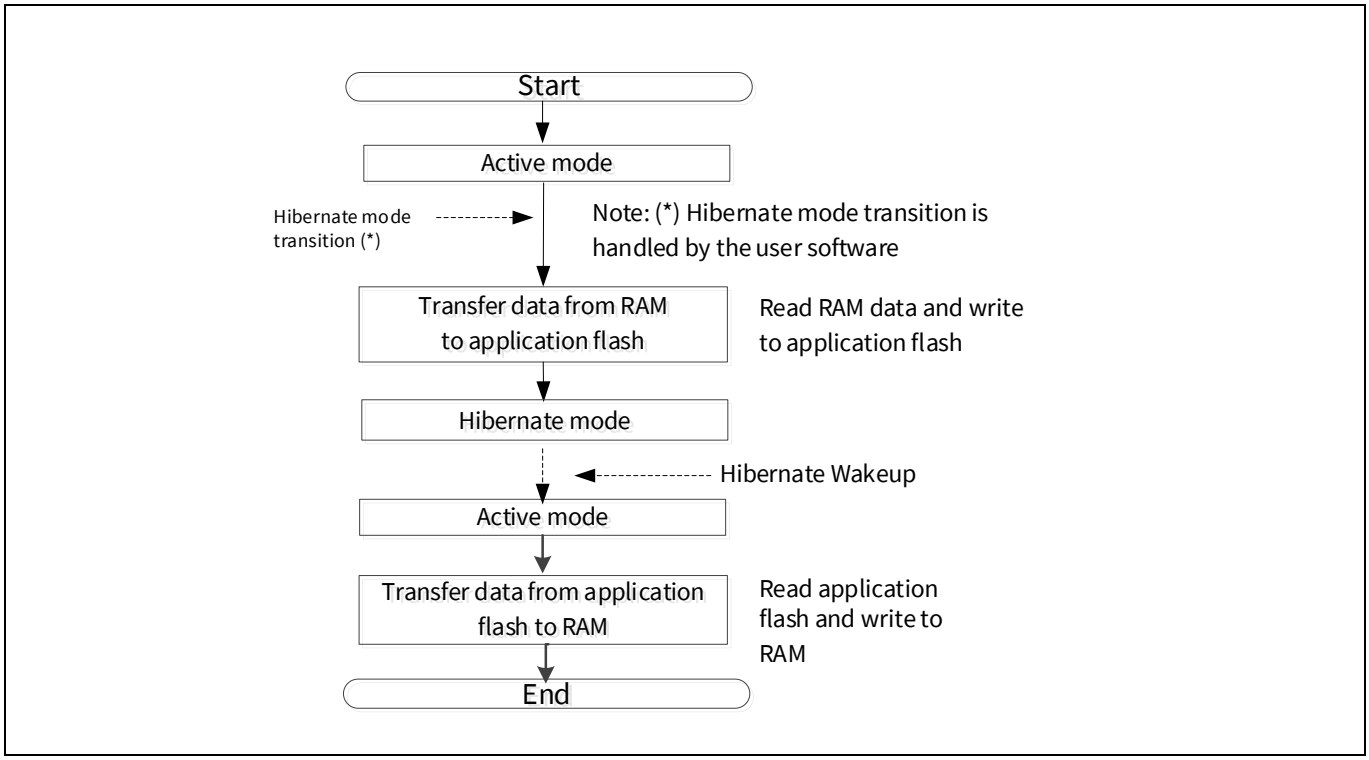

Figure 3 Backup for Hibernate mode

3 RAM retention in reset

This section shows two examples of reset procedures with a block diagram, timing chart, and flowchart. One method uses the low-voltage detection (LVD) interrupt. The other method uses the external reset input signal of the external LVD device.

3.1 Reset using LVD interrupt

In this case, LVD1 and LVD2 are used. LVD1 is a system low-level voltage detector that ensures that a reset occurs with guaranteed RAM0 retention. Also, LVD1 is used to detect a VDDD supply voltage drop. In addition, the user application starts by checking LVD2. LVD2 is used to check whether the VDDD supply voltage has recovered by setting the rising trip point.

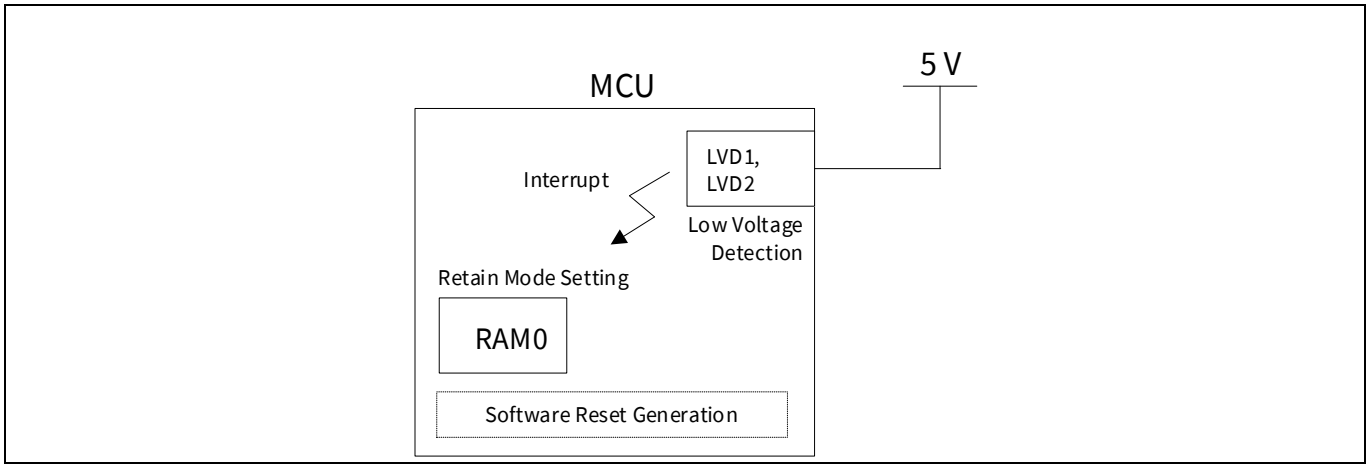

Figure 4 Reset for RAM0 retention with the LVD interrupt method

In **Figure 4**, first, an interrupt is generated when a falling edge is detected from LVD1. After the interrupt has occurred, check the RAM0 status and set it to RETAINED mode. After these steps, perform a software reset. Then, after the MCU starts up from reset, check for VDDD exceeding the LVD2 rising edge with the SRSS_INTR register. Finally, confirm that a BOD reset has not occurred. If a BOD reset occurred, RAM0 data retention is not guaranteed; in such cases, RAM0 data must be discarded.

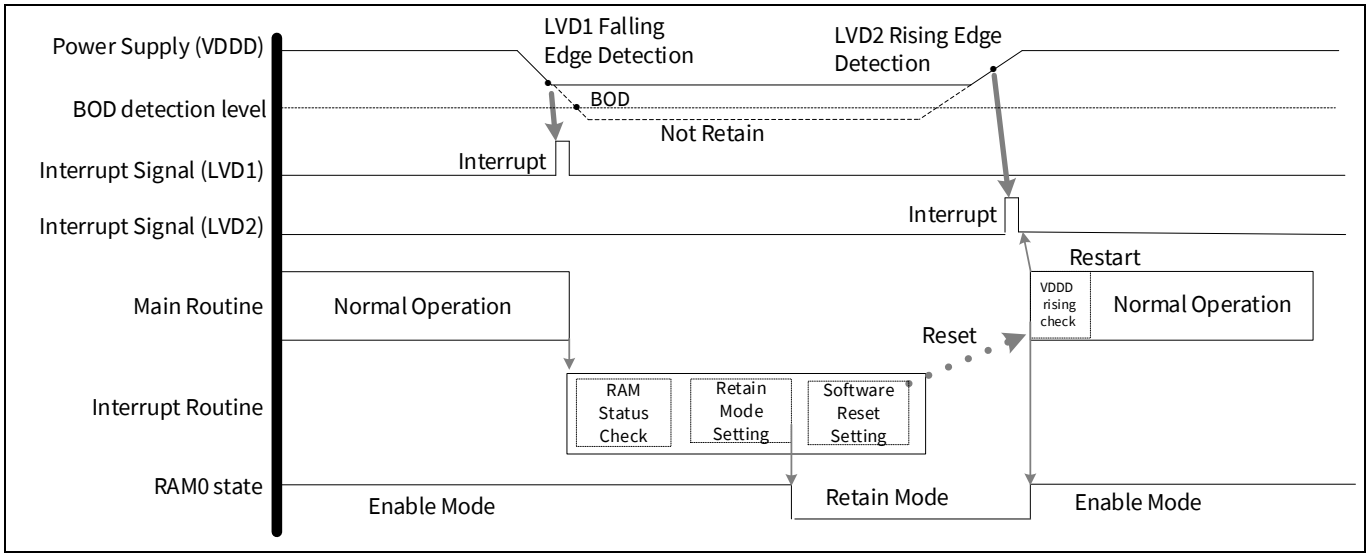

Figure 5 Reset timing chart in RAM0 by LVD interrupt

RAM retention in reset

In **Figure 5**, when VDDD drops, the interrupt routine is called. Then, the interrupt routine checks the RAM0 status, RETAINED mode setting, and software reset setting (reset generation). After the reset is completed, ensure that VDDD is rising with the LVD2 detection method, and resume normal operation.

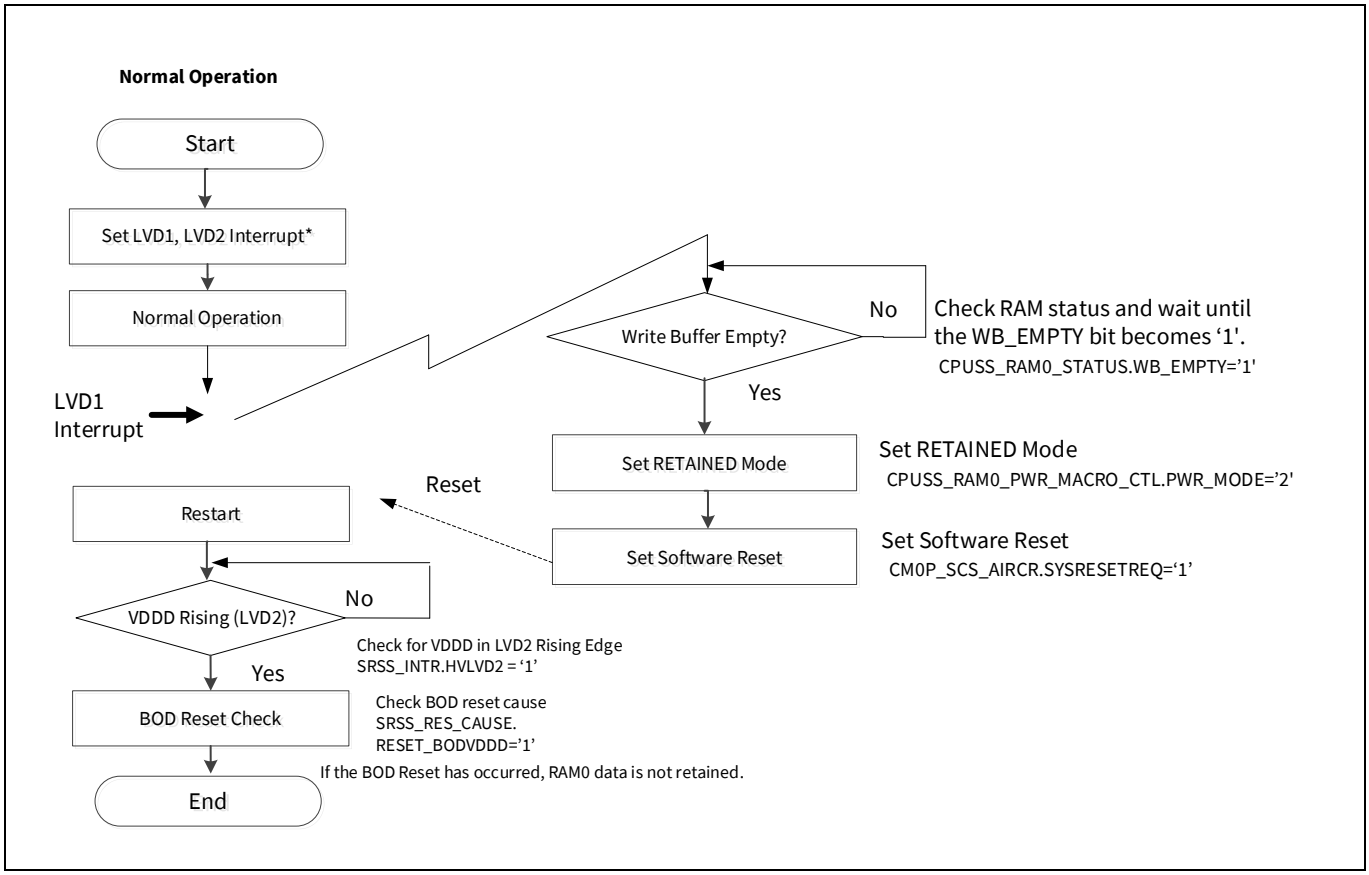

Figure 6 Reset in RAM0 with LVD interrupt

*For details of the interrupt setting, see the "Interrupts" chapter in the **Architecture TRM**.

1. Enable interrupt for LVD1 and LVD2.
2. If low voltage is detected with a voltage drop on VDDD, LVD1 generates an interrupt to the MCU.
3. When the MCU accepts the LVD interrupt, check the write buffer status.
4. If there is data in the write buffer (WB_EMPTY = '0'), wait until the write buffer is empty.
5. When there is no data in the write buffer (WB_EMPTY = '1'), the MCU sets the RETAINED mode and issues a software reset.
6. The MCU checks to see whether the VDDD supply voltage has recovered.
7. If VDDD has recovered, the MCU verifies that a BOD reset has not occurred.
   If a BOD reset has not occurred, RAM0 data is retained; if a BOD reset has occurred, RAM0 data must be discarded.

3.2 Reset using external reset

In this case, low-voltage detection is performed by an external LVD device. **Figure 7** shows the block diagram for the reset procedure for RAM0 retention by using an input signal from an external LVD to a GPIO pin.

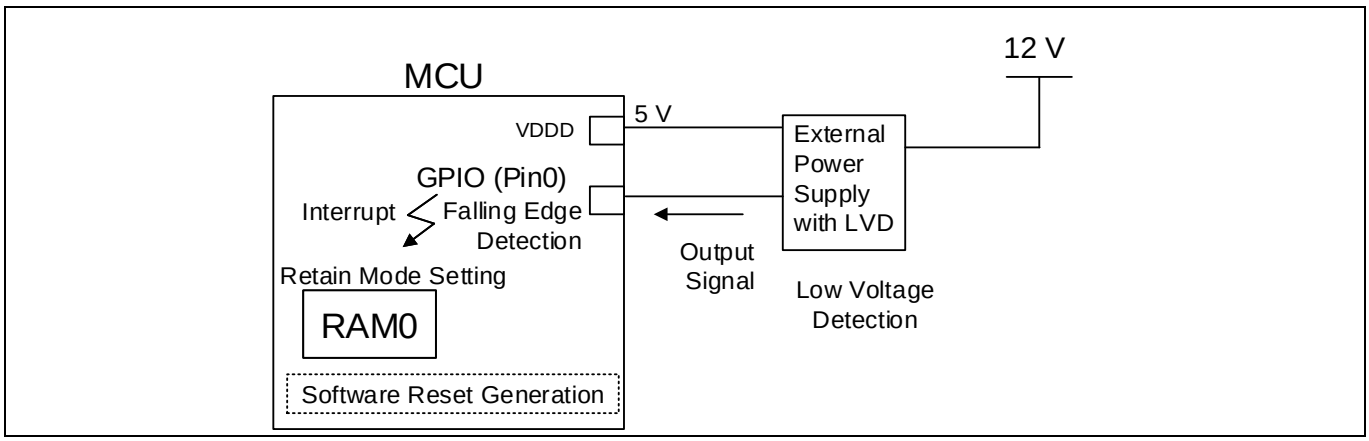

Figure 7 Reset in RAM0 retention by external LVD IC input signal to GPIO

In **Figure 7**, the output of the LVD device is connected to GPIO Pin 0. Pin0 is configured as the interrupt pin, which can generate an interrupt similar to the case of using the LVD. If the supply voltage is 5 V and 1.15 V, use an external LVD device to monitor both 5 V and 1.15 V.

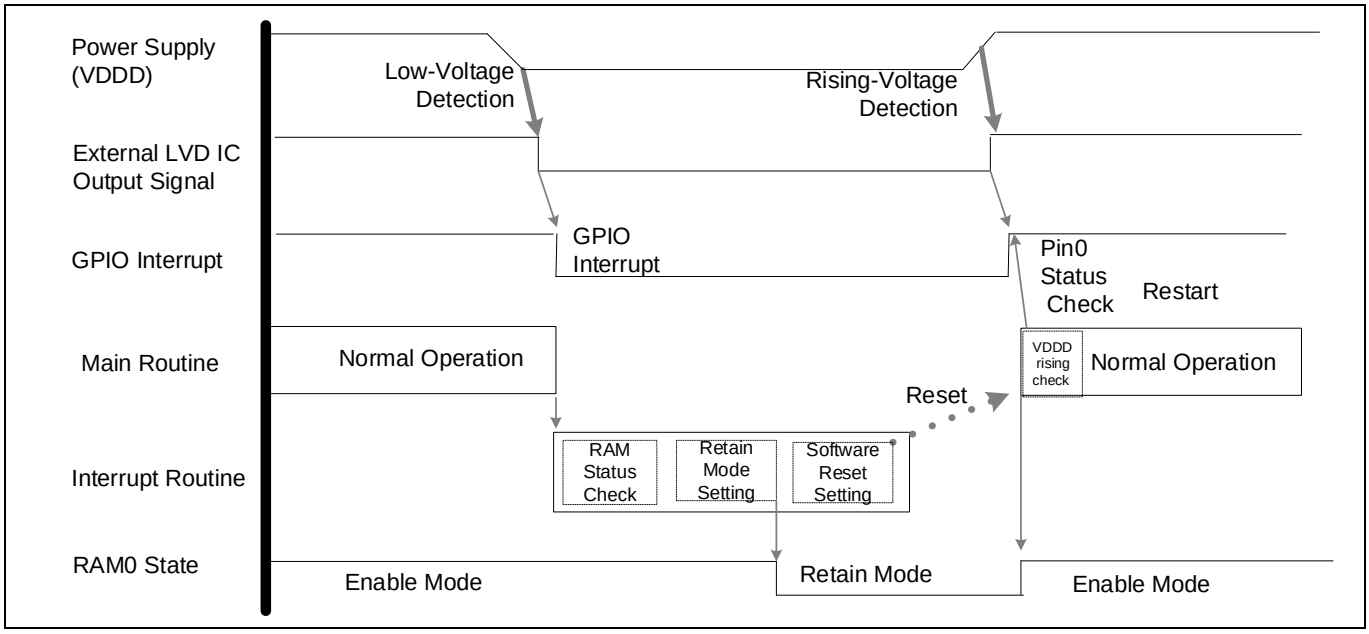

Figure 8 Reset timing chart in RAM0 with an external LVD IC

In **Figure 8**, after detecting low voltage with an external LVD device on VDDD, the external LVD device output is made LOW. GPIO Pin0 detects this falling edge and generates an interrupt. When the interrupt occurs, the MCU operation transitions from the main routine to the interrupt routine. Then, the interrupt routine checks the RAM0 status, RETAINED mode setting, and software reset setting (reset generation). After the MCU starts up from a reset, check for VDDD rising with the GPIO Pin0 status because GPIO Pin0 is connected to the LVD device output, and verify that a BOD reset has not occurred.

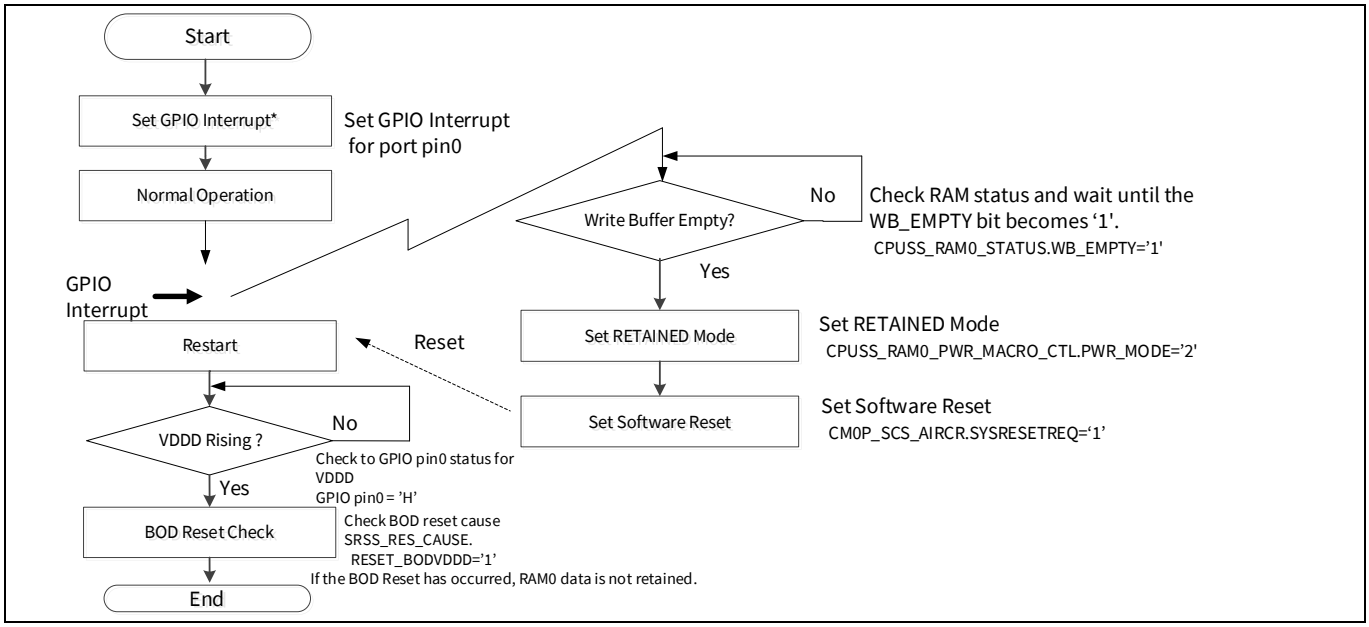

Figure 9 Reset in RAM0 retention with an external LVD IC input signal to GPIO

*For details of interrupt setting, see the "Interrupts" chapter in the **Architecture TRM**.

1. Enable interrupt for GPIO Pin0.
2. If GPIO Pin0 detects the input signal from the external LVD device, GPIO generates an interrupt to the MCU.
3. When the CPU accepts the GPIO interrupt, check the write buffer status.
4. If there is data in the write buffer (WB_EMPTY = '0'), wait until the write buffer is empty.
5. When there is no data in the write buffer (WB_EMPTY = '1'), the CPU sets the RETAINED mode and issues a software reset.
6. The MCU verifies whether the VDDD supply voltage has recovered.
7. When VDDD has recovered, the MCU verifies that a BOD reset has not occurred.
   If a BOD reset has not occurred, the RAM0 data is retained; if a BOD reset has occurred, the RAM0 data must be discarded.

4 RAM retention procedure in low-power mode

The XMC7000 MCU has the following device power modes: Active, Sleep, DeepSleep, and Hibernate; all except Active are low-power modes. This application note describes the DeepSleep and Hibernate mode case.

See the “Device power modes” chapter in the **Architecture TRM**.

4.1 Using DeepSleep mode

In this case, DeepSleep mode transition is used. The DeepSleep mode configuration is performed in Active mode in which the main program is running.

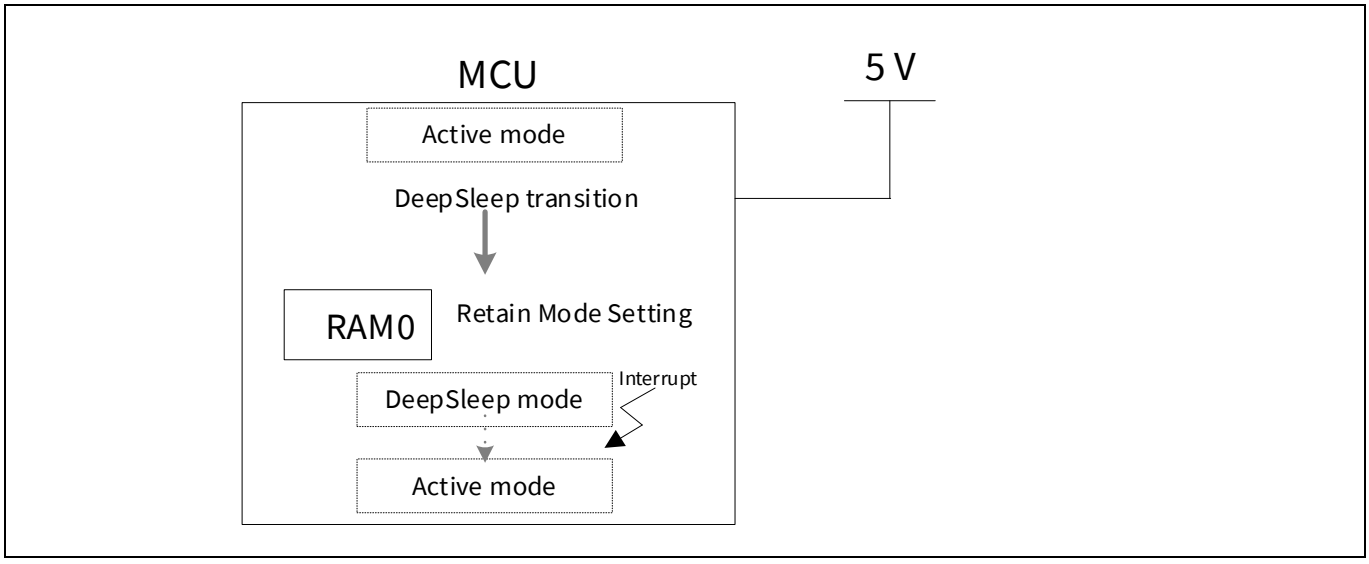

Figure 10 DeepSleep mode transition for RAM0 retention

In **Figure 10**, first, transition to DeepSleep mode from Active mode. During this transition, check the RAM0 status and set the RETAINED mode. Then, the MCU enters DeepSleep mode. If an interrupt occurs while in DeepSleep mode, the MCU returns to Active mode.

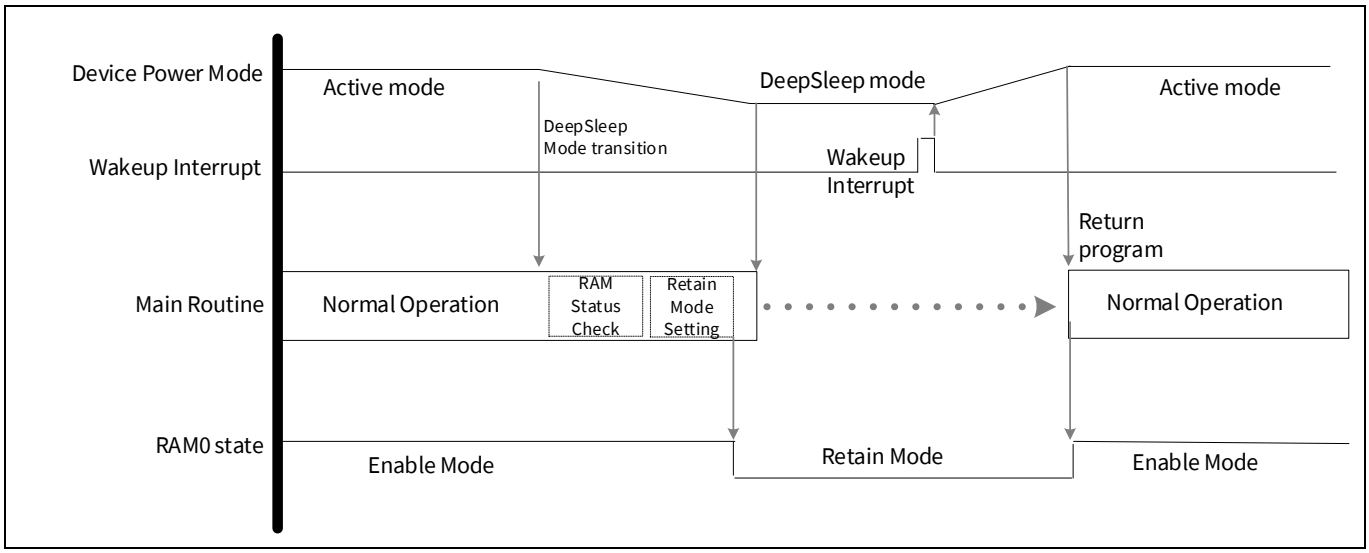

Figure 11 Timing chart of DeepSleep mode transition for RAM0 retention

RAM retention procedure in low-power mode

In **Figure 11**, when transitioning from Active mode to DeepSleep mode, the main routine checks the RAM0 status and sets the RETAINED mode. After that, the MCU goes into DeepSleep mode and the main routine stops. Then, in DeepSleep mode, when an interrupt occurs, it returns to Active mode. The main routine resumes program execution.

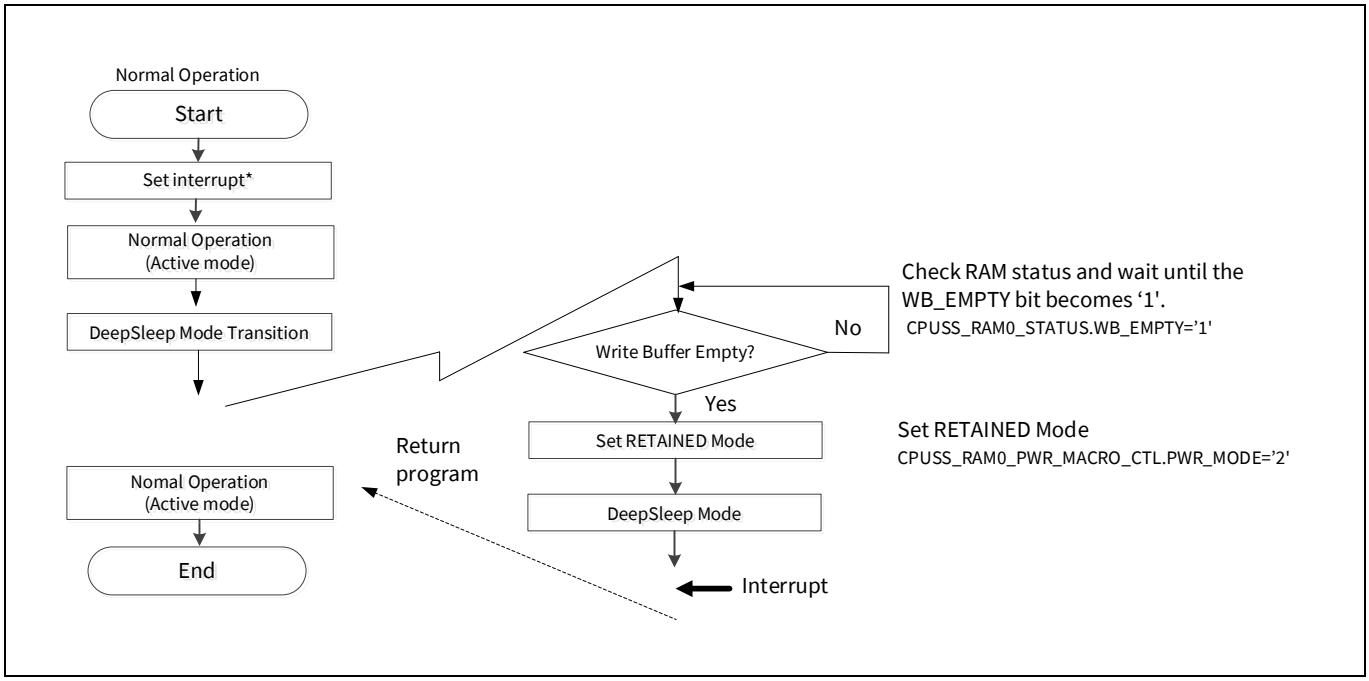

Figure 12 Flowchart of example DeepSleep mode transition procedure for RAM0 retention

*For details of interrupt setting, see the "Interrupts" chapter in the **Architecture TRM**.

1. When in Active mode of normal operation, set Sleep mode.
2. Check the write buffer status.
3. If there is data in the write buffer (WB_EMPTY = '0'), wait until the write buffer is empty.
4. When there is no data in the write buffer (WB_EMPTY = '1'), the MCU sets the RETAINED mode.
5. MCU enters DeepSleep mode.
6. When an interrupt occurs, the MCU changes from DeepSleep mode to Active mode.
7. Program execution resumes.

4.2 Using Hibernate mode

In this case, Hibernate mode transition is used. The Hibernate mode configuration is performed in the Active mode in which the main program is running.

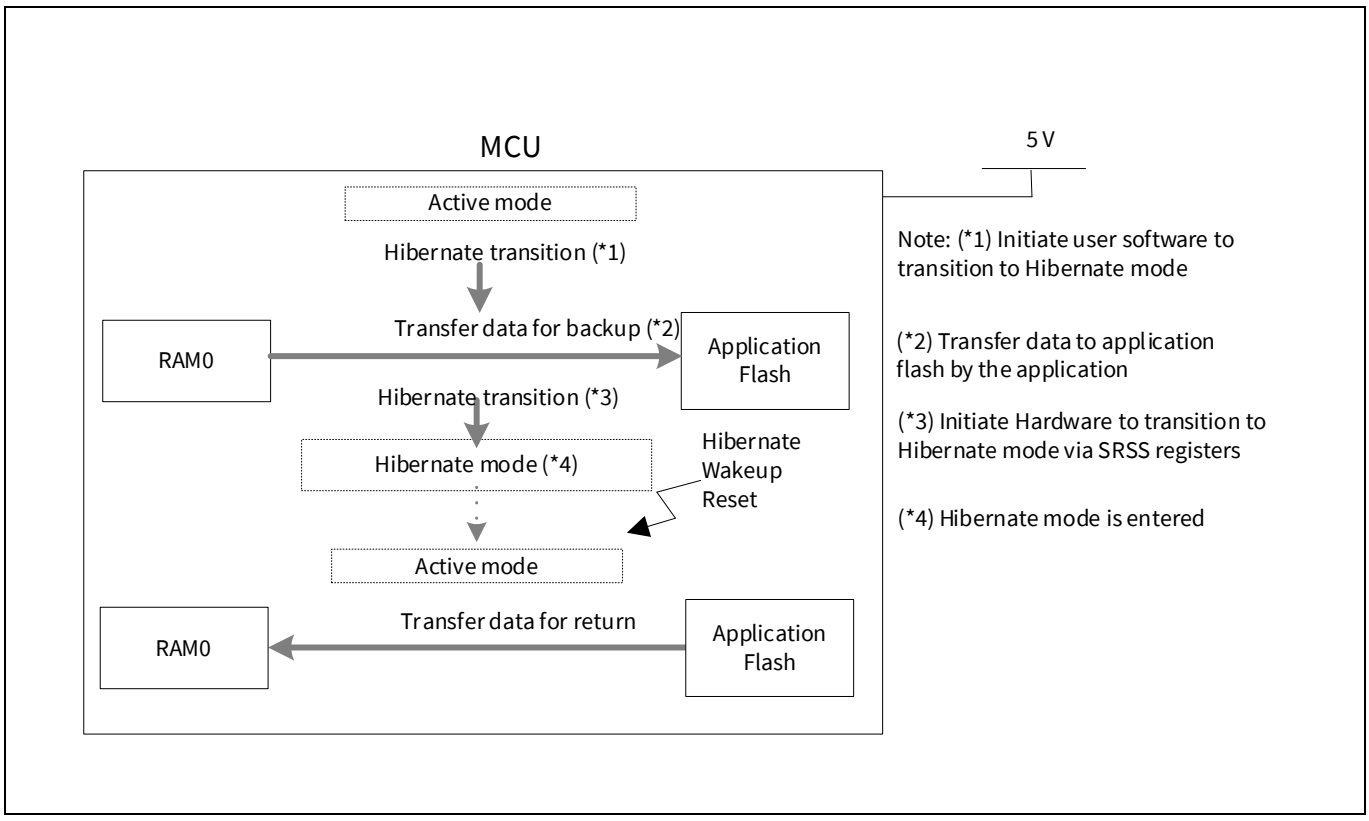

Figure 13 Backup in Hibernate mode transition for memory data

Figure 13 shows the transition to Hibernate mode from Active mode. During this transition, access to the application flash is checked. If the application flash can be accessed, the backup memory data from the RAM is transferred to the application flash. After the data transition is completed, the user software configures the SRSS register to make the MCU enter Hibernate mode. If a Hibernate wakeup reset occurs while in Hibernate mode, the MCU returns to active mode. After moving to active mode, the backup memory data is transferred from the application flash to RAM.

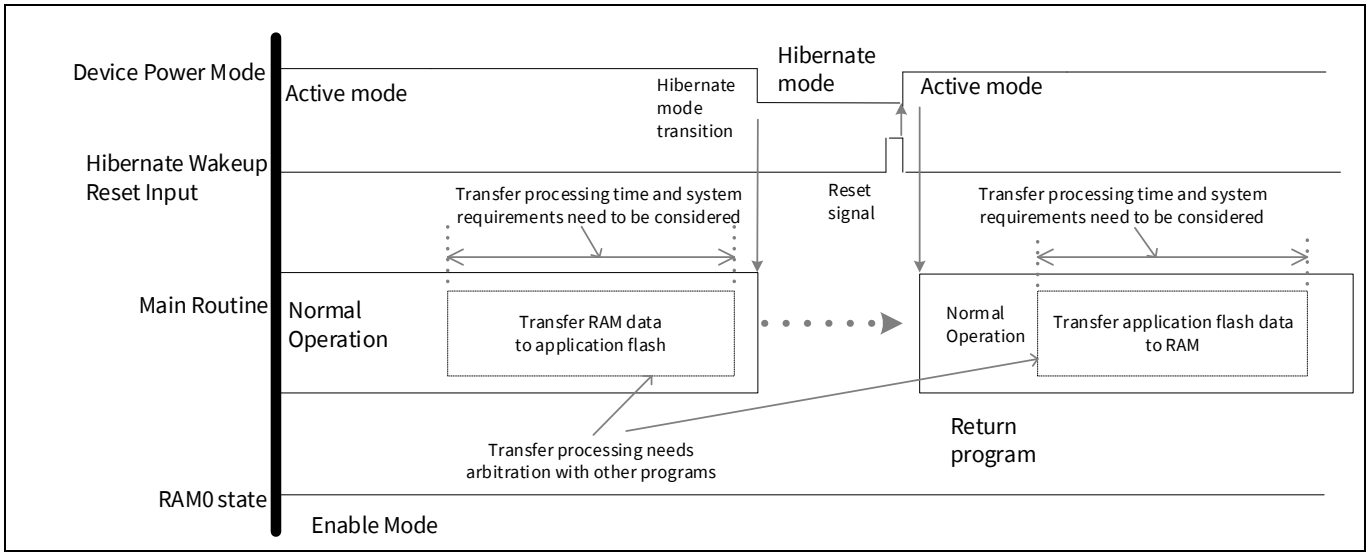

Figure 14 Timing chart of backup in Hibernate mode transition for memory data

In **Figure 14**, when transitioning from Active mode to Hibernate mode, the main program transfers the backup memory data from the RAM to application flash. While writing the application flash, the user application must arbitrate to access. This arbitration is between the RAM and application flash backup access and other application program access. This arbitration is to prevent access to other programs during backup access.

After the arbitration, the MCU goes into Hibernate mode, and the execution of the main routine stops. In Hibernate mode, when a Hibernate wakeup reset occurs, the MCU returns to Active mode. The main routine resumes program execution; the backup memory data is transferred from the application flash to RAM. In this transition, the arbitration with other application programs must be taken into account.

At the time of mode transition, the RAM and application flash transition time for data backup and return must meet the system requirements.

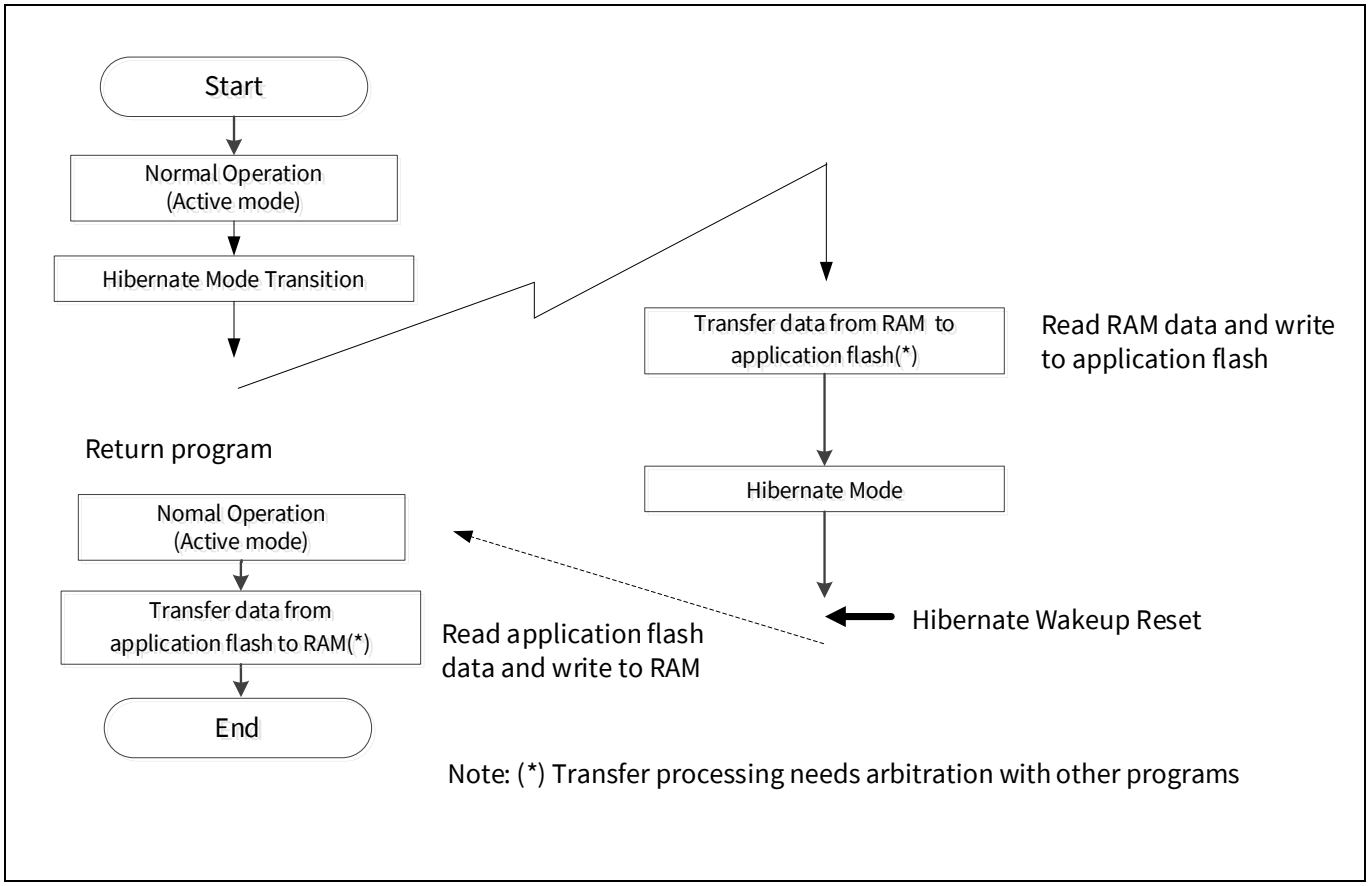

Figure 15 Backup procedure in Hibernate mode transition for memory data

1. When in Active mode of normal operation, set Hibernate mode.
2. Transfer the backup memory data from the RAM to application flash (Read from RAM and Write to the application flash)
3. The MCU enters Hibernate mode.
4. When a Hibernate wakeup reset occurs, the MCU changes from Hibernate mode to Active mode.
5. Program execution resumes. The backup memory data is transferred from the application flash to RAM. (Read to the application flash and Write to the RAM)

5 Glossary

Table 3 Glossary

Terms	Description
CPU	Central Processing Unit
BOD	Brown-Out Detection. See the “Power supply and monitoring” chapter of the **Architecture TRM** for details.
LVD	Low-Voltage Detection. See the “Power supply and monitoring” chapter in the **Architecture TRM** for details.
VDDD	Digital power supply. See the “Power supply and monitoring” chapter of the **Architecture TRM** for details.
GPIO	General purpose input/output. See the “IO system” chapter in the **Architecture TRM** for details.
ECC	Error Correcting Code
CPUSS	CPU subsystem
MCU	Microcontroller Unit

References

The following are the XMC7000 MCU family series datasheets and technical reference manuals. Contact **Technical Support** to obtain these documents.

[1] Device datasheet

- XMC7200 datasheet 32-bit Arm® Cortex®-M7 microcontroller XMC7000 family
- XMC7100 datasheet 32-bit Arm® Cortex®-M7 microcontroller XMC7000 family

[2] Technical reference manual

- XMC7000 MCU family architecture technical reference manual (TRM)
- XMC7200 registers technical reference manual (TRM)
- XMC7100 registers technical reference manual (TRM)

Revision history

Document version	Date of release	Description of changes
**	2022-02-22	Initial release

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2022-02-22

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2022 Infineon Technologies AG. All Rights Reserved.

Do you have a question about this document?

Go to www.infineon.com/support

Document reference

002-34800 Rev. **

IMPORTANT NOTICE

The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (**www.infineon.com**).

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.