Freescale Semiconductor
Application Note

Document Number: AN4531
Rev. 2, 12/2013

Achieving Persistent DRAM on PowerQUICC III and QorIQ Processors

This document describes the requirements and step-by-step procedures for forcing the DRAM into self-refresh mode.

Contents

Benefits of persistent DRAM

Freescale PowerQUICC III and QorIQ processors enable the user to force the DRAM into self-refresh mode, which allows the user to maintain data in DRAM while the processor is not available to refresh the DRAM array. This also eliminates the need for DRAM refreshes by the processor’s memory controller. As a result, the memory controller can be put to sleep.

Practical applications range from enabling persistent storage of trace and crash records in DRAM through a processor reset to persistence of program and data in DRAM through processor sleep and low-power modes.

1 DRAM requirements for self-refresh

DDR DRAM has a self-refresh mode that can be used to retain data in the DRAM without requiring a processor or memory controller to refresh the data. When in this mode, the DRAM does not require external clocking to retain the data.

Before issuing a SELF REFRESH command, the DRAM must be idle, with all banks in the precharge state. Then SELF REFRESH mode can be initiated to the DDR, in a way similar to a REFRESH command, with the exception that the CKE is held low. Afterwards, as long as CKE is maintained low and power supply inputs are stable, the DRAM refreshes itself. At this point, all inputs to the DRAM are "don't care" with the exception of RESET# and CKE.

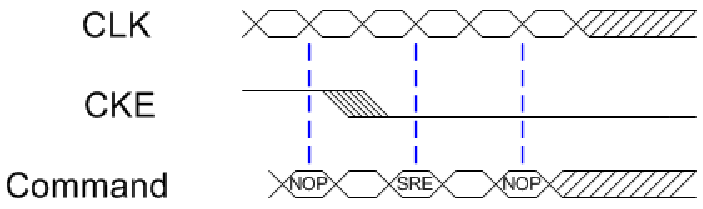

Figure 1. State of CKE and CLK during assertion of self-refresh command to DRAM

2 Initiating self-refresh

The PowerQUICC III and QorIQ family of processors enable DRAM self-refresh mode either through explicit software commands or through the assertion of a panic interrupt.

2.1 Using software to initiate self-refresh

Software can force the DRAM into a self-refresh state at any time. This is particularly convenient when the processor is going to be put into one of the many sleep states available. To enable self-refresh, the processor must set DDR_SDRAM_CFG_2[FRC_SR] = 1. When this bit is set, the DDR controller enters self-refresh mode. The DRAM controller then completes any queued up transactions and puts the DRAM into a self-refresh state. For more information on the DDR SDRAM Control Configuration register 2, see the DDR Memory Controller chapter in the applicable chip reference manual.

Note that, when the DRAM enters a self-refresh state, the DRAM itself is inaccessible. Therefore, if software is expecting to execute instructions after putting the DRAM into a self-refresh state, it must execute from somewhere other than DRAM. The following is an example of this software flow:

1. Copy a small routine to internal SRAM.
2. Jump to internal SRAM.
3. Force DRAM into self-refresh via DDR_SDRAM_CFG_2[FRC_SR].
4. Use the POWMGTCSR register to put the processor to sleep, or request an external reset event.

2.2 Using hardware to initiate self-refresh

Self-refresh can also be initiated through the assertion of a panic interrupt, which may be defined as an interrupt routed by the PIC to the DRAM controller explicitly for the purpose of enabling self-refresh.

On PowerQUICC and QorIQ devices, the panic interrupt is routed to the IRQ_OUT pin. To enable use of the panic interrupt on IRQ_OUT, follow these steps:

1. Route an internal interrupt, messaging interrupt or external interrupt to the IRQ_OUT pin.
   Note that this interrupt is not serviced internally (that is, no ISR is required), and IRQ_OUT does not necessarily need to connect to anything externally.
2. Set DDR_SDRAM_CFG_2[SR_IE] = 1 to enable the DRAM to enter self-refresh mode upon assertion of a panic interrupt. For information on the DDR_SDRAM_CFG_2 register, see the DDR Memory Controller chapter in the applicable chip reference manual.

On later QorIQ and QorIQ Qonverge devices, the panic interrupt is initiated by the internal *sie0* signal. To enable the panic interrupt via *sie0* assertion, follow these steps:

1. Route in internal interrupt, messaging interrupt or external interrupt to the internal *sie0* interrupt signal. This can be routed, for example, from an external pin to the *sie0* signal through the corresponding INT's xILRn register.
2. Set DDR_SDRAM_CFG_2[SR_IE]=1 to enable the DRAM to enter self-refresh mode upon assertion of the panic interrupt. For information on the DDD_SDRAM_CFG_2 register, see the DDR Memory Controller chapter in the applicable chip reference manual.

Please note that the PIC asserts *sie0* to all DDR controllers simultaneously.

3 Reset considerations

DDR3 requires a reset on power up. In order to use DDR self-refresh, hardware must be designed to allow independent control of the processor reset (HRESET or POR) and DDR reset, as DDR reset brings the RAM out of self-refresh state and invalidates the contents.

Software must enable two separate reset flows:

- one for the Power-On state where both the processor and DDR are reset and the DDR is initialized;
- and one where the processor may be reset, but the DDR is left in a self-refresh state, and the DDR itself is not re-initialized.

Additionally, in order to keep DRAM in a self-refresh mode, the CKE signal must be driven low through the entire reset sequence. However, the output CKE signal cannot be guaranteed to be driven low by the memory controller for the duration of HRESET/PORESET assertion. DRAM self-refresh is entered synchronously with clock, but exit from self-refresh is asynchronous to clock. Therefore, when HRESET/PORESET is asserted while DRAM is in self-refresh mode, the CKE signal must be driven low, external to the SoC by hardware on the board. CPLD or FPGA logic, an external FET-switch, or resistor divider on CKE are example hardware solutions that can be used to drive CKE low before and during the assertion of HRESET/PORESET.

3.1 Power-on-reset (POR) considerations

Upon POR, DDR is reset and initialized in accordance with the JEDEC specifications. When employing error checking and correction (ECC), use DDR_SDRAM_CFG_2[D_INIT] to initialize DRAM with a value. This sets the ECC syndrome bits to a known value for the entire DRAM array.

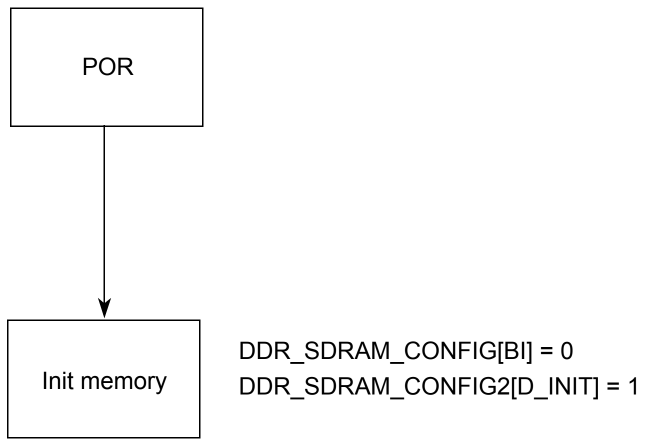

Figure 2. POR flow

3.2 HRESET considerations

For a warm reset with persistent memory, HRESET has been asserted to the processor and DDR reset remains untouched. To preserve the DRAM contents, upon exiting ~HRESET, the processor must not perform a full DDR initialization procedure.

DRAM must not be overwritten, and thus DDR_SDRAM_CFG2[D_INIT] must be kept cleared.

Full DRAM initialization must be bypassed, and so DDR_SDRAM_CFG[BI] must be set. Setting BI directs the memory controller not to reinitialize mode registers within the DRAM devices.

When the memory controller is first initialized, it undergoes various calibration transactions. These transactions read/write an area of memory, and this affects its contents. The location can be set through the DDR_INIT_ADDR and DDR_INIT_EXT_ADDR registers, and should be chosen so as not to corrupt valuable data or instructions that are meant to be preserved through the reset. Approximately 128 bytes are affected, starting at this address. When ECC is used in the design, the syndrome bits will not be in sync for the ~128 bytes starting at the calibration address, and should be reinitialized through a series of writes to that address range before ECC error reporting is turned on.

The two reset flows suggest that software must have a-priori knowledge of what type of reset took place. Unfortunately, this information cannot be determined from any on-chip sources; instead, it must be read from an external source such as a CPLD.

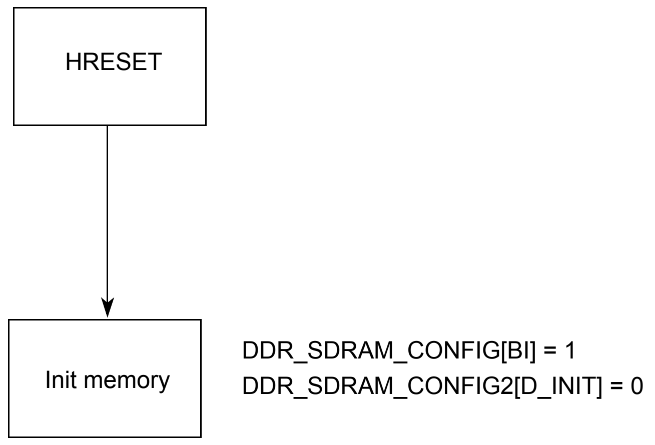

Figure 3. HRESET flow

4 Achieving persistent DRAM across processor sleep

The Freescale PowerQUICC III and QorIQ processors feature dynamic power management to minimize power consumption at several levels. For drastic power reduction, software can shut down clocks to various internal blocks and place the chips into a doze, nap or sleep power-down state. When the processor is placed into the sleep state, the DRAM can be configured to automatically enter a self-refresh mode, enabled by setting DDR_SDRAM_CFG[SREN] when the DDR controller is initially configured.

Upon entering sleep state, the DDR is automatically placed into a self-refresh state. Upon exiting sleep, CKE is reasserted, and the DRAM is taken out of self-refresh mode.

4.1 System design example

One example of usage of persistent memory is for trace data and crash records across watchdog initiated resets. In the case of a watchdog reset, the processor asserts ~HRESET_REQ to request a reset from an external source. Logic could be placed into a CPLD to initiate a panic interrupt, forcing a self refresh, prior to responding to the reset request.

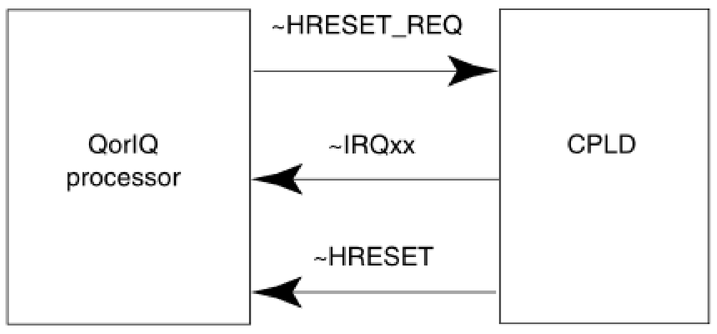

Figure 4. System design example

The processor must first enable the watchdog to assert the external ~HRESET_REQ. TCR[WRC] is defined more specifically for the implementation of the core in the integrated device. The watchdog timer reset control (WRC) field is cleared by the processor reset and set only by software. To prevent errant code from disabling the watchdog reset function, once software sets the WRC field, the field remains set until a reset occurs. For more information on core-level behavior for WRC, see the core reference manual. For SoC-specific information, see the Register Model chapter in the applicable chip reference manual.

The external IRQ must be enabled to route to ~IRQ_OUT. DDR_SDRAM_CFG_2[SR_IE] must be enabled as well.

The watchdog times out two times prior to initiating an ~HRESET_REQ. Upon the first expiration, the watchdog can be configured to trigger a critical interrupt, which may be used by the processor as an opportunity to write crash data into an area of DRAM. Upon the second expiration of the watchdog timer, the watchdog asserts ~HRESET_REQ. The external CPLD has logic inside that asserts the external IRQ, forcing the DRAM into a self-refresh state. Then the CPLD asserts ~HRESET or ~POR to the processor.

The system must be designed so that either HRESET is only used for a reset while memory is persistent or there is an external CPLD register available to determine the cause of reset. The diagram below illustrates the former design, where any HRESET assertion is assumed to leave DRAM in a SELF_REFRESH state.

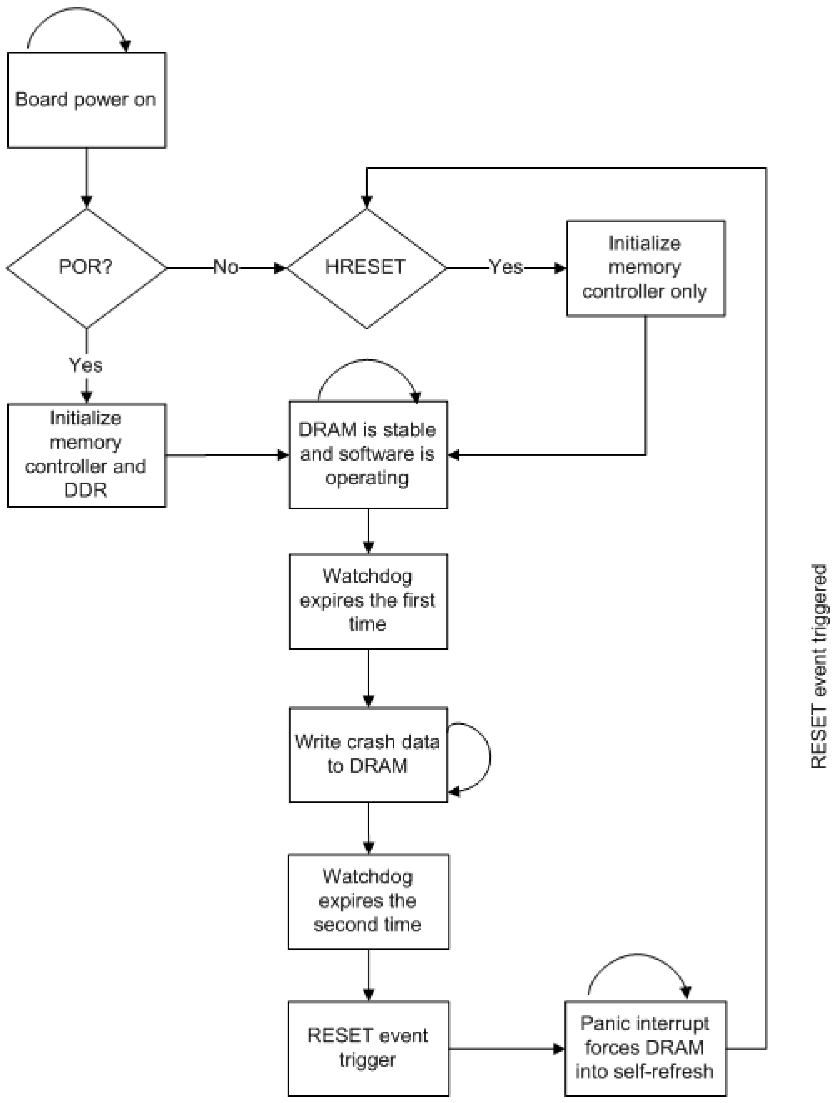
Board power on
POR?
No
HRESET
Yes
Initialize memory controller only
Yes
Initialize memory controller and DDR
DRAM is stable and software is operating
Watchdog expires the first time
Write crash data to DRAM
Watchdog expires the second time
RESET event trigger
Panic interrupt forces DRAM into self-refresh
RESET event triggered

Figure 5. Assertion of HRESET

5 Revision history

This table provides a revision history for this document.

Table 1. Document Revision History

Rev. Number	Date	Substantive Change(s)
2	12/2013	• In Section 2.2, "Using hardware to initiate self-refresh," updated first paragraph and added the second paragraph, the last two paragraphs, and the last two steps. • Added last paragraph in Section 3, "Reset considerations." • In Section 3.2, "HRESET considerations," updated "Up to 128 bytes are affected, starting at this address" to "Approximately 128 bytes are affected, starting at this address."
1	04/2013	• Updated Section 3, "Reset considerations." • In Section 3.1, "Power-on-reset (POR) considerations," updated the first sentence from "Upon POR, DDR is initialized in accordance with the JEDEC specifications" to "Upon POR, DDR is reset and initialized in accordance with the JEDEC specifications." • Rewrote Section 3.2, "HRESET considerations." • Removed Section 5.
0	03/2013	Initial public release

Document Number: AN4531
Rev. 2
12/2013