Scope

The Atmel® | SMART SAMA5D3 series is a high-performance, power-efficient embedded MPU based on the ARM® Cortex®-A5 processor.

The SAMA5D31, SAMA5D33, SAMA5D34, SAMA5D35 and SAMA5D36 eMPUs feature one multi-port DDR2 controller that supports 32-bit DDR2-SDRAM, 32-bit LPDDR1-SDRAM, and 32-bit LPDDR2-SDRAM memories. These memories are called SDRAM in this document.

SAMA5D3x embeds a pad calibration feature that performs bus impedance adaptation, improving signal integrity. This leads to a reduction of overshoots, of Electromagnetic Interference (EMI), of power consumption on I/Os and eliminates the need of serial resistor on data lines.

This application note is intended to help the developer in the design of a system using external memory.

Reference Documents

<table>
<thead>
<tr>
<th>Type</th>
<th>Title</th>
<th>Literature No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datasheet</td>
<td>SAMA5D3 Series</td>
<td>11121 (1)</td>
</tr>
</tbody>
</table>

1. SAMA5D3 Series datasheet available on www.atmel.com
Table of Contents

Scope .................................................................................................................. 1
Reference Documents ......................................................................................... 1
Table of Contents ............................................................................................... 2
1. Multi-port DDR Controller Overview ............................................................ 4
2. Multi-Port DDR Controller Signals Definition ............................................. 5
3. SDRAM Connection on SAMA5D3x ............................................................... 7
   3.1 32-bit Using 2x16-bit DDR2-SDRAM Implementation .......................... 7
       3.1.1 Hardware Configuration ............................................................... 7
       3.1.2 Software Configuration ............................................................... 7
   3.2 DDR2 VREF Signal Considerations ......................................................... 7
   3.3 32-bit Using 2x16-bit LPDDR2-SDRAM Implementation ...................... 8
       3.3.1 Hardware Configuration ............................................................... 8
       3.3.2 Software Configuration ............................................................... 9
   3.4 LPDDR2 VREF Signal Considerations .................................................. 9
4. Layout and Design Constraints ................................................................. 10
   4.1 General Considerations .......................................................................... 10
   4.2 DDR2/LPDDR Bus Interface Controller .............................................. 10
       4.2.1 Signals Group 1 ......................................................................... 10
       4.2.2 Signals Group 2 ......................................................................... 10
       4.2.3 Signals Group 3 ......................................................................... 11
   4.3 EBI Trace Routing Guidelines ............................................................... 11
       4.3.1 Topology About the EBI Bus ...................................................... 11
       4.3.2 Placement .................................................................................... 11
       4.3.3 Bypassing Capacitors ................................................................. 11
       4.3.4 Trace Length .............................................................................. 11
       4.3.5 Trace Spacing ............................................................................ 11
       4.3.6 Via .............................................................................................. 11
       4.3.7 Ground Plane(s) ....................................................................... 12
       4.3.8 Power Plane(s) .......................................................................... 12
       4.3.9 General Considerations for High-Speed Differential Interfaces ........ 12
5. LPDDR2-SDRAM Power-up and Power-off Considerations ....................... 14
   5.1 Power-up Sequence ................................................................................ 14
   5.2 Power-off Sequence ................................................................................ 14
6. Calibration Considerations .......................................................................... 15
   6.1 DDR2-SDRAM or LPDDR1-SDRAM Calibration ..................................... 15
       6.1.1 Hardware .................................................................................. 15
       6.1.2 Software .................................................................................... 15
   6.2 LPDDR2-SDRAM Calibration .................................................................. 15
       6.2.1 Hardware .................................................................................. 15
       6.2.2 Software .................................................................................... 16
7. DDR2 Electromagnetic Compatibility (EMC) Improvement ......................... 17
   7.1 Simultaneous Switching .......................................................................... 17
7.2 Overshoots ................................................................. 17

8. Multi-port DDR Controller Configuration ................................................. 18
  8.1 DDR2-SDRAM Initialization ......................................................... 18
  8.2 LPDDR2-SDRAM Initialization ....................................................... 18
  8.3 Micron® MT47H128M16 DDR2 SDRAM (MPDDRC Configuration Example). ................................................. 18

Revision History ........................................................................................................ 30
1. **Multi-port DDR Controller Overview**

The Multi-port DDR Controller (MPDDRC) extends the memory capabilities of a chip by providing the interface to an external 32-bit SDRAM device. The page size supports ranges from 2048 to 16384, and a number of columns from 256 to 4096. It supports word (32-bit), half-word (16-bit), byte (8-bit) accesses on a 32-bit data path.

The MPDDRC supports a read or write burst length of four locations thanks to the 4-port architecture. It keeps track of the active row in each bank, thus maximizing the SDRAM performance, e.g., the application may be placed in one bank and data in the other banks. So as to optimize performance, it is advisable to avoid accessing different rows in the same bank (Open Bank Policy).

The MPDDRC supports a CAS latency of 2 and 3 and optimizes the read access depending on the frequency.

Self-refresh, power down and deep power down mode features allow to minimize the consumption of SDRAM device.

The MPDDRC I/Os are powered by VDDIODDR. For DDR2-SDRAM and LPDDR1-SDRAM, VDDIODDR is set to 1.8V nominal, for LPDDR2-SDRAM, VDDIODDR is set to 1.2V nominal.

The DDR Chip Select allows to have 512 Mbytes of SDRAM, from address 0x2000 0000 to address 0x4000 0000.
2. Multi-Port DDR Controller Signals Definition

The MPDDRC manages 4-bank and 8-bank SDRAM devices. The signals generated by the controller are defined in Table 2-1.

<table>
<thead>
<tr>
<th>Signal Name</th>
<th>Function</th>
<th>Type</th>
<th>Active Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDR_VREF</td>
<td>Reference Voltage</td>
<td>Input</td>
<td></td>
<td>Used by the input buffers of the DDR2 memories as well as the DDR2 controller to determine logic levels. VREF is specified to be ( \frac{1}{2} ) the power supply voltage and is created using a voltage divider constructed from two 1.5 KΩ, 1% tolerance resistors.</td>
</tr>
<tr>
<td>DDR_CALP</td>
<td>LPDDR2 Calibration Positive Reference Input</td>
<td>Used to calibrate I/O. See calibration section for more details.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDR_CALN</td>
<td>LPDDR2 Calibration Negative Reference Input</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDR_CK, DDR_CKN</td>
<td>DDR2 Differential Clock</td>
<td>Output</td>
<td></td>
<td>Differential clock signals that feed the SDRAM device. All other signals take those two signals as a reference.</td>
</tr>
<tr>
<td>DDR_CKE</td>
<td>DDR2 Clock Enable</td>
<td>Output</td>
<td>High</td>
<td>Acts as an inhibit signal to the DDR device. DDR_CKE remains high during valid DDR2 access (Read, Write, Prech). This signal goes low when the device is in power down mode or in self-refresh mode; a self-refresh command can be issued by the controller (refer to the DDR2 controller self-refresh mode).</td>
</tr>
<tr>
<td>DDR_CS</td>
<td>DDR2 Controller Chip Select Output</td>
<td>Low</td>
<td>When the Chip Select (DDR_CS) is low, the command input is valid. When it is high, the commands are ignored but the operation continues.</td>
<td></td>
</tr>
<tr>
<td>DDR_BA[2..0]</td>
<td>Bank Select</td>
<td>Output</td>
<td>Low</td>
<td>Select the bank to address when a command is input. Read/write or precharge is applied to the bank selected by DDR_BA0, DDR_BA1, or DDR_BA2.</td>
</tr>
<tr>
<td>DDR_WE</td>
<td>DDR2 Write Enable</td>
<td>Output</td>
<td>Low</td>
<td>The Row Address Strobe (DDR_RAS) and the Column Address Strobe (DDR_CAS) will assert to indicate that the corresponding address is present on the bus. The conjunction with Write Enable (DDR_WE) and chip select (SDCS), at the rising edge of the clock (DDR_CK) or the falling edge of the #clock (#DDR_CK), determines the DDR2 operation.</td>
</tr>
<tr>
<td>DDR_RAS - DDR_CAS</td>
<td>Row and Column Signal</td>
<td>Output</td>
<td>Low</td>
<td>SDRAM controller address lines, respectively bounded to [A0:A13] on the microcontroller.</td>
</tr>
<tr>
<td>DDR_A[13..0]</td>
<td>DDR2 Address Bus</td>
<td>Output</td>
<td></td>
<td>SDRAM controller data lines, respectively bounded to [DDR_D31:DDR_D0] on the microcontroller.</td>
</tr>
<tr>
<td>DDR_D[31..0]</td>
<td>DDR2 Data Bus</td>
<td>I/O/-PD</td>
<td></td>
<td>DDR_DQS[0..3]: Data Strobe. The data is sampled on DDR_DQS edges. DDR_DQSN[0..3]: Negative Data Strobe, for LPDDR2-SDRAM. DQSN must be connected to DDR_VREF for DDR2 memories.</td>
</tr>
<tr>
<td>DDR_DQS[3..0], DDR_DQSN[3..0]</td>
<td>Differential Data Strobe</td>
<td>I/O/-PD</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 2-1. SDRAM Controller Signals (Continued)

<table>
<thead>
<tr>
<th>Signal Name</th>
<th>Function</th>
<th>Type</th>
<th>Active Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDR_DQM[3..0]</td>
<td>Write Data Mask</td>
<td>Output</td>
<td></td>
<td>Data is accessed in 32 bits by means of DDR_DQM[3..0], which are respectively the highest to lowest mask bit for the DDR2 data on the bus.</td>
</tr>
</tbody>
</table>
3. **SDRAM Connection on SAMA5D3x**

The SAMA5D3x microprocessor supports 32-bit DDR2 devices on DDR Chip Select area (0x2000 0000 memory zone). This memory area has a length of 512 Mbyte.

The user interface to configure the MPDDR controller is mapped at address 0xFFFF EA00.

Each memory device must use sufficient decoupling to provide an efficient filtering on the power supply rails.

3.1 **32-bit Using 2x16-bit DDR2-SDRAM Implementation**

3.1.1 **Hardware Configuration**

![SDRAM Connection Diagram]

3.1.2 **Software Configuration**

Refer to Section 8.1 “DDR2-SDRAM Initialization” on page 18 for information on the DDR2 initialization sequence.

3.2 **DDR2 VREF Signal Considerations**

DDR_VREF which is half the interface voltage, or 0.9 V, is provided by a voltage divider of 1.8 V using two 1.5 KΩ, 1% tolerance resistors.

DDR_VREF is not a high current supply, but it is important to keep it as quiet as possible with minimal inductance. DQSN[3:0] must be connected to DDR_VREF for DDR2 memories.
3.3 32-bit Using 2x16-bit LPDDR2-SDRAM Implementation

3.3.1 Hardware Configuration
3.3.2 Software Configuration

Refer to Section 8.2 “LPDDR2-SDRAM Initialization” on page 18 for information on the LPDDR2 initialization sequence.

3.4 LPDDR2 VREF Signal Considerations

DDR_VREF which is half the interface voltage, or 0.6 V, is provided by a voltage divider of 1.2 V using two 1.5 KΩ, 1% tolerance resistors.

DDR_VREF is not a high current supply, but it is important to keep it as quiet as possible with minimal inductance.

To reduce noise two VREF pins are needed for LPDDR2-SDRAM: VREFCA and VREFDQ.
4. Layout and Design Constraints

4.1 General Considerations

This section provides routing guidelines for layout and design of a printed circuit board using high-speed memories. The signal integrity rules for high-speed interfaces need to be considered. In fact, it is highly recommended that the board design be simulated to determine optimum layout for signal integrity and quality. Keep in mind that this document can only highlight the most important issues that should be considered when designing a board with high-speed memories. The designer has to take into account the corresponding information (specification, design guidelines, etc.) contained in the documentation for each interface that is to be implemented on board.

4.2 DDR2/LPDDR Bus Interface Controller

Bus signals can be split in three groups:

1. Differential Clock source, VREF middle voltage point for DDR2 reference voltage
   - This first and most critical signal group is set up by only the CK, NCK signals and the VREF reference voltage.

2. Bus, strobe and Mask signals
   - Data bus signals
   - DQS Signals
   - DQM Signals

3. Address, Control signals
   - Address bus signals (DDR_Ax)
   - DDR_BAx signals (Bank select signals)
   - CKE signal, RAS, CAS, NWE and NCS control signals

4.2.1 Signals Group 1

All Group 1 signals are the most critical and shall be routed at first. The Clock is driven in differential mode. Two traces shall be planned to drive this signal to DDR2 packages. The clock traces shall have the same impedance therefore:

- Both traces shall route on outer layer,
- Both traces shall make parallel,
- Clock traces shall use only 2 via/trace.

About VREF, this voltage reference must be considered like a potential victim. This voltage reference traces versus other traces must be stood back from noisy digital traces (in tree dimension: above and below layers and on both sides): Maintain a 15–20 mil clearance from other nets.

This net shall be larger other traces (like a small local voltage plane) and shall be on layer immediately on facing ground layer.

4.2.2 Signals Group 2

All Group 2 signals shall be routed by keeping propagation delay equal as first constraint:

- Route each data group (DQ + DQS + DM) on same layer to match propagation delays and minimize skew between these signals,
- Between DQ and DQM signals, 3 widths minimum space must be set up, but, between DQS (potential victim) and DQ/DQM, 4 widths shall be set up.
4.2.3 Signals Group 3

All Group 3 signals shall be routed by minimizing crosstalk with [DQ, DQS, DQM] ↔ [Addresses, CTRL Bus]: Maintain a gap between the two groups and do not interlace them.

Table 4-1. Example of PCB Stackup

<table>
<thead>
<tr>
<th>Layer</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer 1 (Top)</td>
<td>Signal</td>
<td>Differential and critical signals: oscillators, quartz, clock, DDR_VREF, DDR_CLK / DDR_NCLK, CLK_EBI / NCLK_EBI</td>
</tr>
<tr>
<td>Layer 2</td>
<td>GND</td>
<td>GND</td>
</tr>
<tr>
<td>Layer 3</td>
<td>Signal</td>
<td>Addresses and data buses</td>
</tr>
<tr>
<td>Layer 4</td>
<td>Plane Power and Signal</td>
<td>Non-critical signal</td>
</tr>
<tr>
<td>Layer 5</td>
<td>Signal</td>
<td>Free</td>
</tr>
<tr>
<td>Layer 6</td>
<td>Signal</td>
<td>Free</td>
</tr>
<tr>
<td>Layer 7</td>
<td>Plane Power and Signal</td>
<td>Non-critical signal</td>
</tr>
<tr>
<td>Layer 8</td>
<td>Signal</td>
<td>Addresses and data buses</td>
</tr>
<tr>
<td>Layer 9</td>
<td>GND</td>
<td>GND</td>
</tr>
<tr>
<td>Layer 10 (Bottom)</td>
<td>Signal</td>
<td>Differential and critical signals: Same as Layer 1 (Top)</td>
</tr>
</tbody>
</table>

4.3 EBI Trace Routing Guidelines

4.3.1 Topology About the EBI Bus

Bus impedance: Maintain an impedance of 50 ohms to ±10%.

4.3.2 Placement

Place the highest-speed/highest-current components as far from the I/O connections as possible.

4.3.3 Bypassing Capacitors

Keep all surface traces that run between the pads of the decoupling capacitors and their vias as short and wide as possible. Use as small a body size for a decoupling capacitor as you can afford and minimize the length of all connections from the capacitor pads to the power and ground planes.

4.3.4 Trace Length

Keep the time delay of stubs less than 20% of the rise time of the fastest signals. Route the shorter path between MPU and resistor networks dedicated to the memories.

4.3.5 Trace Spacing

For microstrip or stripline transmission lines, keep the spacing between adjacent signal paths at least twice the line width. Keep all traces at least five line widths from the edge of the board.

4.3.6 Via

Use vias as large in diameter as practical when routing to power or ground planes.
4.3.7 Ground Plane(s)
Follow the return path of each signal and keep the width of the return path under each signal path at least as wide, and preferably at least three times as wide, as the signal trace.
Route signal traces around rather than across return-path discontinuities.

4.3.8 Power Plane(s)
- Minimize the loop inductance between the power and ground paths.
- Allocate power and ground planes on adjacent layers with as thin a dielectric as you can afford.
- Route the power and ground planes as close as possible to the surface where the decoupling capacitors are mounted.
- Supply voltages must be composed of planes only, not traces. Short connections (= 8 mil) are commonly used to attach vias to planes. Any connections required from supply voltages to vias for device pins or decoupling capacitors should be as short and as wide as possible to minimize trace impedance (20 mil trace width).

4.3.9 General Considerations for High-Speed Differential Interfaces
The following is a list of suggestions for designing with high-speed differential signals.
- Use controlled impedance PCB traces that match the specified differential impedance.
- Keep the trace lengths of the differential signal pairs as short as possible.
- The differential signal pair traces should be trace-length matched and the maximum trace-length mismatch should not exceed the specified values. Match each differential pair per segment.
- Maintain parallelism and symmetry between differential signals with the trace spacing needed to achieve the specified differential impedance.
- Maintain maximum possible separation between the differential pairs and any high-speed clocks/periodic signals (CMOS/TTL) and any connector leaving the PCB (such as, I/O connectors, control and signal headers, or power connectors).
- Route differential signals on the signal layer nearest to the ground plane using a minimum of vias and corners. This will reduce signal reflections and impedance changes. Use GND stitching vias when changing layers.
- Route CMOS/TTL and differential signals on a different layer(s), which should be isolated by the power and ground planes.
- Avoid tight bends. When it becomes necessary to turn 90°, use two 45° turns or an arc instead of making a single 90° turn.
- Do not route traces under crystals, crystal oscillators, clock synthesizers, magnetic devices or ICs that use, and/or generate, clocks.
- Stubs on differential signals should be avoided due to the fact that stubs will cause signal reflections and affect signal quality.
- Keep the length of high-speed clock and periodic signal traces that run parallel to high-speed signal lines at a minimum to avoid crosstalk. Based on EMI testing experience, the minimum suggested spacing to clock signals is 50 mil.
- Use a minimum of 20 mil spacing between the differential signal pairs and other signal traces for optimal signal quality. This helps to prevent crosstalk.
- Route all traces over continuous planes (VCC or GND) with no interruptions.
Figure 4-1. Schematic SAMA5D3x-CM Memory
5. LPDDR2-SDRAM Power-up and Power-off Considerations

5.1 Power-up Sequence
A specific sequence must be used to power up the LPDDR2-SDRAM device, this procedure is mandatory. Power-up and initialization by means other than those specified will result in undefined operation. Refer to the LPDDR2-SDRAM datasheet for full details and timings.

5.2 Power-off Sequence
A specific sequence must be used to power off the LPDDR2-SDRAM device, this procedure is mandatory. Power-off by means other than those specified will result in uncontrolled power-off.

The VDDIODDR power fail must be handled at system level (IRQ or FIQ). When this event occurs the LPDDR2 power-off sequence is to be applied using the LPDDR2_PWOFF bit (bit 3 in MPDDRC_LPR) before the VDDIODDR power-off.

Uncontrolled power-off sequence can be applied only up to 400 times in the life of a LPDDR2-SDRAM device. Refer to the LPDDR2-SDRAM datasheet for full details and timings.
6. Calibration Considerations

SAMA5D3x embeds a pad calibration feature that performs bus impedance adaptation, improving signal integrity. This leads to
- reduction of overshoots
- reduction of Electromagnetic Interference (EMI)
- reduction of power consumption on I/Os
- eliminating the need of serial resistor on data lines

6.1 DDR2-SDRAM or LPDDR1-SDRAM Calibration

6.1.1 Hardware

DDR2 or LPDDR1 calibration requires connecting a 200Ω resistor on DDR_CALP to GND and on DDR_CALN to VDDIODDR.

6.1.2 Software

DDR2 or LPDDR1 software calibration is to be done only once and requires following steps:
1. Set RDIV field in MPDDRC_IO_CALIBR register, according to the board impedance
2. Calculate TZQIO value using the formula $TZQIO = (DDRCLK \times 20 \text{ ns}) + 1$
3. Set TZQIO time in MPDDRC_IO_CALIBR
4. Activate calibration by setting the 5th bit in High Speed Register (0xFFFFEA24)

6.2 LPDDR2-SDRAM Calibration

6.2.1 Hardware

LPDDR2 calibration requires to connect 240Ω resistor on DDR_CALP to GND and DDR_CALN to VDDIODDR.
6.2.2 Software

LPDDR2 software calibration is to be done only once and requires following steps:

1. Set RDIV field in MPDDRC_IO_CALIBR register, according to the board impedance
2. Calculate TZQIO value using the formula TZQIO = (DDRCLK × 20 ns) + 1
3. Set TZQIO time in MPDDRC_IO_CALIBR
4. Program Short Calibration Time with ZQCS field in LPDDR2_TIM_CAL, according to the LPDDR2-SDRAM datasheet
5. Calculate the calibration pulse over Process Voltage Temperature (PVT) according to the refresh rate, the temperature and voltage expected change, the temperature and voltage sensitivities defined in LPDDR2-SDRAM datasheet, using the formula ZQCorrection / ((TSens × Tdriftrate) + (VSens × Vdriftrate))
6. Set the value in field COUNT_CAL in LPDDR2_CAL_MMR4 register

For example, if TSens = 0.75%/°C, VSens = 0.2%/mV, Tdriftrate = 1°C/sec and driftrate = 15mV/sec, then the interval between ZQCS commands is calculated as 1.5 / ((0.75 × 1) + (0.2 × 15)) = 0.4 sec.

This LPDDR2-SDRAM devices require a calibration every 0.4s.

The value to be loaded depends on average time between REFRESH commands, tREF.

For an LPDDR2-SDRAM with a time between refresh of 7.8 µs, the value of the Calibration Timer Count bit is programmed (0.4/7.8 × 10^-6) = 0xC852.
7. DDR2 Electromagnetic Compatibility (EMC) Improvement

7.1 Simultaneous Switching

Simultaneous switching is the worst enemy of EMC at device operation level. SAMA5D3x embeds pad calibration feature that performs bus impedance adaptation improving signal integrity, reducing current driven and so power consumption.

7.2 Overshoots

Overshoots occur when the current driven is too high. SAMA5D3x embeds pad calibration feature that performs bus impedance adaptation improving signal integrity, reducing overshoots.
8. Multi-port DDR Controller Configuration

8.1 DDR2-SDRAM Initialization
The DDR2-SDRAM initialization sequence is described in the section “DDR2-SDRAM Initialization” of the SAMA5D3 Series datasheet. For an example of initialization, see the “DDR2 Initialization Code Example” on page 20.

8.2 LPDDR2-SDRAM Initialization
The LPDDR2-SDRAM initialization sequence is described in the section “Low-power DDR2-SDRAM Initialization” of the SAMA5D3 Series datasheet.

8.3 Micron® MT47H128M16 DDR2 SDRAM (MPDDRC Configuration Example)
The Micron MT47H128M16 are 256 MB DDR2 SDRAM devices arranged as 16 Mbit × 16 × 8 banks with a CAS latency of 3 at 134 MHz. These devices are featured on the SAMA5D3x-CM.

Table 8-1 gives the delay in ns extracted from the DDR2 SDRAM datasheet, the corresponding number of cycles at 134 MHz, and the field to program these values accordingly in a system running at 536 MHz for Processor Clock and 134 MHz for System Clock.

Table 8-1. MPDDRC Configuration Example with Micron MT47H128M16

<table>
<thead>
<tr>
<th>Description</th>
<th>System Configuration</th>
<th>Value in Micron Datasheet</th>
<th>Value in SAMA5D3 Datasheet</th>
<th>Register</th>
<th>Bit or Field</th>
<th>Register or Field Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>PLL Frequency</td>
<td>1072 MHz</td>
<td></td>
<td>CKGR_PLLAR</td>
<td>0x215C3F01</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Processor / Bus Clock</td>
<td>536 / 134 MHz</td>
<td></td>
<td>PMC_MCKR</td>
<td>0x00001202</td>
<td></td>
</tr>
<tr>
<td></td>
<td>System Clock</td>
<td>DDR clock enable</td>
<td></td>
<td>PMC_SCER</td>
<td>0x00000005</td>
<td></td>
</tr>
<tr>
<td>DDR2 Device</td>
<td>MPDDRC Configuration Register(^{(2)})</td>
<td>MPDDRC_CR</td>
<td>0x13D</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Number of Columns</td>
<td>10</td>
<td>10</td>
<td>MPDDRC_CR</td>
<td>NC</td>
<td>0x1</td>
</tr>
<tr>
<td></td>
<td>Number of Rows</td>
<td>14</td>
<td>14</td>
<td>MPDDRC_CR</td>
<td>NR</td>
<td>0x3</td>
</tr>
<tr>
<td></td>
<td>CAS Latency</td>
<td>3</td>
<td>3 cycles</td>
<td>MPDDRC_CR</td>
<td>CAS</td>
<td>0x3</td>
</tr>
<tr>
<td></td>
<td>Reset DLL</td>
<td>Disable</td>
<td>Disable DLL</td>
<td>MPDDRC_CR</td>
<td>DLL</td>
<td>0x0</td>
</tr>
<tr>
<td></td>
<td>Drive Strength (DDR2 only)</td>
<td>Weak</td>
<td>Weak</td>
<td>MPDDRC_CR</td>
<td>DIC_DS</td>
<td>0x1</td>
</tr>
<tr>
<td></td>
<td>Output Driver Impedance control</td>
<td>Normal</td>
<td>Normal</td>
<td>MPDDRC_CR</td>
<td>DIC/DS</td>
<td>0x0</td>
</tr>
<tr>
<td></td>
<td>Disable DLL</td>
<td>No</td>
<td>No</td>
<td>MPDDRC_CR</td>
<td>DIS_DLL</td>
<td>0x0</td>
</tr>
<tr>
<td></td>
<td>Off-Chip Driver</td>
<td>(1)</td>
<td>MPDDRC_CR</td>
<td>OCD</td>
<td>0x0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mask Data is shared</td>
<td>Not shared</td>
<td>Not shared</td>
<td>MPDDRC_CR</td>
<td>DMQS</td>
<td>0x0</td>
</tr>
<tr>
<td></td>
<td>Enable Read Measure</td>
<td>Disabled</td>
<td>Disabled</td>
<td>MPDDRC_CR</td>
<td>ENRDM</td>
<td>0x0</td>
</tr>
<tr>
<td></td>
<td>MPDDRC I/O Calibration Register(^{(2)})</td>
<td>MPDDRC_IO_CALIBR</td>
<td>0x404</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Resistor Divider</td>
<td>50Ω</td>
<td>MPDDRC_IO_CALIBR</td>
<td>RDIV</td>
<td>0x4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IO Calibration</td>
<td></td>
<td>MPDDRC_IO_CALIBR</td>
<td>TZQIO</td>
<td>0x4</td>
<td></td>
</tr>
</tbody>
</table>
### Table 8-1. MPDDRC Configuration Example with Micron MT47H128M16 (Continued)

<table>
<thead>
<tr>
<th>Description</th>
<th>System Configuration</th>
<th>Value in Micron Datasheet</th>
<th>Number of Cycles at 134 MHz</th>
<th>Value in SAMA5D3 Datasheet</th>
<th>Register</th>
<th>Bit or Field</th>
<th>Register or Field Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MPDDRC Timing Parameter 0 Register</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACTIVATE to PRECHARGE Time (delay)</td>
<td></td>
<td>45 ns</td>
<td>6</td>
<td>MPDDRC_TPR0</td>
<td>TRAS</td>
<td>0x6</td>
<td></td>
</tr>
<tr>
<td>ACTIVATE to READ/WRITE Time (delay)</td>
<td></td>
<td>15 ns</td>
<td>2</td>
<td>MPDDRC_TPR0</td>
<td>TRCD</td>
<td>0x2</td>
<td></td>
</tr>
<tr>
<td>Last DATA-IN to PRECHARGE Time (delay)</td>
<td></td>
<td>15 ns</td>
<td>2</td>
<td>MPDDRC_TPR0</td>
<td>TWR</td>
<td>0x2</td>
<td></td>
</tr>
<tr>
<td>REFRESH to ACTIVATE Time (delay)</td>
<td></td>
<td>55 ns</td>
<td>8</td>
<td>MPDDRC_TPR0</td>
<td>TRC</td>
<td>0x8</td>
<td></td>
</tr>
<tr>
<td>PRECHARGE to ACTIVATE Time (delay)</td>
<td></td>
<td>15 ns</td>
<td>2</td>
<td>MPDDRC_TPR0</td>
<td>TRP</td>
<td>0x2</td>
<td></td>
</tr>
<tr>
<td>ACTIVE bankA to ACTIVE BankB (delay)</td>
<td></td>
<td>10 ns</td>
<td>2</td>
<td>MPDDRC_TPR0</td>
<td>TRRD</td>
<td>0x2</td>
<td></td>
</tr>
<tr>
<td>Internal Write to Read Delay</td>
<td></td>
<td>7.5 ns</td>
<td>1</td>
<td>MPDDRC_TPR0</td>
<td>TWTR</td>
<td>0x1</td>
<td></td>
</tr>
<tr>
<td>Load Mode Register Command to ACTIVE or REFRESH Command (delay)</td>
<td></td>
<td>2 cycles</td>
<td>2</td>
<td>MPDDRC_TPR0</td>
<td>TMRD</td>
<td>0x2</td>
<td></td>
</tr>
<tr>
<td><strong>MPDDRC Timing Parameter 1 Register</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Row Cycle Delay</td>
<td></td>
<td>197.5 ns</td>
<td>27</td>
<td>MPDDRC_TPR1</td>
<td>TRFC</td>
<td>0x1B</td>
<td></td>
</tr>
<tr>
<td>Exit Self Refresh Delay to Non-Read Command</td>
<td></td>
<td>TRFC + 10 ns</td>
<td>28</td>
<td>MPDDRC_TPR1</td>
<td>TXSNR</td>
<td>0x1C</td>
<td></td>
</tr>
<tr>
<td>Exit Self Refresh Delay to Read Command</td>
<td></td>
<td>200 cycles</td>
<td>200</td>
<td>MPDDRC_TPR1</td>
<td>TXSRD</td>
<td>0xC8</td>
<td></td>
</tr>
<tr>
<td>Exit Power-down Delay to First Command</td>
<td></td>
<td>2 cycles</td>
<td>2</td>
<td>MPDDRC_TPR1</td>
<td>TXP</td>
<td>0x2</td>
<td></td>
</tr>
<tr>
<td><strong>MPDDRC Timing Parameter 2 Register</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exit Active Power Down Delay to Read Command (Fast Exit)</td>
<td></td>
<td>2 cycles</td>
<td>2</td>
<td>MPDDRC_TPR2</td>
<td>TXARD</td>
<td>0x2</td>
<td></td>
</tr>
<tr>
<td>Exit Active Power Down Delay to Read Command (Slow Exit)</td>
<td></td>
<td></td>
<td>7</td>
<td>MPDDRC_TPR2</td>
<td>TXARDS</td>
<td>0x7</td>
<td></td>
</tr>
<tr>
<td>Row Precharge All Delay</td>
<td></td>
<td>2</td>
<td></td>
<td>MPDDRC_TPR2</td>
<td>TRPA</td>
<td>0x2</td>
<td></td>
</tr>
<tr>
<td>Read to Precharge</td>
<td></td>
<td>15 ns</td>
<td>2</td>
<td>MPDDRC_TPR2</td>
<td>TRTP</td>
<td>0x2</td>
<td></td>
</tr>
<tr>
<td><strong>MPDDRC Memory Device Register</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Memory Device</td>
<td></td>
<td>DDR2-SDRAM</td>
<td></td>
<td>MPDDRC_MD</td>
<td>MD</td>
<td>0x6</td>
<td></td>
</tr>
<tr>
<td>Data Bus Width</td>
<td></td>
<td>32 bits</td>
<td></td>
<td>MPDDRC_MD</td>
<td>DBW</td>
<td>0x0</td>
<td></td>
</tr>
<tr>
<td><strong>MPDDRC Refresh Timer Register - Timer Count</strong></td>
<td></td>
<td></td>
<td>7.8 µs</td>
<td>MPDDRC_RTR</td>
<td></td>
<td>0x410</td>
<td></td>
</tr>
</tbody>
</table>

Notes:
1. OCD is not supported, but it is a mandatory step in the DDR2 initialization phase.
2. Any bits or fields of this register not listed in the table must remain unchanged.
Appendix A. DDR2 Initialization Code Example

This appendix provides the following example of the DDR2 initialization code, associated to the different steps of the DDR2-SDRAM initialization sequence:

```c
int ddram_init(unsigned int ddram_controller_address, unsigned int ddram_address, struct SDdramConfig *ddram_config)
{
    volatile unsigned int i;
    unsigned int cr = 0;

    // Initialization Step 1: Program the memory device type
    // Configure the DDR controller
    write_ddramc(ddram_controller_address, HMPDDRC_MDR, ddram_config->ddramc_mdr);
    // Program the DDR Controller
    write_ddramc(ddram_controller_address, HMPDDRC_CR, ddram_config->ddramc_cr);

    // Initialization Step 2: assume timings for 7.5 ns min clock period
    write_ddramc(ddram_controller_address, HMPDDRC_T0PR, ddram_config->ddramc_t0pr);
    // psDDRC->HMPDDRC_T1PR
    write_ddramc(ddram_controller_address, HMPDDRC_T1PR, ddram_config->ddramc_t1pr);
    // psDDRC->HMPDDRC_T2PR
    write_ddramc(ddram_controller_address, HMPDDRC_T2PR, ddram_config->ddramc_t2pr);

    // Initialization Step 3: NOP command -> allow to enable clk
    write_ddramc(ddram_controller_address, HMPDDRC_MR, AT91C_DDRC2_MODE_NOP_CMD);
    *((unsigned volatile int*) ddram_address) = 0;
    // Initialization Step 3 (must wait 200 µs) (6 core cycles per iteration, core is at 536 MHz:
    // min 17,733 loops)
    for (i = 0; i < 17800; i++) {
        asm("    nop");
    }

    // Initialization Step 4: A NOP command is issued to the DDR2-SDRAM
    // NOP command -> allow to enable cke
    write_ddramc(ddram_controller_address, HMPDDRC_MR, AT91C_DDRC2_MODE_NOP_CMD);
    *((unsigned volatile int*) ddram_address) = 0;
    // wait 400 ns min
    for (i = 0; i < 250; i++) {
        asm("    nop");
    }
}
```
// Initialization Step 5: Set All Bank Precharge
write_ddramc(ddram_controller_address, HMPDDRC_MR, AT91C_DDRC2_MODE_PRCGALL_CMD);
*{(unsigned volatile int*) ddram_address) = 0;
// wait 400 ns min
for (i = 0; i < 250; i++) {
  asm("   nop");
}

// Initialization Step 6: Set EMR operation (EMRS2)
write_ddramc(ddram_controller_address, HMPDDRC_MR, AT91C_DDRC2_MODE_EXT_LMR_CMD);
*{(unsigned int *)(ddram_address + 0x4000000)) = 0;
// wait 2 cycles min
for (i = 0; i < 100; i++) {
  asm("   nop");
}

// Initialization Step 7: Set EMR operation (EMRS3)
write_ddramc(ddram_controller_address, HMPDDRC_MR, AT91C_DDRC2_MODE_EXT_LMR_CMD);
*{(unsigned int *)(ddram_address + 0x6000000)) = 0;
// wait 2 cycles min
for (i = 0; i < 100; i++) {
  asm("   nop");
}

// Initialization Step 8: Set EMR operation (EMRS1)
write_ddramc(ddram_controller_address, HMPDDRC_MR, AT91C_DDRC2_MODE_EXT_LMR_CMD);
*{(unsigned int *)(ddram_address + 0x2000000)) = 0;
// wait 200 cycles min
for (i = 0; i < 10000; i++) {
  asm("   nop");
}

// Initialization Step 9: enable DLL reset
cr = read_ddramc(ddram_controller_address, HMPDDRC_CR);
write_ddramc(ddram_controller_address, HMPDDRC_CR, cr | AT91C_DDRC2_DLL_RESET_ENABLED);

// Initialization Step 10: reset DLL
write_ddramc(ddram_controller_address, HMPDDRC_MR, AT91C_DDRC2_MODE_EXT_LMR_CMD);
*{(unsigned volatile int*) ddram_address)) = 0;
// wait 2 cycles min
for (i = 0; i < 100; i++) {
  asm("   nop");
}
Initialization Step 11: Set All Bank Precharge

```c
write_ddramc(ddram_controller_address, HMPDDRC_MR, AT91C_DDRC2_MODE_PRCGALL_CMD);
*((unsigned volatile int*) ddram_address)) = 0;
```

// wait 400 ns min
for (i = 0; i < 250; i++) {
    asm("    nop");
}

Initialization Step 12: Two auto-refresh (CBR) cycles are provided. Program the auto refresh command (CBR) into the Mode Register.

```c
write_ddramc(ddram_controller_address, HMPDDRC_MR, AT91C_DDRC2_MODE_RFSH_CMD);
*((unsigned volatile int*) ddram_address)) = 0;
```

// wait 10 cycles min
for (i = 0; i < 100; i++) {
    asm("    nop");
}

// Set 2nd CBR
write_ddramc(ddram_controller_address, HMPDDRC_MR, AT91C_DDRC2_MODE_RFSH_CMD);
*((unsigned volatile int*) ddram_address)) = 0;

// wait 10 cycles min
for (i = 0; i < 100; i++) {
    asm("    nop");
}

Initialization Step 13: Program DLL field into the Configuration Register to low (Disable DLL reset).

```c
cr = read_ddramc(ddram_controller_address, HMPDDRC_CR);
write_ddramc(ddram_controller_address, HMPDDRC_CR, cr & (~AT91C_DDRC2_DLL_RESET_ENABLED));
```

Initialization Step 14: A Mode Register set (MRS) cycle is issued to program the parameters of the DDR2-SDRAM devices.

```c
write_ddramc(ddram_controller_address, HMPDDRC_MR, AT91C_DDRC2_MODE_LMR_CMD);
*((unsigned volatile int*) ddram_address)) = 0;
```

Initialization Step 15: Program OCD field into the Configuration Register to high (OCD calibration default).

```c
cr = read_ddramc(ddram_controller_address, HMPDDRC_CR);
write_ddramc(ddram_controller_address, HMPDDRC_CR, cr | AT91C_DDRC2_OCD_DEFAULT);
```

Initialization Step 16: An Extended Mode Register set (EMRS1) cycle is issued to OCD default value.

```c
write_ddramc(ddram_controller_address, HMPDDRC_MR, AT91C_DDRC2_MODE_EXT_LMR_CMD);
*((unsigned int*) (ddram_address + 0x2000000)) = 0;
```

// wait 2 cycles min
for (i = 0; i < 100; i++) {
    asm("    nop");
}
Initialization Step 17: Program OCD field into the Configuration Register to low (OCD calibration mode exit). Write a 1 to DIC_DS field to use DDR2 weak drive strength.

```c
cr = read_ddramc(ddram_controller_address, HMPDDRC_CR);
write_ddramc(ddram_controller_address, HMPDDRC_CR, cr & (~AT91C_DDRC2_OCD_EXIT));
write_ddramc(ddram_controller_address, HMPDDRC_CR, cr | (AT91C_DDRC2_WEAKSTRENGTH));
```

Initialization Step 18: An Extended Mode Register set (EMRS1) cycle is issued to enable OCD exit.

```c
write_ddramc(ddram_controller_address, HMPDDRC_MR, AT91C_DDRC2_MODE_EXT_LMR_CMD);
*((unsigned int*) (ddram_address + 0x600000)) = 0;
```

// wait 2 cycles min
for (i = 0; i < 100; i++) {
    asm("    nop");
}

Initialization Step 19, 20: A mode Normal command is provided. Program the Normal mode into Mode Register.

```c
write_ddramc(ddram_controller_address, HMPDDRC_MR, AT91C_DDRC2_MODE_NORMAL_CMD);
*((unsigned volatile int*) ddram_address)) = 0;
```

Initialization Step 21: Write the refresh rate into the count field in the Refresh Timer Register.

```c
// Set Refresh timer
write_ddramc(ddram_controller_address, HMPDDRC_RTR, ddram_config->ddramc_rtr);
// OK, now we are ready to work on the DDRSDR
// wait for the end of calibration
for (i = 0; i < 500; i++) {
    asm("    nop");
}
return 0;
```
Appendix B. LPDDR2 Initialization Code Example

This appendix provides the following example of the LPDDR2 initialization code, associated to the different steps of the LPDDR2-SDRAM initialization sequence:

```c
void LPDDR2_MT42L128M16D1_Initialise( LPDDR2 psst_ddr2 )
{
    /****************************************************************************/
    // Initialization Step 1
    // Program the memory device type into the Memory Device Register
    /****************************************************************************/
    // Memory device = LPDDR2 => MPDDRC_MD_MD_LPDDR2_SDRAM
    // Data bus width = 32 bits => 0x0 (The system is in 64 bits, thus memory data bus width should be 32 bits)
    MPDDRC->MPDDRC_MD = MPDDRC_MD_MD_LPDDR2_SDRAM ;// LPDDR2

    /****************************************************************************/
    // Initialization Step 2
    // Program the features of Low-power DDR2-SDRAM device into the Timing Register
    // (asynchronous timing, trc, tras, etc.) and into the Configuration Register (number of columns, rows, banks, CAS latency and output drive strength) (see Section 8.3 on page 35, Section 8.4 on page 39 and Section 8.5 on page 41).
    /****************************************************************************/
    // MPDDRC Configuration Register
    // NC = 0x0. Number of column to address is 9 (extract from memory data sheet)
    // NR = 0x2. Number of row to address is 13 (extract from memory data sheet)
    // CAS latency = 3. FPGA platform runs at 30 MHz (depends on the frequency, check memory data sheet)
    // No DLL in LPDDR2 devices => DLL, DIS_DLL and DIC_DS = 0
    // ZQ = 0. ZQ_INIT calibration will be performed later
    // OCD = 0
    // DQS = 0. Bus is not shared
    // NDQS = 0. LPDDR2 uses DQS and NDQS
    // DECOD = 0. Sequential decoding is chosen (may changed after the initialization)
    // UNAL = 1; Unaligned accesses will be performed
    MPDDRC->MPDDRC_CR = (psst_ddr2.n_col|     // 9 col + 8 COL supported or not
                  psst_ddr2.n_row  | // 14 row
                  MPDDRC_CR_CAS_3_LPDDR2    |  // CAS 3
                  MPDDRC_CR_NB_8) |   // 8 banks
                  MPDDRC_CR_UNAL_SUPPORTED|     // Unaligned accesses
                  MPDDRC_CR_ENRDM_ON;
```

---

24 Implementation of DDR2 and LPDDR2 on SAMA5D3x Devices [APPLICATION NOTE]
Atmel-11172B-ATARM-Implementation-of-DDR2-and-LPDDR2-on-SAMA5D3x-Devices-ApplicationNote_26-Feb-15
// Write the LPDDR2 drive strength according to the PCB. It should be set according to RDIV field in MPDDRC I/O Calibration Register
// DS Write-only OP<3:0>
// 0000B: reserved
// 0001B: 34.3-ohm typical
// 0010B: 40-ohm typical (default)
// 0011B: 48-ohm typical
// 0100B: 60-ohm typical
// 0101B: reserved for 68.6-ohm typical
// 0110B: 80-ohm typical
// 0111B: 120-ohm typical (optional)
// All others: reserved

MPDDRC->MPDDRC_LPDDR2_LPR=MPDDRC_LPDDR2_LPR_DS(0x3);

// MPDDRC Timing Parameter

MPDDRC->MPDDRC_TPR0 = MPDDRC_TPR0_TRAS(psst_ddr2.t_tras) | /*03-TRAS tRAS Row active time*/ 
              MPDDRC_TPR0_TRCD(psst_ddr2.t_trcd)  | /*04 -TRC tRCD RAS-to-CAS delay*/  
              MPDDRC_TPR0_TWR(psst_ddr2.t_twr)   | /*05 -TWR tWR WRITE recovery time */  
              MPDDRC_TPR0_TRC(psst_ddr2.t_trc)   | /*06 -TRC tRC ACTI-to-ACTIVT command period*/  
              MPDDRC_TPR0_TRP(psst_ddr2.t_trp)   | /*07 -TRP tRPpb Row precharge time */  
              MPDDRC_TPR0_TRRD(psst_ddr2.t_trrd) | /*08 -TRRD tRRD Active bank a to active bank b*/  
              MPDDRC_TPR0_TWTR(psst_ddr2.t_twtr) | /*09 -TWTR-tWTR Internal WRITE-to-READ command delay*/  
              MPDDRC_TPR0_TMRD(psst_ddr2.t_tmrd)/10 -TMRD-tMRD *;

MPDDRC->MPDDRC_TPR1 = MPDDRC_TPR1_TRFC(psst_ddr2.t_trfc) | /*11 -TRFC tRFCab Refresh cycle time */  
              MPDDRC_TPR1_TXSNR(psst_ddr2.t_txsnr) | /*12 -TXSNR SELF REFRESH exit to next valid delay */  
              MPDDRC_TPR1_TXSRD(psst_ddr2.t_txsrd) | /*13-TXSRD Exit Self Refresh*/  
              MPDDRC_TPR1_TXP(psst_ddr2.t_txp)   | /*14 -TXP-tXP Exit power-down */  

MPDDRC->MPDDRC_TPR2 = MPDDRC_TPR2_TXARD(psst_ddr2.t_txard) | /*15 TXARD-txARD */  
              MPDDRC_TPR2_TXARDS(psst_ddr2.t_tards) | /*16 TXARDS-txARDs */  
              MPDDRC_TPR2_TRPA(psst_ddr2.t_trpa)   | /*17 TRPA-tRPpb Row precharge time (all banks) */  
              MPDDRC_TPR2_TRTP(psst_ddr2.t_trtp)   | /*18 TRTP-tTRP */  
              MPDDRC_TPR2_TFAW(psst_ddr2.t_tfaw)  | /*19 TFAW--tFAW */

MPDDRC->MPDDRC_LPR= 0x00000000; // Set low power register to normal mode

// Initialization Step 3
// An NOP command is issued to the Low-power DDR2-SDRAM. Program the NOP
command into the Mode Register, the application must set the MODE (MDDRC Command Mode) field to 1 in the Mode Register (see Section 8.1 on page 32). Perform a write access to any Low-power DDR2-SDRAM address to acknowledge this command. Now, clocks which drive Low-power DDR2-SDRAM devices are enabled. A minimum pause of 100 ns must be observed to precede any signal toggle.

******************************************************************************************

```c
// Enable Clock output
MPDDRC->MPDDRC_MR = MPDDRC_MR_MODE_NOP_CMD;
*(unsigned int *)DDR_CS_ADDR = 0x00000000; // Access to memory
Wait (0xFFFF); // Delay loop (at least 100 ns)
```

******************************************************************************************

// Initialization Step 4

An NOP command is issued to the Low-power DDR2-SDRAM. Program the NOP command into the Mode Register, the application must set MODE to 1 in the Mode Register (see Section 8.1 on page 32). Perform a write access to any Low-power DDR2-SDRAM address to acknowledge this command. Now, CKE is driven high. A minimum pause of 200 ns must be satisfied before Reset Command.

******************************************************************************************

```c
// Drive CKE high
MPDDRC->MPDDRC_MR = MPDDRC_MR_MODE_NOP_CMD;
*(unsigned int *)DDR_CS_ADDR = 0x00000000; // Access to memory
Wait (0xFFFF); // Delay loop (at least 200 us)
```

******************************************************************************************

// Initialization Step 5

A reset command is issued to the Low-power DDR2-SDRAM. Program LPDDR2_CMD in the MODE (MDDRC Command Mode) and MRS (Mode Register Select LPDDR2) field of the Mode Register, the application must set MODE to 7 and MRS to 63. (see Section 8.1 on page 32). Perform a write access to any Low-power DDR2-SDRAM address to acknowledge this command. Now, the reset command is issued. A minimum pause of 1 us must be satisfied before any commands.

******************************************************************************************

```c
// Issue Reset Command
MPDDRC->MPDDRC_MR = MPDDRC_MR_MODE_LPDDR2_CMD; // MODE = 0x7 and MRS = 0x3F
*(unsigned int *)DDR_CS_ADDR = 0x00000000; // Access to memory
Wait (0xFFFF); // Delay loop (at least 1 us)
```

******************************************************************************************

// Initialization Step 6

A Mode Register Read command is issued to the Low-power DDR2-SDRAM. Program LPDDR2_CMD in the MODE and MRS field of the Mode Register, the application must set MODE to 7 and must set MRS field to 0. (see Section 8.1 on
// page 32). Perform a write access to any Low-power DDR2-SDRAM address to
// acknowledge this command. Now, the Mode Register Read command is issued.
// A minimum pause of 10 is must be satisfied before any commands.
****************************************************************************************
// Mode Register Read command. MODE = 0x7 and MRS = 0x00
MPDDRC->MPDDRC_MR = MPDDRC_MR_MODE_LPDDR2_CMD | MPDDRC_MR_MRS( 0x00);
*(unsigned int *)DDR_CS_ADDR = 0x00000000;// Access to memory
Wait (0xFFFF);// Delay loop (at least 1 us)

Initialization Step 7
A calibration command is issued to the Low-power DDR2-SDRAM. Program the type
of calibration into the Configuration Register, ZQ field, RESET value (see Section 8.3
"MPDDRC Configuration Register" on page 37). In the Mode Register, program the
MODE field to LPDDR2_CMD value, and the MRS field; the application must set
MODE to 7 and MRS to 10 (see Section 8.1 "MPDDRC Mode Register" on page 34).
Perform a write access to any Low-power DDR2-SDRAM address to acknowledge
this command. Now, the ZQ Calibration command is issued. Program the type of calibration
into the Configuration Register, ZQ field,
****************************************************************************************
MPDDRC->MPDDRC_CR&=~MPDDRC_CR_ZQ_Msk;
MPDDRC->MPDDRC_CR|= MPDDRC_CR_ZQ_RESET;
// Mode Register Read command. MODE = 0x7 and MRS = 0x0A

Initialization Step 8
// A Mode Register Write command is issued to the Low-power DDR2-SDRAM. Program
// LPDDR2_CMD in the MODE and MRS field in the Mode Register, the
// application must set MODE to 7 and must set MRS field to 0.5 (see Section 8.1 on
// page 32). The Mode Register Write command cycle is issued to program the parameters
// of the Low-power DDR2-SDRAM devices, in particular burst length. Perform a
// write access to any Low-power DDR2-SDRAM address to acknowledge this command.
// Now, the Mode Register Write command is issued.
****************************************************************************************
// Programm LPDDR2 parameters MODE = 0x7 and MRS = 0x01
MPDDRC->MPDDRC_MR = MPDDRC_MR_MODE_LPDDR2_CMD | MPDDRC_MR_MRS( 0x01);
*(unsigned int *)DDR_CS_ADDR= 0x00000000;// Access to memory
Wait (0xFFFF);// Add a delay loop (not is the programmer datasheet)

Initialization Step 9
// Mode Register Write Command is issued to the Low-power DDR2-SDRAM. Program
// LPPDR2_CMD in the MODE and MRS field in the Mode Register, the
// application must set MODE to 7 and must set MRS field to 2. (see Section 8.1 on
// page 32). The Mode Register Write command cycle is issued to program the parameters
// of the Low-power DDR2-SDRAM devices, in particular CAS latency. Perform a
// write access to any Low-power DDR2-SDRAM address to acknowledge this command.
// Now, the Mode Register Write command is issued.

Programm LPDDR2 CAS MODE = 0x7 and MRS = 0x02
MPDDRC->MPDDRC_MR= MPDDRC_MR_MODE_LPDDR2_CMD |MPDDRC_MR_MRS( 0x02);

*(unsigned int *)DDR_CS_ADDR= 0x00000000;// Access to memory
Wait (0xFFFF);// Add a delay loop (not is the programmer datasheet)

Initialization Step 10
// A Mode Register Write Command is issued to the Low-power DDR2-SDRAM. Program
// LPPDR2_CMD in the MODE and MRS field of the Mode Register, the
// application must set MODE to 7 and must set MRS field to 3. (see Section 8.1 on
// page 32). The Mode Register Write command cycle is issued to program the parameters
// of the Low-power DDR2-SDRAM devices, in particular Drive Strength and Slew
// Rate. Perform a write access to any Low-power DDR2-SDRAM address to acknowledge
// this command. Now, the Mode Register Write command is issued.

Programm LPDDR2 DS MODE = 0x7 and MRS = 0x03
MPDDRC->MPDDRC_MR= MPDDRC_MR_MODE_LPDDR2_CMD |MPDDRC_MR_MRS( 0x03); //0x00000307;

*(unsigned int *)DDR_CS_ADDR= 0x00000000;// Access to memory
Wait (0xFFFF);// Add a delay loop (not is the programmer datasheet)

Initialization Step 11
// A Mode Register Write Command is issued to the Low-power DDR2-SDRAM. Program
// LPPDR2_CMD in the MODE and MRS field of the Mode Register, the
// application must set MODE to 7 and must set MRS field to 16. (see Section 8.1 on
// page 32). Mode Register Write command cycle is issued to program the parameters
// of the Low-power DDR2-SDRAM devices, in particular Partial Array Self Refresh
// (PASR). Perform a write access to any Low-power DDR2-SDRAM address to
// acknowledge this command. Now, the Mode Register Write command is issued.
Programm LPDDR2 PASR MODE = 0x7 and MRS = 0x10
MPDDRC->MPDDRC_MR=MPDDRC_MR_MODE_LPDDR2_CMD | MPDDRC_MR_MRS( 0x10); // 0x00001007;
*(unsigned int *)DDR_CS_ADDR= 0x00000000; // Access to memory
Wait (0xFFFF); // Add a delay loop (not is the programmer datasheet)

Initialization Step 12
Write the refresh rate into the COUNT field in the Refresh Timer register (see page // 33). (Refresh rate = delay between refresh cycles). The Low-power DDR2-SDRAM // device requires a refresh every 7.81 ìs. With a 100 MHz frequency, the refresh timer // count register must to be set with (7.81/100 MHz) = 781 i.e. 0x030d.

MPDDRC->MPDDRC_RTR&=~MPDDRC_RTR_COUNT_Msk;
MPDDRC->MPDDRC_RTR |=MPDDRC_RTR_COUNT(psst_ddr2.t_refresh);
//MPDDRC->MPDDRC_RTR|= MPDDRC_RTR_ADJ_REF ; // MR4 READ enabled
MPDDRC->MPDDRC_MR= 0x00000000; // Set Normal mode
*(unsigned int *)DDR_CS_ADDR= 0x00000000; // Perform

Wait (0xFFFF);
// Launch short ZQ calibration

MPDDRC->MPDDRC_CR&= ~ (MPDDRC_CR_ZQ_Msk); // Enable short calibration in the CR
MPDDRC->MPDDRC_CR |= (MPDDRC_CR_ZQ_SHORT);
MPDDRC->MPDDRC_CR |= MPDDRC_CR_DLL_RESET_ENABLED;

*(unsigned int *)DDR_CS_ADDR= 0x00000000; // Perform
// Calculate ZQS: search for tZQCS in the memory datasheet => tZQCS = 180 ns
MPDDRC->MPDDRC_LPDDR2_TIM_CAL = MPDDRC_LPDDR2_TIM_CAL_ZQCS(psst_ddr2.t_tZQCS);
## Revision History

<table>
<thead>
<tr>
<th>Doc. Rev.</th>
<th>Date</th>
<th>Changes</th>
</tr>
</thead>
</table>
| 11172B   | 26-Feb-15 | Reformatted document  
General editorial changes throughout  
Revised content of “Scope” (now includes mention of SAMA5D36 device)  
Added “Reference Documents”  
**Section 1. “Multi-port DDR Controller Overview”: replaced “The MPDDRC supports a CAS latency of 2, 3, 4, 5 or 6” with “The MPDDRC supports a CAS latency of 2 and 3”**  
**Table 2-1 “SDRAM Controller Signals”: reorganized content; removed “Frequency” column**  
Renamed Section 4. “Layout and Design Constraints” (was “SDRAM Signal Routing Considerations”) and revised content  
Revised Section 8.1 “DDR2-SDRAM Initialization”  
Revised Section 8.2 “LPDDR2-SDRAM Initialization”  
**Section 8.3 “Micron® MT47H128M16 DDR2 SDRAM (MPDDRC Configuration Example)”:**  
- in first sentence, corrected “32 Mbit × 16 × 8 banks” to “16 Mbit × 16 × 8 banks”  
- replaced all instances of “132 MHz“ or “133 MHz” with “134 MHz”  
- changed processor clock speed from “528 MHz“ to “536 MHz”  
- changed PLL frequency from “1056 MHz” to “1072 MHz”  
**Table 8-1 “MPDDRC Configuration Example with Micron MT47H128M16”:**  
- reformatted table and reorganized content  
- updated register names  
- deleted “EBI Chip Select Assignment” rows  
- added “Drive Strength (DDR2 only)” to MPDDRC_CR parameters  
- changed “7 µs” to “7.8 µs” as timer count period for MPDDRC_RTR  
**Appendix A. “DDR2 Initialization Code Example”: added title and updated introductory sentence; in Initialization Step 3, changed instance of “528 MHz” to “536 MHz”; updated Initialization Step 17 to include DDR2 weak drive strength**  
Added Appendix B. “LPDDR2 Initialization Code Example” |
| 11172A   | 19-Oct-2012 | First issue                  |