

Implementation of DDR2 on AT91SAM9G45 Devices

1. Scope

The AT91SAM9G45 microprocessor features:

- One multi-port DDR2 controller that supports 16-bit DDR2 or 16-bit LP-DDR memories only
- One single-port DDR2 controller that supports 16-bit DDR2, 16-bit LP-DDR, 16- or 32-bit SDR or LP-SDR memories through the EBI

The purpose of this document is to help the developer in the design of a system utilizing DDR2. Each DDR2 controller is described separately.



AT91 ARM Thumb-based Microcontrollers

Application Note

6492A-ATARM-22-Sep-09



2. Multi-Port DDR2 Controller Overview

The DDR2 controller (DDR2C) extends the memory capabilities of a chip by providing the interface to an external 16-bit DDR2 device. The page size supports ranges from 2048 to 16384, and a number of columns from 256 to 4096. It supports byte (8-bit) and half-word (16-bit) accesses.

The DDR2 controller 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 DDR2 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 DDR2 controller only supports a CAS latency of 3; it optimizes the read access according to the operating frequency.

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

The multi-port DDR2 controller I/Os are powered by VDDIOM0. For DDR2, VDDIOM0 is set to 1.8V nominal.

3. Multi-Port DDR2 Controller Signals Definition

The DDR2 controller is capable of managing 4-bank DDR2 devices. The signals generated by the controller are defined below.

Table 3-1. DDR2 Controller Signals

DDR_D0 - DDR_D15	Data Bus	I/O		VDDIOM0	Pulled-up input at reset
DDR_A0 - DDR_A13	Address Bus	Output		VDDIOM0	0 at reset
DDR_CLK - #DDR_CLK	DDR2 Differential Clock Input	Output		VDDIOM0	
DDR_CKE	DDR2 Clock Enable	Output	High	VDDIOM0	
DDR_CS	DDR2 Chip Select	Output	Low	VDDIOM0	
DDR_WE	DDR2 Write Enable	Output	Low	VDDIOM0	
DDR_RAS - DDR_CAS	Row and Column Signal	Output	Low	VDDIOM0	
DDR_DQM[0..1]	Write Data Mask	Output		VDDIOM0	
DDR_DQS[0..1]	Data Strobe	Output		VDDIOM0	
DDR_BA0 - DDR_BA1	Bank Select	Output		VDDIOM0	
DDR_VREF	Reference Voltage	Input		VDDIOM0	

Where:

DDR_D0 - DDR_D15 are DDR2 controller data lines, respectively bounded to [DDR_D15:DDR_D0] on the microcontroller.

DDR_A0 - DDR_A13 are DDR2 controller address lines, respectively bounded to [A0:A13] on the microcontroller.

DDR_CLK - #DDR_CLK are the differential clock signals that feed the DDR2 device. All other signals take those two signals as a reference.

To reach the 133 MHz speed on these pins loaded with a 10 pF equivalent capacitor, a dedicated high speed pin is necessary; it cannot be multiplexed on a PIO line (lower frequency).

DDR_CKE 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).

DDR_CS: 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.

DDR_RAS - DDR_CAS, DDR_WE: 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.

DDR_DQM[0..1]: Data is accessed in 8 or 16 bits by means of DDR_DQM[1..0], which are respectively the highest to lowest mask bit for the DDR2 data on the bus.

DDR_DQS[0..1]: Data strobe. The data is sampled on DDR_DQS edges.

DDR_BA0 - DDR_BA1 select the bank to address when a command is input. Read/write or pre-charge is applied to the bank selected by DDR_BA0 and DDR_BA1.

DDR_VREF is 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 1K Ω , 1% tolerance resistors.

4. EBI DDR2 Controller Overview

The EBI embeds a single-port DDR2 controller (DDR2SDRC) that extends the memory capabilities of a chip by providing the interface to 16-bit DDR2, 16-bit LP-DDR, 16-bit or 32-bit SDR or LD-SDR external devices. The page size supports ranges from 2048 to 16384, and a number of columns from 256 to 4096. It supports byte (8-bit), half-word (16-bit) and word (32-bit) accesses.

The DDR2 controller supports a read or write burst length of one location. It keeps track of the active row in each bank, thus maximizing DDR2 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 DDR2 controller supports a CAS latency of 3 for DDR2, and 2 or 3 for SDR. It permits to optimize the read access according to the frequency.

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

The DDR2 controller I/Os are powered by VDDIOM1. For DDR2, VDDIOM1 is to be set to 1.8V nominal.

5. EBI DDR2 Controller Signals Definition

The DDR2 controller is capable of managing four-bank DDR2 devices. The signals generated by the controller are defined below (refer to the EBI Section on the product Datasheet).

Table 5-1. EBI Controller Signals

D0 - D31	Data Bus	I/O		VDDIOM1	Pulled-up input at reset
A0 - A25	Address Bus	Output		VDDIOM1	0 at reset
SDCK - #SDCK	DDR2/SDRAM Differential Clock	Output		VDDIOM1	
SDCKE	DDR2/SDRAM Clock Enable	Output	High	VDDIOM1	
SDCS	DDR2/SDRAM Controller Chip Select	Output	Low	VDDIOM1	
BA0 - BA1	Bank Select	Output		VDDIOM1	
SDWE	DDR2/SDRAM Write Enable	Output	Low	VDDIOM1	
RAS - CAS	Row and Column Signal	Output	Low	VDDIOM1	
SDA10	SDRAM Address 10 Line	Output		VDDIOM1	
DQS[0..1]	Data Strobe	Output		VDDIOM1	
DQM[0..3]	Write Data Mask	Output		VDDIOM1	

Where:

D0 - D31 are DDR2 controller data lines, respectively bounded to [D31:D0] on the microcontroller.

A0 - A12 are DDR/SDR controller address lines, respectively bounded to [A2:A14] of the microcontroller except for A10 (SDA10) which is not bounded to A12.

A[12:0] lines allow to address up to 11 columns and 13 rows. A[25:13] lines are not used.

SDCK - #SDCK are the Differential Clock signals that feed the DDR2 device. All other signals take those two signals as a reference. All SDRAM input signals are sampled on the positive edge of SDCK.

SDCKE acts as an inhibit signal to the SDRAM device. SDCKE remains high during a valid SDRAM 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 SDRAM controller self-refresh mode).

SDCS: When the chip select SDCS is low, the command input is valid. When it is high, the commands are ignored but the operation continues.

BA0 - BA1 selects the bank to address when a command is input. Read/write or precharge is applied to the bank selected by BA0 and BA1.

RAS - CAS, SDWE: The row address strobe (RAS), column address strobe (CAS) will assert to indicate that the corresponding address is present on the bus. The conjunction with write Enable (SDWE) and chip select (SDCS) at the rising edge of the clock (SDCK) determines the SDRAM operation.

SDA10 acts as a DDR/SDR address line but is also used as the auto-precharge command bit. An AT91 product outputs a dedicated SDA10 signal.

DQS[0..1]: Data strobe. The data is sampled on DDR_DQS edges.

DQM[0..3]: Data is accessed in 8 or 16 bit by means of DDR_DQM[1:0] which are respectively the highest to lowest mask bit for the DDR2 data on the bus. DQM[3:2] are used in case of an SDR implementation.

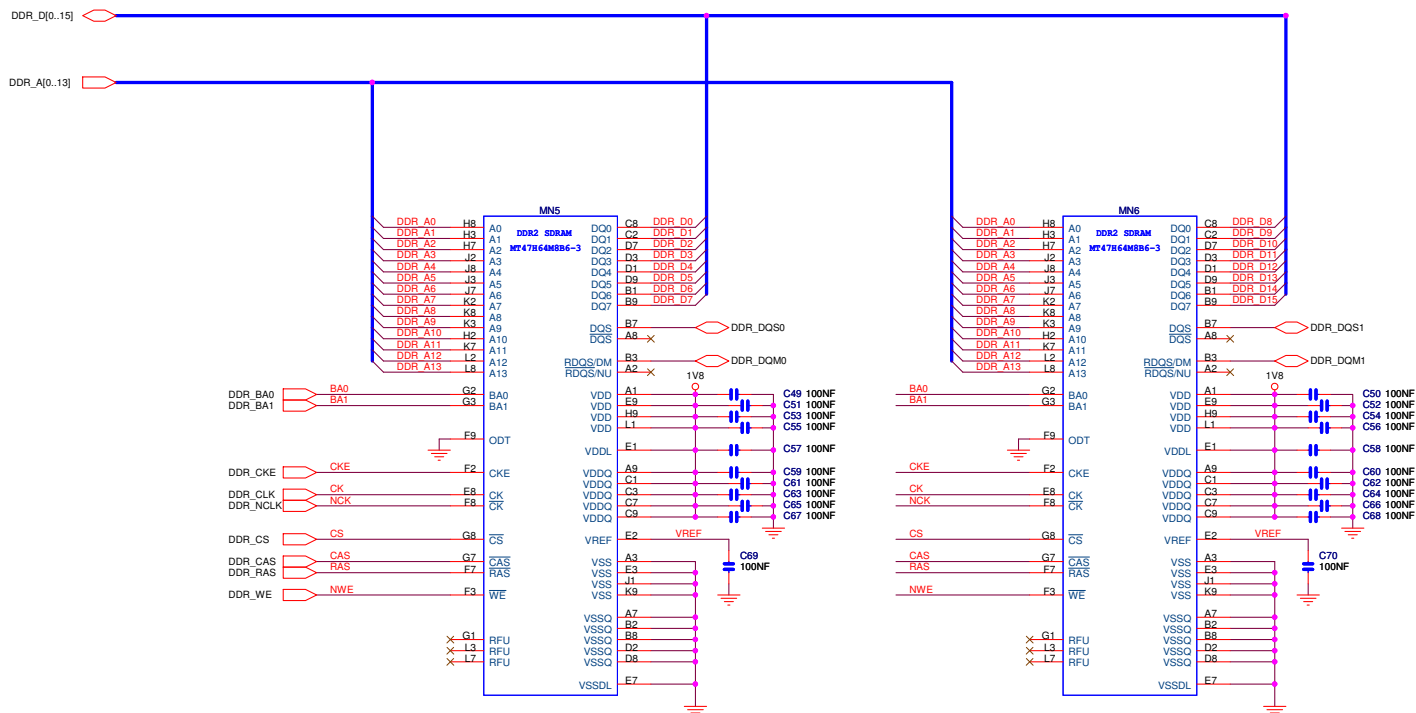
6. DDR2 Connexion on AT91SAM9 Multi-Port Controller

The AT91SAM9G45 microprocessor supports 16-bit DDR2 devices on DDR/LPDDR Chip Select area (0x70000000 memory zone). The user interface to configure the DDR2 controller is mapped at address 0xFFFF E600.

Each DDR2 device must use sufficient decoupling to provide an efficient filtering on the power supply rails.

6.1 16-bit Using 2 x 8-bit DDR2 Implementation

6.1.1 Hardware Configuration



6.1.2 Software Configuration

The DDR2 initialization sequence is described in [Section 11.1 “DDR2-SDRAM Initialization”](#) on [page 13](#).

7. DDR2 Connexion on AT91SAM9 EBI Controller

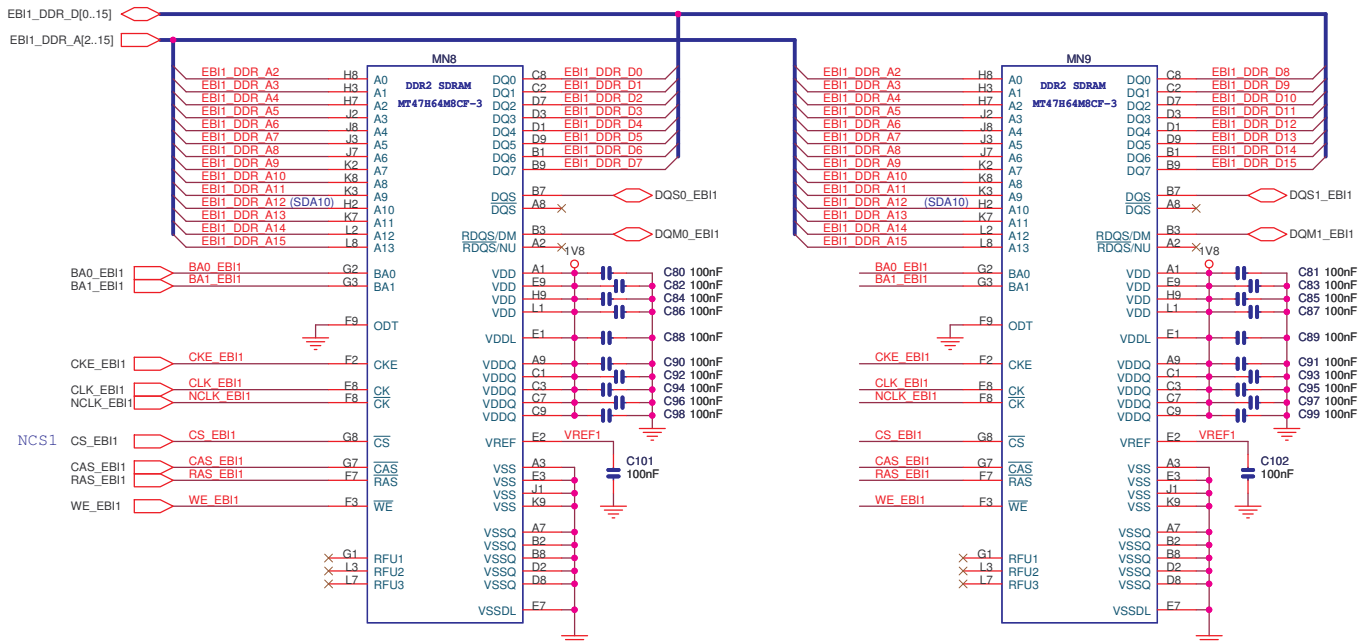
The AT91SAM9G45 microcontroller supports 16-bit DDR2 devices on one Chip Select area (NCS1). The user interface to configure the DDR2 controller is mapped at address 0xFFFF E400.

The 32-bit interface can be achieved by a single 32-bit SDRAM device or two 16-bit SDRAM devices.

Each DDR2 device must use sufficient decoupling to provide an efficient filtering on the power supply rails.

7.1 16-bit Using 2 x 8-bit DDR2 Implementation

7.1.1 Hardware Configuration



7.1.2 Software Configuration

The following configuration has to be performed:

- Assign the EBI CS1 to the DDR2 controller by setting the bit EBI_CS1A in the EBI Chip Select Assignment Register located in the bus matrix memory space
- Initialize the DDR2 controller accordingly to the DDR2 device and the system bus frequency.

The Data Bus Width is to be programmed to 16 bits.

The DDR2 initialization sequence is described in [Section 11.1 "DDR2-SDRAM Initialization"](#) on page 13.

8. DDR2 Signal Routing Considerations

The critical high speed signal is associated with the DDR2. The following are general guidelines for designing a DDR2 interface with AT91SAM9 products with a targeted speed of 133 MHz on SDCK/#SDCK:

- At first, position the DDR2 devices as close to the processor as possible. A longer trace will increase the rise time and the fall time of the signals. The setup time of signals generated by the AT91 Microcontroller will decrease with an increased trace length.
- Keep the DDR2 clocks and control lines as short as possible.
- Keep the DDR2 address and data lines as short as possible.
- For a proper DDR2 operation at 133 MHz, a bus impedance adaptation is necessary. 10 to 30 Ohms series resistors can be placed on all the switching signals to limit the current flow into each outputs. The resistor is to be located near the processor. The need for series termination resistors and their specific value on the signals are better determined by simulation, using IBIS models and the specific design PCB layout. On SAM9G45-EKES and SAM9M10G45-EK, the adaptation is achieved with a 27-Ohm serial resistor.
- To support maximum speeds, reasonable DDR2 loading constraints must be followed. For high-speed operation, the maximum load cannot exceed 30 pF on address and data buses, and 10 pF on SDCK and #SDCK. The user must consider all the devices connected on the different buses to calculate the system load.
- Use sufficient decoupling scheme for memory devices. It is recommended to use low ESR 0.01 μ F and 0.1 μ F decoupling capacitors in parallel. An additional 0.001 μ F decoupling capacitor is recommended to minimize ground bounce and to filter high frequency noise.

9. DDR2 Electromagnetic Compatibility Improvement

9.1 Simultaneous Switching

Simultaneous switching is the worst enemy of EMI at device operation level. The AT91SAM9G45 microprocessor embeds Delay Controller on High Speed signals.

These delays are applied to address A[15:0] and Data D[15:0]. They are controlled in dedicated registers in DDR2 controller, PIO controller and Static Memory controller respectively for DDR2 signal, High Speed MCI and EBI signals. Refer to the Product Datasheet for more details.

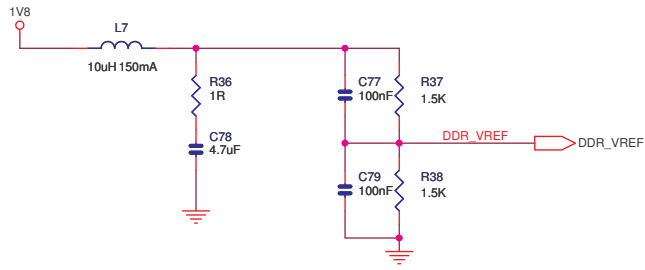
9.2 Over-Shoots

Over-shoots occur when the current driven is too high. The AT91SAM9G45 microprocessor embeds drive control on memory signals. Refer to the Product Datasheet for more details.

10. DDR2 VREF Signal Considerations

DDR_VREF is used by the input buffers of the DDR2 memories and the DDR2 controller to determine logic levels. VREF is specified to be 0.9V ($\frac{1}{2}$ the power supply voltage) and is created using a voltage divider constructed from two 1.5 kOhm, 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.



11. DDR2 Controller Configuration

11.1 DDR2-SDRAM Initialization

The initialization sequence is generated by software. The DDR2-SDRAM devices are initialized by the following sequence:

(For a register description, see DDR2-SDRAM Initialization section on the AT91SAM9G45 Datasheet)

(For an example of initialization, see the “Appendix”)

1. Program the memory device type into the Memory Device Register.
2. Program the DDR2-SDRAM device features 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).
3. A NOP command is issued to the DDR2-SDRAM. Program the NOP command into the Mode Register. The application must set Mode to 1 in the Mode Register and perform a write access to any DDR2-SDRAM address to acknowledge this command. Now clocks which drive DDR2-SDRAM device are enabled.

A minimum pause of 200 μ s is provided to precede any signal toggle.

4. An NOP command is issued to the DDR2-SDRAM. Program the NOP command into the Mode Register. The application must set Mode to 1 in the Mode Register and perform a write access to any DDR2-SDRAM address to acknowledge this command. Now CKE is driven high.
5. An all banks precharge command is issued to the DDR2-SDRAM. Program all banks precharge command into the Mode Register. The application must set Mode to 2 in the Mode Register and perform a write access to any DDR2-SDRAM address to acknowledge this command.
6. An Extended Mode Register set (EMRS2) cycle is issued to choose between commercial or high temperature operations. The application must set Mode to 5 in the Mode Register and perform a write access to the DDR2-SDRAM to acknowledge this command. The write address must be chosen so that BA[1] is set to 1 and BA[0] is set to 0. For example, with a 16-bit 128-MB DDR2-SDRAM (12 rows, 9 columns, 4 banks) bank address, the DDR2-SDRAM write access should be done at address 0x20800000*.
* This address is for example purposes only. The real address depends on the implementation in the product.
7. An Extended Mode Register set (EMRS3) cycle is issued to set all registers to “0”. The application must set Mode to 5 in the Mode Register and perform a write access to the DDR2-SDRAM to acknowledge this command. The write address must be chosen so that BA[1] is set to 1 and BA[0] is set to 1. For example, with a 16-bit 128-MB DDR2-SDRAM (12 rows, 9 columns, 4 banks) bank address, the DDR2-SDRAM write access should be done at address 0x20C00000.
8. An Extended Mode Register set (EMRS1) cycle is issued to enable DLL. The application must set Mode to 5 in the Mode Register and perform a write access to the DDR2-SDRAM to acknowledge this command. The write address must be chosen so that BA[1] and BA[0] are set to 0. For example, with a 16-bit 128-MB DDR2-SDRAM (12 rows, 9 columns, 4 banks) bank address, the DDR2-SDRAM write access should be done at address 0x20800000 or 0x20400000.

An additional 200 cycles of clock are required for locking DLL.

9. Program DLL field into the Configuration Register to high (Enable DLL reset).

10. A Mode Register set (MRS) cycle is issued to reset DLL. The application must set Mode to 3 in the Mode Register and perform a write access to the DDR2-SDRAM to acknowledge this command. The write address must be chosen so that BA[1:0] bits are set to 0. For example, with a 16-bit 128-MB DDR2-SDRAM (12 rows, 9 columns, 4 banks) bank address, the SDRAM write access should be done at address 0x20000000.
11. An all banks precharge command is issued to the DDR2-SDRAM. Program all banks precharge command into the Mode Register. The application must set Mode to 2 in the Mode Register and perform a write access to any DDR2-SDRAM address to acknowledge this command.
12. Two auto-refresh (CBR) cycles are provided. Program the auto refresh command (CBR) into the Mode Register. The application must set Mode to 4 in the Mode Register and perform a write access to any DDR2-SDRAM location twice, to acknowledge these commands.
13. Program DLL field into the Configuration Register to low (Disable DLL reset).
14. A Mode Register Set (MRS) cycle is issued to program the parameters of the DDR2-SDRAM devices, in particular CAS latency, burst length, and to disable DLL reset. The application must set Mode to 3 in the Mode Register and perform a write access to the DDR2-SDRAM to acknowledge this command. The write address must be chosen so that BA[1:0] are set to 0. For example, with a 16-bit 128-MB SDRAM (12 rows, 9 columns, 4 banks) bank address, the SDRAM write access should be done at address 0x20000000.
15. Program OCD field into the Configuration Register to high (OCD calibration default).
16. An Extended Mode Register set (EMRS1) cycle is issued to OCD default value. The application must set Mode to 5 in the Mode Register and perform a write access to the DDR2-SDRAM to acknowledge this command. The write address must be chosen so that BA[1] is set to 0 and BA[0] is set to 1. For example, with a 16-bit 128 MB DDR2-SDRAM (12 rows, 9 columns, 4 banks) bank address, the DDR2-SDRAM write access should be done at address 0x20400000.
17. Program OCD field into the Configuration Register to low (OCD calibration mode exit).
18. An Extended Mode Register Set (EMRS1) cycle is issued to enable OCD exit. The application must set Mode to 5 in the Mode Register and perform a write access to the DDR2-SDRAM to acknowledge this command. The write address must be chosen so that BA[1] is set to 1 and BA[0] is set to 1. For example, with a 16-bit 128 MB DDR2-SDRAM (12 rows, 9 columns, 4 banks) bank address, the DDR2-SDRAM write access should be done at address or 0x20400000.
19. A Normal mode command is provided. Program the Normal mode into Mode Register. Perform a write access to any DDR2-SDRAM address to acknowledge this command.
20. Perform a write access to any DDR2-SDRAM address.
21. Write the refresh rate into the count field in the Refresh Timer register (see page 32). (Refresh rate = delay between refresh cycles). The DDR2-SDRAM device requires a refresh every 15.625 μ s or 7.81 μ s. With a 100 MHz frequency, the refresh count will be either $(15.625 / 100 \text{ MHz}) = 1562$, i.e. 0x061A, or $(7.81 / 100 \text{ MHz}) = 781$, i.e. 0x030d.

After initialization, the DDR2-SDRAM devices are fully functional.

11.2 Micron MT47H64M8

The Micron MT47H64M8 are 64 MB devices arranged as 16 Mbit x 8 x 4 banks with a CAS latency of 3 at 133 MHz. These devices are featured on the AT91SAM9G45-EKES and AT91SAM9M10G45-EK evaluation kits.

The following table gives the delay in ns extracted from the DDR2-SDRAM datasheet, the corresponding number of cycles at 133 MHz, and the field to program these values accordingly.

Description			Register/Field	Value
System				
PLL Frequency		800 MHz	PMC_PLLAR	0x20c73f03
Processor / Bus Clock		400 / 133 MHz	PMC_MCKR	0x00001302
System Clock		DDR clock enable	PMC_SCER	0x00000005
EBI Chip Select Assignment		DDR/SDR	EBI_CSA	
			EBI_CS1A	0x2
DDR2 Device	Value in Datasheet	Value in Datasheet	Register/Field	Value
DDRSDR Configuration Register			DDRSDR_CR	0x3d
Number of Columns	10	10	NC	0x1
Number of Rows	14	14	NR	0x3
CAS Latency	3	3 cycles	CAS	0x3
Reset DLL		Disable	DLL	0x0
Output Driver Impedance control		Normal	DIC/DS	0x0
Disable DLL		No	DIS_DLL	0x0
Off-Chip Driver		(1)	OCD	0x0
Mask Data is shared		not shared	DMQS	0x0
Enable Read Measure		disabled	ENRDM	0x0
	Delay in Datasheet (ns)	Number of Cycles at 133 MHz	Register/Field	Value
DDRSDR Timing 0 Register			DDRSDR_T0PR	0x21128226
ACTIVATE to PRECHARGE time	45	6 cycles	TRAS	0x6
ACTIVATE to READ/WRITE time	15	2 cycles	TRCD	0x2
Last DATA-IN to PRECHARGE time	15	2 cycles	TWR	0x2
REFRESH to ACTIVATE time	55	8 cycles	TRC	0x8
PRECHARGE to ACTIVATE time	15	2 cycles	TRP	0x2
ACTIVE bankA to ACTIVE BankB	7.5	1 cycle	TRRD	0x1
Internal Write to Read Delay	7.5	1 cycle	TWTR	0x1
Load Mode Register Command to ACTIVE or REFRESH Command	2 cycles	2 cycles	TMRD	0x2
DDRSDR Timing 1 Register			DDRSDR_T1PR	0x02c8100e
Row Cycle delay	105	14 cycles	TRFC	0xe
Exit Self Refresh Delay to Non-Read Command	TRFC+10	16 cycles	TXSNR	0x10

Description			Register/Field	Value
Exit Self Refresh Delay to Read Command	200 cycles	200 cycles	TXSRD	0xc8
Exit Power-down Delay to First Command	2 cycles	2 cycles	TXP	0x2
DDRSDBC Timing 2 Register			DDRSDBC_T2PR	0x00000107 2
Exit Active Power Down Delay to Read Command (Fast Exit)	2 cycles	2 cycles	TXARD	0x2
Exit Active Power Down Delay to Read Command (Slow Exit)		7 cycles	TXARDS	0x7
Row Precharge All Delay		0 cycle	TRPA	0x0
Read to Precharge	7.5	1 cycle	TRTP	0x1
DDRSDBC Memory Device Register			DDRSDBC_MD	0x00000016
Memory Device		DDR2-SDRAM	MD	0x6
Data Bus Width		16 bits	DBW	0x1
SDRAM Refresh Timer Register - Timer Count		7μs	SDRAMC_TR	0x410

Note: 1. OCD is not supported, but it is a mandatory step in the DDR2 initialization phase.

12. Appendix

Here is an example of the DDR2 initialization code, associated to the different steps introduced in [Section 11.1 “DDR2-SDRAM Initialization”](#) on page 13:

```

/*-----
/* \fn    ddram_init
/* \brief Initialization of the DDR Controller
/*-----
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, HDDRSDRC2_MDR, ddram_config->ddramc_mdr);
    // Program the DDR Controller
    write_ddramc(ddram_controller_address, HDDRSDRC2_CR, ddram_config->ddramc_cr);

    // Initialization Step 2: assume timings for 7.5 ns min clock period
    write_ddramc(ddram_controller_address, HDDRSDRC2_T0PR, ddram_config->ddramc_t0pr);
    // pSDDRC->HDDRSDRC2_T1PR
    write_ddramc(ddram_controller_address, HDDRSDRC2_T1PR, ddram_config->ddramc_t1pr);
    // pSDDRC->HDDRSDRC2_T2PR
    write_ddramc(ddram_controller_address, HDDRSDRC2_T2PR, ddram_config->ddramc_t2pr);

    // Initialization Step 3: NOP command -> allow to enable clk
    write_ddramc(ddram_controller_address, HDDRSDRC2_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 396 MHz:
    // min 13,200 loops)
    for (i = 0; i < 13300; i++) {
        asm("    nop");
    }

    // Initialization Step 4: An NOP command is issued to the DDR2-SDRAM
    // NOP command -> allow to enable cke
    write_ddramc(ddram_controller_address, HDDRSDRC2_MR, AT91C_DDRC2_MODE_NOP_CMD);
    *((unsigned volatile int*) ddram_address) = 0;
    // wait 400 ns min
    for (i = 0; i < 100; i++) {
        asm("    nop");
    }
}

```

```
// Initialization Step 5: Set All Bank Precharge
write_ddramc(ddram_controller_address, HDDRSDRC2_MR, AT91C_DDRC2_MODE_PRCGALL_CMD);
*((unsigned volatile int*) ddram_address) = 0;
// wait 400 ns min
for (i = 0; i < 100; i++) {
    asm("    nop");
}

// Initialization Step 6: Set EMR operation (EMRS2)
write_ddramc(ddram_controller_address, HDDRSDRC2_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, HDDRSDRC2_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, HDDRSDRC2_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, HDDRSDRC2_CR);
write_ddramc(ddram_controller_address, HDDRSDRC2_CR, cr | AT91C_DDRC2_DLL_RESET_ENABLED);

// Initialization Step 10: reset DLL
write_ddramc(ddram_controller_address, HDDRSDRC2_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
write_ddramc(ddram_controller_address, HDDRSDRC2_MR, AT91C_DDRC2_MODE_PRCGALL_CMD);
*((unsigned volatile int*) ddram_address) = 0;
// wait 400 ns min
for (i = 0; i < 100; i++) {
    asm("    nop");
}

// Initialization Step 12: Two auto-refresh (CBR) cycles are provided. Program the auto refresh
// command (CBR) into the Mode Register.
write_ddramc(ddram_controller_address, HDDRSDRC2_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, HDDRSDRC2_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).
cr = read_ddramc(ddram_controller_address, HDDRSDRC2_CR);
write_ddramc(ddram_controller_address, HDDRSDRC2_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.
write_ddramc(ddram_controller_address, HDDRSDRC2_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).
cr = read_ddramc(ddram_controller_address, HDDRSDRC2_CR);
write_ddramc(ddram_controller_address, HDDRSDRC2_CR, cr | AT91C_DDRC2_OCD_DEFAULT);

// Initialization Step 16: An Extended Mode Register set (EMRS1) cycle is issued to OCD default
// value.
write_ddramc(ddram_controller_address, HDDRSDRC2_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).
cr = read_ddramc(ddram_controller_address, HDDRSDRC2_CR);
write_ddramc(ddram_controller_address, HDDRSDRC2_CR, cr & (~AT91C_DDRC2_OCD_EXIT));

// Initialization Step 18: An Extended Mode Register set (EMRS1) cycle is issued to enable OCD
// exit.
write_ddramc(ddram_controller_address, HDDRSDRC2_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 19, 20: A mode Normal command is provided. Program the Normal mode into
// Mode Register.
write_ddramc(ddram_controller_address, HDDRSDRC2_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. The DDR2-SDRAM device requires a refresh every 15.625  $\mu$ s or 7.81  $\mu$ s. With a
// 100MHz frequency, the refresh timer count register must to be set with (15.625 x 100 MHz) =
// 1562 i.e. 0x061A or (7.81 x 100MHz) = 781 i.e. 0x030d.
// Set Refresh timer
write_ddramc(ddram_controller_address, HDDRSDRC2_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;
}

```

Revision History

Doc. Rev	Comments	Change Request Ref.
6492A	First issue	



Headquarters

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

International

Atmel Asia
Unit 1-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
Hong Kong
Tel: (852) 2245-6100
Fax: (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-
Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Product Contact

Web Site
www.atmel.com
www.atmel.com/AT91SAM

Technical Support
[AT91SAM Support](#)
[Atmel technical support](#)

Sales Contacts
www.atmel.com/contacts/

Literature Requests
www.atmel.com/literature

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. **EXCEPT AS SET FORTH IN ATMEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL'S WEB SITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

© 2009 Atmel Corporation. All rights reserved. Atmel®, Atmel logo and combinations thereof, and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.