Initializing the DM814x for DDR3/DDR2/LPDDR1 Memory Access

Jeff Cobb *Texas Instruments*

ABSTRACT

This document describes the procedure to initialize the DM814x device for accessing external DDR3/DDR2/LPDDR1 memories. The files attached to this report are required to determine the parameters that will configure and optimize performance of the DDR memory interface.

Contents

Figures

1 Configuring the DDR3/DDR2/LPDDR1 Controller for DDR3 Operation

The steps below describe how to configure the DDR3/DDR2/LPDDR1 Controller for DDR3 operation. Optimizing the timing configuration for DDR3 is a multiple step process that is specific to the board design.

The procedure described below relies on several spreadsheets and other files, which are available in the zip file associated with this application report.

1.1 Determining Initial Control Register Values for DDR3 Operation

The first step in the process is to determine the proper EMIF4 register settings for DDR3 operation at the desired speed. For the illustration purposes, this document will use the Micron MT41J64M16-125 DDR3 SDRAM as an example.

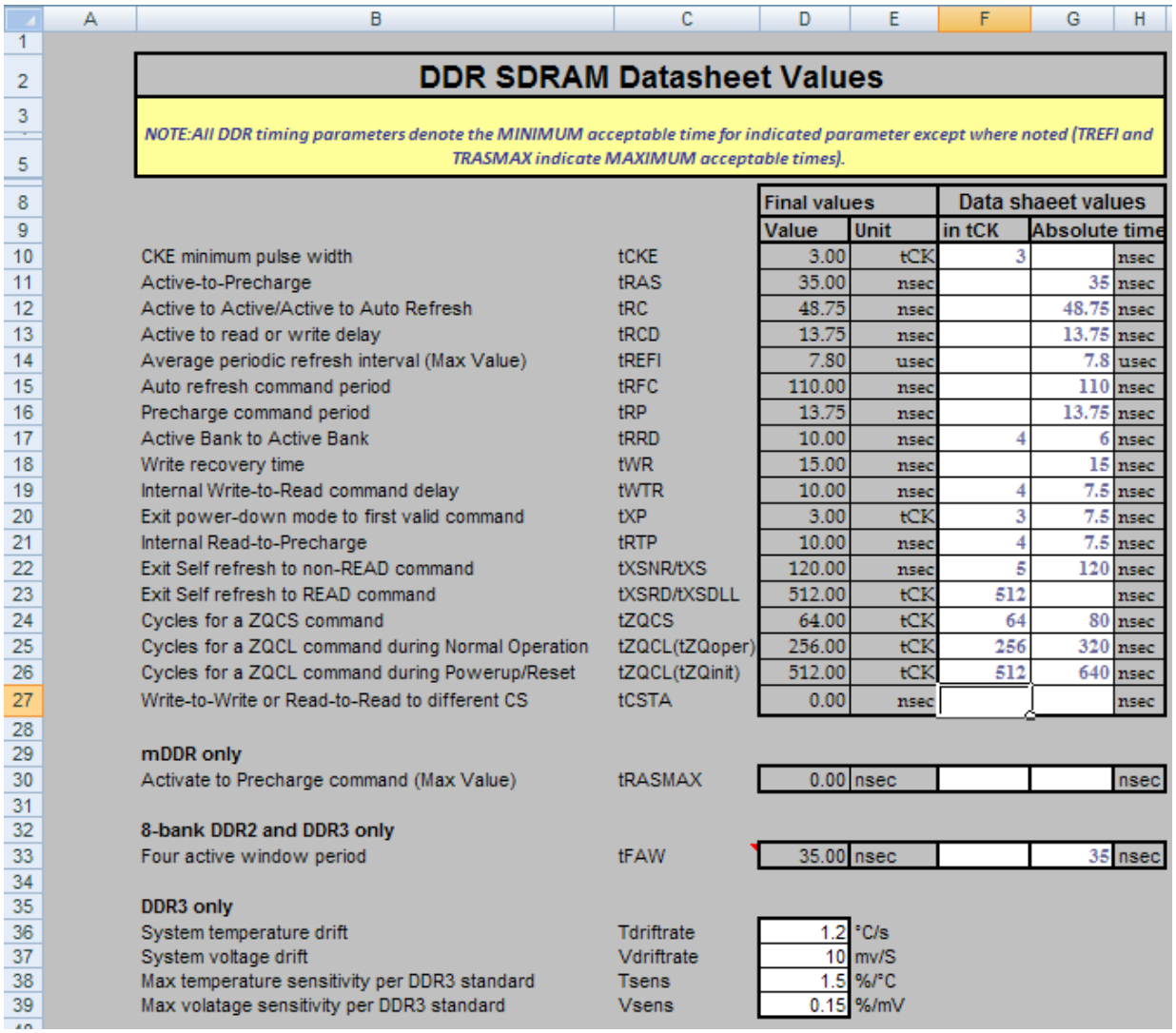
To determine the correct register settings for the EMIF4, the ***TI814x_EMIF4_Register_settings.xls*** spreadsheet is used.

1.1.1 Configuring the DDR3 Timing Values

The DDR3 timing values are strictly dependent on the datasheet values for the memory used. Use the following steps to calculate the register settings for the DDR3 timings.

1) Open the ***TI814x_EMIF4_Register_settings.xls*** spreadsheet and click on the DRAM tab.

2) On the left side of the worksheet, the timings from the DDR3 memory datasheet are entered in the white cells. Data may only be entered in the white cells. On the right side of the worksheet is a DDR3 example configuration showing which parameters should be entered from the DDR3 datasheet. Enter the timing parameters from the DDR3 datasheet. No input is required in the section marked “mDDR only”. Timings defined in units of CK are entered under the “in tCK” column. Timings in units of ns are entered under the “Absolute time” column.



DDR SDRAM Datasheet Values

NOTE:All DDR timing parameters denote the MINIMUM acceptable time for indicated parameter except where noted (TREFI and TRASMAX indicate MAXIMUM acceptable times).

		Final values		Data shaeet values		
		Value	Unit	in tCK	Absolute time	
CKE minimum pulse width	tCKE	3.00	tCK	3		nsec
Active-to-Precharge	tRAS	35.00	nsec		35	nsec
Active to Active/Active to Auto Refresh	tRC	48.75	nsec		48.75	nsec
Active to read or write delay	tRCD	13.75	nsec		13.75	nsec
Average periodic refresh interval (Max Value)	tREFI	7.80	usec		7.8	usec
Auto refresh command period	tRFC	110.00	nsec		110	nsec
Precharge command period	tRP	13.75	nsec		13.75	nsec
Active Bank to Active Bank	tRRD	10.00	nsec	4	6	nsec
Write recovery time	tWR	15.00	nsec		15	nsec
Internal Write-to-Read command delay	tWTR	10.00	nsec	4	7.5	nsec
Exit power-down mode to first valid command	tXP	3.00	tCK	3	7.5	nsec
Internal Read-to-Precharge	tRTP	10.00	nsec	4	7.5	nsec
Exit Self refresh to non-READ command	tXSNR/tXS	120.00	nsec	5	120	nsec
Exit Self refresh to READ command	tXSRD/tXSDLL	512.00	tCK	512		nsec
Cycles for a ZQCS command	tZQCS	64.00	tCK	64	80	nsec
Cycles for a ZQCL command during Normal Operation	tZQCL(tZQoper)	256.00	tCK	256	320	nsec
Cycles for a ZQCL command during Powerup/Reset	tZQCL(tZQinit)	512.00	tCK	512	640	nsec
Write-to-Write or Read-to-Read to different CS	tCSTA	0.00	nsec			nsec
mDDR only						
Activate to Precharge command (Max Value)	tRASMAX	0.00	nsec			nsec
8-bank DDR2 and DDR3 only						
Four active window period	tFAW	35.00	nsec		35	nsec
DDR3 only						
System temperature drift	Tdriftrate	1.2	°C/s			
System voltage drift	Vdriftrate	10	mv/S			
Max temperature sensitivity per DDR3 standard	Tsens	1.5	%/°C			
Max volatage sensitivity per DDR3 standard	Vsens	0.15	%/mV			

Table 1. EMIF4 Register Settings.xls DRAM Worksheet

3) Once the timings have been entered in the DRAM worksheet, click on the EMIF4 worksheet. The top of this worksheet shows the register values to program in the DDR control registers based on the timing entered in the DRAM worksheet and speed and mode selections made in this worksheet.

4) In the Memory Configuration section, select **DDR3** as the memory type.

5) In the Memory Configuration section, enter the desired DDR Frequency. For illustration 400 MHz will be used.

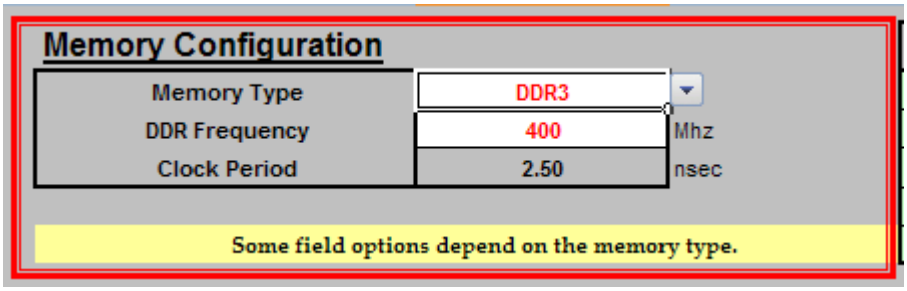


Table 2. EMIF4 Register Settings.xls EMIF4 Worksheet – Memory Configuration

1.1.2 Selecting the DDR3 Configuration Register Values

While the timing values entered in the previous section will automatically populate the **SDTIMR1/2/3** registers, the remaining DDR controller configurations registers must be selected manually by setting the white fields in the EMIF4 worksheet.

Currently each DDR controller on the DM814x supports up to 8Gb of DDR3 with the restrictions shown in Table 3 below.

Data Width	Devices (max)	Chip Selects (max)	Address Bits (max)
8-bit	4	1	15
16-bit	2	1	15

Table 3. Allowed DDR3 Configurations for each EMIF

Based on these allowed configurations and the layout restrictions specified in the device datasheet, many of the parameters must always be programmed to a certain value. Others depend on the memory device itself or optional features of the DM814x.

Table 4 below shows the DDR controller registers that can be configured from the spreadsheet. Other registers not listed below do not need to be configured and can be left at their default values.

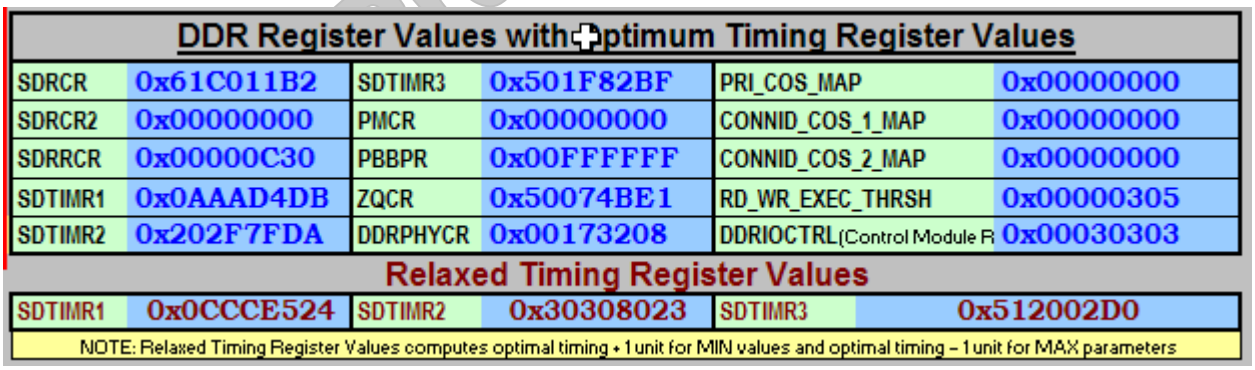
Register	Field	Value	Notes
SDRCR	**IBANKPOS**	Internal Bank Position 0	
	DDRTERM	RZQ_DIV4	
	DDQS	Differential	
	DYNODT	RZQ_DIV2	
	DLL Enable	Enable	
	DRIVE	RZQ_DIV6	
	CWL	-	See DDR Datasheet
	NM	-	Set according to how many data bits are used per DDR controller
	CL	-	See DDR Datasheet
	RSIZE	-	This field is not used since IBANKPOS=0 and EBANKPOS=0
	IBANK	-	See DDR Datasheet
	EBANK	1CS only	
	PAGESIZE		See DDR Datasheet
SDRCR2	**EBANKPOS**	Position 0	
SDRRCR	**INITREF_DIS**	Enable	
	SRT	Normal	
	ASR	Manual	
	PASR	Full Array	
PMCR	**-**	Default	* To take advantage of the power saving features, see the TRM for details on each mode.
PBBPR	**COS1_COUNT**	0xFE	
	COS2_COUNT	0xFE	
	PRI_OLD_COUNT	0xFE	
ZQCR	**ZQ_CS1EN**	ZQ Calibration Disabled	
	ZQ_CS0EN	ZQ Calibration Enabled	
	ZQ_DUALCALEN	ZQ Dual Calibration Disabled	
	ZQ_SEFXITEN	ZQCL after SR Exit	
DDRPHYCR	**EN_DYN_PWRDN**	Powered Up only during read	
	RDEYE_LVL_DIS	Disabled	
	GATE_LVL_DIS	Disabled	

	WR_LVL_DIS	Disabled	
	PHY_RST	OUT_OF_RESET	
	IDLE_LODT	100Ohms	
	RD_LODT	50Ohms	
	RD_LATENCY	CL Plus 3	
PRI_COS_MAP	-	Default	*
CONNID_COS_1_MAP	-	Default	*
CONNID_COS_2_MAP	-	Default	*
RD_WR_EXEC_THRSH	-	Default	*
DDRIOCTRL	**CS_SLEW**	Fastest_SlewRate	
	CS_IMPEDANCE	50Ohms_8mA	
	CMD_SLEW	Fastest_SlewRate	
	CMD_IMPEDANCE	50Ohms_8mA	
	DATA_SLEW	Fastest_SlewRate	
	DATA_IMPEDANCE	50Ohms_8mA	

***** These registers may all be left in their default state and do not need to be configured

Table 4. DDR Controller Register Settings

Once everything has been configured, the top of the worksheet shows a summary of the required values to be programmed in these registers. The “Optimum Timing Register Values” shows the highest performance (most aggressive) timing values. The “Relaxed Timing Register Values” add some additional margin to the timing values as described in the yellow box at the bottom. These values should be used in the GEL file and/or the code used to initialize the DDR controller.



DDR Register Values with Optimum Timing Register Values					
SDRCR	0x61C011B2	SDTIMR3	0x501F82BF	PRI_COS_MAP	0x00000000
SDRCR2	0x00000000	PMCR	0x00000000	CONNID_COS_1_MAP	0x00000000
SDRRCR	0x00000C30	PBBPR	0x00FFFFFF	CONNID_COS_2_MAP	0x00000000
SDTIMR1	0x0AAAD4DB	ZQCR	0x50074BE1	RD_WR_EXEC_THRSH	0x00000305
SDTIMR2	0x202F7FDA	DDRPHYCR	0x00173208	DDRIOCTRL(Control Module R	0x00030303

Relaxed Timing Register Values					
SDTIMR1	0x0CCCE524	SDTIMR2	0x30308023	SDTIMR3	0x512002D0

NOTE: Relaxed Timing Register Values computes optimal timing + 1 unit for MIN values and optimal timing – 1 unit for MAX parameters

Table 5. EMIF4 Register Settings.xls EMIF4 Worksheet – DDR Register Values

1.2 Performing Software (Slave Ratio) Leveling

Although the calculations above will provide an initial configuration for the memory controller, DDR3 software (slave ratio) leveling is recommended to optimize the timings in the PHY to compensate for board trace effects. The software leveling is board-specific and relies on information about board design to compute the appropriate configuration.

Software leveling computes three timing components per byte lane:

- Read DQS Slave Ratio
- Read DQS Gate Slave Ratio
- Write DQS Slave Ratio

These values are calculated by a program called ***DDR3_DDR2_slave_ratio_search_TI814x.out***. This program varies DQS delay values to determine the minimum and maximum DQS delays that still operate properly. Once these limits are found, the program produces the mid-point values for these ratios as the optimum timing. ***DDR3_DDR2_slave_ratio_search_TI814x.out*** runs under Code Composer Studio (CCS) to perform the test.

1.2.1 Using the Ratio_Seed_TI814x.xls Spreadsheet

DDR3_DDR2_slave_ratio_search_TI814x.out requires seed values to begin the test. The seed values are obtained by using the ***Ratio_Seed_TI814x.xls*** spreadsheet. The spreadsheet is shown in the figure below. The green cells allow data entry.

	A	B	C	D	E
1	**Parameters**				
2	*DDR3 clock frequency*	400	MHz		
3	*Invert Clkout*	0			
4					
5	**Trace Length (inches)**				
6		*Byte 0*	*Byte 1*	*Byte 2*	*Byte 3*
7	*CK trace*	3.5	3	2.5	2
8	*DQS trace*	2	1.8	1.6	1.4
9					
10					
11	**Seed values (per byte lane)**				
12	*WR DQS*	1B	16	10	B
13	*RD DQS*	34	34	34	34
14	*RD DQS GATE*	BE	AD	9D	8C
15					
16	**Seed Values to input to program**				
17	*WR DQS*	**13**			
18	*RD DQS*	**34**			
19	*RD DQS GATE*	**A5**			

Table 6. Ratio_Seed_TI814x.xls Spreadsheet View

The steps to obtain the seed values are:

1) Enter the desired DDR clock frequency (in cell B2)
2) Select 0 (non-inverted) for the Invert Clkout field

3) Provide the length of the DDR clock traces (average of differential pair) to each of the byte lanes in cells B7-E7. The length is entered in units of inches (1 inch = 1000 mils = 25.4 mm).

4) Provide the length of the DDR DQS traces (average of differential pair) to each of the byte lanes in cells B8-E8. The length is entered in units of inches.

5) The spreadsheet will calculate seed values per byte lane and a set of 3 hexadecimal seed values at the bottom to be entered into ***DDR3_DDR2_slave_ratio_search_TI814x.out*** when it runs, as explained in the following section.

1.2.2 Running the Software Leveling Program

Follow the steps below using Code Composer Studio version 4 or higher to load the program:

1) With the JTAG emulator connected to the board, open the target configuration corresponding to the board

2) In Debug view, right click on the Cortex A8 and click “Connect Target”

3) Load the attached GEL file “Ti814x_ddr.gel”

4) Select DDR3_EMIF0_EMIF1_400MHz_Config from the Scripts menu

5) Click View -> Registers

6) Expand CPSR

7) Select “M” and set it to 0x13

8) Go to Tools -> ARM Advanced Features and select NEON enabled

9) Select Target -> Load Program and load the **DDR3_DDR2_slave_ratio_search_TI814x.out**

10) Run the program. If there is no output, halt the processor, reset the “M” field in the CPSR to 0x13, and reload and run the program

The program will request the ratio values calculated earlier. See the figure below for an example of how the program is run.

```
Run 2:

 Enter the Seed Read DQS Gate Ratio Value in Hex to search the RD DQS Gate Window
 0x106

 Enter the Seed Read DQS Ratio Value in Hex to search the RD DQS Ratio Window
 0x23

 Enter the Seed Write DQS Ratio Value in Hex to search the Write DQS Ratio Window
 0x86

 Enter the input file Name
 seed value out.txt


*********************************************************
           The Slave Ratio Search Program Values are
*********************************************************
PARAMETER                MAX     MIN    OPTIMUM       RANGE
*********************************************************
Read DQS                 66       e         3a        58
Read DQS GATE            170      5c        e6        114
Write DQS                44       0         22        44
Write DATA               68       1c        42        4c
*********************************************************

===== END OF TEST ========
```

Table 7. Running DDR3_DDR2_slave_ratio_search_TI814x.out

The slave ratio search values are saved into the specified output file. They will be used later when initializing the DDR controller.

1.3 Initializing the Device

The steps to initialize the DDR controller and other necessary device components are outlined in the device Technical Reference Manual (TRM). The GEL file included with this document (Ti814x_ddr.gel) performs this initialization. In this section, we will go through the code in the GEL file and explain the procedures needed to initialize the device.

1.3.1 Configure the PLLs

First, the PLLs must be configured for the desired frequency. The maximum supported frequency for DDR3 at OPP 100% is 400 MHz (DDR3-800). To program the DDRPLL to output this frequency, use the following settings:

CLKINP = 20 MHz, **N** = 19, **M** = 800, **M2** = 2

These values give the following output frequency:

20 MHz / (19 + 1) * 800 / 2 = 400 MHz

Slower supported DDR frequencies can be used by lowering the value of **M**.

The following code from the GEL file initializes the DDRPLL for 400 MHz:

```
/* Set PLL to Idle mode */
WR_MEM_32(DDRPLL_CLKCTRL, RD_MEM_32(DDRPLL_CLKCTRL)|0x00800000);
while (( (RD_MEM_32(DDRPLL_STATUS)) & 0x00000101) != 0x00000101);
/* Set SELFREQDCO */
read_clkctrl = RD_MEM_32(Base_Address+CLKCTRL);
WR_MEM_32(Base_Address+CLKCTRL,(read_clkctrl & 0xffffe3ff) | CLKCTRL_VAL);
/* Reset PLL core */
WR_MEM_32(DDRPLL_CLKCTRL, RD_MEM_32(DDRPLL_CLKCTRL)& 0xfffffffe);
wait_delay(3);
/* Load M2NDIV and MN2DIV registers and latch both values */
WR_MEM_32((DDRPLL_M2NDIV),m2nval);
WR_MEM_32((DDRPLL_MN2DIV),mn2val);
wait_delay(3);
WR_MEM_32((DDRPLL_TENABLEDIV),0x1);
wait_delay(3);
WR_MEM_32((DDRPLL_TENABLEDIV),0x0);
wait_delay(3);
WR_MEM_32((DDRPLL_TENABLE),0x1);
wait_delay(3);
WR_MEM_32((DDRPLL_TENABLE),0x0);
wait_delay(3);
/* Take PLL core out of reset and set DCO clock frequency range (HS2)*/
read_clkctrl = RD_MEM_32(DDRPLL_CLKCTRL);
WR_MEM_32(DDRPLL_CLKCTRL,(read_clkctrl & 0xff7fe3ff) | CLKCTRL_VAL);
read_clkctrl = RD_MEM_32(DDRPLL_CLKCTRL);
/* Poll for the frequency and phase lock to occur */
while (( (RD_MEM_32(DDRPLL_STATUS)) & 0x00000600) != 0x00000600);
wait_delay(10);
```

1.3.2 Configure the PRCM

Next the clocks for the DMM must be enable using the PRCM as follows:

```
/*Enable DMM Clock*/

WR_MEM_32(CM_DEFAULT_DMM_CLKCTRL,       0x2);

/*Poll for Module is functional*/

while(RD_MEM_32(CM_DEFAULT_DMM_CLKCTRL)!=0x2);
```

1.3.3 Configure the Control Module Registers

The DDR IO Control registers (DDRIOCTRL) should be set according to Table 4:

```
WR_MEM_32(DDR0_IO_CTRL,0x00030303);

WR_MEM_32(DDR1_IO_CTRL,0x00030303);
```

This sets all fields to the values recommended in the TRM for DDR3 operation and should not need to be customized.

1.3.4 Perform VTP Calibration

In order to compensate for voltage, temperature, and process variations, the IO driver impedance can be configured automatically. There are two modules that must be initialized using the following sequence:

```
// Write 1 to ENABLE bit

WR_MEM_32(DDR_VTP_CTRL_0, RD_MEM_32(DDR_VTP_CTRL_0) | 0x00000040 );

WR_MEM_32(DDR_VTP_CTRL_1, RD_MEM_32(DDR_VTP_CTRL_1) | 0x00000040 );

// Write 0 to CLRZ bit

WR_MEM_32(DDR_VTP_CTRL_0, RD_MEM_32(DDR_VTP_CTRL_0) & 0xfffffffe );

WR_MEM_32(DDR_VTP_CTRL_1, RD_MEM_32(DDR_VTP_CTRL_1) & 0xfffffffe );

// Write 1 to CLRZ bit

WR_MEM_32(DDR_VTP_CTRL_0, RD_MEM_32(DDR_VTP_CTRL_0) | 0x00000001 );

WR_MEM_32(DDR_VTP_CTRL_1, RD_MEM_32(DDR_VTP_CTRL_1) | 0x00000001 );

// Read VTP control registers & check READY bits

while( (RD_MEM_32(DDR_VTP_CTRL_0) & 0x00000020) != 0x20);

while( (RD_MEM_32(DDR_VTP_CTRL_1) & 0x00000020) != 0x20);
```

The default values of 0x3 for the FILTER field and 0x0 for the LOCK field are recommended and used in the example above.

1.3.5 Configure the PHY Registers

The PHYs contain a number of macros which must all be initialized according to both the mode of operation and the board layout characteristics, determined in the previous sections.

First, initialize the Command macros as follows for each PHY:

```
WR_MEM_32(CMD0_REG_PHY0_INVERT_CLKOUT_0, 0);

WR_MEM_32(CMD1_REG_PHY0_INVERT_CLKOUT_0, 0);

WR_MEM_32(CMD2_REG_PHY0_INVERT_CLKOUT_0, 0);


WR_MEM_32(CMD0_REG_PHY0_CTRL_SLAVE_RATIO_0,(0x80));

WR_MEM_32(CMD1_REG_PHY0_CTRL_SLAVE_RATIO_0,(0x80));

WR_MEM_32(CMD2_REG_PHY0_CTRL_SLAVE_RATIO_0,(0x80));


WR_MEM_32(CMD0_REG_PHY0_DLL_LOCK_DIFF_0,0x4);

WR_MEM_32(CMD1_REG_PHY0_DLL_LOCK_DIFF_0,0x4);

WR_MEM_32(CMD2_REG_PHY0_DLL_LOCK_DIFF_0,0x4);
```

For normal operation, the non-inverted clock is passed to the DRAM. The default value of 0x80 should be used for CMD_SLAVE_RATIO. The DLL lock difference should also use its default value of 0x4.

Next, each of the Data macros must be configured as follows:

```
       WR_MEM_32((DATA0_REG_PHY0_RD_DQS_SLAVE_RATIO_0 + BaseAddrOffset), (rd_dqs_cs0
<< 10 | rd_dqs_cs0));

       WR_MEM_32((DATA0_REG_PHY0_WR_DQS_SLAVE_RATIO_0 + BaseAddrOffset), (wr_dqs_cs0
<< 10 | wr_dqs_cs0));

       WR_MEM_32((DATA0_REG_PHY0_WRLVL_INIT_RATIO_0 + BaseAddrOffset),   0);

       WR_MEM_32((DATA0_REG_PHY0_GATELVL_INIT_RATIO_0 + BaseAddrOffset), 0);

       WR_MEM_32((DATA0_REG_PHY0_RD_DQS_GATE_SLAVE_RATIO_0 +
BaseAddrOffset),(RD_DQS_GATE_cs0 << 10  | RD_DQS_GATE_cs0));

       WR_MEM_32((DATA0_REG_PHY0_WR_DATA_SLAVE_RATIO_0 + BaseAddrOffset),(wr_data_cs0
<< 10 | wr_data_cs0));
```

The values for the read and write leveling come from the software leveling program output. The Write and Read Leveling Init Ratios should all be set to 0x0.

1.3.6 Configure the DMM

The DMM LISA MAP registers should be configured according to the memory configuration used on the board. For optimal performance, it is recommended to use interleaving between the two controllers.

For example, the EVM uses a total of 8Gb of DDR3, with 4Gb attached to each controller. Therefore the DMM LISA MAP registers are set as follows:

```
/*Program the DMM to Access EMIF0 and EMIF1*/

 WR_MEM_32(DMM_LISA_MAP_0, 0x80640300);
```

```
WR_MEM_32(DMM_LISA_MAP_1, 0x80640300);

WR_MEM_32(DMM_LISA_MAP_2, 0x80640300);

WR_MEM_32(DMM_LISA_MAP_3, 0x80640300);

while(RD_MEM_32(DMM_LISA_MAP_0)!=0x80640300);

while(RD_MEM_32(DMM_LISA_MAP_1)!=0x80640300);

while(RD_MEM_32(DMM_LISA_MAP_2)!=0x80640300);

while(RD_MEM_32(DMM_LISA_MAP_3)!=0x80640300);

WR_MEM_32(DMM_PAT_VIEW_MAP_BASE, 0x80000000);
```

This configures the two controllers to interleave data every 128 bytes, from address 0x80000000 to 0xBFFFFFFF. Details on how this is configured are explained in the DMM section of the TRM.

1.3.7 Program the DDR Controller MMRs

Finally, the DDR controller registers are programmed according to the values calculated using the spreadsheet in section 1.1, with the exception of steps 2 and 4 below. The registers and corresponding shadow registers should be programmed in the following order:

1. **DDRPHYCR**, **SDRTIM1**, **SDRTIM2**, **SDRTIM3**, **SDRCR**, **SDRCR2, PBBPR**
2. **SDRRCR** (with **SDRRCR = 0x10004000**)
3. **ZQCR**
4. **SDRRCR** (with **SDRRCR = 0x00004000**)
5. **SDRRCR** (with **SDRRCR** programmed with value from spreadsheet)

The DDR memory should now be accessible.

2 Configuring the DDR3/DDR2/LPDDR1 Controller for DDR2 Operation

The steps below describe how to configure the DDR3/DDR2/LPDDR1 Controller for DDR2 operation. Optimizing the timing configuration for DDR2 is a multiple step process that is specific to the board design.

The procedure described below relies on several spreadsheets and other files, which are available in the zip file associated with this application report.

2.1 Determining Initial Control Register Values for DDR2 Operation

The first step in the process is to determine the proper EMIF4 register settings for DDR2 operation at the desired speed. For the illustration purposes, this document will use the ISSI IS43DR16640 DDR2 SDRAM as an example.

To determine the correct register settings for the EMIF4, the ***TI814x_EMIF4_Register_settings.xls*** spreadsheet is used.

2.1.1 Configuring the DDR2 Timing Values

The DDR2 timing values are strictly dependent on the datasheet values for the memory used. Use the following steps to calculate the register settings for the DDR2 timings.

1. Open the ***TI814x_EMIF4_Register_settings.xls*** spreadsheet and click on the DRAM tab.

2. On the left side of the worksheet, the timings from the DDR2 memory datasheet are entered in the white cells. Data may only be entered in the white cells. On the right side of the worksheet is a DDR2 example configuration showing which parameters should be entered from the DDR2 datasheet. Enter the timing parameters from the DDR2 datasheet. No input is required in the section marked “mDDR only”. Timings defined in units of CK are entered under the “in tCK” column. Timings in units of ns are entered under the “Absolute time” column.

DDR SDRAM Datasheet Values

NOTE:All DDR timing parameters denote the MINIMUM acceptable time for indicated parameter except where noted (TREFI and TRASMAX indicate MAXIMUM acceptable times).

		Final values		Data shaeet values		
		Value	Unit	in tCK	Absolute time	
CKE minimum pulse width	tCKE	3.00	tCK	3		nsec
Active-to-Precharge	tRAS	45.00	nsec		45	nsec
Active to Active/Active to Auto Refresh	tRC	57.50	nsec		57.5	nsec
Active to read or write delay	tRCD	12.50	nsec		12.5	nsec
Average periodic refresh interval (Max Value)	tREFI	7.80	usec		7.8	usec
Auto refresh command period	tRFC	127.50	nsec		127.5	nsec
Precharge command period	tRP	12.50	nsec		12.5	nsec
Active Bank to Active Bank	tRRD	7.50	nsec		7.5	nsec
Write recovery time	tWR	15.00	nsec		15	nsec
Internal Write-to-Read command delay	tWTR	7.50	nsec		7.5	nsec
Exit power-down mode to first valid command	tXP	2.00	tCK	2		nsec
Internal Read-to-Precharge	tRTP	7.50	nsec		7.5	nsec
Exit Self refresh to non-READ command	tXSNR/tXS	137.50	nsec		137.5	nsec
Exit Self refresh to READ command	tXSRD/tXSDLL	200.00	tCK	200		nsec
Cycles for a ZQCS command	tZQCS	0.00	tCK			nsec
Cycles for a ZQCL command during Normal Operation	tZQCL(tZQoper)	0.00	tCK			nsec
Cycles for a ZQCL command during Powerup/Reset	tZQCL(tZQinit)	0.00	tCK			nsec
Write-to-Write or Read-to-Read to different CS	tCSTA	0.00	nsec			nsec
mDDR only						
Activate to Precharge command (Max Value)	tRASMAX	0.00	nsec			nsec
8-bank DDR2 and DDR3 only						
Four active window period	tFAW	50.00	nsec		50	nsec

EMIF4 Register Settings.xls DRAM Worksheet

3. Once the timings have been entered in the DRAM worksheet, click on the EMIF4 worksheet. The top of this worksheet shows the register values to program in the DDR control registers based on the timing entered in the DRAM worksheet and speed and mode selections made in this worksheet.

4. In the Memory Configuration section, select **DDR2** as the memory type.

5. In the Memory Configuration section, enter the desired DDR Frequency. For illustration 400 MHz will be used.

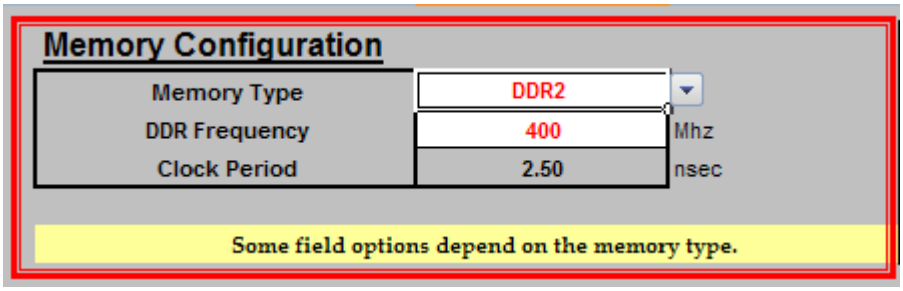


Table 8. EMIF4 Register Settings.xls EMIF4 Worksheet – Memory Configuration

2.1.2 Selecting the DDR2 Configuration Register Values

While the timing values entered in the previous section will automatically populate the **SDTIMR1/2/3** registers, the remaining DDR controller configurations registers must be selected manually by setting the white fields in the EMIF4 worksheet.

Currently each DDR controller on the DM814x supports up to 8Gb of DDR2 with the restrictions shown in Table 9 below.

Data Width	Devices (max)	Chip Selects (max)	Address Bits (max)
16-bit	2	1	15

Table 9. Allowed DDR2 Configurations for each EMIF

Based on these allowed configurations and the layout restrictions specified in the device datasheet, many of the parameters must always be programmed to a certain value. Others depend on the memory device itself or optional features of the DM814x.

Table 10 below shows the DDR controller registers that can be configured from the spreadsheet. Other registers not listed below do not need to be configured and can be left at their default values.

Register	Field	Value	Notes
SDRCR	**IBANKPOS**	Internal Bank Position 0	
	DDRTERM	50 Ohm	
	DDQS	Differential	
	DLL Enable	Enable	
	DRIVE	Full	
	NM	32 Bits	
	CL	-	See DDR Datasheet

	RSIZE	-	This field is not used since IBANKPOS=0 and EBANKPOS=0
	IBANK	-	See DDR Datasheet
	EBANK	1CS only	
	PAGESIZE		See DDR Datasheet
SDRCR2	**EBANKPOS**	Position 0	
SDRRCR	**INITREF_DIS**	Enable	
	SRT	Normal	
	PASR	Full Array	
PMCR	**-**	Default	* To take advantage of the power saving features, see the TRM for details on each mode.
PBBPR	**COS1_COUNT**	0xFE	
	COS2_COUNT	0xFE	
	PRI_OLD_COUNT	0xFE	
DDRPHYCR	**EN_DYN_PWRDN**	Powered Up only during read	
	RDEYE_LVL_DIS	Disabled	
	GATE_LVL_DIS	Disabled	
	WR_LVL_DIS	Disabled	
	PHY_RST	OUT_OF_RESET	
	IDLE_LODT	100Ohms	
	RD_LODT	50Ohms	
	RD_LATENCY	CL Plus 4	
PRI_COS_MAP	**-**	Default	*
CONNID_COS_1_MAP	**-**	Default	*
CONNID_COS_2_MAP	**-**	Default	*
RD_WR_EXEC_THRSH	**-**	Default	*
DDRIOCTRL	**CS_SLEW**	Fastest_SlewRate	
	CS_IMPEDANCE	50Ohms_8mA	
	CMD_SLEW	Fastest_SlewRate	
	CMD_IMPEDANCE	50Ohms_8mA	
	DATA_SLEW	Fastest_SlewRate	
	DATA_IMPEDANCE	50Ohms_8mA	

***** These registers may all be left in their default state and do not need to be configured

Table 10. DDR Controller Register Settings

Once everything has been configured, the top of the worksheet shows a summary of the required values to be programmed in these registers. The “Optimum Timing Register Values” shows the highest performance (most aggressive) timing values. The “Relaxed Timing Register Values” add some additional margin to the timing values as described in the yellow box at the bottom. These values should be used in the GEL file and/or the code used to initialize the DDR controller.

DDR Register Values with Optimum Timing Register Values					
SDRCR	0x438011B2	SDTIMR3	0x5000032F	PRI_COS_MAP	0x00000000
SDRCR2	0x00000000	PMCR	0x00000000	CONNID_COS_1_MAP	0x00000000
SDRRCR	0x00000C30	PBBPR	0x00FEFEFE	CONNID_COS_2_MAP	0x00000000
SDTIMR1	0x088B159A	ZQCR	NA	RD_WR_EXEC_THRSH	0x00000305
SDTIMR2	0x203631D2	DDRPHYCR	0x00173209	DDRIOCTRL(Control Module R	0x00030303
Relaxed Timing Register Values					
SDTIMR1	0x0AAD25E3	SDTIMR2	0x3037321B	SDTIMR3	0x51008340
NOTE: Relaxed Timing Register Values computes optimal timing + 1 unit for MIN values and optimal timing – 1 unit for MAX parameters					

Table 11. EMIF4 Register Settings.xls EMIF4 Worksheet – DDR Register Values

2.2 Performing Software (Slave Ratio) Leveling

Although the calculations above will provide an initial configuration for the memory controller, DDR2 software (slave ratio) leveling is recommended to optimize the timings in the PHY to compensate for board trace effects. The software leveling is board-specific and relies on information about board design to compute the appropriate configuration.

Software leveling computes three timing components per byte lane:

- Read DQS Slave Ratio
- Read DQS Gate Slave Ratio
- Write DQS Slave Ratio

These values are calculated by a program called ***DDR3_DDR2_slave_ratio_search_TI814x.out***. This program varies DQS delay values to determine the minimum and maximum DQS delays that still operate properly. Once these limits are found, the program produces the mid-point values for these ratios as the optimum timing. ***DDR3_DDR2_slave_ratio_search_TI814x.out*** runs under Code Composer Studio (CCS) to perform the test.

2.2.1 Using the Ratio_Seed_TI814x.xls Spreadsheet

DDR3_DDR2_slave_ratio_search_TI814x.out requires seed values to begin the test. The seed values are obtained by using the ***Ratio_Seed_TI814x.xls*** spreadsheet. The spreadsheet is shown in the figure below. The green cells allow data entry.

	A	B	C	D	E
1	**Parameters**				
2	*DDR3 clock frequency*	400	MHz		
3	*Invert Clkout*	0			
4					
5	**Trace Length (inches)**				
6		*Byte 0*	*Byte 1*	*Byte 2*	*Byte 3*
7	*CK trace*	3.5	3	2.5	2
8	*DQS trace*	2	1.8	1.6	1.4
9					
10					
11	**Seed values (per byte lane)**				
12	*WR DQS*	1B	16	10	B
13	*RD DQS*	34	34	34	34
14	*RD DQS GATE*	BE	AD	9D	8C
15					
16	**Seed Values to input to program**				
17	*WR DQS*	**13**			
18	*RD DQS*	**34**			
19	*RD DQS GATE*	**A5**			

Table 12. Ratio_Seed_TI814x.xls Spreadsheet View

The steps to obtain the seed values are:

1) Enter the desired DDR clock frequency (in cell B2)

2) Select 0 (non-inverted) for the Invert Clkout field

3) Provide the length of the DDR clock traces (average of differential pair) to each of the byte lanes in cells B7-E7. The length is entered in units of inches (1 inch = 1000 mils = 25.4 mm).

4) Provide the length of the DDR DQS traces (average of differential pair) to each of the byte lanes in cells B8-E8. The length is entered in units of inches.

5) The spreadsheet will calculate seed values per byte lane and a set of 3 hexadecimal seed values at the bottom to be entered into ***DDR3_DDR2_slave_ratio_search_TI814x.out*** when it runs, as explained in the following section.

2.2.2 Running the Software Leveling Program

Follow the steps below using Code Composer Studio version 4 or higher to load the program:

1. With the JTAG emulator connected to the board, open the target configuration corresponding to the board

2. In Debug view, right click on the Cortex A8 and click “Connect Target”

3. Load the attached GEL file “Ti814x_ddr.gel”

4. Select DDR2_EMIF0_EMIF1_400MHz_Config from the Scripts menu
5. Click View -> Registers
6. Expand CPSR
7. Select “M” and set it to 0x13
8. Go to Tools -> ARM Advanced Features and select NEON enabled
9. Select Target -> Load Program and load the **DDR3_DDR2_slave_ratio_search_TI814x.out**
10. Run the program. If there is no output, halt the processor, reset the “M” field in the CPSR to 0x13, and reload and run the program

The program will request the ratio values calculated earlier. See the figure below for an example of how the program is run.

```
Run 2:

 Enter the Seed Read DQS Gate Ratio Value in Hex to search the RD DQS Gate Window
 0x106

 Enter the Seed Read DQS Ratio Value in Hex to search the RD DQS Ratio Window
 0x23

 Enter the Seed Write DQS Ratio Value in Hex to search the Write DQS Ratio Window
 0x86

 Enter the input file Name
 seed value out.txt


*********************************************************
           The Slave Ratio Search Program Values are
*********************************************************
PARAMETER                MAX      MIN     OPTIMUM        RANGE
*********************************************************
Read DQS                 66        e          3a         58
Read DQS GATE            170       5c         e6        114
Write DQS                44        0          22         44
Write DATA               68        1c         42         4c
*********************************************************

===== END OF TEST ========
```

Table 13. Running DDR3_DDR2_slave_ratio_search_TI814x.out

The slave ratio search values are saved into the specified output file. They will be used later when initializing the DDR controller.

2.3 Initializing the Device

The steps to initialize the DDR controller and other necessary device components are outlined in the device Technical Reference Manual (TRM). The GEL file included with this document (Ti814x_ddr.gel) performs this initialization. In this section, we will go through the code in the GEL file and explain the procedures needed to initialize the device.

2.3.1 Configure the PLLs

First, the PLLs must be configured for the desired frequency. The maximum supported frequency for DDR2 at OPP 100% is 400 MHz (DDR2-800). To program the DDRPLL to output this frequency, use the following settings:

CLKINP = 20 MHz, **N** = 19, **M** = 800, **M2** = 2

These values give the following output frequency:

20 MHz / (19 + 1) * 800 / 2 = 400 MHz

Slower supported DDR frequencies can be used by lowering the value of **M**.

The following code from the GEL file initializes the DDRPLL for 400 MHz:

```
/* Set PLL to Idle mode */
WR_MEM_32(DDRPLL_CLKCTRL, RD_MEM_32(DDRPLL_CLKCTRL)|0x00800000);
while (( (RD_MEM_32(DDRPLL_STATUS)) & 0x00000101) != 0x00000101);
/* Set SELFREQDCO */
read_clkctrl = RD_MEM_32(Base_Address+CLKCTRL);
WR_MEM_32(Base_Address+CLKCTRL,(read_clkctrl & 0xffffe3ff) | CLKCTRL_VAL);
/* Reset PLL core */
WR_MEM_32(DDRPLL_CLKCTRL, RD_MEM_32(DDRPLL_CLKCTRL)& 0xfffffffe);
wait_delay(3);
/* Load M2NDIV and MN2DIV registers and latch both values */
WR_MEM_32((DDRPLL_M2NDIV),m2nval);
WR_MEM_32((DDRPLL_MN2DIV),mn2val);
wait_delay(3);
WR_MEM_32((DDRPLL_TENABLEDIV),0x1);
wait_delay(3);
WR_MEM_32((DDRPLL_TENABLEDIV),0x0);
wait_delay(3);
WR_MEM_32((DDRPLL_TENABLE),0x1);
wait_delay(3);
```

```
WR_MEM_32((DDRPLL_TENABLE),0x0);

wait_delay(3);

/* Take PLL core out of reset and set DCO clock frequency range (HS2)*/

read_clkctrl = RD_MEM_32(DDRPLL_CLKCTRL);

WR_MEM_32(DDRPLL_CLKCTRL,(read_clkctrl & 0xff7fe3ff) | CLKCTRL_VAL);

read_clkctrl = RD_MEM_32(DDRPLL_CLKCTRL);

/* Poll for the frequency and phase lock to occur */

while (( (RD_MEM_32(DDRPLL_STATUS)) & 0x00000600) != 0x00000600);

wait_delay(10);
```

Note that the wait delays "**wait_delay(3)**" are only required for PLL initialization done from a GEL file. PLL initialization done from user code does not require the wait delays.

2.3.2 Configure the PRCM

Next the clocks for the DMM must be enabled using the PRCM as follows:

```
/*Enable DMM Clock*/

WR_MEM_32(CM_DEFAULT_DMM_CLKCTRL,       0x2);

/*Poll for Module is functional*/

while(RD_MEM_32(CM_DEFAULT_DMM_CLKCTRL)!=0x2);
```


2.3.3 Configure the Control Module Registers

The DDR IO Control registers (DDRIOCTRL) should be set according to Table 10:

```
 WR_MEM_32(DDR0_IO_CTRL,0x00030303);

 WR_MEM_32(DDR1_IO_CTRL,0x00030303);
```

This sets all fields to the values recommended in the TRM for DDR2 operation and should not need to be customized.

2.3.4 Perform VTP Calibration

In order to compensate for voltage, temperature, and process variations, the IO driver impedance can be configured automatically. There are two modules that must be initialized using the following sequence:

```
// Write 1 to ENABLE bit

WR_MEM_32(DDR_VTP_CTRL_0, RD_MEM_32(DDR_VTP_CTRL_0) | 0x00000040 );

WR_MEM_32(DDR_VTP_CTRL_1, RD_MEM_32(DDR_VTP_CTRL_1) | 0x00000040 );

// Write 0 to CLRZ bit
```

```
WR_MEM_32(DDR_VTP_CTRL_0, RD_MEM_32(DDR_VTP_CTRL_0) & 0xfffffffe );

WR_MEM_32(DDR_VTP_CTRL_1, RD_MEM_32(DDR_VTP_CTRL_1) & 0xfffffffe );

// Write 1 to CLRZ bit

WR_MEM_32(DDR_VTP_CTRL_0, RD_MEM_32(DDR_VTP_CTRL_0) | 0x00000001 );

WR_MEM_32(DDR_VTP_CTRL_1, RD_MEM_32(DDR_VTP_CTRL_1) | 0x00000001 );

// Read VTP control registers & check READY bits

while( (RD_MEM_32(DDR_VTP_CTRL_0) & 0x00000020) != 0x20);

while( (RD_MEM_32(DDR_VTP_CTRL_1) & 0x00000020) != 0x20);
```

The default values of 0x3 for the FILTER field and 0x0 for the LOCK field are recommended and used in the example above.

2.3.5 Configure the PHY Registers

The PHYs contain a number of macros which must all be initialized according to both the mode of operation and the board layout characteristics, determined in the previous sections.

First, initialize the Command macros as follows for each PHY:

```
WR_MEM_32(CMD0_REG_PHY0_INVERT_CLKOUT_0, 0);

WR_MEM_32(CMD1_REG_PHY0_INVERT_CLKOUT_0, 0);

WR_MEM_32(CMD2_REG_PHY0_INVERT_CLKOUT_0, 0);


WR_MEM_32(CMD0_REG_PHY0_CTRL_SLAVE_RATIO_0,(0x80));

WR_MEM_32(CMD1_REG_PHY0_CTRL_SLAVE_RATIO_0,(0x80));

WR_MEM_32(CMD2_REG_PHY0_CTRL_SLAVE_RATIO_0,(0x80));


WR_MEM_32(CMD0_REG_PHY0_DLL_LOCK_DIFF_0,0x4);

WR_MEM_32(CMD1_REG_PHY0_DLL_LOCK_DIFF_0,0x4);

WR_MEM_32(CMD2_REG_PHY0_DLL_LOCK_DIFF_0,0x4);
```

For normal operation, the non-inverted clock is passed to the DRAM. The default value of 0x80 should be used for CMD_SLAVE_RATIO. The DLL lock difference should also use its default value of 0x4.

Next, each of the Data macros must be configured as follows:

```
       WR_MEM_32((DATA0_REG_PHY0_RD_DQS_SLAVE_RATIO_0 + BaseAddrOffset), (rd_dqs_cs0
<< 10 | rd_dqs_cs0));

       WR_MEM_32((DATA0_REG_PHY0_WR_DQS_SLAVE_RATIO_0 + BaseAddrOffset), (wr_dqs_cs0
<< 10 | wr_dqs_cs0));

       WR_MEM_32((DATA0_REG_PHY0_WRLVL_INIT_RATIO_0 + BaseAddrOffset),   0);
```

```
        WR_MEM_32((DATA0_REG_PHY0_GATELVL_INIT_RATIO_0 + BaseAddrOffset), 0);

        WR_MEM_32((DATA0_REG_PHY0_RD_DQS_GATE_SLAVE_RATIO_0 +
BaseAddrOffset),(RD_DQS_GATE_cs0 << 10  | RD_DQS_GATE_cs0));

        WR_MEM_32((DATA0_REG_PHY0_WR_DATA_SLAVE_RATIO_0 + BaseAddrOffset),(wr_data_cs0
<< 10 | wr_data_cs0));
```

The values for the read and write leveling come from the software leveling program output. The Write and Read Leveling Init Ratios should all be set to 0x0.

2.3.6 Configure the DMM

The DMM LISA MAP registers should be configured according to the memory configuration used on the board. For optimal performance, it is recommended to use interleaving between the two controllers.

For example, the EVM uses a total of 8Gb of DDR2, with 4Gb attached to each controller. Therefore the DMM LISA MAP registers are set as follows:

```
/*Program the DMM to Access EMIF0 and EMIF1*/

 WR_MEM_32(DMM_LISA_MAP_0, 0x80640300);

 WR_MEM_32(DMM_LISA_MAP_1, 0x80640300);

 WR_MEM_32(DMM_LISA_MAP_2, 0x80640300);

 WR_MEM_32(DMM_LISA_MAP_3, 0x80640300);

 while(RD_MEM_32(DMM_LISA_MAP_0)!=0x80640300);

 while(RD_MEM_32(DMM_LISA_MAP_1)!=0x80640300);

 while(RD_MEM_32(DMM_LISA_MAP_2)!=0x80640300);

 while(RD_MEM_32(DMM_LISA_MAP_3)!=0x80640300);

 WR_MEM_32(DMM_PAT_VIEW_MAP_BASE, 0x80000000);
```

This configures the two controllers to interleave data every 128 bytes, from address 0x80000000 to 0xBFFFFFFF. Details on how this is configured are explained in the DMM section of the TRM.

2.3.7 Program the DDR Controller MMRs

Finally, the DDR controller registers are programmed according to the values calculated using the spreadsheet in section 2.1, with the exception of steps 2 and 3 below. The registers and corresponding shadow registers should be programmed in the following order:

1. **DDRPHYCR**, **SDRTIM1**, **SDRTIM2**, **SDRTIM3**, **SDRCR**, **SDRCR2, PBBPR**
2. **SDRRCR** (with **SDRRCR = 0x10004000**)
3. **SDRRCR** (with **SDRRCR = 0x00004000**)
4. **SDRRCR** (with **SDRRCR** programmed with value from spreadsheet)

The DDR memory should now be accessible.